Bayesian Context Aggregation for Neural Processes

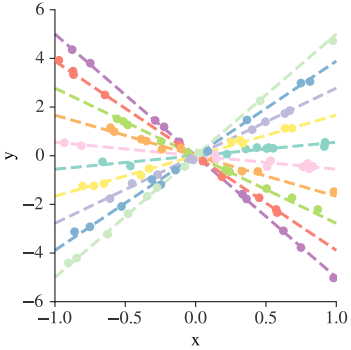
ICLR 2021

Michael Volpp[1,2], Fabian Flürenbrock[1], Lukas Grossberger[1], Christian Daniel[1], Gerhard Neumann[2]

[1]Bosch Center for Artificial Intelligence, Renningen, Germany
[2]Karlsruhe Institute of Technology, Karlsruhe, Germany

Probabilistic Regression as a Multi-task Learning Problem

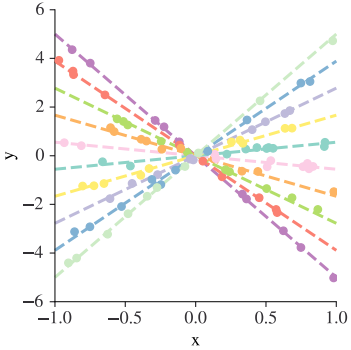
- Family $\mathcal{F}$ of functions $f_\ell : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ ("tasks") with some form of shared structure



ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*

Probabilistic Regression as a Multi-task Learning Problem

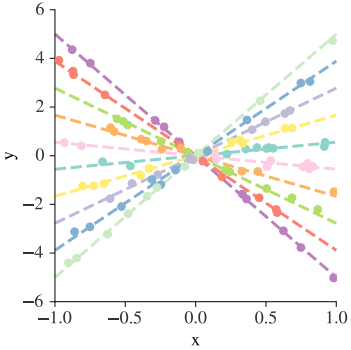
- Family $\mathcal{F}$ of functions $f_\ell : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ ("tasks") with some form of shared structure
- Noisy evaluations $\mathcal{D}_\ell = \{(x_{\ell,i}, y_{\ell,i})\}_i$ with $y_{\ell,i} = f_\ell(x_{\ell,i}) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$



ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*

Probabilistic Regression as a Multi-task Learning Problem

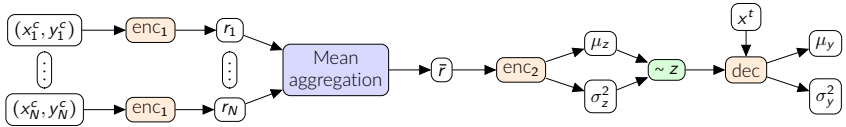
- Family $\mathcal{F}$ of functions $f_\ell : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ ("tasks") with some form of shared structure
- Noisy evaluations $\mathcal{D}_\ell = \{(x_{\ell,i}, y_{\ell,i})\}_i$ with $y_{\ell,i} = f_\ell(x_{\ell,i}) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$
- Learn the predictive distribution $p(y_\ell^t | x_\ell^t, \mathcal{D}_\ell^c)$ over target outputs, conditioned on a context set $\mathcal{D}_\ell^c \subset \mathcal{D}_\ell$



ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*

The Neural Process[1]

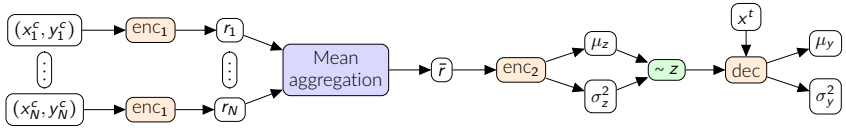
Neural network (NN)-based multi-task learning architecture



[1] Garnelo et al., *Conditional Neural Processes*, ICML 2018, Garnelo et al. *Neural Processes*. arxiv/1807.01622 2018

The Neural Process[1]

Neural network (NN)-based multi-task learning architecture

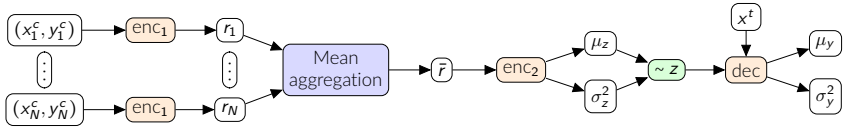


1. Infer latent representation $p(z|\mathcal{D}^c)$ of the target function from $\mathcal{D}^c = \{(x_n^c, y_n^c)\}_{n=1}^N$

[1] Garnelo et al., *Conditional Neural Processes*, ICML 2018, Garnelo et al. *Neural Processes*. arxiv/1807.01622 2018

The Neural Process[1]

Neural network (NN)-based multi-task learning architecture

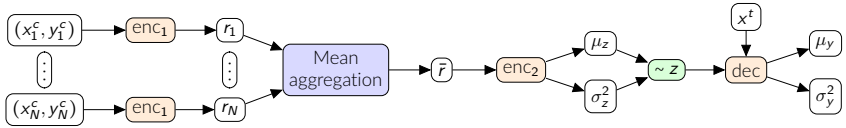


1. Infer latent representation $p(z|\mathcal{D}^c)$ of the target function from $\mathcal{D}^c = \{(x_n^c, y_n^c)\}_{n=1}^N$
   a. Map each context tuple (x_n^c, y_n^c) onto a latent observation $r_n = \text{enc}_1(x_n^c, y_n^c)$

[1] Garnelo et al., *Conditional Neural Processes*, ICML 2018, Garnelo et al. *Neural Processes*. arxiv/1807.01622 2018

The Neural Process[1]

Neural network (NN)-based multi-task learning architecture

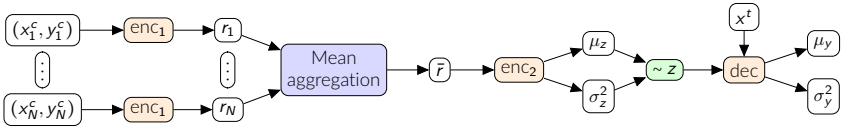


1. Infer latent representation $p(z|\mathcal{D}^c)$ of the target function from $\mathcal{D}^c = \{(x_n^c, y_n^c)\}_{n=1}^N$
   a. Map each context tuple (x_n^c, y_n^c) onto a latent observation $r_n = \text{enc}_1(x_n^c, y_n^c)$
   b. Form an aggregated latent observation $\bar{r}$ using *mean aggregation*: $\bar{r} = \frac{1}{N}\sum_{n=1}^{N} r_n$

[1] Garnelo et al., *Conditional Neural Processes*, ICML 2018, Garnelo et al. *Neural Processes*. arxiv/1807.01622 2018

The Neural Process[1]

Neural network (NN)-based multi-task learning architecture

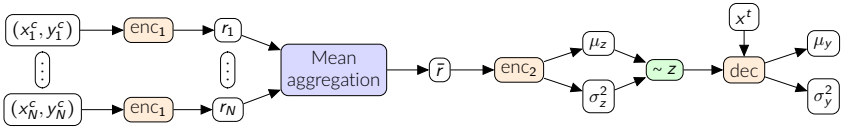


1. Infer latent representation $p(z|\mathcal{D}^c)$ of the target function from $\mathcal{D}^c = \{(x_n^c, y_n^c)\}_{n=1}^N$
   a. Map each context tuple (x_n^c, y_n^c) onto a latent observation $r_n = \text{enc}_1(x_n^c, y_n^c)$
   b. Form an aggregated latent observation $\bar{r}$ using *mean aggregation*: $\bar{r} = \frac{1}{N}\sum_{n=1}^{N} r_n$
   c. Map $\bar{r}$ onto the parameters of the latent distribution: $(\mu_z, \sigma_z^2) = \text{enc}_2(\bar{r})$

[1] Garnelo et al., *Conditional Neural Processes*, ICML 2018, Garnelo et al. *Neural Processes*. arxiv/1807.01622 2018

The Neural Process[1]

Neural network (NN)-based multi-task learning architecture

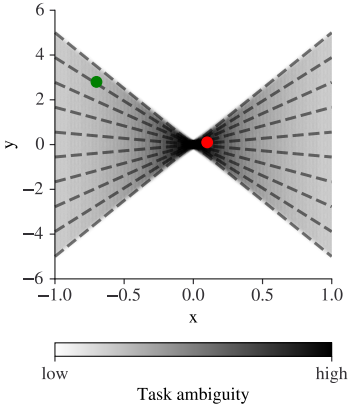


1. Infer latent representation $p(z|\mathcal{D}^c)$ of the target function from $\mathcal{D}^c = \{(x_n^c, y_n^c)\}_{n=1}^N$
   a. Map each context tuple (x_n^c, y_n^c) onto a latent observation $r_n = \text{enc}_1(x_n^c, y_n^c)$
   b. Form an aggregated latent observation $\bar{r}$ using *mean aggregation*: $\bar{r} = \frac{1}{N}\sum_{n=1}^{N} r_n$
   c. Map $\bar{r}$ onto the parameters of the latent distribution: $(\mu_z, \sigma_z^2) = \text{enc}_2(\bar{r})$
2. Map samples $z \sim p(z|\mathcal{D}^c)$ onto a Gaussian output distribution: $(\mu_y, \sigma_y^2) = \text{dec}(z, x^t)$

[1] Garnelo et al., *Conditional Neural Processes*, ICML 2018, Garnelo et al. *Neural Processes*. arxiv/1807.01622 2018

Context Aggregation and Task Ambiguity

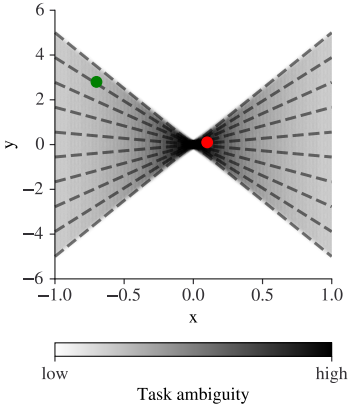
- Different areas in the (x, y)-space can have different *task ambiguity* (TA)



ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*

Context Aggregation and Task Ambiguity

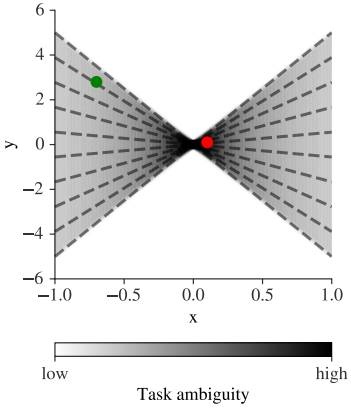
- Different areas in the (x, y)-space can have different *task ambiguity* (TA)
- Context tuples (x_n^c, y_n^c) with
  - high TA should have little influence on $p(z|\mathcal{D}^c)$
  - low TA should have large influence on $p(z|\mathcal{D}^c)$



ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*

Context Aggregation and Task Ambiguity

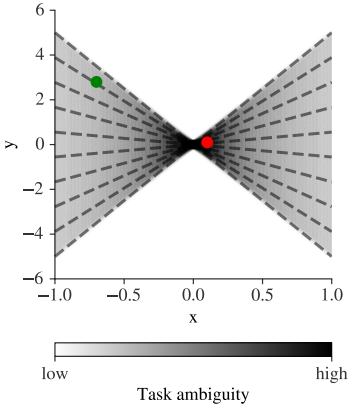
- Different areas in the (x, y)-space can have different *task ambiguity* (TA)
- Context tuples (x_n^c, y_n^c) with
  - high TA should have little influence on $p(z|\mathcal{D}^c)$
  - low TA should have large influence on $p(z|\mathcal{D}^c)$
- Mean aggregation assigns the same weight to all context tuples: $\bar{r} = \frac{1}{N} \sum_{n=1}^{N} r_n$



ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*

Context Aggregation and Task Ambiguity

- Different areas in the (x, y)-space can have different *task ambiguity* (TA)
- Context tuples (x_n^c, y_n^c) with
  - high TA should have little influence on $p(z|\mathcal{D}^c)$
  - low TA should have large influence on $p(z|\mathcal{D}^c)$
- Mean aggregation assigns the same weight to all context tuples: $\bar{r} = \frac{1}{N} \sum_{n=1}^{N} r_n$



How to efficiently incorporate task ambiguity into NP parameter inference?

ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*

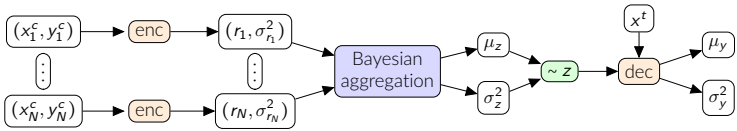
Bayesian Context Aggregation

Context aggregation and parameter inference should be treated as one holistic mechanism!
Directly aggregate the context data into the statistical description of $\boldsymbol{z}$*!*

Bayesian Context Aggregation

Context aggregation and parameter inference should be treated as one holistic mechanism!
Directly aggregate the context data into the statistical description of z*!*

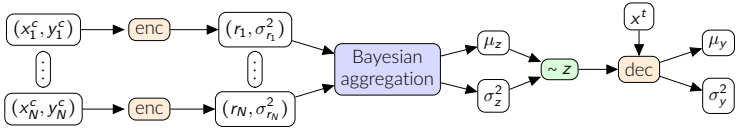
- NP with our *Bayesian aggregation (BA)*:



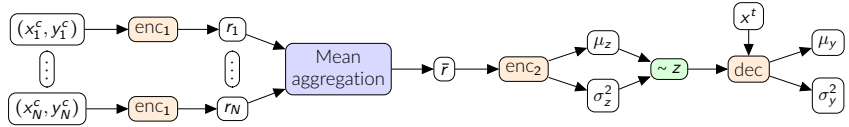
Bayesian Context Aggregation

Context aggregation and parameter inference should be treated as one holistic mechanism!
Directly aggregate the context data into the statistical description of z*!*

- NP with our *Bayesian aggregation (BA)*:

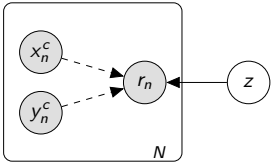


- Compare: NP with traditional mean aggregation (MA):



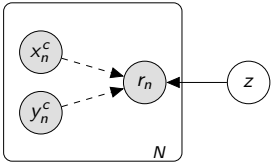
Bayesian Context Aggregation

- Context aggregation as Bayesian inference



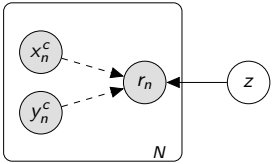
Bayesian Context Aggregation

- Context aggregation as Bayesian inference
- Observation model: $p(r_n|z) = \mathcal{N}\left(r_n|z, \text{diag}\left(\sigma^2_{r_n}\right)\right)$



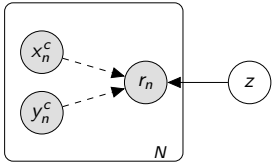
Bayesian Context Aggregation

- Context aggregation as Bayesian inference
- Observation model: $p(r_n|z) = \mathcal{N}\left(r_n|z, \text{diag}\left(\sigma^2_{r_n}\right)\right)$
- Encoder learns: $(r_n, \sigma^2_{r_n}) = \text{enc}(x^c_n, y^c_n)$



Bayesian Context Aggregation

- Context aggregation as Bayesian inference
- Observation model: $p(r_n|z) = \mathcal{N}\left(r_n|z, \text{diag}\left(\sigma^2_{r_n}\right)\right)$
- Encoder learns: $(r_n, \sigma^2_{r_n}) = \text{enc}(x^c_n, y^c_n)$
- Latent posterior: $p(z|\{r_n\}) = \mathcal{N}(z|\mu_z, \text{diag}(\sigma^2_z))$

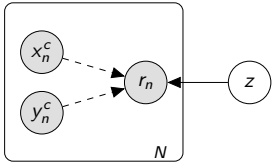


$$\sigma^2_z = \left[\sum_{n=1}^{N}\left(\sigma^2_{r_n}\right)^{-1}\right]^{-1}, \quad \mu_z = \sum_{n=1}^{N}\frac{\sigma^2_z}{\sigma^2_{r_n}} r_n \qquad \text{(for each latent dim.)}$$

ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*

Bayesian Context Aggregation

- Context aggregation as Bayesian inference
- Observation model: $p(r_n|z) = \mathcal{N}\left(r_n|z, \text{diag}\left(\sigma^2_{r_n}\right)\right)$
- Encoder learns: $(r_n, \sigma^2_{r_n}) = \text{enc}\,(x^c_n, y^c_n)$
- Latent posterior: $p(z\,|\,\{r_n\}) = \mathcal{N}(z|\mu_z, \text{diag}(\sigma^2_z))$

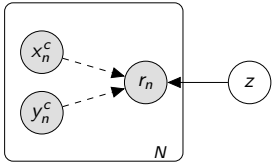


$$\sigma^2_z = \left[\sum_{n=1}^{N}\left(\sigma^2_{r_n}\right)^{-1}\right]^{-1}, \quad \mu_z = \sum_{n=1}^{N}\frac{\sigma^2_z}{\sigma^2_{r_n}} r_n \qquad \text{(for each latent dim.)}$$

- r_n enters μ_z with learned weight $\sigma^2_z/\sigma^2_{r_n}$

Bayesian Context Aggregation

- Context aggregation as Bayesian inference
- Observation model: $p(r_n|z) = \mathcal{N}\left(r_n|z, \text{diag}\left(\sigma^2_{r_n}\right)\right)$
- Encoder learns: $(r_n, \sigma^2_{r_n}) = \text{enc}(x^c_n, y^c_n)$
- Latent posterior: $p(z|\{r_n\}) = \mathcal{N}(z|\mu_z, \text{diag}(\sigma^2_z))$

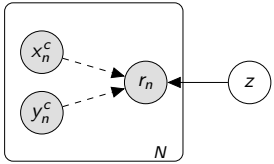


$$\sigma^2_z = \left[\sum_{n=1}^{N}\left(\sigma^2_{r_n}\right)^{-1}\right]^{-1}, \quad \mu_z = \sum_{n=1}^{N}\frac{\sigma^2_z}{\sigma^2_{r_n}} r_n \qquad \text{(for each latent dim.)}$$

- r_n enters μ_z with learned weight $\sigma^2_z/\sigma^2_{r_n}$
- Principled quantification of task ambiguity

Bayesian Context Aggregation

- Context aggregation as Bayesian inference
- Observation model: $p(r_n|z) = \mathcal{N}\left(r_n|z, \text{diag}\left(\sigma^2_{r_n}\right)\right)$
- Encoder learns: $(r_n, \sigma^2_{r_n}) = \text{enc}(x^c_n, y^c_n)$
- Latent posterior: $p(z|\{r_n\}) = \mathcal{N}(z|\mu_z, \text{diag}(\sigma^2_z))$

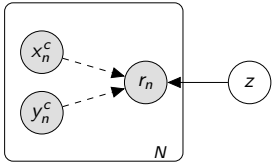


$$\sigma^2_z = \left[\sum_{n=1}^{N}\left(\sigma^2_{r_n}\right)^{-1}\right]^{-1}, \quad \mu_z = \sum_{n=1}^{N}\frac{\sigma^2_z}{\sigma^2_{r_n}} r_n \qquad \text{(for each latent dim.)}$$

- r_n enters μ_z with learned weight $\sigma^2_z/\sigma^2_{r_n}$
- Principled quantification of task ambiguity
- Only marginal computational overhead

Bayesian Context Aggregation

- Context aggregation as Bayesian inference
- Observation model: $p(r_n|z) = \mathcal{N}\left(r_n|z, \text{diag}\left(\sigma^2_{r_n}\right)\right)$
- Encoder learns: $(r_n, \sigma^2_{r_n}) = \text{enc}(x^c_n, y^c_n)$
- Latent posterior: $p(z|\{r_n\}) = \mathcal{N}(z|\mu_z, \text{diag}(\sigma^2_z))$

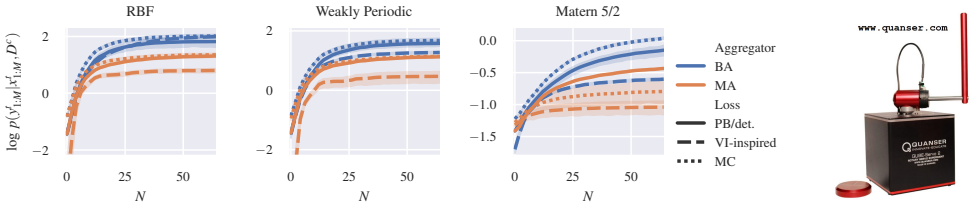


$$\sigma^2_z = \left[\sum_{n=1}^{N}\left(\sigma^2_{r_n}\right)^{-1}\right]^{-1}, \quad \mu_z = \sum_{n=1}^{N}\frac{\sigma^2_z}{\sigma^2_{r_n}} r_n \qquad \text{(for each latent dim.)}$$

- r_n enters μ_z with learned weight $\sigma^2_z/\sigma^2_{r_n}$
- Principled quantification of task ambiguity
- Only marginal computational overhead
- Compatible with existing NP architectures

Experiments

	PB/det.		VI		MC	
	BA	MA (CNP)	BA	MA (LP-NP)	BA	MA
RBF GP	**1.37 ± 0.15**	0.94 ± 0.04	**1.40 ± 0.04**	0.45 ± 0.12	**1.62 ± 0.05**	1.07 ± 0.05
Weakly Periodic GP	**1.13 ± 0.08**	0.76 ± 0.02	**0.89 ± 0.03**	0.07 ± 0.14	**1.30 ± 0.06**	0.85 ± 0.04
Matern-5/2 GP	**−0.50 ± 0.07**	−0.68 ± 0.01	**−0.79 ± 0.01**	−1.09 ± 0.10	**−0.33 ± 0.01**	−0.90 ± 0.15
Furuta Dynamics	**7.50 ± 0.27**	7.06 ± 0.12	**7.32 ± 0.18**	5.57 ± 0.21	**8.25 ± 0.33**	7.55 ± 0.24



ICLR 2021 | M. Volpp, F. Flürenbrock, L. Grossberger, C. Daniel, G. Neumann, *Bayesian Context Aggregation for Neural Processes*