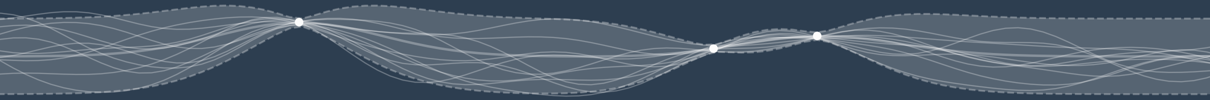


Autoregressive Conditional Neural Processes

Wessel Bruinsma

Microsoft Research AI4Science

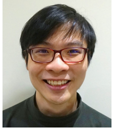
Research Talk
Center for Basic Machine Learning Research in Life Science (MLLS)
Copenhagen, 25 Jan 2024

Collaborators

Wessel Bruinsma[*,1,2]

Stratis Markou[*,2]

James Requeima[*,2]



Andrew Foong[*,1]

Tom Andersson[3]

Anna Vaughan[2]

Anthony Buonomo[2]

J. Scott Hosking[3,4]

Rich Turner[1,2]

[*]Equal contribution

[1]Microsoft Research AI4Science, [2]University of Cambridge, [3]British Antarctic Survey, [4]The Alan Turing Institute

Outline

- Autoregressive Conditional Neural Processes

Wessel P. Bruinsma, Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom R. Andersson, Anna Vaughan, Anthony Buonomo, J. Scott Hosking, and Richard E. Turner (2023). "Autoregressive Conditional Neural Processes". In: *Proceedings of the 11th International Conference on Learning Representations*. eprint: `https://arxiv.org/abs/2303.14468`

`wessel.ai/pdf/arcnps`

Published as a conference paper at ICLR 2023

AUTOREGRESSIVE CONDITIONAL NEURAL PROCESSES

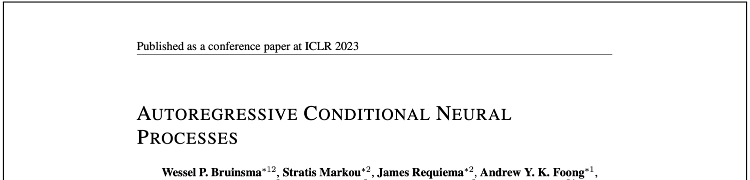
Wessel P. Bruinsma[*12], **Stratis Markou**[*2], **James Requeima**[*2], **Andrew Y. K. Foong**[*1],

Outline

- Introduction to Neural Processes
- Autoregressive Conditional Neural Processes

Wessel P. Bruinsma, Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom R. Andersson, Anna Vaughan, Anthony Buonomo, J. Scott Hosking, and Richard E. Turner (2023). "Autoregressive Conditional Neural Processes". In: *Proceedings of the 11th International Conference on Learning Representations*. eprint: `https://arxiv.org/abs/2303.14468`

`wessel.ai/pdf/arcnps`


Published as a conference paper at ICLR 2023

AUTOREGRESSIVE CONDITIONAL NEURAL PROCESSES

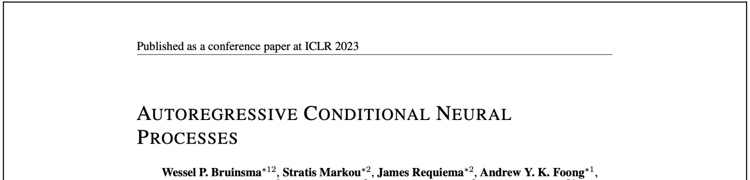
Wessel P. Bruinsma[*12], **Stratis Markou**[*2], **James Requiema**[*2], **Andrew Y. K. Foong**[*1],

Outline

- Introduction to Neural Processes
- Autoregressive Conditional Neural Processes
- Prediction Map Approximation: A Theoretical Analysis

Wessel P. Bruinsma, Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom R. Andersson, Anna Vaughan, Anthony Buonomo, J. Scott Hosking, and Richard E. Turner (2023). "Autoregressive Conditional Neural Processes". In: *Proceedings of the 11th International Conference on Learning Representations*. eprint: https://arxiv.org/abs/2303.14468

`wessel.ai/pdf/arcnps`



Published as a conference paper at ICLR 2023

AUTOREGRESSIVE CONDITIONAL NEURAL PROCESSES

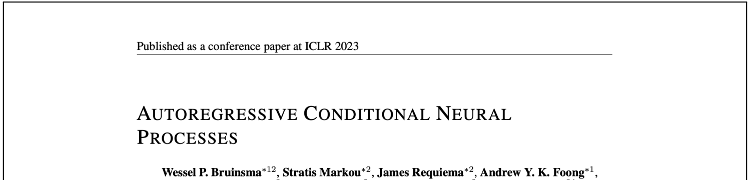
Wessel P. Bruinsma[*12], **Stratis Markou**[*2], **James Requiema**[*2], **Andrew Y. K. Foong**[*1],

Outline

- Introduction to Neural Processes
- Autoregressive Conditional Neural Processes
- Prediction Map Approximation: A Theoretical Analysis
- Conclusion

Wessel P. Bruinsma, Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom R. Andersson, Anna Vaughan, Anthony Buonomo, J. Scott Hosking, and Richard E. Turner (2023). "Autoregressive Conditional Neural Processes". In: *Proceedings of the 11th International Conference on Learning Representations*. eprint: `https://arxiv.org/abs/2303.14468`

`wessel.ai/pdf/arcnps`



Published as a conference paper at ICLR 2023

AUTOREGRESSIVE CONDITIONAL NEURAL PROCESSES

Wessel P. Bruinsma[*12], **Stratis Markou**[*2], **James Requiema**[*2], **Andrew Y. K. Foong**[*1],

Today's Message

- ✗ A specific model
- ✗ Restricted to a meta-learning setting
- ✗ Very unique and new

Today's Message

- ✗ A specific model
- ✗ Restricted to a meta-learning setting
- ✗ Very unique and new

Neural processes:

- a flexible collection of architectural neural network techniques
- for general supervised learning problems.

Today's Message

✗ A specific model

✗ Restricted to a meta-learning setting

✗ Very unique and new

Neural processes:

- a flexible collection of architectural neural network techniques
- for general supervised learning problems.

↑

e.g., multidimensional irregular off-the-grid data

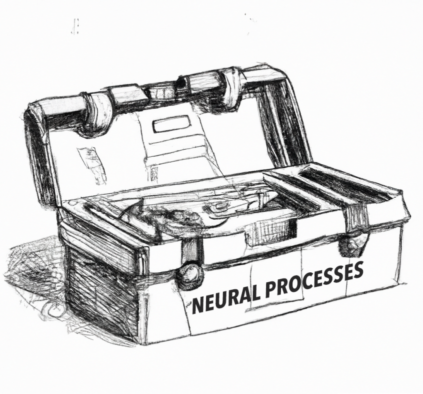
Today's Message

- ✗ A specific model
- ✗ Restricted to a meta-learning setting
- ✗ Very unique and new

Neural processes:

- a flexible collection of architectural neural network techniques
- for general supervised learning problems.

e.g., multidimensional irregular off-the-grid data



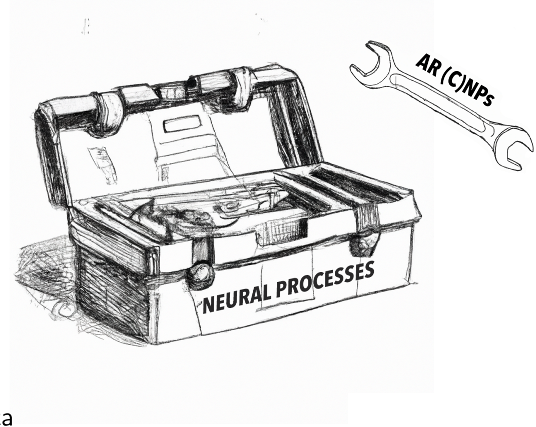
Today's Message

- ✗ A specific model
- ✗ Restricted to a meta-learning setting
- ✗ Very unique and new

Neural processes:

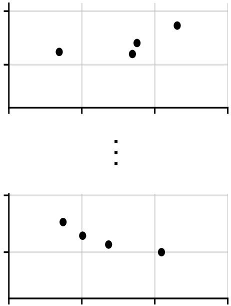
- a flexible collection of architectural neural network techniques
- for general supervised learning problems.

↑ *e.g.*, multidimensional irregular off-the-grid data

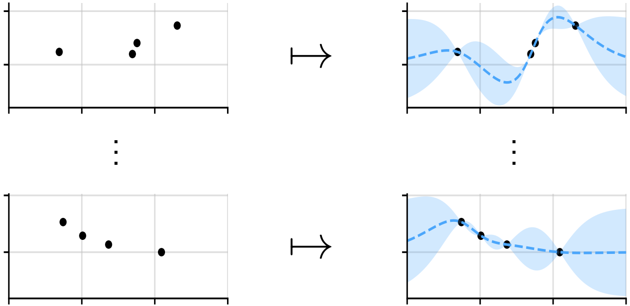


Introduction to Neural Processes

Meta-Learning and Neural Processes: Learning to Predict



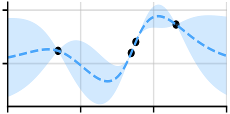
Meta-Learning and Neural Processes: Learning to Predict



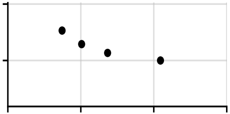
Meta-Learning and Neural Processes: Learning to Predict

$$\pi: \quad \text{data sets } \mathcal{D} \quad \longrightarrow \quad \text{predictions } \mathcal{P}$$

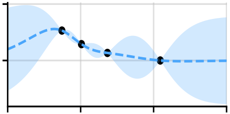
$\stackrel{\pi}{\longmapsto}$



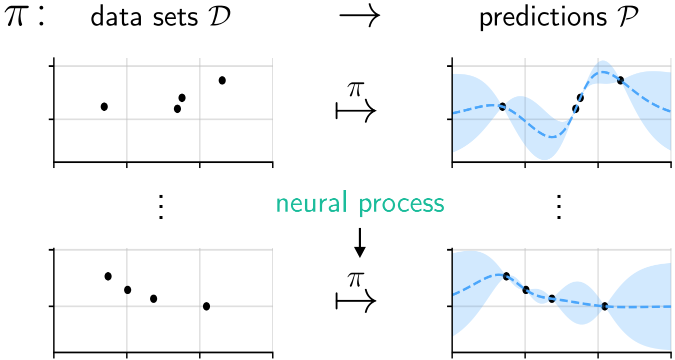
$\vdots$ $\qquad\qquad$ $\vdots$



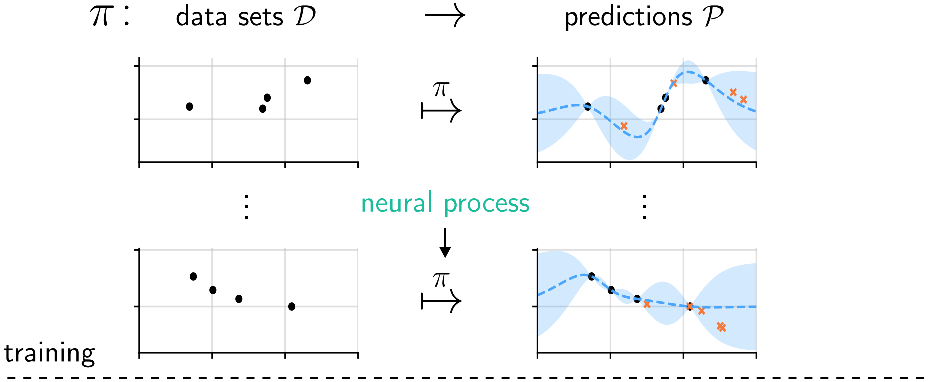
$\stackrel{\pi}{\longmapsto}$



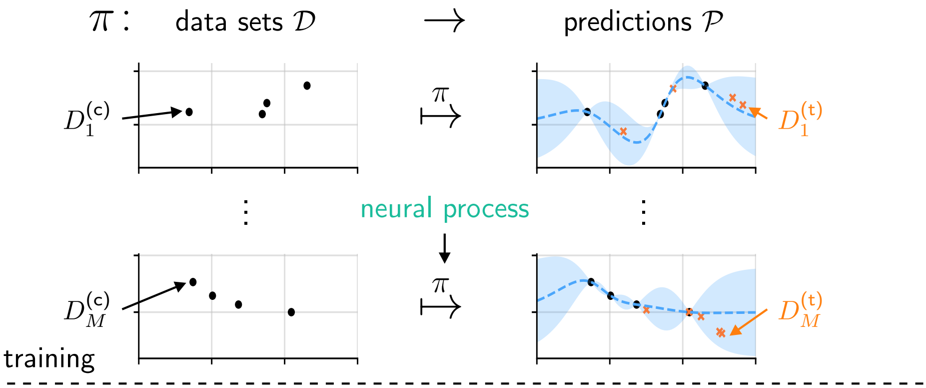
Meta-Learning and Neural Processes: Learning to Predict



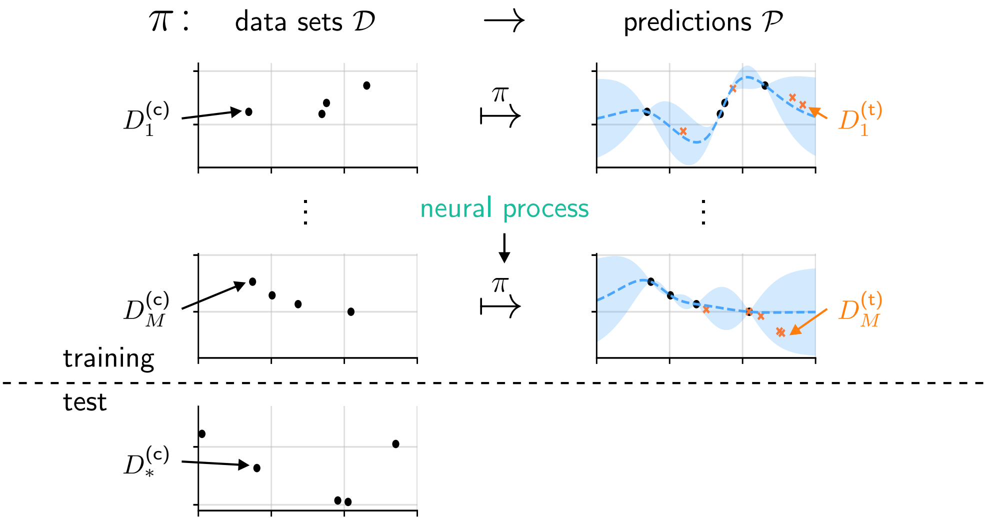
Meta-Learning and Neural Processes: Learning to Predict



Meta-Learning and Neural Processes: Learning to Predict

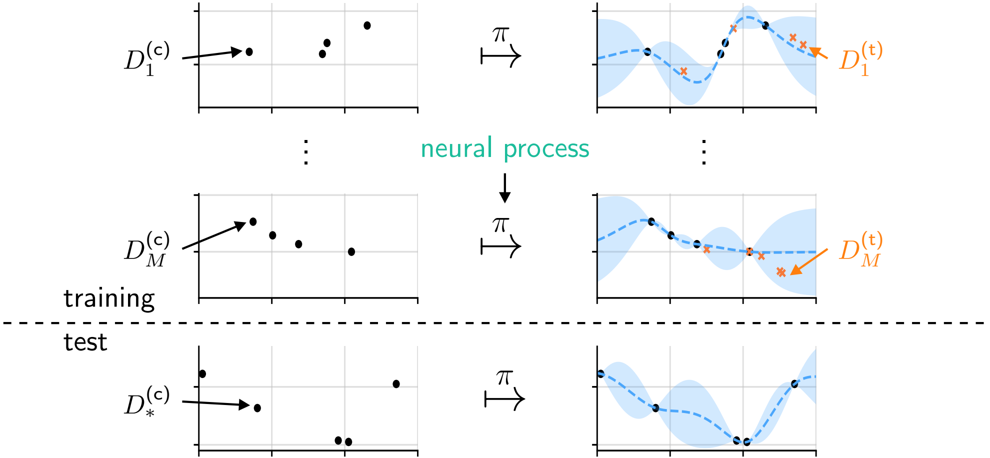


Meta-Learning and Neural Processes: Learning to Predict



Meta-Learning and Neural Processes: Learning to Predict

π : data sets $\mathcal{D}$ $\longrightarrow$ predictions $\mathcal{P}$



Definitions and Notation

- A neural process is a function $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$ parametrised with neural networks.

Definitions and Notation

intuitively, (μ, σ^2) at test inputs; rigorously, the space of all stochastic processes

- A neural process is a function $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$ parametrised with neural networks.

Definitions and Notation

intuitively, (μ, σ^2) at test inputs; rigorously, the space of all stochastic processes

- A neural process is a function $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$ parametrised with neural networks.
- $q_\theta(\mathbf{y} \mid \mathbf{x}, D)$: the density of $\pi_\theta(D)$ at $\mathbf{x}$.

Definitions and Notation

intuitively, (μ, σ^2) at test inputs; rigorously,
the space of all stochastic processes

- A neural process is a function $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$ parametrised with neural networks.
- $q_\theta(\mathbf{y} \mid \mathbf{x}, D)$: the density of $\pi_\theta(D)$ at $\mathbf{x}$.

- NPs operate in setting of meta-learning, with meta–data sets:

$$(D_m)_{m=1}^{M} \quad \text{with} \quad D_m = D_m^{(\mathsf{c})} \cup D_m^{(\mathsf{t})}.$$

$D_m^{(\mathsf{c})} = (\mathbf{x}_m^{(\mathsf{c})}, \mathbf{y}_m^{(\mathsf{c})})$ is the context set; $D_m^{(\mathsf{t})} = (\mathbf{x}_m^{(\mathsf{t})}, \mathbf{y}_m^{(\mathsf{t})})$ is the target set.

Definitions and Notation

intuitively, (μ, σ^2) at test inputs; rigorously,
the space of all stochastic processes

- A neural process is a function $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$ parametrised with neural networks.
- $q_\theta(\mathbf{y} \mid \mathbf{x}, D)$: the density of $\pi_\theta(D)$ at $\mathbf{x}$.

- NPs operate in setting of meta-learning, with meta–data sets:

$$(D_m)_{m=1}^{M} \quad \text{with} \quad D_m = D_m^{(\mathsf{c})} \cup D_m^{(\mathsf{t})}.$$

$D_m^{(\mathsf{c})} = (\mathbf{x}_m^{(\mathsf{c})}, \mathbf{y}_m^{(\mathsf{c})})$ is the context set; $D_m^{(\mathsf{t})} = (\mathbf{x}_m^{(\mathsf{t})}, \mathbf{y}_m^{(\mathsf{t})})$ is the target set.

- Training with MLE:

$$\hat{\theta} \in \operatorname*{arg\,max}_{\theta \in \Theta} \sum_{m=1}^{M} \log q_\theta(\mathbf{y}_m^{(\mathsf{t})} \ell \mid \mathbf{x}_m^{(\mathsf{t})}, D_m^{(\mathsf{c})}).$$

Definitions and Notation

intuitively, (μ, σ^2) at test inputs; rigorously, the space of all stochastic processes

- A neural process is a function $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$ parametrised with neural networks.
- $q_\theta(\mathbf{y} \mid \mathbf{x}, D)$: the density of $\pi_\theta(D)$ at $\mathbf{x}$.

- NPs operate in setting of meta-learning, with meta–data sets:

$$(D_m)_{m=1}^M \quad \text{with} \quad D_m = D_m^{(\mathsf{c})} \cup D_m^{(\mathsf{t})}.$$

$D_m^{(\mathsf{c})} = (\mathbf{x}_m^{(\mathsf{c})}, \mathbf{y}_m^{(\mathsf{c})})$ is the context set; $D_m^{(\mathsf{t})} = (\mathbf{x}_m^{(\mathsf{t})}, \mathbf{y}_m^{(\mathsf{t})})$ is the target set.

- Training with MLE:

will omit when clear from context

$$\hat{\theta} \in \operatorname*{arg\,max}_{\theta \in \Theta} \sum_{m=1}^M \log q_\theta(\mathbf{y}_m^{(\mathsf{t})} \ell \mid \mathbf{x}_m^{(\mathsf{t})}, D_m^{(\mathsf{c})}).$$

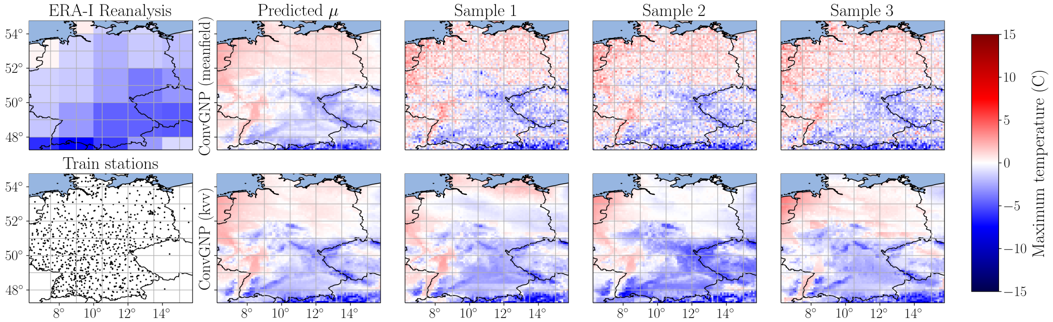
The Appeal of Neural Processes

- ✓ Extremely versatile and flexible
- ✓ Fast, probabilistic predictions
- ✓ Simple to train
- ✓ Work very well in practice

The Appeal of Neural Processes

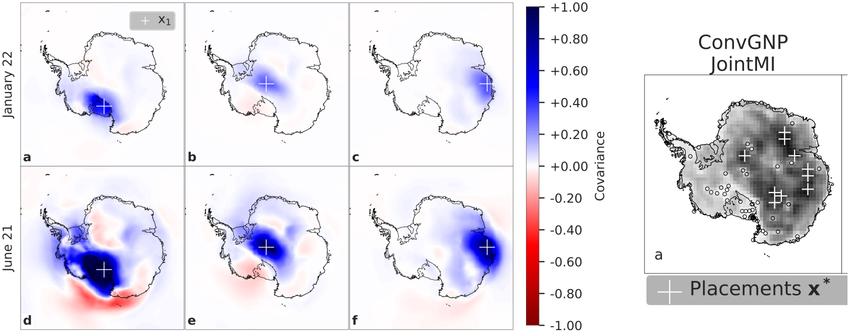
- ✓ Extremely versatile and flexible
- ✓ Fast, probabilistic predictions
- ✓ Simple to train
- ✓ Work very well in practice

- Climate model downscaling (Markou et al., 2022):



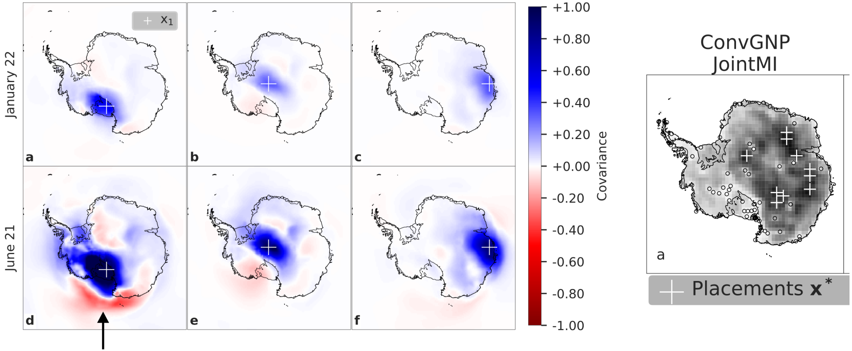
The Appeal of Neural Processes (2)

- Environmental sensor placement in Antarctica (Andersson et al., 2023):



The Appeal of Neural Processes (2)

- Environmental sensor placement in Antarctica (Andersson et al., 2023):



positive covariance extends to edge of Ross ice shelf when sea ice is present

Two Axes of Neural Process Design

- Starting point: want to parametrise $\pi_\theta : \mathcal{D} \to \mathcal{P}$.

Two Axes of Neural Process Design

"prediction map"

- Starting point: want to parametrise $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$.

Two Axes of Neural Process Design

"prediction map"

- Starting point: want to parametrise $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$.

1. Choose the form of the predictions.

Two Axes of Neural Process Design

"prediction map"

- Starting point: want to parametrise $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$.

1. Choose the form of the predictions.

- For example, $q(y \,|\, x, D) = \mathcal{N}(y \,|\, \mu_\theta(x, D), \sigma^2_\theta(x, D))$.

Two Axes of Neural Process Design

“prediction map”

- Starting point: want to parametrise $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$.

1. Choose the form of the predictions.

- For example, $q(y \mid x, D) = \mathcal{N}(y \mid \mu_\theta(x, D), \sigma^2_\theta(x, D))$.

$\mu_\theta \colon \mathcal{X} \times \mathcal{D} \to \mathbb{R}$
“mean function”

$\sigma^2_\theta \colon \mathcal{X} \times \mathcal{D} \to [0, \infty)$
“variance function”

Two Axes of Neural Process Design

"prediction map"

- Starting point: want to parametrise $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$.

1. Choose the form of the predictions.

- For example, $q(y \mid x, D) = \mathcal{N}(y \mid \mu_\theta(x, D), \sigma_\theta^2(x, D))$.

$\mu_\theta \colon \mathcal{X} \times \mathcal{D} \to \mathbb{R}$
"mean function"

$\sigma_\theta^2 \colon \mathcal{X} \times \mathcal{D} \to [0, \infty)$
"variance function"

2. Parametrise these parameter functions with a neural network architecture.

Two Axes of Neural Process Design

“prediction map”

- Starting point: want to parametrise $\pi_\theta \colon \mathcal{D} \to \mathcal{P}$.

1. Choose the form of the predictions.

- For example, $q(y \mid x, D) = \mathcal{N}(y \mid \mu_\theta(x, D), \sigma^2_\theta(x, D))$.

$\mu_\theta \colon \mathcal{X} \times \mathcal{D} \to \mathbb{R}$
“mean function”

$\sigma^2_\theta \colon \mathcal{X} \times \mathcal{D} \to [0, \infty)$
“variance function”

2. Parametrise these parameter functions with a neural network architecture.

- How do we parametrise functions on $\mathcal{D}$?

First Axis: Form of Prediction

- Conditional neural processes (CNPs; Garnelo, Rosenbaum, et al., 2018):

$$q(\mathbf{y} \mid D) = \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D) \\ \mu_2(D) \end{bmatrix}, \begin{bmatrix} \sigma_1^2(D) & 0 \\ 0 & \sigma_2^2(D) \end{bmatrix}\right).$$

$\sigma^2(x_2, D)$

First Axis: Form of Prediction

- Conditional neural processes (CNPs; Garnelo, Rosenbaum, et al., 2018):

$$q(\mathbf{y} \mid D) = \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D) \\ \mu_2(D) \end{bmatrix}, \begin{bmatrix} \sigma_1^2(D) & 0 \\ 0 & \sigma_2^2(D) \end{bmatrix}\right).$$

$\sigma^2(x_2, D)$

- Latent-variable neural processes (LNPs; Garnelo, Schwarz, et al., 2018):

$$q(\mathbf{y} \mid D) = \int \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D, \mathbf{z}) \\ \mu_2(D, \mathbf{z}) \end{bmatrix}, \begin{bmatrix} \sigma_2^2(D, \mathbf{z}) & 0 \\ 0 & \sigma_2^2(D, \mathbf{z}) \end{bmatrix}\right) q(\mathbf{z} \mid D) \, \mathrm{d}\mathbf{z}.$$

First Axis: Form of Prediction

- Conditional neural processes (CNPs; Garnelo, Rosenbaum, et al., 2018):

$$q(\mathbf{y} \,|\, D) = \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D) \\ \mu_2(D) \end{bmatrix}, \begin{bmatrix} \sigma_1^2(D) & 0 \\ 0 & \sigma_2^2(D) \end{bmatrix}\right).$$

$\sigma^2(x_2, D)$

- Latent-variable neural processes (LNPs; Garnelo, Schwarz, et al., 2018):

$$q(\mathbf{y} \,|\, D) = \int \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D, \mathbf{z}) \\ \mu_2(D, \mathbf{z}) \end{bmatrix}, \begin{bmatrix} \sigma_2^2(D, \mathbf{z}) & 0 \\ 0 & \sigma_2^2(D, \mathbf{z}) \end{bmatrix}\right) q(\mathbf{z} \,|\, D) \, \mathrm{d}\mathbf{z}.$$

- Gaussian neural processes (GNPs; Markou et al., 2022):

$$q(\mathbf{y} \,|\, D) = \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D) \\ \mu_2(D) \end{bmatrix}, \begin{bmatrix} \Sigma_{11}(D) & \Sigma_{12}(D) \\ \Sigma_{21}(D) & \Sigma_{22}(D) \end{bmatrix}\right).$$

First Axis: Form of Prediction

- Conditional neural processes (CNPs; Garnelo, Rosenbaum, et al., 2018):

$$q(\mathbf{y} \mid D) = \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D) \\ \mu_2(D) \end{bmatrix}, \begin{bmatrix} \sigma_1^2(D) & 0 \\ 0 & \sigma_2^2(D) \end{bmatrix}\right).$$

$\sigma^2(x_2, D)$

- Latent-variable neural processes (LNPs; Garnelo, Schwarz, et al., 2018):

$$q(\mathbf{y} \mid D) = \int \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D, \mathbf{z}) \\ \mu_2(D, \mathbf{z}) \end{bmatrix}, \begin{bmatrix} \sigma_2^2(D, \mathbf{z}) & 0 \\ 0 & \sigma_2^2(D, \mathbf{z}) \end{bmatrix}\right) q(\mathbf{z} \mid D) \, \mathrm{d}\mathbf{z}.$$

- Gaussian neural processes (GNPs; Markou et al., 2022):

$$q(\mathbf{y} \mid D) = \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \,\middle|\, \begin{bmatrix} \mu_1(D) \\ \mu_2(D) \end{bmatrix}, \begin{bmatrix} \Sigma_{11}(D) & \Sigma_{12}(D) \\ \Sigma_{21}(D) & \Sigma_{22}(D) \end{bmatrix}\right).$$

- Non-Gaussian distributions, mixture distributions, normalising flows... much more!

Second Axis: Neural Network Architecture

e.g., $\mu_\theta \colon \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

- Parametrise parameter functions of the form $f_\theta \colon \mathcal{X} \times \mathcal{D} \to Z$.

Second Axis: Neural Network Architecture

e.g., $\mu_\theta : \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

- Parametrise parameter functions of the form $f_\theta : \mathcal{X} \times \mathcal{D} \to Z$.

General parametrisation of f_θ:

Second Axis: Neural Network Architecture

e.g., $\mu_\theta : \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

- Parametrise parameter functions of the form $f_\theta : \mathcal{X} \times \mathcal{D} \to Z$.

General parametrisation of f_θ:

- Deep set[1]: CNP[2], NP[3].
- Transformer[4]: ANP[6], TNP[6], LBANP[7].

[1] Zaheer et al. (2017) and Edwards et al. (2017); [2] Garnelo, Rosenbaum, et al. (2018); [3] Garnelo, Schwarz, et al. (2018); [4] Vaswani et al. (2017); [5] Kim et al. (2019); [6] Nguyen and Grover (2022); [7] Feng et al. (2023);

Second Axis: Neural Network Architecture

e.g., $\mu_\theta : \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

- Parametrise parameter functions of the form $f_\theta : \mathcal{X} \times \mathcal{D} \to Z$.

General parametrisation of f_θ:

- Deep set[1]: CNP[2], NP[3].
- Transformer[4]: ANP[6], TNP[6], LBANP[7].

$\mathsf{T} \circ f_\theta = f_\theta \circ \mathsf{T}$ for all T in symmetry group:

[1]Zaheer et al. (2017) and Edwards et al. (2017); [2]Garnelo, Rosenbaum, et al. (2018); [3]Garnelo, Schwarz, et al. (2018); [4]Vaswani et al. (2017); [5]Kim et al. (2019); [6]Nguyen and Grover (2022); [7]Feng et al. (2023);

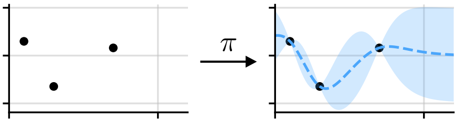
Second Axis: Neural Network Architecture

e.g., $\mu_\theta : \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

- Parametrise parameter functions of the form $f_\theta : \mathcal{X} \times \mathcal{D} \to Z$.

General parametrisation of f_θ:

- Deep set[1]: CNP[2], NP[3].
- Transformer[4]: ANP[6], TNP[6], LBANP[7].



$\mathrm{T} \circ f_\theta = f_\theta \circ \mathrm{T}$ for all T in symmetry group:

[1]Zaheer et al. (2017) and Edwards et al. (2017); [2]Garnelo, Rosenbaum, et al. (2018); [3]Garnelo, Schwarz, et al. (2018); [4]Vaswani et al. (2017); [5]Kim et al. (2019); [6]Nguyen and Grover (2022); [7]Feng et al. (2023);

Second Axis: Neural Network Architecture

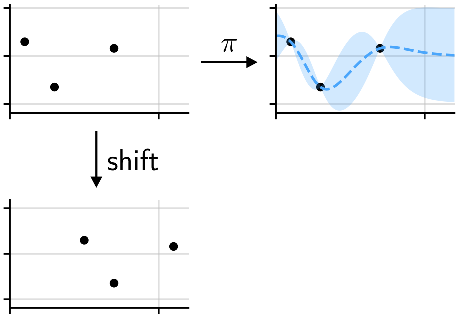
e.g., $\mu_\theta \colon \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

- Parametrise parameter functions of the form $f_\theta \colon \mathcal{X} \times \mathcal{D} \to Z$.

General parametrisation of f_θ:

- Deep set[1]: CNP[2], NP[3].
- Transformer[4]: ANP[6], TNP[6], LBANP[7].

$\mathsf{T} \circ f_\theta = f_\theta \circ \mathsf{T}$ for all T in symmetry group:



[1] Zaheer et al. (2017) and Edwards et al. (2017); [2] Garnelo, Rosenbaum, et al. (2018); [3] Garnelo, Schwarz, et al. (2018); [4] Vaswani et al. (2017); [5] Kim et al. (2019); [6] Nguyen and Grover (2022); [7] Feng et al. (2023);

Second Axis: Neural Network Architecture

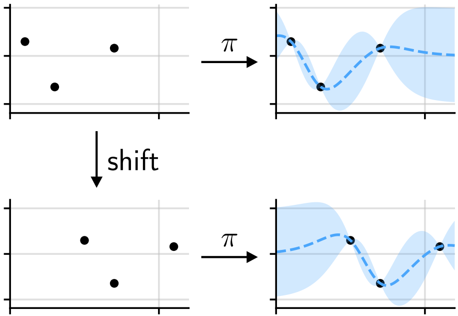
e.g., $\mu_\theta \colon \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

- Parametrise parameter functions of the form $f_\theta \colon \mathcal{X} \times \mathcal{D} \to Z$.

General parametrisation of f_θ:

- Deep set[1]: CNP[2], NP[3].
- Transformer[4]: ANP[6], TNP[6], LBANP[7].

$\mathsf{T} \circ f_\theta = f_\theta \circ \mathsf{T}$ for all T in symmetry group:



[1] Zaheer et al. (2017) and Edwards et al. (2017); [2] Garnelo, Rosenbaum, et al. (2018); [3] Garnelo, Schwarz, et al. (2018); [4] Vaswani et al. (2017); [5] Kim et al. (2019); [6] Nguyen and Grover (2022); [7] Feng et al. (2023);

Second Axis: Neural Network Architecture

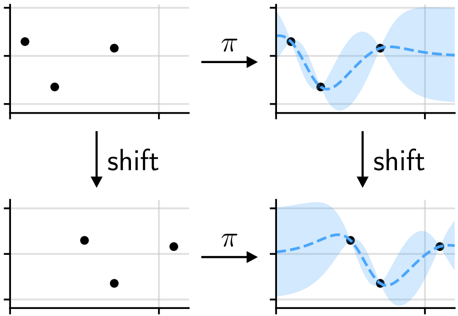
e.g., $\mu_\theta : \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

- Parametrise parameter functions of the form $f_\theta : \mathcal{X} \times \mathcal{D} \to Z$.

General parametrisation of f_θ:

- Deep set[1]: CNP[2], NP[3].
- Transformer[4]: ANP[6], TNP[6], LBANP[7].

$\mathsf{T} \circ f_\theta = f_\theta \circ \mathsf{T}$ for all T in symmetry group:



[1] Zaheer et al. (2017) and Edwards et al. (2017); [2] Garnelo, Rosenbaum, et al. (2018); [3] Garnelo, Schwarz, et al. (2018); [4] Vaswani et al. (2017); [5] Kim et al. (2019); [6] Nguyen and Grover (2022); [7] Feng et al. (2023);

Second Axis: Neural Network Architecture

e.g., $\mu_\theta : \mathcal{X} \times \mathcal{D} \to \mathbb{R}$

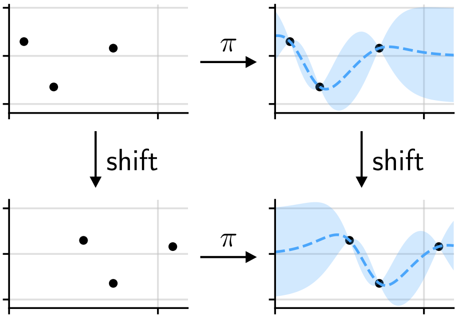
- Parametrise parameter functions of the form $f_\theta : \mathcal{X} \times \mathcal{D} \to Z$.

General parametrisation of f_θ:

- Deep set[1]: CNP[2], NP[3].
- Transformer[4]: ANP[6], TNP[6], LBANP[7].

$\mathrm{T} \circ f_\theta = f_\theta \circ \mathrm{T}$ for all T in symmetry group:

- Equivariance w.r.t. context data D: ConvCNP[8], EquivCNP[9], RCNP[10].
- Equivariance w.r.t. input x: SteerCNP[11].

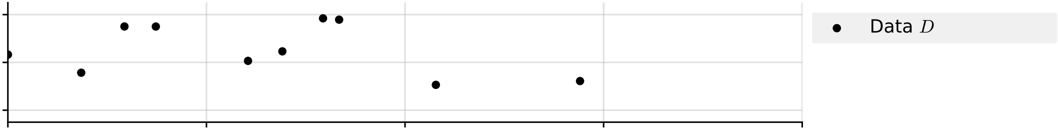


[1]Zaheer et al. (2017) and Edwards et al. (2017); [2]Garnelo, Rosenbaum, et al. (2018); [3]Garnelo, Schwarz, et al. (2018); [4]Vaswani et al. (2017); [5]Kim et al. (2019); [6]Nguyen and Grover (2022); [7]Feng et al. (2023); [8]Gordon et al. (2020); [9]Kawano et al. (2021); [10]Huang et al. (2023); [11]Holderrieth et al. (2021).

Autoregressive Neural Processes

Neural Processes Are Not Without Challenges...

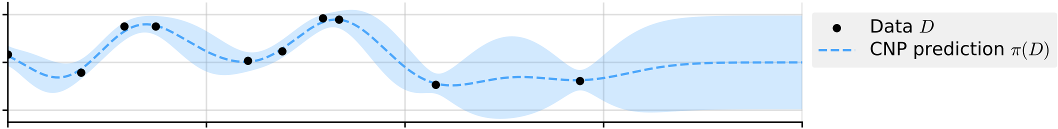
- Prediction by a Conditional Neural Process (CNP):



CNPs

Neural Processes Are Not Without Challenges...

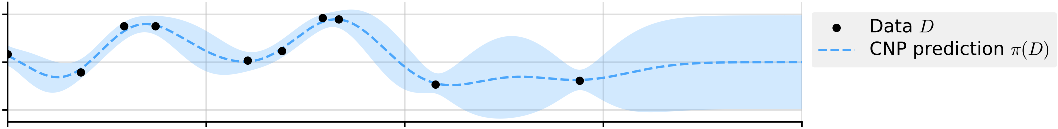
- Prediction by a Conditional Neural Process (CNP):



CNPs

Neural Processes Are Not Without Challenges...

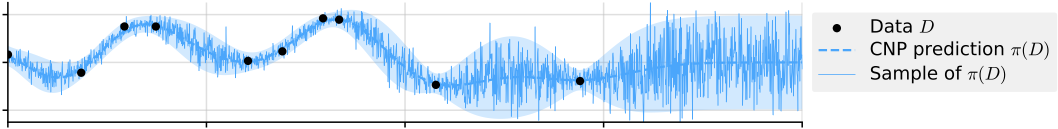
- Prediction by a Conditional Neural Process (CNP):



	Non-Gaussian predictions	Exact training
CNPs	✓	✓

Neural Processes Are Not Without Challenges...

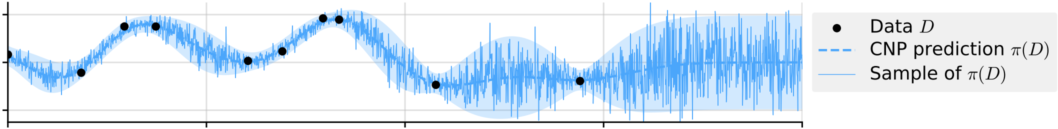
- Prediction by a Conditional Neural Process (CNP):



	Non-Gaussian predictions	Exact training
CNPs	✓	✓

Neural Processes Are Not Without Challenges...

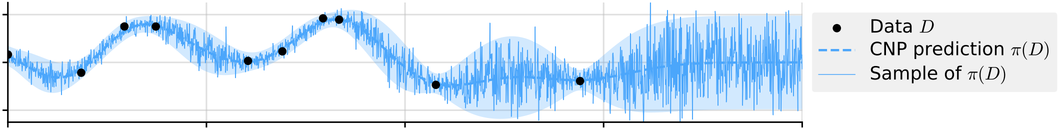
- Prediction by a Conditional Neural Process (CNP):



	Correlated predictions	Non-Gaussian predictions	Exact training
CNPs	✗	✓	✓

Neural Processes Are Not Without Challenges...

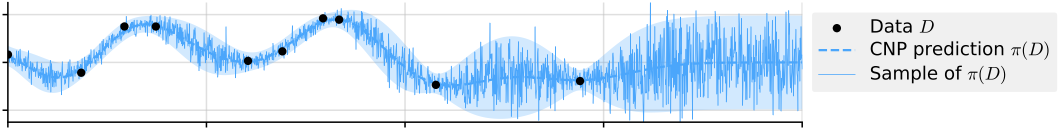
- Prediction by a Conditional Neural Process (CNP):



	Correlated predictions	Non-Gaussian predictions	Exact training
CNPs	✗	✓	✓
Latent-variable NPs	✓	✓	✗

Neural Processes Are Not Without Challenges...

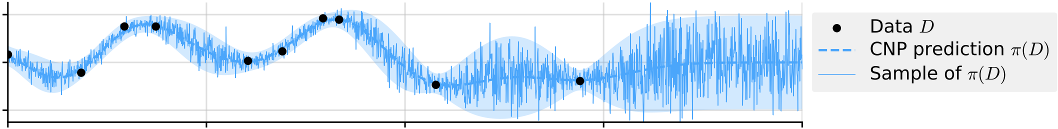
- Prediction by a Conditional Neural Process (CNP):



	Correlated predictions	Non-Gaussian predictions	Exact training
CNPs	✗	✓	✓
Gaussian NPs	✓	✗	✓
Latent-variable NPs	✓	✓	✗

Neural Processes Are Not Without Challenges...

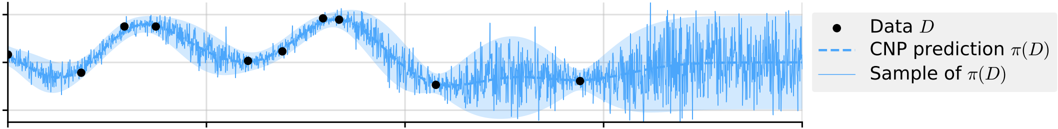
- Prediction by a Conditional Neural Process (CNP):



	Correlated predictions	Non-Gaussian predictions	Exact training
CNPs	✗	✓	✓
Gaussian NPs	✓	✗	✓
Latent-variable NPs	✓	✓	✗
Autoregressive CNPs (AR CNPs)	✓	✓	✓

Neural Processes Are Not Without Challenges...

- Prediction by a Conditional Neural Process (CNP):



	Correlated predictions	Non-Gaussian predictions	Exact training	Consistent predictions
CNPs	✗	✓	✓	✓
Gaussian NPs	✓	✗	✓	✓
Latent-variable NPs	✓	✓	✗	✓
Autoregressive CNPs (AR CNPs)	✓	✓	✓	✗

Autoregressive Conditional Neural Processes

- Idea: feed output of CNP back into the model in an autoregressive fashion:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:3} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D)q^{(\text{CNP})}(y_3 \mid y_1, y_2, D).$$

Autoregressive Conditional Neural Processes

- Idea: feed output of CNP back into the model in an autoregressive fashion:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:3} \,|\, D) = q^{(\text{CNP})}(y_1 \,|\, D)q^{(\text{CNP})}(y_2 \,|\, y_1, D)q^{(\text{CNP})}(y_3 \,|\, y_1, y_2, D).$$

- AR modelling certainly not new, but not yet explored for NPs.

Autoregressive Conditional Neural Processes

- Idea: feed output of CNP back into the model in an autoregressive fashion:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:3} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D)q^{(\text{CNP})}(y_3 \mid y_1, y_2, D).$$

- AR modelling certainly not new, but not yet explored for NPs.

✓ Correlated and non-Gaussian predictions!

Autoregressive Conditional Neural Processes

- Idea: feed output of CNP back into the model in an autoregressive fashion:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:3} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D)q^{(\text{CNP})}(y_3 \mid y_1, y_2, D).$$

- AR modelling certainly not new, but not yet explored for NPs.

✓ Correlated and non-Gaussian predictions!

✓ No modifications to model or training procedure!

Autoregressive Conditional Neural Processes

- Idea: feed output of CNP back into the model in an autoregressive fashion:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:3} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D)q^{(\text{CNP})}(y_3 \mid y_1, y_2, D).$$

- AR modelling certainly not new, but not yet explored for NPs.

✓ Correlated and non-Gaussian predictions!

✓ No modifications to model or training procedure!

✗ Predictions depend on number and order of data (predictions no longer consistent)

Autoregressive Conditional Neural Processes

- Idea: feed output of CNP back into the model in an autoregressive fashion:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:3} \mid D) = q^{(\text{CNP})}(y_1 \mid D) q^{(\text{CNP})}(y_2 \mid y_1, D) q^{(\text{CNP})}(y_3 \mid y_1, y_2, D).$$

- AR modelling certainly not new, but not yet explored for NPs.

✓ Correlated and non-Gaussian predictions!

✓ No modifications to model or training procedure!

✗ Predictions depend on number and order of data (predictions no longer consistent)

✗ Requires multiple forward passes of CNP

↳ only run CNPs with Gaussian marginals in AR mode: computationally cheapest class

Autoregressive Conditional Neural Processes

- Idea: feed output of CNP back into the model in an autoregressive fashion:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:3} \mid D) = q^{(\text{CNP})}(y_1 \mid D) q^{(\text{CNP})}(y_2 \mid y_1, D) q^{(\text{CNP})}(y_3 \mid y_1, y_2, D).$$

- AR modelling certainly not new, but not yet explored for NPs.

✓ Correlated and non-Gaussian predictions!

✓ No modifications to model or training procedure!

✗ Predictions depend on number and order of data (predictions no longer consistent)

✗ Requires multiple forward passes of CNP (Prop. 2.2 offers a partial remedy!)

↳ only run CNPs with Gaussian marginals in AR mode: computationally cheapest class

AR CNPs as a Neural Density Estimator

PixelCNN (Oord et al., 2016): the pixels of an image from top left to bottom right.

AR CNPs as a Neural Density Estimator

PixelCNN (Oord et al., 2016):	the pixels of an image	from top left to bottom right.
AR CNPs:	all target points	in any order.

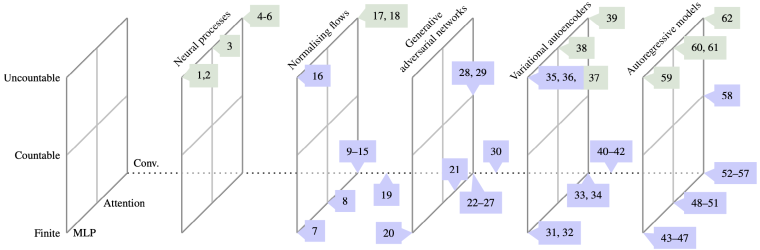
AR CNPs as a Neural Density Estimator

PixelCNN (Oord et al., 2016):	the pixels of an image	from top left to bottom right.
AR CNPs:	$\underbrace{\text{all target points}}_{\text{uncountably many!}}$	in any order.

PixelCNN (Oord et al., 2016): the pixels of an image from top left to bottom right.

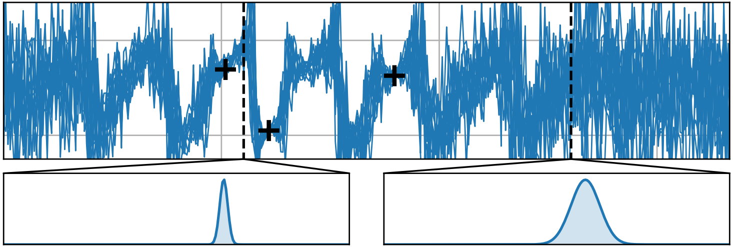
AR CNPs: $\underbrace{\text{all target points}}_{\text{uncountably many!}}$ in any order.

- A slightly insane diagram in the paper:

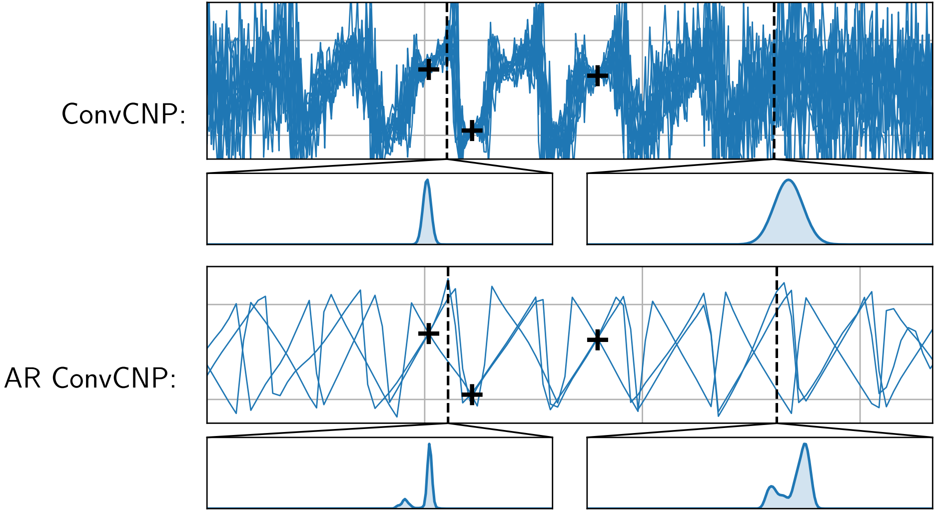


Example: ConvCNP (Gordon et al., 2020) on Sawtooth Data

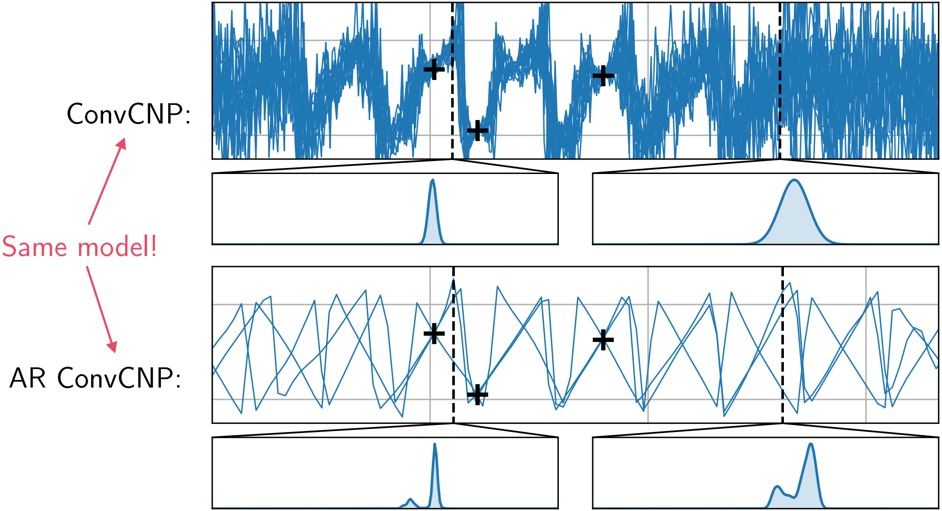
ConvCNP:



Example: ConvCNP (Gordon et al., 2020) on Sawtooth Data



Example: ConvCNP (Gordon et al., 2020) on Sawtooth Data



Big Surprise: AR ConvCNP Performs Really Well

	EQ Norm. KL to truth (↓ better)		Sawtooth Norm. log-lik. (↑ better)		Mixture Norm. log-lik. (↑ better)	
	$d_x, d_y = 1$	$d_x, d_y = 2$	$d_x, d_y = 1$	$d_x, d_y = 2$	$d_x, d_y = 1$	$d_x, d_y = 2$
ConvCNP	0.41 ±0.01	0.41 ±0.00	2.38±0.04	0.12 ±0.01	−0.23±0.04	−0.85 ±0.01
ConvCNP (AR)	0.01 ±0.00	**0.03** ±0.00	**3.60**±0.01	**0.38** ±0.00	**0.45**±0.04	**−0.62** ±0.01
ConvGNP	0.01 ±0.00	0.19 ±0.00	2.62±0.05	0.26 ±0.01	−0.24±0.02	−0.74 ±0.01
FullConvGNP	**0.00** ±0.00		2.16±0.04		−0.05±0.03	
ConvLNP (ML)	0.25 ±0.01	0.39 ±0.00	3.06±0.04	0.31 ±0.01	−0.06±0.03	−0.78 ±0.02
ConvLNP (ELBO)	0.06 ±0.00	0.79 ±0.00	3.51±0.02	0.04 ±0.00	0.12±0.04	−0.92 ±0.01
Diagonal GP	0.40 ±0.01	0.40 ±0.00				
Trivial	1.19 ±0.00	0.79 ±0.00	−0.18±0.00	−0.32 ±0.00	−1.32±0.00	−1.46 ±0.00

Some Intuition

$$q^{(\textsf{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\textsf{CNP})}(y_1 \mid D)q^{(\textsf{CNP})}(y_2 \mid y_1, D) \cdots q^{(\textsf{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

Some Intuition

$$q^{\textsf{(AR CNP)}}(\mathbf{y}_{1:100} \mid D) = q^{\textsf{(CNP)}}(y_1 \mid D) q^{\textsf{(CNP)}}(y_2 \mid y_1, D) \cdots q^{\textsf{(CNP)}}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

- $q^{\textsf{(CNP)}}(y_1 \mid D)$ likely a poor approximation.

Some Intuition

$$q^{\text{(AR CNP)}}(\mathbf{y}_{1:100} \mid D) = q^{\text{(CNP)}}(y_1 \mid D) q^{\text{(CNP)}}(y_2 \mid y_1, D) \cdots q^{\text{(CNP)}}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

- $q^{\text{(CNP)}}(y_1 \mid D)$ likely a poor approximation.
- Insight: when conditioned on many observations, the true data becomes Gaussian:

$$p(f \mid \text{many observations}) \text{ is approximately Gaussian.} \quad (\approx \text{Bernstein–von Mises})$$

Some Intuition

$$\xrightarrow{\text{Gaussian approx. becomes more accurate}}$$

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(y_1 \mid D) q^{(\text{CNP})}(y_2 \mid y_1, D) \cdots q^{(\text{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

- $q^{(\text{CNP})}(y_1 \mid D)$ likely a poor approximation.
- Insight: when conditioned on many observations, the true data becomes Gaussian:

$$p(f \mid \text{many observations}) \text{ is approximately Gaussian.} \qquad (\approx \text{Bernstein–von Mises})$$

$\Rightarrow$ $q^{(\text{CNP})}(y_i \mid \mathbf{y}_{1:(i-1)}, D)$ more accurate as i increases!

Some Intuition

$$\xrightarrow{\text{Gaussian approx. becomes more accurate}}$$

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(y_1 \mid D) q^{(\text{CNP})}(y_2 \mid y_1, D) \cdots q^{(\text{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

- $q^{(\text{CNP})}(y_1 \mid D)$ likely a poor approximation.
- Insight: when conditioned on many observations, the true data becomes Gaussian:

$$p(f \mid \text{many observations}) \text{ is approximately Gaussian.} \quad (\approx \text{Bernstein–von Mises})$$

$\Rightarrow$ $q^{(\text{CNP})}(y_i \mid \mathbf{y}_{1:(i-1)}, D)$ more accurate as i increases!

- First few AR steps poor, then become more accurate.

Some Intuition

$$\xrightarrow{\text{Gaussian approx. becomes more accurate}}$$

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D) \cdots q^{(\text{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

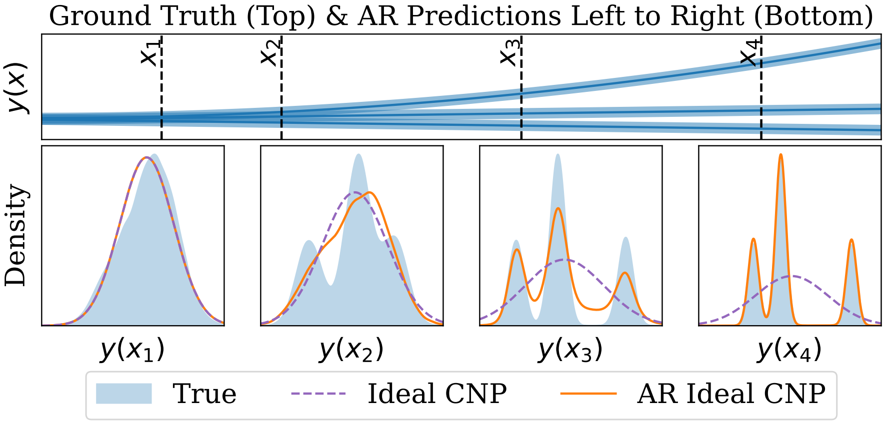
- $q^{(\text{CNP})}(y_1 \mid D)$ likely a poor approximation.
- Insight: when conditioned on many observations, the true data becomes Gaussian:

$$p(f \mid \text{many observations}) \text{ is approximately Gaussian.} \quad (\approx \text{Bernstein–von Mises})$$

$\Rightarrow$ $q^{(\text{CNP})}(y_i \mid \mathbf{y}_{1:(i-1)}, D)$ more accurate as i increases!

- First few AR steps poor, then become more accurate.
- Different random order for every sample: average out first few bad AR steps.

Some Intuition (2)



AR Sampling Design Space

Naively AR sampling 100 target points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D) \cdots q^{(\text{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

AR Sampling Design Space

Naively AR sampling 100 target points:

$$q^{(\mathsf{AR\ CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\mathsf{CNP})}(y_1 \mid D)\, q^{(\mathsf{CNP})}(y_2 \mid y_1, D) \cdots q^{(\mathsf{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

✓ Prediction over $\mathbf{y}_{1:100}$ correlated!

✗ Requires 100 model forwards

AR Sampling Design Space

Naively AR sampling 100 target points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(y_1 \mid D) q^{(\text{CNP})}(y_2 \mid y_1, D) \cdots q^{(\text{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

✓ Prediction over $\mathbf{y}_{1:100}$ correlated!

✗ Requires 100 model forwards

Sample in blocks of 10 points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(\mathbf{y}_{1:10} \mid D) q^{(\text{CNP})}(\mathbf{y}_{11:20} \mid \mathbf{y}_{1:10}, D) \cdots q^{(\text{CNP})}(\mathbf{y}_{91:100} \mid \mathbf{y}_{1:90}, D).$$

AR Sampling Design Space

Naively AR sampling 100 target points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D) \cdots q^{(\text{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

✓ Prediction over $\mathbf{y}_{1:100}$ correlated!

✗ Requires 100 model forwards

Sample in blocks of 10 points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(\mathbf{y}_{1:10} \mid D)q^{(\text{CNP})}(\mathbf{y}_{11:20} \mid \mathbf{y}_{1:10}, D) \cdots q^{(\text{CNP})}(\mathbf{y}_{91:100} \mid \mathbf{y}_{1:90}, D).$$

✗ No correlations within a block

✓ Only 10 model forwards!

AR Sampling Design Space

Naively AR sampling 100 target points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D) \cdots q^{(\text{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

✓ Prediction over $\mathbf{y}_{1:100}$ correlated!

✗ Requires 100 model forwards

Sample in blocks of 10 points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(\mathbf{y}_{1:10} \mid D)q^{(\text{CNP})}(\mathbf{y}_{11:20} \mid \mathbf{y}_{1:10}, D) \cdots q^{(\text{CNP})}(\mathbf{y}_{91:100} \mid \mathbf{y}_{1:90}, D).$$

✗ No correlations within a block

✓ Only 10 model forwards!

Run a GNP in AR mode:

$$q^{(\text{AR GNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{GNP})}(\mathbf{y}_{1:10} \mid D)q^{(\text{GNP})}(\mathbf{y}_{11:20} \mid \mathbf{y}_{1:10}, D) \cdots q^{(\text{GNP})}(\mathbf{y}_{91:100} \mid \mathbf{y}_{1:90}, D).$$

AR Sampling Design Space

Naively AR sampling 100 target points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(y_1 \mid D)q^{(\text{CNP})}(y_2 \mid y_1, D) \cdots q^{(\text{CNP})}(y_{100} \mid \mathbf{y}_{1:99}, D).$$

✓ Prediction over $\mathbf{y}_{1:100}$ correlated!

✗ Requires 100 model forwards

Sample in blocks of 10 points:

$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{CNP})}(\mathbf{y}_{1:10} \mid D)q^{(\text{CNP})}(\mathbf{y}_{11:20} \mid \mathbf{y}_{1:10}, D) \cdots q^{(\text{CNP})}(\mathbf{y}_{91:100} \mid \mathbf{y}_{1:90}, D).$$

✗ No correlations within a block

✓ Only 10 model forwards!

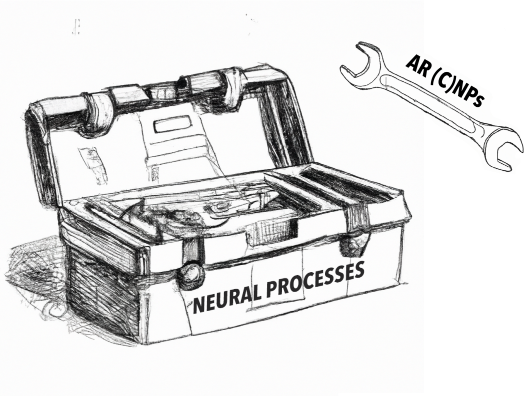
Run a GNP in AR mode:

$$q^{(\text{AR GNP})}(\mathbf{y}_{1:100} \mid D) = q^{(\text{GNP})}(\mathbf{y}_{1:10} \mid D)q^{(\text{GNP})}(\mathbf{y}_{11:20} \mid \mathbf{y}_{1:10}, D) \cdots q^{(\text{GNP})}(\mathbf{y}_{91:100} \mid \mathbf{y}_{1:90}, D).$$

✓ Correlations within a block!

✓ Only 10 model forwards!

Today's Message



AR (C)NPs equip the NP toolbox with a new tool where modelling complexity and computational expense at training time can be traded for computational expense at test time.

Experiment: Cloud Cover Over Antarctica

- Cloud cover is in $[0,1]$, so use categorical–Beta mixture prediction:

$$q(\mathbf{y} \mid \mathbf{x}, D) = \prod_{i=1}^{|\mathbf{y}|} \begin{cases} p_{0,\theta} & \text{if } y_i = 0, \\ p_{1,\theta} & \text{if } y_i = 1, \\ (1 - p_{0,\theta} - p_{1,\theta}) \operatorname{Beta}(y_i; \alpha_\theta, \beta_\theta) & \text{if } y_i \in (0,1). \end{cases}$$

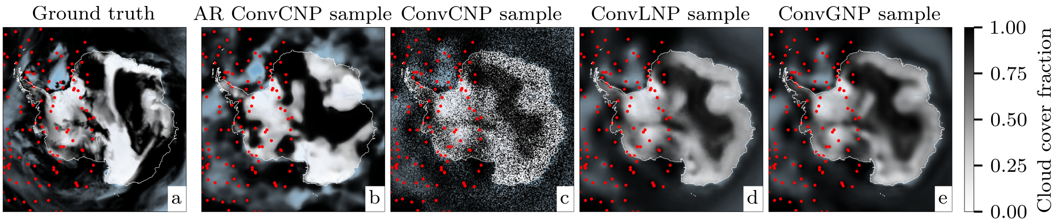
$\beta_\theta = \beta_\theta(x_i, D)$, *et cetera*

Experiment: Cloud Cover Over Antarctica

- Cloud cover is in $[0, 1]$, so use categorical–Beta mixture prediction:

$$q(\mathbf{y} \,|\, \mathbf{x}, D) = \prod_{i=1}^{|\mathbf{y}|} \begin{cases} p_{0,\theta} & \text{if } y_i = 0, \\ p_{1,\theta} & \text{if } y_i = 1, \\ (1 - p_{0,\theta} - p_{1,\theta}) \, \mathrm{Beta}(y_i; \alpha_\theta, \beta_\theta) & \text{if } y_i \in (0, 1). \end{cases}$$

$\beta_\theta = \beta_\theta(x_i, D)$, *et cetera*



Prediction Map Approximation: A Theoretical Analysis

Some Burning Theoretical Questions

AR CNPs:

- Guarantees about the performance of AR CNPs w.r.t. to other NPs?

Some Burning Theoretical Questions

AR CNPs:

- Guarantees about the performance of AR CNPs w.r.t. to other NPs?
- Do predictions of AR CNPs converge to the ground truth in some sense?

Some Burning Theoretical Questions

AR CNPs:

- Guarantees about the performance of AR CNPs w.r.t. to other NPs?
- Do predictions of AR CNPs converge to the ground truth in some sense?

NPs in general:

- CNPs are iffy. Can we establish rigorous theoretical foundations without issues?

Some Burning Theoretical Questions

AR CNPs:

- Guarantees about the performance of AR CNPs w.r.t. to other NPs?
- Do predictions of AR CNPs converge to the ground truth in some sense?

NPs in general:

- CNPs are iffy. Can we establish rigorous theoretical foundations without issues?
- In the limit of infinite data and network capacity, what do neural processes converge to?

Some Burning Theoretical Questions

AR CNPs:

- Guarantees about the performance of AR CNPs w.r.t. to other NPs?
- Do predictions of AR CNPs converge to the ground truth in some sense?

NPs in general:

- CNPs are iffy. Can we establish rigorous theoretical foundations without issues?
- In the limit of infinite data and network capacity, what do neural processes converge to?
- Convergence in which sense? Under what conditions? Rate of convergence?

Some Burning Theoretical Questions

AR CNPs:

- Guarantees about the performance of AR CNPs w.r.t. to other NPs?
- Do predictions of AR CNPs converge to the ground truth in some sense?

NPs in general:

- CNPs are iffy. Can we establish rigorous theoretical foundations without issues?
- In the limit of infinite data and network capacity, what do neural processes converge to?
- Convergence in which sense? Under what conditions? Rate of convergence?

Wessel P. Bruinsma (2022). "Convolutional Conditional Neural Processes". PhD thesis. Department of Engineering, University of Cambridge. DOI: 10.17863/CAM.100216. URL: https://www.repository.cam.ac.uk/handle/1810/354383

Sketch of Theoretical Analysis

e.g., CNPs choose all GPs with independent predictions

- Prediction map: $\pi\colon \mathcal{D} \to \mathcal{Q}$.

- Posit a ground truth stochastic process f, possibly non-Gaussian.
- Posterior prediction map: $\pi_f\colon \mathcal{D} \to \mathcal{P}$, $\pi_f(D) = p(f \mid D)$.
- Approximate π_f with a neural process $\pi_\theta\colon \mathcal{D} \to \mathcal{Q}$.
- Do this by minimising the neural process objective $\mathcal{L}_{\mathsf{NP}}$:

$f(\mathbf{x}) + \boldsymbol{\varepsilon} \sim P^\sigma_{\mathbf{x}}\mu$

with $f \sim \mu$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$

$$\hat{\theta} \in \arg\min \mathcal{L}_{\mathsf{NP}}(\pi_f, \pi_\theta), \quad \mathcal{L}_{\mathsf{NP}}(\pi_f, \pi_\theta) = \mathbb{E}_{p(\mathbf{x})p(D)}[\mathrm{KL}(P^{\sigma_f}_{\mathbf{x}}\pi_f(D), P^{\sigma}_{\mathbf{x}}\pi_\theta(D))].$$

- Study minimisers for CNPs and GNPs.

convergence to minimiser (consistency); compare minimisers for CNPs and GNPs

Some Results From Bruinsma (2022)

- Sec 3.2: Rigorous theoretical foundations are possible.
- Props 3.11: Neural process objective is well defined.

Some Results From Bruinsma (2022)

- Sec 3.2: Rigorous theoretical foundations are possible.
- Props 3.11: Neural process objective is well defined.
- Props 3.26 and 3.27: Identification of minimiser of $\mathcal{L}_{\text{NP}}$ for CNPs and GNPs.

Some Results From Bruinsma (2022)

- Sec 3.2: Rigorous theoretical foundations are possible.
- Props 3.11: Neural process objective is well defined.
- Props 3.26 and 3.27: Identification of minimiser of $\mathcal{L}_{\mathrm{NP}}$ for CNPs and GNPs.

$\Rightarrow$ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.

Some Results From Bruinsma (2022)

- Sec 3.2: Rigorous theoretical foundations are possible.
- Props 3.11: Neural process objective is well defined.
- Props 3.26 and 3.27: Identification of minimiser of $\mathcal{L}_{\mathrm{NP}}$ for CNPs and GNPs.

$\Rightarrow$ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.

$\Rightarrow$ CNPs cannot disentangle epistemic and aleatoric uncertainty, but GNPs can!

Some Results From Bruinsma (2022)

- Sec 3.2: Rigorous theoretical foundations are possible.
- Props 3.11: Neural process objective is well defined.
- Props 3.26 and 3.27: Identification of minimiser of $\mathcal{L}_{\mathrm{NP}}$ for CNPs and GNPs.

$\Rightarrow$ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.

$\Rightarrow$ CNPs cannot disentangle epistemic and aleatoric uncertainty, but GNPs can!

- Props 3.34 and 3.35: Precise conditions for consistency of CNPs and GNPs.

Some Results From Bruinsma (2022)

- Sec 3.2: Rigorous theoretical foundations are possible.
- Props 3.11: Neural process objective is well defined.
- Props 3.26 and 3.27: Identification of minimiser of $\mathcal{L}_{\mathrm{NP}}$ for CNPs and GNPs.

$\Rightarrow$ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.

$\Rightarrow$ CNPs cannot disentangle epistemic and aleatoric uncertainty, but GNPs can!

- Props 3.34 and 3.35: Precise conditions for consistency of CNPs and GNPs.
- Thm 5.7: Translation-equivariant NPs generalise spatially.

Some Results From Bruinsma (2022)

- Sec 3.2: Rigorous theoretical foundations are possible.
- Props 3.11: Neural process objective is well defined.
- Props 3.26 and 3.27: Identification of minimiser of $\mathcal{L}_{\mathrm{NP}}$ for CNPs and GNPs.

$\Rightarrow$ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.

$\Rightarrow$ CNPs cannot disentangle epistemic and aleatoric uncertainty, but GNPs can!

- Props 3.34 and 3.35: Precise conditions for consistency of CNPs and GNPs.
- Thm 5.7: Translation-equivariant NPs generalise spatially.
- Thm 5.15: In the limit of infinite data, AR CNPs always outperform GNPs.

Some Results From Bruinsma (2022)

- Sec 3.2: Rigorous theoretical foundations are possible.
- Props 3.11: Neural process objective is well defined.
- Props 3.26 and 3.27: Identification of minimiser of $\mathcal{L}_{\mathrm{NP}}$ for CNPs and GNPs.

$\Rightarrow$ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.

$\Rightarrow$ CNPs cannot disentangle epistemic and aleatoric uncertainty, but GNPs can!

- Props 3.34 and 3.35: Precise conditions for consistency of CNPs and GNPs.
- Thm 5.7: Translation-equivariant NPs generalise spatially.
- Thm 5.15: In the limit of infinite data, AR CNPs always outperform GNPs.

Many results obvious...

But exciting that all can be established in one unifying theoretical framework!

Unresolved Questions

- Do predictions of AR CNPs converge to the ground truth in some sense?

Unresolved Questions

- Do predictions of AR CNPs converge to the ground truth in some sense?

- Some theoretical wrinkles for ConvDeepSets (Gordon et al., 2020) to iron out.
- Unclarity around the representation capacity of ConvGNPs (Markou et al., 2022).

Unresolved Questions

- Do predictions of AR CNPs converge to the ground truth in some sense?

- Some theoretical wrinkles for ConvDeepSets (Gordon et al., 2020) to iron out.
- Unclarity around the representation capacity of ConvGNPs (Markou et al., 2022).

- Rate of convergence w.r.t. meta–data set size M? Suspect $1/\sqrt{M}$.

Unresolved Questions

- Do predictions of AR CNPs converge to the ground truth in some sense?

- Some theoretical wrinkles for ConvDeepSets (Gordon et al., 2020) to iron out.
- Unclarity around the representation capacity of ConvGNPs (Markou et al., 2022).

- Rate of convergence w.r.t. meta–data set size M? Suspect $1/\sqrt{M}$.
- Analysis in setting of infinitely wide neural networks. Finite widths?

Unresolved Questions

- Do predictions of AR CNPs converge to the ground truth in some sense?

- Some theoretical wrinkles for ConvDeepSets (Gordon et al., 2020) to iron out.
- Unclarity around the representation capacity of ConvGNPs (Markou et al., 2022).

- Rate of convergence w.r.t. meta–data set size M? Suspect $1/\sqrt{M}$.
- Analysis in setting of infinitely wide neural networks. Finite widths?

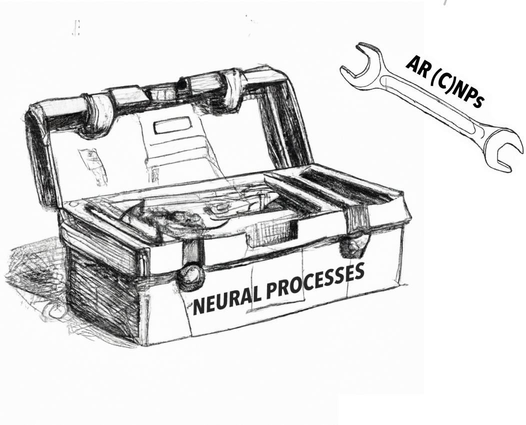
- Approximate equivariances? Got some preliminary results!

Conclusion

Wrapping Up

Neural processes:

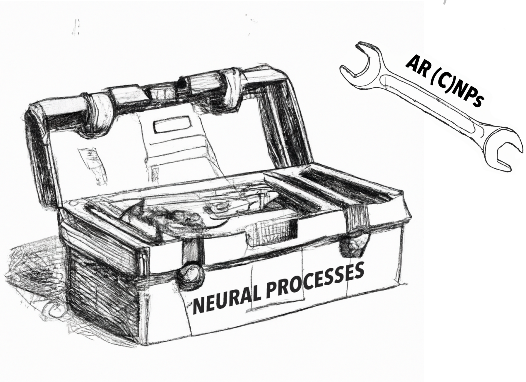
- a flexible collection of architectural neural network techniques
- for general supervised learning problems.



Wrapping Up

Neural processes:

- a flexible collection of architectural neural network techniques
- for general supervised learning problems.

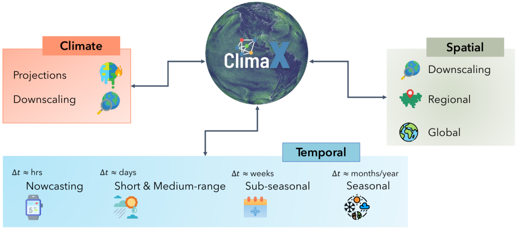


Paper:	This presentation:	Code:
`wessel.ai/pdf/arcnps`	`wessel.ai/pdf/arcnps-mlls`	`github.com/wesselb/neuralprocesses`

Would you like to collaborate? Reach out at `hi@wessel.ai`

Microsoft Research

- Member of the PDE Team within the AI4Science initiative at Microsoft Research
- We're building a foundation model for weather and climate prediction:



Nguyen, Brandstetter, et al. (2023)

Interested? Reach out at `wbruinsma@microsoft.com`

Appendix

References

Andersson, Tom R., Wessel P. Bruinsma, Stratis Markou, James Requeima, Alejandro Coca-Castro, Anna Vaughan, Anna-Louise Ellis, Matthew Lazzara, Daniel C. Jones, J. Scott Hosking, and Richard E. Turner (2023). "Active Learning With Convolutional Gaussian Neural Processes for Environmental Sensor Placement". In: *Environmental Data Science*. eprint: `https://arxiv.org/abs/2211.10381`.

Bruinsma, Wessel P. (2022). "Convolutional Conditional Neural Processes". PhD thesis. Department of Engineering, University of Cambridge. DOI: `10.17863/CAM.100216`. URL: `https://www.repository.cam.ac.uk/handle/1810/354383`.

Bruinsma, Wessel P., Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom R. Andersson, Anna Vaughan, Anthony Buonomo, J. Scott Hosking, and Richard E. Turner (2023). "Autoregressive Conditional Neural Processes". In: *Proceedings of the 11th International Conference on Learning Representations*. eprint: `https://arxiv.org/abs/2303.14468`.

References (2)

Edwards, H. and A. Storkey (2017). "Towards a Neural Statistician". In: *Proceedings of the 5th International Conference on Learning Representations*. eprint: `https://arxiv.org/abs/1606.02185`.

Feng, Leo, Hossein Hajimirsadeghi, Yoshua Bengio, and Mohamed Osama Ahmed (2023). "Latent Bottlenecked Attentive Neural Processes". In: *Proceedings of the 11th International Conference on Learning Representations*. eprint: `https://arxiv.org/abs/2211.08458`.

Garnelo, M., D. Rosenbaum, C. J. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. Whye Teh, D. J. Rezende, and S. M. A. Eslami (2018). "Conditional Neural Processes". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR. eprint: `https://arxiv.org/abs/1807.01613`.

Garnelo, M., J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. M. A. Eslami, and Y. Whye Teh (2018). "Neural Processes". In: *35th International Conference on Machine Learning. Theoretical Foundations and Applications of Deep Generative Models Workshop*. eprint: `https://arxiv.org/abs/1807.01622`.

References (3)

Gordon, Jonathan, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner (2020). "Convolutional Conditional Neural Processes". In: *Proceedings of the 8th International Conference on Learning Representations*. eprint: `https://arxiv.org/abs/1910.13556`.

Holderrieth, Peter, Michael Hutchinson, and Yee Whye Teh (2021). "Equivariant Learning of Stochastic Fields: Gaussian Processes And Steerable Conditional Neural Processes". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR. eprint: `https://arxiv.org/abs/2011.12916`.

Huang, Daolang, Manuel Haussmann, Ulpu Remes, ST John, Grégoire Clarté, Kevin Sebastian Luck, Samuel Kaski, and Luigi Acerbi (2023). "Practical Equivariances via Relational Conditional Neural Processes". In: *arXiv:2306.10915*. eprint: `https://arxiv.org/abs/2306.10915`.

References (4)

Kawano, Makoto, Wataru Kumagai, Akiyoshi Sannai, Yusuke Iwasawa, and Yutaka Matsuo (2021). "Group Equivariant Conditional Neural Processes". In: *Proceedings of the 9th International Conference on Learning Representations*. URL: https://openreview.net/forum?id=e8W-hsu_q5.

Kim, H., A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. Whye Teh (2019). "Attentive Neural Processes". In: *Proceedings of the 7th International Conference on Learning Representations*. eprint: https://arxiv.org/abs/1901.05761.

Markou, Stratis, James Requeima, Wessel P. Bruinsma, Anna Vaughan, and Richard E. Turner (2022). "Practical Conditional Neural Processes Via Tractable Dependent Predictions". In: *Proceedings of the 10th International Conference on Learning Representations*. eprint: https://arxiv.org/abs/2203.08775.

Nguyen, Tung, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover (2023). "ClimaX: A Foundation Model for Weather and Climate". In: eprint: https://arxiv.org/abs/2301.10343.

References (5)

Nguyen, Tung and Aditya Grover (2022). "Transformer Neural Processes: Uncertainty-Aware Meta Learning Via Sequence Modeling". In: eprint: `https://arxiv.org/abs/2207.04179`.

Oord, Aaron van den, Nal Kalchbrenner, and Koray Kavukcuoglu (2016). "Pixel Recurrent Neural Networks". In: *Proceedings of the 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. PMLR. eprint: `https://arxiv.org/abs/1601.06759`.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). "Attention Is All You Need". In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc. eprint: `https://arxiv.org/abs/1706.03762`.

Zaheer, M., S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola (2017). "Deep Sets". In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc. eprint: `https://arxiv.org/abs/1703.06114`.