uai2023

Exploiting Inferential Structure in Neural Processes

Dharmesh Tailor[1], Emtiyaz Khan[2], Eric Nalisnick[1]

[1]University of Amsterdam (Netherlands) [2]RIKEN AIP (Japan)

University of Amsterdam

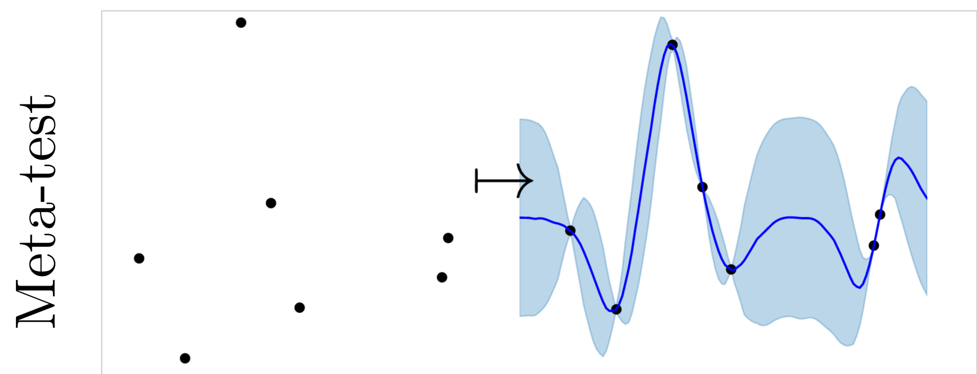
RIKEN AIP

We enrich the latent variable of Neural Processes with **structured priors** (e.g. with multiple modes, heavy-tails, *etc.*) and provide a framework that directly translates such distributional assumptions into an aggregation strategy for the context set.

Neural Processes as Meta-Learning Approximate Inference

Taken from: https://yanndubs.github.io/Neural-Process-Family/

Train on many, small datasets with **context-target** splits

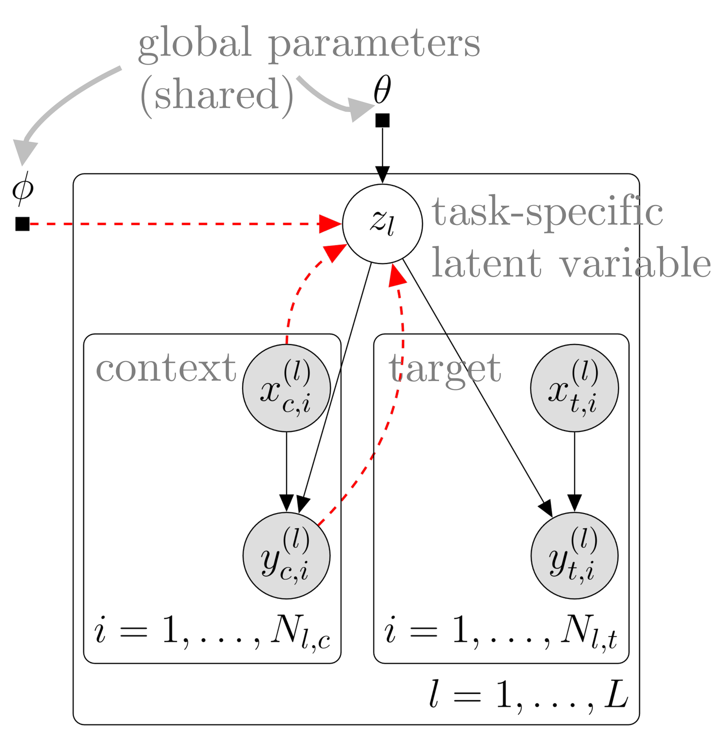


Rapidly adapt to new tasks at test-time given context set

- Seek to learn an approximate distribution over task-specific variables (via **amortization**) which gives rise to a posterior predictive

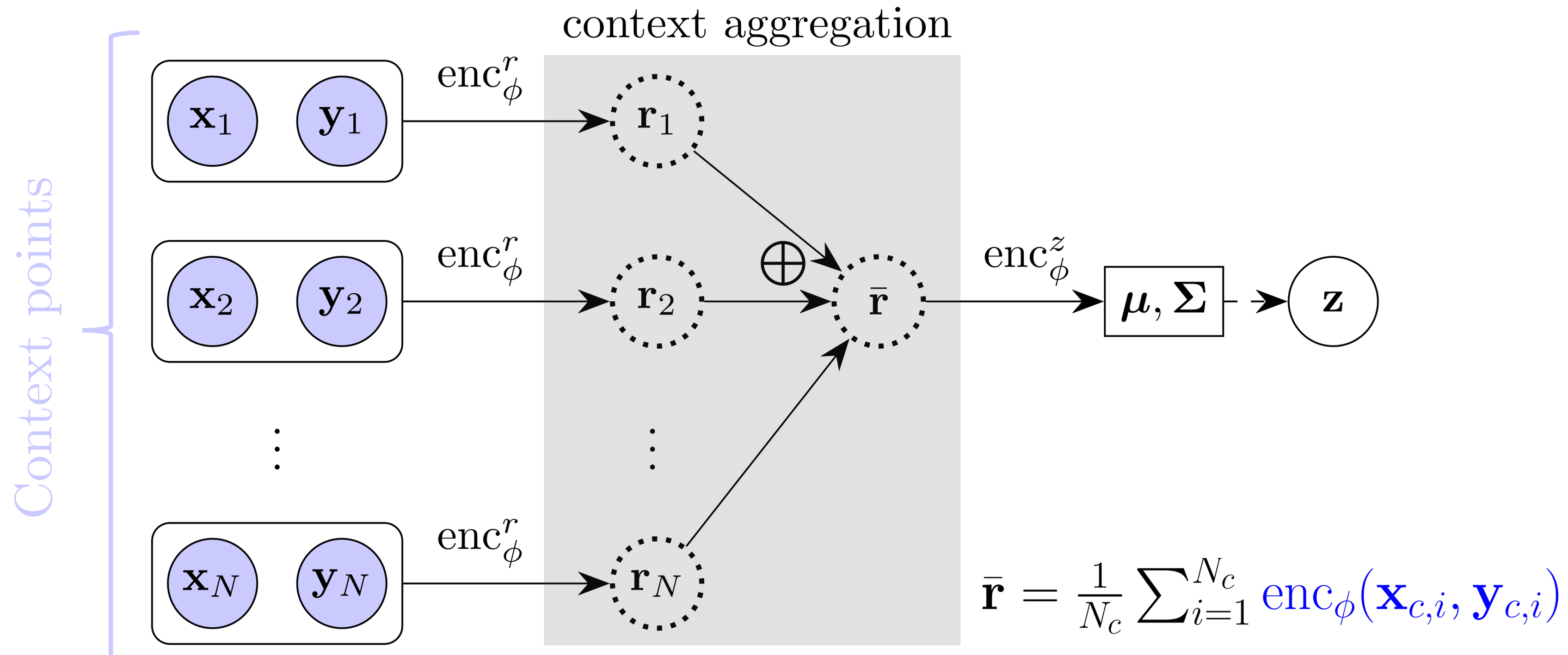
$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathcal{D}_c)}\left[p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{z})\right]$$

- Train all parameters end-to-end using lower-bound to conditional marginal likelihood across all tasks.



Standard multi-task LVM

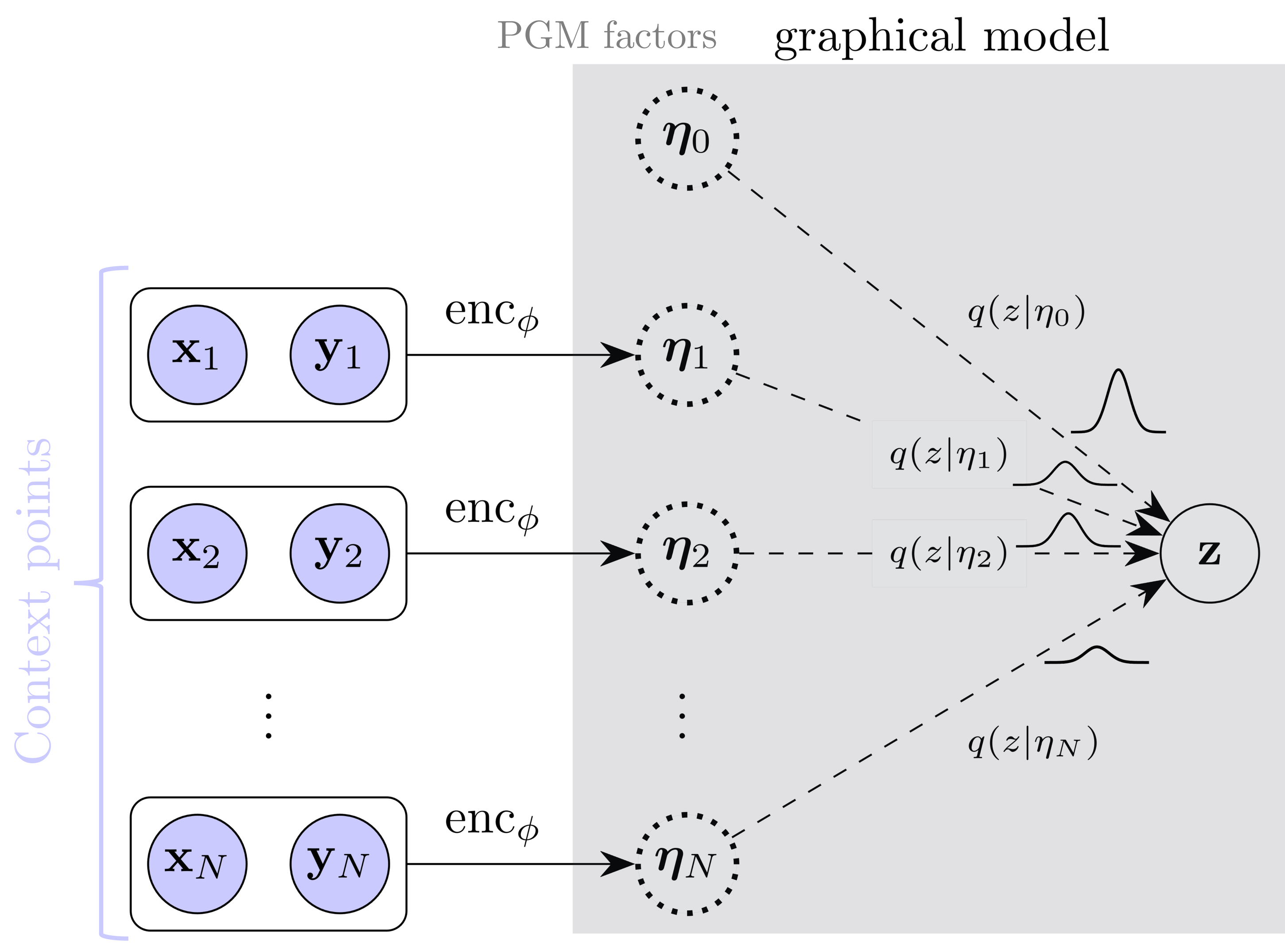
Sum-Decomposition Network



$$\bar{\mathbf{r}}=\frac{1}{N_c}\sum_{i=1}^{N_c}\text{enc}_\phi(\mathbf{x}_{c,i},\mathbf{y}_{c,i})$$

1. NN amortized wrt context points
2. Permutation-invariant aggregation operation
3. Additional network to map to variational parameters

Structured Inference Network



1. Reframe context embeddings as **neural sufficient statistics**, which is aggregated alongside prior
2. Aggregation operation directly follows from distributional assumptions
3. No need for additional network since aggregation occurs directly in space of latent variables

$$q_\phi(\mathbf{z}\,|\,\mathcal{D}_c)=\frac{1}{Z_c(\phi)}\underbrace{\left[\prod_{i=1}^{N_c}q(\mathbf{z}\,|\,\text{enc}_\phi(\mathbf{x}_{c,i},\mathbf{y}_{c,i}))\right]}_{\text{NN factors}}\times\underbrace{\left[q(\mathbf{z};\phi_0)\right]}_{\text{Prior}}$$

- Conjugate case:

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

- Also extends to the non-conjugate setting by maximizing the lower-bound

$$\log Z_c(\phi)\geq\mathbb{E}_{\tilde{q}(\mathbf{z})}\left[\log q_\phi(\mathbf{z},\mathcal{D}_c)\right]+\mathcal{H}\left(\tilde{q}(\mathbf{z})\right)$$

Bayesian Context Aggregation

I. Gaussian prior → Bayesian Aggregation *[Volpp et. al., 2020]*

PGM $q_\phi(\mathbf{z}\,|\,\mathcal{D}_c)=\frac{1}{Z_c(\phi)}\left[\prod_{i=1}^{N_c}\mathbb{N}(\mathbf{z}\,|\,\mathbf{m}_{c,i},\mathbf{V}_{c,i})\right]\times\left[\mathbb{N}(\mathbf{z}\,|\,\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)\right]$

Aggregation = Conjugate-computations:

$$\tilde{\boldsymbol{\Sigma}}^{-1}=\sum_{i=1}^{N_c}\mathbf{V}_{c,i}^{-1}+\boldsymbol{\Sigma}_0^{-1}$$

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

Weighted aggregation

II. Mixture of Gaussian prior → *Mixture* Bayesian Aggregation

PGM: $q_\phi(\mathbf{z}\,|\,\mathcal{D}_c)=\frac{1}{Z_c(\phi)}\left[\prod_{i=1}^{N_c}\mathbb{N}(\mathbf{z}\,|\,\mathbf{m}_{c,i},\mathbf{V}_{c,i})\right]\times\left[\sum_{k=1}^{K}\pi_k\mathbb{N}(\mathbf{z}\,|\,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right]$

Aggregation = Conjugate-computations:

$$\tilde{\boldsymbol{\Sigma}}_k^{-1}=\sum_{i=1}^{N_c}\mathbf{V}_{c,i}^{-1}+\boldsymbol{\Sigma}_k^{-1}$$

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

$$\tilde{\pi}_k=\pi_k\,C_k\,/\,Z_c$$

III. Heavy-tail assumptions → *Robust* Bayesian Aggregation

PGM: $q_\phi(\mathbf{z},\alpha,\boldsymbol{\beta})=\left[\prod_{i=1}^{N_c}\mathbb{N}(\mathbf{z}|\mathbf{m}_{c,i},\beta_i^{-1}\mathbf{V}_{c,i})\right]\times\left[\mathbb{N}(\mathbf{z}|\mathbf{0},\alpha^{-1}\mathbf{I})\mathcal{G}(\alpha|a_0,b_0)\prod_{i=1}^{N_c}\mathcal{G}(\beta_i|c_0,c_0)\right]$

Aggregation = Coordinate-ascent VI:

$$\tilde{\boldsymbol{\Sigma}}^{-1}=\sum_{i=1}^{N_c}\mathbb{E}[\beta_i]\mathbf{V}_{c,i}^{-1}+\mathbb{E}[\alpha]\mathbf{I}$$

$$\tilde{\boldsymbol{\mu}}=\tilde{\boldsymbol{\Sigma}}\sum_{i=1}^{N_c}\mathbb{E}[\beta_i]\mathbf{V}_{c,i}^{-1}\mathbf{m}_{c,i},$$

$$\mathbb{E}[\alpha]=\left(a_0+\tfrac{D}{2}\right)/\left(b_0+\tfrac{1}{2}\text{tr}(\tilde{\boldsymbol{\mu}}\tilde{\boldsymbol{\mu}}^\top+\tilde{\boldsymbol{\Sigma}})\right)$$

$$\mathbb{E}[\beta_i]=\left(c_0+\tfrac{D}{2}\right)/\left(c_0+\tfrac{1}{2}\left(\mathbf{m}_{c,i}^\top\mathbf{V}_{c,i}^{-1}\mathbf{m}_{c,i}-2\mathbf{m}_{c,i}^\top\mathbf{V}_{c,i}^{-1}\tilde{\boldsymbol{\mu}}+\text{tr}\left(\mathbf{V}_{c,i}^{-1}(\tilde{\boldsymbol{\mu}}\tilde{\boldsymbol{\mu}}^\top+\tilde{\boldsymbol{\Sigma}})\right)\right)\right)$$

- Coordinate-wise updates remain fully-differentiable and we backprop through the unrolled steps for gradient-based learning

Experiments

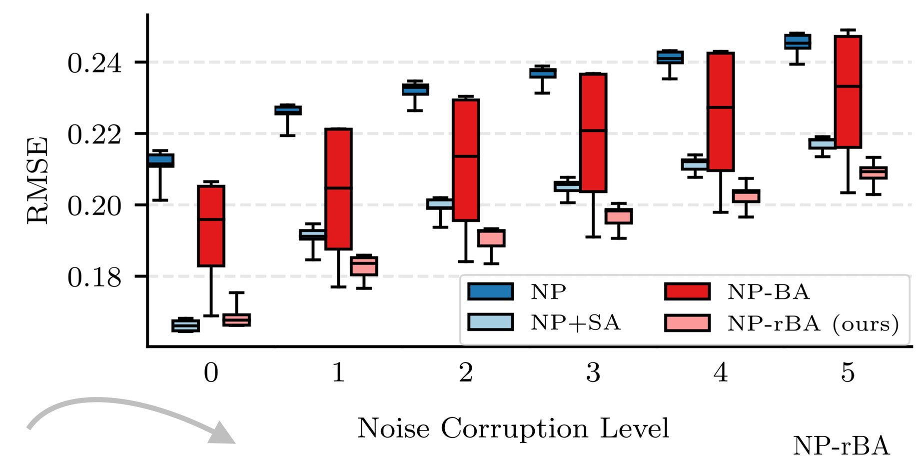
- We compare against NP-based models with a single latent path (i.e. no deterministic path) and without task-specific contextual representations (e.g. ANP)

Self-attention

	RMSE ↓ Seen classes (0-9)		Unseen classes (10-46)	
	context	target	context	target
NP	0.201±0.018	0.218±0.014	0.244±0.014	0.265±0.009
NP+SA	0.127±0.002	**0.165**±0.001	0.177±0.002	0.224±0.002
NP-BA	0.154±0.027	0.193±0.014	0.193±0.033	0.238±0.018
NP-mBA (K=2)	0.128±0.033	0.181±0.017	0.162±0.038	0.221±0.020
NP-mBA (K=3)	0.128±0.031	0.180±0.015	0.162±0.038	0.221±0.020
NP-mBA (K=5)	**0.122**±0.025	0.177±0.011	**0.155**±0.031	**0.217**±0.016

(Image completion – EMNIST)

Boost with increasing number of mixture components



Pixel corruptions to context set at test-time