Exploiting Inferential Structure in Neural Processes

Abstract

Neural Processes represent distributions over functions for which the uncertainty is induced by an aggregated context encoding. This utilizes existing set-based representation learning architectures whereby context points are first embedded individually and then pooled into a single representation. The pooling step is often implemented by a simple mean-aggregation which has been shown to limit the expressiveness and result in underfitting. We propose a novel context aggregation mechanism that combines flexible gradient-based learning with structured representations via the machinery of probabilistic graphical models. The aggregation operation is exactly determined by the choice of distributional assumptions and inference routine, whilst remaining fully-differentiable. The previously proposed *Bayesian aggregation* (BA) mechanism can be recovered from Gaussian assumptions. We demonstrate the generality of our framework by using structured exponential-family priors—in particular, mixture and Student-t assumptions—resulting in improvements over BA in terms of function modelling and test-time robustness.

1 INTRODUCTION

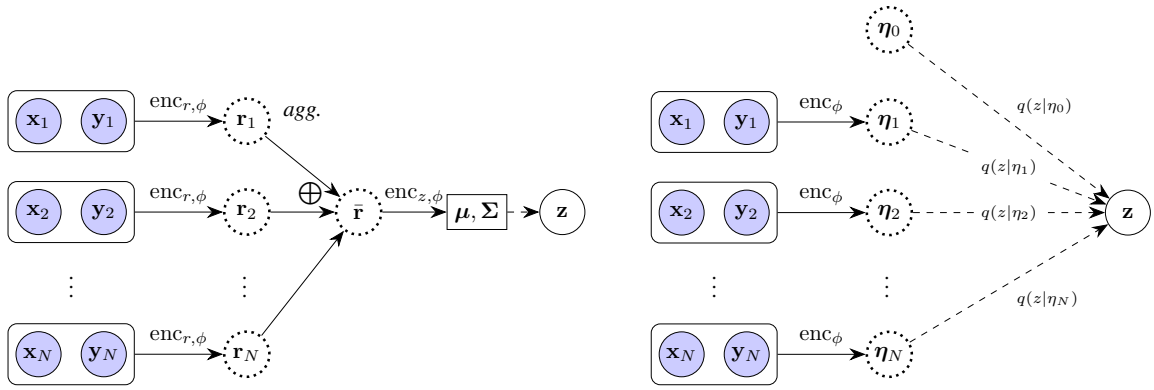
Neural Processes (NPs) [Garnelo et al., 2018a,b] are stochastic processes parameterized by neural networks. They can be conveniently trained via a meta-learning framework by exposing them to many related datasets: each dataset is presented as a context-target split, enabling NPs to perform test-time predictions by aggregating a set of context points. This extends the amortization procedure of Variational Autoencoder (VAE) [Kingma and Welling, 2013] where, instead of separately encoding each example into the latent space, a set of multiple examples are simultaneously encoded. Such mapping is convenient for problems which naturally utilize sets of examples [Wagstaff et al., 2019], for example, handling a set of 3D objects, a set of multiple agents, or just a point cloud.

Despite this, there are several shortcomings of existing context aggregation strategies. Often, special architectures are used to satisfy symmetry properties [Bloem-Reddy and Teh, 2020], for example, invariances under different permutation of the set elements. In its most basic form this reduces to a *sum-decomposition network* (SDN) [Edwards and Storkey, 2017, Zaheer et al., 2017]. As shown in Fig. 1, a shared network is used to embed individual context points, followed by an aggregation operation that pools all embeddings into a global contextual representation. A further network then transforms this representation to the parameters of the latent variable. Unfortunately, such structures can underfit [Kim et al., 2019] and suffer from limited representation power [Wagstaff et al., 2019].

Part of this deficiency is attributed to the equal weights assigned to every context embedding [Kim et al., 2019]. Previous attempts at alleviating this have sought to increase the expressiveness by incorporating attention modules [Kim et al., 2019] or utilize a transformer-based architecture [Nguyen and Grover, 2022]. However, such approaches increase expressiveness at the expense of an increase in parameterization due to additional networks or completely replace the existing architecture stack. Is it possible to do this without any additional networks as well as maintain the simplicity of SDN? Our goal in this paper is to show that it is.

We propose a new context aggregation mechanism which does not require additional networks. This is made by possible by a novel framing of individual context embeddings as *potentials* specifying a probabilistic graphical model (PGM). Context aggregation then naturally arises from inference on the PGM. We can now have greater expressiveness through the choice of distributional assumptions and inference strategies; this does not require any increase in the number of

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.



(a) Sum-Decomposition Network

(b) Structured Inference Network

Figure 1: A set of context points $(\mathbf{x}_i, \mathbf{y}_i)$ can be aggregated using a sum-decomposition architecture, shown on the left. A shared network $\mathrm{enc}_{r,\phi}$ is used for all followed by a pre-specified aggregation operation that pools all embeddings. The output is passed to a further network $\mathrm{enc}_{z,\phi}$. We propose an alternative, shown on the right, that does not require an additional network and rather uses inference on a probabilistic graphical model (PGM) to automatically aggregate the embeddings. Different types of exponential-family distributions and mixtures, denoted here by $q(z|\eta_i)$, can be used resulting in different aggregation mechanisms.

parameters. Specifically, we use the method by [Lin et al., 2018] referred to as *structured inference network* where inference can be performed with a mix of message-passing scheme [Winn et al., 2005] and gradient-based method, bringing the best of the two worlds together. The method can be seen as a generalization of the recently proposed Bayesian Aggregation (BA) mechanism [Volpp et al., 2020] which is recovered as a special case when simple Gaussian factors are used in the PGM. Our approach allows for much more expressive PGMs from which novel context aggregation mechanisms can be derived. In particular, we consider mixture and Student's t assumptions, which give rise to two variants of BA, which we term *mixture* and *iterative* Bayesian Aggregation. We demonstrate improvements to the functional modelling and test-time robustness compared with BA without any increase in the number of parameters.

2 BACKGROUND

Data NPs assume a partition of the data into a *context* set and a *target* set. The former is used by the model to seed adaptation. The latter is a set of points from the same domain as the context set and for which we will make predictions. Specifically, we denote the context set for the lth task as $\mathcal{D}_c^{(l)} = \{\mathbf{x}_{c,i}^{(l)}, \mathbf{y}_{c,i}^{(l)}\}_{i=1}^{N_{l,c}}$, where $\mathbf{x}$ denotes a feature vector and $\mathbf{y}$ the corresponding response vector. The target set for the lth task is denoted similarly as $\mathcal{D}_t^{(l)} = \{\mathbf{x}_{t,i}^{(l)}, \mathbf{y}_{t,i}^{(l)}\}_{i=1}^{N_{l,t}}$. At test time, for a new task l_*, we observe $\mathcal{D}_c^{(l_*)}$ and $\{\mathbf{x}_{t,i}^{(l_*)}\}_{i=1}^{N_{l_*,t}}$. The target responses $\{\mathbf{y}_{t,i}^{(l_*)}\}_{i=1}^{N_{l_*,t}}$ are unobserved, and our goal is to predict them.

Neural Processes *Neural processes* [Garnelo et al., 2018b] frame few-shot learning as a multi-task learning problem [Heskes, 2000], employing a conditional latent variable model with context/target splits on task-specific datasets as shown in Fig. 2. Training amounts to maximising the following *conditional* marginal likelihood across L tasks:

$$\sum_{l=1}^{L} \log p_\theta(\mathcal{D}_t^{(l)} \,|\, \mathcal{D}_c^{(l)}) \tag{1}$$

$$= \sum_{l=1}^{L} \log \int p_\theta(\mathcal{D}_t^{(l)} \,|\, \mathbf{z}_l)\; p_\theta(\mathbf{z}_l \,|\, \mathcal{D}_c^{(l)})\; d\mathbf{z}_l \tag{2}$$

$$\gtrapprox \sum_{l=1}^{L} \mathbb{E}_{q_\phi(\mathbf{z}|\mathcal{D}_c^{(l)} \cup \mathcal{D}_t^{(l)})} \left[\log p_\theta(\mathcal{D}_t^{(l)} \,|\, \mathbf{z})\right] \tag{3}$$

$$- \mathrm{D}_{\mathrm{KL}} \left[q_\phi(\mathbf{z}|\mathcal{D}_c^{(l)} \cup \mathcal{D}_t^{(l)}) \,\|\, q_\phi(\mathbf{z}|\mathcal{D}_c^{(l)})\right]$$

where $\mathbf{z}_l$ is the task-specific latent variable and θ is the global parameter that is shared across tasks. The marginalization over task-specific latent variables is typically intractable hence approximate inference is required:

$$p_\theta(\mathbf{z} \,|\, \mathcal{D}_c) = \frac{1}{p_\theta(\mathcal{D}_c)} \prod_{i=1}^{N_c} p_\theta(\mathbf{y}_{c,i} \,|\, \mathbf{x}_{c,i}, \mathbf{z})\; p_\theta(\mathbf{z}) \tag{4}$$

$$\approx q_\phi(\mathbf{z} \,|\, \mathcal{D}_c). \tag{5}$$

We have dropped task indices for the sake of notational simplicity. The variational approximation is amortised, meaning we will parameterize the local approximations with an inference model. For a Gaussian approximation, the mean and variance are parameterized by neural networks (NNs) that take as input sets of datapoints: $q_\phi(\mathbf{z} \,|\, \mathcal{D}_c) = \mathcal{N}(\mathbf{z} \,|\, \widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}})$

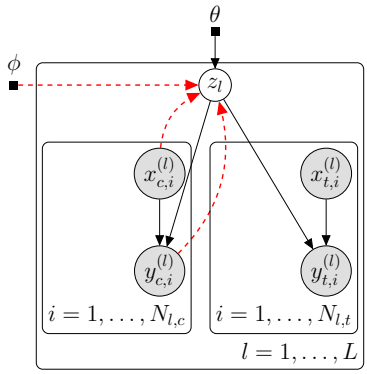


Figure 2: Generative process for multi-task learning with context/target splits along with the variational approximation (- - →).

with $\widetilde{\boldsymbol{\mu}} = \text{enc}_\phi^m(\mathcal{D}_c)$ and $\widetilde{\boldsymbol{\Sigma}} = \text{enc}_\phi^v(\mathcal{D}_c)$. Throughout we assume covariance matrices have diagonal structure, resulting in factorized Gaussian distributions.

Neural Process Extensions NPs come in two variants, the aforementioned latent NP [Garnelo et al., 2018b] and the conditional NP (CNP) [Garnelo et al., 2018a]. The CNP can be obtained from the latent NP by approximating the latent posterior with a Dirac delta distribution, $p_\theta(\mathbf{z}|\mathcal{D}_c) = \delta(|\mathbf{z} - \hat{\mathbf{z}}|)$. Training CNPs then corresponds to a maximum likelihood meta-training procedure. Most subsequent work has improved upon NPs through NN architectural modifications such as attention [Kim et al., 2019, 2022], convolutions [Gordon et al., 2020, Foong et al., 2020], mixtures [Wang and Van Hoof, 2022], equivariance [Kawano et al., 2021], and adaptation [Requeima et al., 2019]. Comparatively fewer works have attempted to improve the distributional assumptions of the latent variable. Two works that have have attempted to do so by employing hierarchical [Wang and Van Hoof, 2020] and non-parametric [Flam-Shepherd et al., 2018] formulations.

Sum-Decomposition Networks The inference networks for NPs must have at least two properties. The first is that they make no assumptions about the size of the context set. The second is that the encoder be invariant to the ordering of context points. A common way to satisfy these criteria is by having the encoder take the form of a sum-decomposition network [Edwards and Storkey, 2017, Zaheer et al., 2017]:

$$\mathbf{r}_i = f_\phi(\mathbf{x}_{c,i}, \mathbf{y}_{c,i}), \quad \bar{\mathbf{r}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{r}_i \tag{6}$$

where $\mathbf{r}_i$ are datapoint-wise encodings given by a NN which are then aggregated. The aggregation operation is typically taken to be a simple average in NPs but other operators are valid as long as they are permutation-invariant. Thus the amortisation goes one level further with parameter sharing across context points. Finally, the aggregated representation $\bar{\mathbf{r}}$ is passed to further NNs to give the variational parameters. In the case of Gaussian posterior, there are two further NNs predicting the mean and variance: $\widetilde{\boldsymbol{\mu}} = \mu_\phi(\bar{\mathbf{r}})$, $\widetilde{\boldsymbol{\Sigma}} = \Sigma_\phi(\bar{\mathbf{r}})$.

Bayesian Context Aggregation Volpp et al. [2020] propose a novel aggregation mechanism, derived by constructing a surrogate conditional latent variable (CLV) model in which the datapoint-wise encodings $\mathbf{r}_i$ are interpreted as noisy observations of the underlying Gaussian latent variable. Another encoder network then evaluates the observation noise for each datapoint $\boldsymbol{\sigma}_i^2 = f_{\phi_{\text{sigma}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i})$. By choosing a Gaussian prior with mean $\boldsymbol{\mu}_0$ and variance $\boldsymbol{\sigma}_0^2$, Bayesian inference in this surrogate CLV results in the following aggregation mechanism:

$$\widetilde{\boldsymbol{\sigma}}^2 = \left[\boldsymbol{\sigma}_0^{-2} + \sum_{i=1}^{N_c} \boldsymbol{\sigma}_i^{-2} \right]^{-1} \tag{7}$$

$$\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_0 + \widetilde{\boldsymbol{\sigma}}^2 \circ \sum_{i=1}^{N_c} \left(\mathbf{r}_i - \boldsymbol{\mu}_0 \right) / \boldsymbol{\sigma}_i^2 \tag{8}$$

where $\mathbf{a} \circ \mathbf{b}$ and $\mathbf{a}/\mathbf{b}$ denote element-wise product and division respectively between vector $\mathbf{a}$ and $\mathbf{b}$. Aggregation operates directly in the latent-space which forgoes the need for further NNs. Bayesian aggregation is a strict generalization of mean aggregation since the latter is recovered when a non-informative prior is used along with uniform observation variances. Eq. (8) can be seen as re-weighting context points, with the weights given by $\boldsymbol{\sigma}_i^{-2}$. This has some resemblance to self-attention mechanisms that have been adapted for neural processes [Kim et al., 2019] however the key difference is that the weights are computed without consideration of other context points.

3 STRUCTURED INFERENCE NETWORKS

Despite the flexibility of the sum-decomposition architecture, its use is more for computational convenience than due to a strong connection to the variational approximation (Eq. (4)). The computation is clearly localized to each data point, similar to the factorization of the likelihood, but the additional NN-based transform make them uninterpretable. Furthermore, there is no explicit presence of a prior (whose underlying structure could be exploited). Below describe our approach that incorporates the structure of the graphical model into the encoder whilst still allowing fast amortized inference.

Exponential Family Let us first consider the case where the likelihood and prior in Eq. (4) are given by *conjugate* exponential-family distributions. The posterior can then be obtained analytically [Wainwright et al., 2008]. The prior is expressed as $p(\mathbf{z}) = h(\mathbf{z}) \exp\left[\langle \mathbf{T}(\mathbf{z}), \boldsymbol{\eta}_0 \rangle - A(\boldsymbol{\eta}_0)\right]$, where $\boldsymbol{\eta}_0$ is the natural parameters, $\mathbf{T}(\mathbf{z})$ is the sufficient statistics,

$h(\mathbf{z})$ is the base measure and $A(\boldsymbol{\eta}_0)$ is the log-partition function. Due to conjugacy, the likelihood can be expressed in the same form as the prior. Excluding the feature vector for simplicity and droping the subscript indicating the context set for notational clarity, we have:

$$p(\mathbf{y}_i|\mathbf{z}) = h(\mathbf{y}_i)\exp\left[\langle \mathbf{T}(\mathbf{y}_i), \boldsymbol{\eta}_y(\mathbf{z})\rangle - A\left(\boldsymbol{\eta}_y(\mathbf{z})\right)\right] \quad (9)$$

$$\propto \exp\left[\langle \mathbf{T}(\mathbf{z}), \boldsymbol{\eta}_i(\mathbf{y}_i)\rangle\right] \quad (10)$$

where $\mathbf{T}(\mathbf{z}) := (\boldsymbol{\eta}_y(\mathbf{z}), -A\left(\boldsymbol{\eta}_y(\mathbf{z})\right))$ and $\boldsymbol{\eta}_i(\mathbf{y}_i) := (\mathbf{T}(\mathbf{y}_i), 1)$. Then the posterior distribution is,

$$p(\mathbf{z}|\mathcal{D}) \propto h(\mathbf{z})\exp\left[\left\langle \mathbf{T}(\mathbf{z}), \boldsymbol{\eta}_0 + \sum_{i=1}^{N}\boldsymbol{\eta}_i(\mathbf{y}_i)\right\rangle\right] \quad (11)$$

where it is clear that the posterior natural parameters are given by adding the *sufficient statistics* of $\{\mathbf{y}_i\}_{i=1}^N$ to the prior natural parameters.

Neural Sufficient Statistics With conjugacy in hand, we can construct variational distributions that embed expressive recognition networks within these tractable computations:

$$q_\phi(\mathbf{z}|\mathcal{D}_c) \propto \exp\left[\left\langle \mathbf{T}(\mathbf{z}), \boldsymbol{\eta}_0 + \sum_{i=1}^{N_c}\mathbf{f}_\phi(\mathbf{x}_{c,i}, \mathbf{y}_{c,i})\right\rangle\right] \quad (12)$$

where $\mathbf{f}_\phi(\cdot)$ is a flexible neural network for automatic and amortized construction of summary statistics termed *neural sufficient statistics* [Wu et al., 2020].

Eq. (12) is an instance of a *structured inference network* [Lin et al., 2018, Johnson et al., 2016] that reflects the likelihood's factorization (Eq. (4)) and the presence of a prior. We can rewrite this as:

$$q_\phi(\mathbf{z}\,|\,\mathcal{D}_c) = \frac{1}{Z_c(\phi)}\underbrace{\left[\prod_{i=1}^{N_c} q(\mathbf{z}\,|\,f_{\phi_{\text{NN}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i}))\right]}_{\text{NN factor}} \times \underbrace{\left[q(\mathbf{z}; \phi_{\text{PGM}})\right]}_{\text{prior}} \quad (13)$$

where $Z_c(\phi)$ is the normalization constant. The neural sufficient statistics give rise to NN factors that are conjugate to the prior by construction. The framework is flexible to also allow for the prior natural parameters to be fitted, $\phi := \{\phi_{\text{NN}}, \phi_{\text{PGM}}\}$. The neural sufficient statistics may not necessarily be in canonical form. In the case of Gaussian factors, $f_{\phi_{\text{NN}}}(\cdot)$ may output the mean and variance, and therefore a further transformation to natural parameters is required. The neural sufficient statistics view is further supported by considering the VI objective used to train NPs. After substituting the structured inference network and simplifying, the resulting objective contains a term that resembles the entropy on the individual NN factors (for derivation see the appendix).

These structured inference networks have several important differences from the sum-decomposition network. The first is the level at which aggregation is performed. Instead of having the intermediate representation $\bar{\mathbf{r}}$ (Eq. (6))—which has no interpretation—the encoder produces the NN factors. The variational parameters are then computed directly from these factors instead of having to be computed by a black-box NN-based transform (similar to Volpp et al. [2020]). Not only does this aid in interpretability, but it also results in fewer NN parameters to estimate. The second change is the introduction of an explicit prior distribution whose parameters are aggregated along with the datapoint-wise representations. The third change is the aggregation strategy is determined directly from the parameterization chosen for the exponential-family. If operating in natural parameters (as shown above), the aggregation is a simple sum—thus recovering mean aggregation. However, as we will demonstrate below, other parameterizations lead to non-trivial pooling operations.

Gaussian Prior For a Gaussian prior, the resulting conjugate exponential-family distribution for the NN factors is also Gaussian:

$$q_\phi(\mathbf{z}\,|\,\mathcal{D}_c) = \frac{1}{Z_c(\phi)}\prod_{i=1}^{N_c}\mathcal{N}(\mathbf{z}\,|\,\mathbf{m}_i, \mathbf{V}_i)\,\mathcal{N}(\mathbf{z}\,|\,\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \quad (14)$$

where $\mathbf{m}_i = f^{(1)}_{\phi_{\text{NN}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i})$ and $\mathbf{V}_i = f^{(2)}_{\phi_{\text{NN}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i})$ are the mean and variance parameterized by a NN. Let $q_\phi(\mathbf{z}\,|\,\mathcal{D}_c) = \mathcal{N}(\mathbf{z}; \widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}})$ then the posterior moments are given by,

$$\widetilde{\boldsymbol{\Sigma}}^{-1} = \sum_{i=1}^{N_c}\mathbf{V}_i^{-1} + \boldsymbol{\Sigma}_0^{-1} \quad (15)$$

$$\widetilde{\boldsymbol{\mu}} = \widetilde{\boldsymbol{\Sigma}}\left(\sum_{i=1}^{N_c}\mathbf{V}_i^{-1}\mathbf{m}_i + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right). \quad (16)$$

The normalization constant $Z_c(\phi)$ is also available in closed-form. Since we assume diagonal covariance matrices, the computations remain tractable, i.e. no expensive matrix operations. The Gaussian-based procedure above is equivalent to the Bayesian aggregation mechanism [Volpp et al., 2020]. This can be seen by a straightforward manipulation of Eq. (16) to give the incremental update form of Eq. (8) (see appendix for details).

Mixture of Gaussian prior We can consider a more expressive prior distribution such as a mixture of Gaussian (MoG) which despite being a conditionally-conjugate exponential-family distribution, results in closed-form up-

dates when the NN factors are Gaussian densities,

$$q_\phi(\mathbf{z} \mid \mathcal{D}_c) = \frac{1}{Z_c(\phi)} \prod_{i=1}^{N_c} \mathcal{N}(\mathbf{z} \mid \mathbf{m}_i, \mathbf{V}_i) \left[\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right] \tag{17}$$

$$= \sum_{k=1}^{K} \widetilde{\pi}_k \mathcal{N}(\mathbf{z} \mid \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k) \tag{18}$$

Consequently, the posterior also takes a MoG form in which the expressions for the mean and variance of the mixture components take a similar form to Eqs. (15) and (16). There is also an update expression for the mixing proportions (see the appendix for further details). A MoG prior may be a beneficial modelling assumption if we expect the data to arise from multiple sources. We refer to this approach as *mixture* Bayesian Aggregation (mBA), a generalization of BA that is recovered when the number of mixture components K is set to 1.

4 BEYOND CONJUGACY

The conjugate case is attractive due to its analytic properties, but it may be a poor modelling assumption for real-world data. By dropping the need for conjugacy between the NN factors and the prior, we can allow for a wider range of distributional assumptions, better tailoring the NP to the problem and assumptions at hand. Yet non-conjugacy implies the posterior can no longer be expressed analytically since the normalization constant $Z_c(\phi)$ is intractable. Below we address the conditionally conjugate and fully non-conjugate cases.

Conditional Conjugacy Consider the case in which the NN factors and prior specify a conditionally-conjugate exponential-family system. We start by introducing additional latent variables $\boldsymbol{\alpha}, \boldsymbol{\beta}$.

$$q_\phi(\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{D}_c) = \underbrace{\left[\prod_{i=1}^{N_c} q(\mathbf{z} \mid f_{\phi_{\text{NN}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i}), \beta_i) \right]}_{\text{NN factor}} \times \underbrace{\left[q(\mathbf{z}|\boldsymbol{\alpha})\; q(\boldsymbol{\alpha}; \phi_{\text{PGM}})\; q(\boldsymbol{\beta}; \phi_{\text{PGM}}) \right]}_{\text{prior}} \tag{19}$$

followed by an approximating distribution $\tilde{q} \in \mathcal{Q}$ that belongs to the structured family of mean-field distributions:

$$\tilde{q}(\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \tilde{q}(\mathbf{z})\; \tilde{q}(\boldsymbol{\alpha}) \prod_{i=1}^{N_c} \tilde{q}(\beta_i) \tag{20}$$

This gives rise to simple, closed-form, iterative updates for the variational posterior:

$$\tilde{q}(\mathbf{z}) \propto \exp\left\{ \mathbb{E}_{\tilde{q}(\boldsymbol{\beta})\tilde{q}(\boldsymbol{\alpha})} \left[\log q_\phi(\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{D}_c) \right] \right\} \tag{21}$$

with analogous expressions for $\tilde{q}(\boldsymbol{\alpha})$ and $\tilde{q}(\boldsymbol{\beta})$. This is referred to as coordinate ascent variational inference (CAVI) [Blei et al., 2017] (itself a special case of variational message passing [Winn et al., 2005]) in which each factor of the variational density is optimized whilst holding the others fixed. This is a fixed-point algorithm that is expected to converge very quickly.

Example: robust regression We specify the following distributions in Eq. (19),

$$q(\mathbf{z} \mid f_{\phi_{\text{NN}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i}), \beta_i) = \mathcal{N}(\mathbf{z} \mid \mathbf{m}_i, \beta_i^{-1}\mathbf{V}_i) \tag{22}$$

$$q(\mathbf{z}|\alpha) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \alpha^{-1}\mathbf{I}) \tag{23}$$

$$q(\alpha) = \mathcal{G}(\alpha|a_0, b_0) \tag{24}$$

$$q(\beta_i) = \mathcal{G}(\beta_i|c_0, c_0) \tag{25}$$

where $\{\mathbf{m}_i, \mathbf{V}_i\} = f_{\phi_{\text{NN}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i})$ and $\phi_{\text{PGM}} = \{a_0, b_0, c_0\}$. Eqs. (22) and (25) together can be viewed as the hierarchical form of the Student's t distribution,

$$\mathcal{T}(\mathbf{z}|\mathbf{m}_i, \mathbf{V}_i, c_0) = \int \mathcal{N}(\mathbf{z} \mid \mathbf{m}_i, \beta_i^{-1}\mathbf{V}_i)\; \mathcal{G}(\beta_i|c_0, c_0)\; d\beta_i$$

However, the marginal form would unfortunately render the updates in Eq. (21) intractable. Instead we restrict our attention to the joint specification and follow previous work on robust Bayesian interpolation [Tipping and Lawrence, 2005] which demonstrated robustness to outliers and corruptions in the targets with this form. The prior is also specified in hierarchical form with the hyperprior Eq. (24) parameters initialized to be very small values (i.e. uninformative hyperprior).

Variational Message Passing Now we can specify the iterative updates in Eq. (21) explicitly for these distributional assumptions. We have $\tilde{q}(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}})$ with,

$$\widetilde{\boldsymbol{\Sigma}}^{-1} = \sum_{i=1}^{N_c} \mathbb{E}[\beta_i]\mathbf{V}_i^{-1} + \mathbb{E}[\alpha]\mathbf{I} \tag{26}$$

$$\widetilde{\boldsymbol{\mu}} = \widetilde{\boldsymbol{\Sigma}} \sum_{i=1}^{N_c} \mathbb{E}[\beta_i]\mathbf{V}_i^{-1}\mathbf{m}_i \tag{27}$$

$\tilde{q}(\boldsymbol{\beta}) = \prod_{i=1}^{N_c} \text{Gamma}(\beta_i|\widetilde{c}, \widetilde{d}_i)$ with,

$$\widetilde{c} = c_0 + \frac{D}{2} \tag{28}$$

$$\widetilde{d}_i = c_0 + \frac{1}{2}\Big(\mathbf{m}_i^\top \mathbf{V}_i^{-1}\mathbf{m}_i - 2\mathbf{m}_i^\top \mathbf{V}_i^{-1}\mathbb{E}[\mathbf{z}] + \text{tr}(\mathbf{V}_i^{-1}\mathbb{E}[\mathbf{z}\mathbf{z}^\top]) \Big) \tag{29}$$

Table 1: *mixture Bayesian Aggregation (mBA)*. We observe a boost in performance with increasing number of components (K), as well as exceeding the performance of NP with Self-Attention (NP+SA) on OOD data. This is demonstrated on the task of 2D image completion (EMNIST).

	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	$\mathbf{0.977}_{\pm 0.006}$	$\mathbf{0.840}_{\pm 0.005}$	$0.823_{\pm 0.007}$	$0.609_{\pm 0.008}$	$0.127_{\pm 0.002}$	$\mathbf{0.165}_{\pm 0.001}$	$0.177_{\pm 0.002}$	$0.224_{\pm 0.002}$
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)	$0.975_{\pm 0.083}$	$0.792_{\pm 0.064}$	$\mathbf{0.883}_{\pm 0.105}$	$\mathbf{0.642}_{\pm 0.084}$	$\mathbf{0.122}_{\pm 0.025}$	$0.177_{\pm 0.011}$	$\mathbf{0.155}_{\pm 0.031}$	$\mathbf{0.217}_{\pm 0.016}$

and finally $\tilde{q}(\alpha) = \text{Gamma}(\alpha|\widetilde{a}, \widetilde{b})$ with,

$$\widetilde{a} = a_0 + \frac{D}{2} \tag{30}$$

$$\widetilde{b} = b_0 + \frac{1}{2}\mathbb{E}[\mathbf{z}^\top \mathbf{z}] \tag{31}$$

The expectations required to evaluate the above updates are given by,

$$\mathbb{E}[\alpha] = \frac{\widetilde{a}}{\widetilde{b}}, \quad \mathbb{E}[\beta_i] = \frac{\widetilde{c}}{\widetilde{d}_i} \tag{32}$$

$$\mathbb{E}[\mathbf{z}] = \widetilde{\boldsymbol{\mu}}, \quad \mathbb{E}[\mathbf{z}\mathbf{z}^\top] = \widetilde{\boldsymbol{\mu}}\widetilde{\boldsymbol{\mu}}^\top + \widetilde{\boldsymbol{\Sigma}} \tag{33}$$

It is clear that the updates (specifically the Gaussian posterior moments in Eqs. (26) and (27)) are a generalization of the Bayesian aggregation mechanism, which is recovered when the Gamma variational parameters are set to be $\widetilde{a} = \widetilde{b} = \widetilde{c} = \widetilde{d}_i = 1$. If we initialize the parameters as such then running VMP steps can be interpreted as reweighting the neural sufficient statistics where the weights are given by the 1st moment of the variational noise distibution. The context-dependent variability in this term is given by Eq. (29) which is evaluated using the encoding of the given context point as well as the merged representation. Viewed as a weighted aggregation scheme, this has close resemblance to self-attention which evaluates a similarity function (e.g. Euclidean norm, dot-product etc.) between all pairs of context points. In the case of corruptions to the context set, such outliers can be downweighted, improving the robustness of neural processes. We refer to our approach as *iterative* Bayesian aggregation (iBA).

General Non-Conjugacy Lastly, we turn our attention to the fully non-conjugate case. Here there is no structure to exploit for inference, but we could proceed by forming a lower bound on the evidence, $\mathcal{L}_{\text{SIN}}(\tilde{q}(\mathbf{z})) \leq \log Z_c(\phi)$, where $\mathcal{L}_{\text{SIN}}$ is defined as:

$$\begin{aligned}\mathcal{L}_{\text{SIN}}(\tilde{q}(\mathbf{z})) =& \mathbb{E}_{\tilde{q}(\mathbf{z})}\left[\sum_{i=1}^{N_c} \log q(\mathbf{z} \mid f_{\phi_{\text{NN}}}(\mathbf{x}_{c,i}, \mathbf{y}_{c,i}))\right] \\ &- \text{D}_{\text{KL}}\left[\tilde{q}(\mathbf{z}) \,\|\, q(\mathbf{z}; \phi_{\text{PGM}})\right]\end{aligned} \tag{34}$$

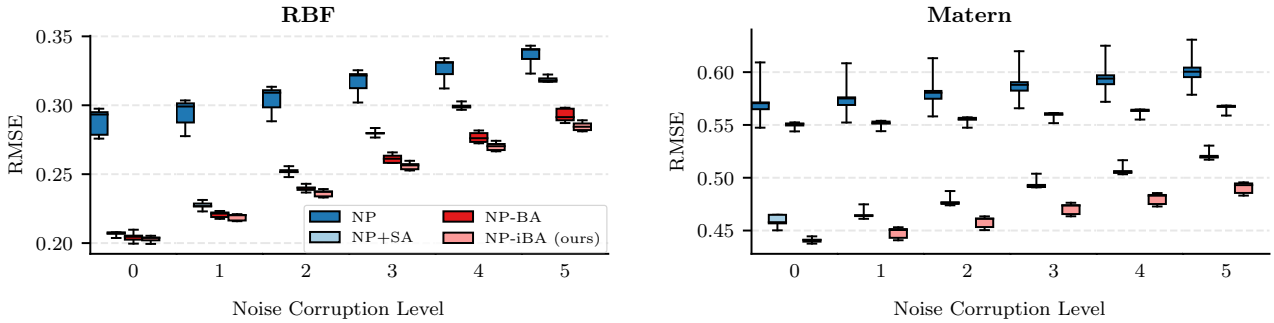
Now the approximating distribution $\tilde{q}(\mathbf{z})$ depends implicitly on the neural sufficient statistics and prior parameters. While this approach would be generally applicable, we cannot see a viable implementation that preserves the core motivations to use NP—namely, fast adaptation. Every forward propagation through the NP would require solving the novel stochastic optimization problem that is maximizing Eq. (34). Thus, we leave solving the fully non-conjugate case to future work.

5 EXPERIMENTS

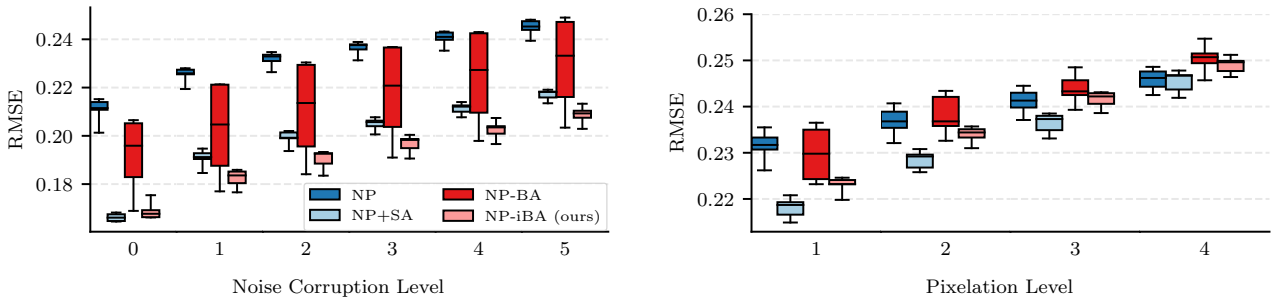
We conduct experiments to assess the performance of our *mixture* and *iterative* variants of Bayesian Aggregation (BA) against vanilla BA [Volpp et al., 2020]. We also benchmark against NPs with regular encoder architecture (NP) [Garnelo et al., 2018b] and an NP with self-attention module (NP+SA) [Kim et al., 2019]. Similar to Volpp et al. [2020], we only evaluate on NP-based models with a single latent path (i.e. no deterministic path). We do not consider encoder architectures that evaluate task-specific contextual representations such as Attentive NP [Kim et al., 2019] and leave the combination of BA variants with cross-attention style mechanisms to future work. The decoder network architecture is fixed across all models such that any performance differences can be attributed solely to the encoder architecture. To reduce the effects of differences in network architecture between BA and non-BA encoders, we keep the BA encoder size close to but less that of the non-BA architectures. Each model is trained with 5 different seeds and we report the mean and standard deviations from these runs. Please refer to the appendix for further details on the model specification and training configuration.

5.1 TASKS: MODELLING FUNCTIONS

1-D Regression We train on functional samples drawn from Gaussian Process (GP) priors with RBF and Matern-5/2 kernel. The specification of the RBF kernel is taken from Lee et al. [2020] where both the lengthscale and output vari-



(a) 1D regression with functional samples from GP kernel.



(b) 2D image completion on EMNIST dataset.

Figure 3: *iterative Bayesian Aggregation (iBA).* We demonstrate gains to test-time robustness in the presence of corruptions to the context set as compared with vanilla BA. In (a), context set function values are perturbed by heavy-tail noise of varying magnitude. In (b), we consider common image noise perturbations (left) and image pixelation (right), of varying intensity. Except for pixelation, NP-iBA shows improved robustness over NP with Self-Attention (NP+SA).

ance is varied (GP hyperparameters). The matern-5/2 kernel is taken from Gordon et al. [2020] where the hyperparameters are fixed.

Image completion A subset of pixels for a given image is presented with the task to predict the remaining pixels. Therefore, we can understand the inputs to be coordinates of each pixel and the pixel intensities representing the targets (i.e. 2-D regression). We consider the EMNIST dataset [Cohen et al., 2017] – a dataset of handwritten digits and letters comprising of grayscale 28x28 images. During training, we restrict images to the first 10 classes. We evaluate on two settings: in-distribution data (held-out images from the first 10 classes) and out-of-distribution (OOD) data (images from the remaining 37 classes).

5.2 MIXTURE BAYESIAN AGGREGATION

We initialize the mixture prior in the PGM as follows: the prior mixing proportions are set to uniform, the prior variance is set to 1 and the prior means are sampled from a zero-mean Gaussian with standard deviation 0.1. The latter setting is to encourage diversity in the posterior mixture components and prevent collapse to a unimodal Gaussian. The task of image completion on the EMNIST dataset is considered. Table 1 clearly demonstrates the utility of additional mixing components with improved function modelling as the number of components is increased from 2 to 5. This is also shown in dependence of the context set size in Fig. 4. The reconstruction error is also indicated ("context"). On OOD data, NP-mBA even exceeds the performance of NP with Self-Attention (NP+SA) for all numbers of components considered. This is despite NP+SA having a considerably larger number of parameters (see Table 2). We also observe little change in the run-time when the number of components is increased.

5.3 ITERATIVE BAYESIAN AGGREGATION

We demonstrate the improved robustness of our iterative Bayesian Aggregation (NP-iBA) to corruptions in the context sets at test-time. For the 1D regression task, we extend the *model-data* mismatch setting from Lee et al. [2020] where the function values are corrupted by Student's t noise of increasing magnitude, $\varepsilon \sim \gamma \cdot \mathcal{T}(2.1)$ with $\gamma \in [0.05, 0.08, 0.11, 0.13, 0.15]$. For the image completion setting, we only consider the in-distribution data and perform noise corruption and pixelation as specified in [Hendrycks

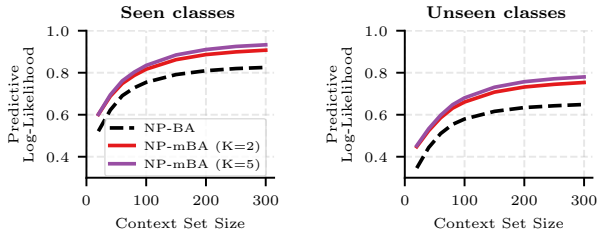


Figure 4: *mixture Bayesian Aggregation (mBA)*. Across a range of context set sizes, there is a gain in performance over vanilla Bayesian Aggregation. This is demonstrated on the task of 2D image completion (EMNIST). Error bars are omitted for visual clarity.

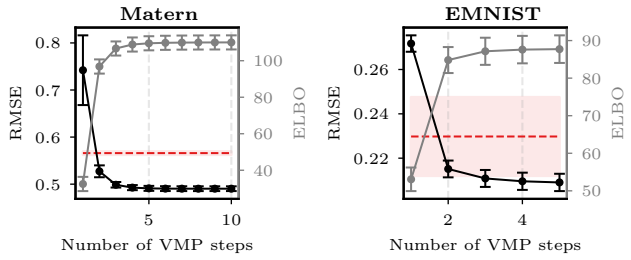


Figure 5: *Context aggregation via variational message-passing*. By increasing the number of VMP steps, we observe a monotonic improvement in both the accuracy of the target function (RMSE —●—) and on the lower bound to the evidence of the PGM (ELBO —●—). With just 2 steps, there is a gain over vanilla Bayesian Aggregation (- -). This is shown for test-time corruption to the context set of the highest intensity. During training, a fixed number of steps is used (10 and 5 for Matern and EMNIST respectively).

and Dietterich]. For the noise corruption, we average over four different types of noise, namely Gaussian, Shot, Poisson and Impulse noise, of increasing severity. Pixelation involves downsampling the image to different reduced resolutions and then upsampling back to the original resolution. For our NP-iBA, we run the message passing algorithm for 5 steps (at both train and test time) except for Matern where we run for 10 steps.

In Fig. 3, we observe a consistent gain over vanilla BA across the different tasks and corruptions. With the exception of corruption by pixelation, NP-iBA also shows improvement over NP with Self-Attention (NP+SA). Fig. 5 demonstrates speed-accuracy trade-off where increasing the number of steps at test-time leads to monotonic improvement in robustness.

Table 2: Number of model parameters and training time (s). Run-time is measured on the image completion experiment on a GTX 1080, for a single epoch (batch size 100). NP-mBA is run with 5 components and NP-iBA with 5 VMP steps.

	# Parameters	Training (s)
NP	248,962	8.1
NP+SA	282114	11.2
NP-BA		10.0
NP-mBA	166,914	16.3 (*)
NP-iBA		16.1

(*) 10 latent samples

6 CONCLUSION

We propose structured inference networks for context aggregation in Neural Processes (NPs). This change is attractive for several reasons: (i) the local encodings now have a clear interpretation as neural sufficient statistics, (ii) the aggregation step is predetermined by and follows from the probabilistic assumptions, and (iii) structured priors are straightforward to incorporate. We show that an existing context aggregation mechanism, Bayesian Aggregation (BA), is recovered by imposing Gaussianity assumptions in the PGM. By imposing different modelling assumptions, we demonstrate alternative context aggregation mechanisms can be derived. In particular, we consider, (1) Mixture of Gaussian prior and (2) Student's t assumptions, which give rise to two novel variants of BA, namely *mixture* and *iterative* Bayesian Aggregation. We demonstrate improvements to the functional modelling and test-time robustness of NPs without any increase in the parameterization of the encoder. For future work, we look to consider more general PGM structure and address the fully non-conjugate setting.

An implementation of the methods presented in this paper will be made available under the MIT License on GitHub upon acceptance.

References

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *The Journal of Machine Learning Research*, 21(1):3535–3595, 2020.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

Harrison Edwards and Amos Storkey. Towards a neural statistician. In *International Conference on Learning Representations*, 2017.

Daniel Flam-Shepherd, Yuxiang Gao, and Zhaoyu Guo. Stick-breaking neural latent variable models. *NeurIPS Workshop on Bayesian Deep Learning*, 2018.

Andrew YK Foong, Wessel P Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard E Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. *arXiv preprint arXiv:2007.01332*, 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, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=Skey4eBYPS`.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

TM Heskes. Empirical bayes for learning to learn. 2000.

Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29, 2016.

Makoto Kawano, Wataru Kumagai, Akiyoshi Sannai, Yusuke Iwasawa, and Yutaka Matsuo. Group equivariant conditional neural processes. In *International Conference on Learning Representations*, 2021.

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

Mingyu Kim, Kyeong Ryeol Go, and Se-Young Yun. Neural processes with stochastic attention: Paying more attention to the context dataset. In *International Conference on Learning Representations*, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Juho Lee, Yoonho Lee, Jungtaek Kim, Eunho Yang, Sung Ju Hwang, and Yee Whye Teh. Bootstrapping neural processes. *arXiv preprint arXiv:2008.02956*, 2020.

Wu Lin, Nicolas Hubacher, and Mohammad Emtiyaz Khan. Variational message passing with structured inference networks. In *International Conference on Learning Representations*, 2018.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *International Conference on Machine Learning*, pages 16569–16594. PMLR, 2022.

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 E Tipping and Neil D Lawrence. Variational inference for student-t models: Robust bayesian interpolation and generalised component analysis. *Neurocomputing*, 69(1-3):123–141, 2005.

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

Edward Wagstaff, Fabian Fuchs, Martin Engelcke, Ingmar Posner, and Michael A Osborne. On the limitations of representing functions on sets. In *International Conference on Machine Learning*, pages 6487–6494. PMLR, 2019.

Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2): 1–305, 2008.

Qi Wang and Herke Van Hoof. Doubly stochastic variational inference for neural processes with hierarchical latent variables. In *International Conference on Machine Learning*, pages 10018–10028. PMLR, 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.

John Winn, Christopher M Bishop, and Tommi Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6(4), 2005.

Hao Wu, Heiko Zimmermann, Eli Sennesh, Tuan Anh Le, and Jan-Willem Van De Meent. Amortized population Gibbs samplers with neural sufficient statistics. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10421–10431. PMLR, 2020.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.