



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^{*12}**



**Stratis
Markou^{*2}**



**James
Requeima^{*2}**



**Andrew
Foong^{*1}**



**Tom
Andersson³**



**Anna
Vaughan²**



**Anthony
Buonomo²**



**J. Scott
Hosking³⁴**



**Rich
Turner¹²**

***Equal contribution**

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

- 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},

- 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 Requeima^{*2}, Andrew Y. K. Foong^{*1},

- 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 Requeima^{*2}, Andrew Y. K. Foong^{*1},

- 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 Requeima^{*2}, Andrew Y. K. Foong^{*1},

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

- ✗ 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.

- ✗ 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

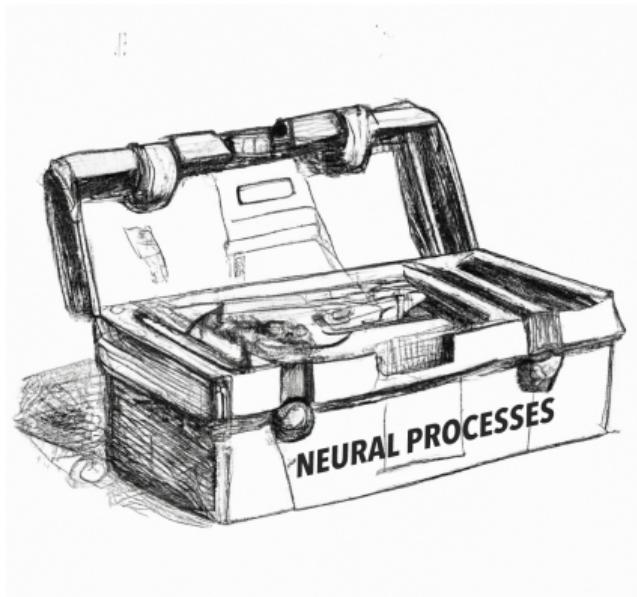
- ✗ A specific model
- ✗ Restricted to a meta-learning setting
- ✗ Very unique and new

Neural processes:

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



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



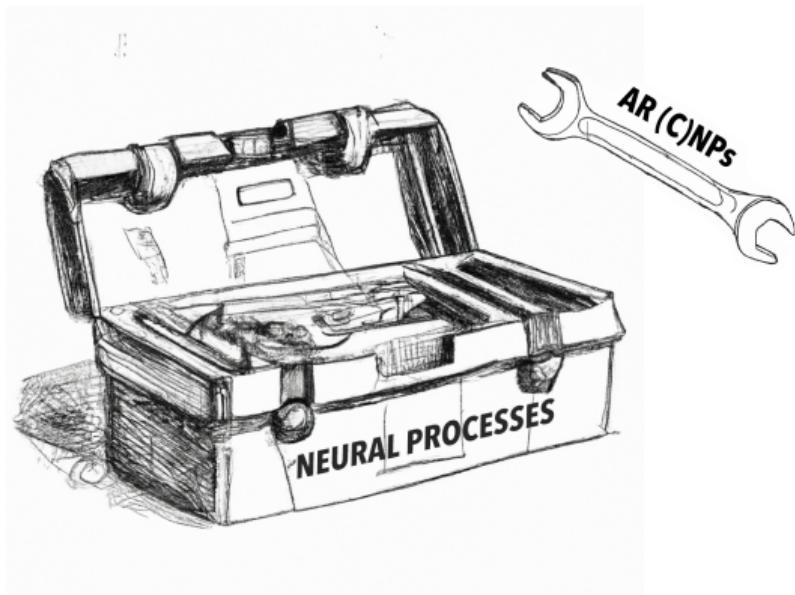
- ✗ A specific model
- ✗ Restricted to a meta-learning setting
- ✗ Very unique and new

Neural processes:

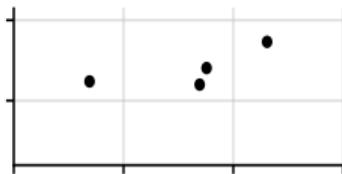
- a flexible collection of architectural neural network techniques
- for general supervised learning problems.



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



Introduction to Neural Processes

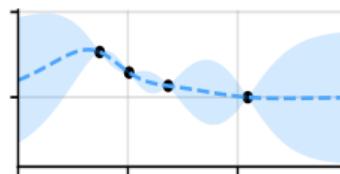
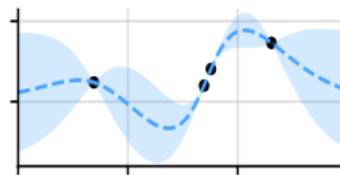
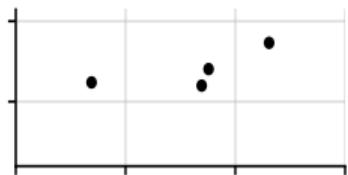


⋮



Meta-Learning and Neural Processes: Learning to Predict

3/25



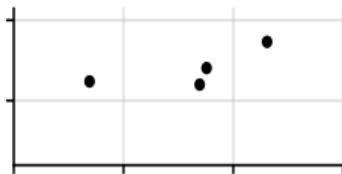
Meta-Learning and Neural Processes: Learning to Predict

3/25

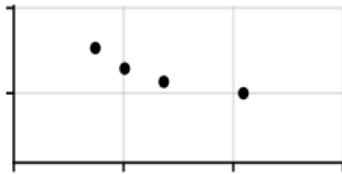
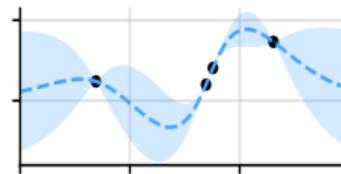
π : data sets \mathcal{D}

→

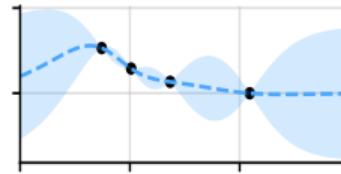
predictions \mathcal{P}



$\pi \rightarrow$

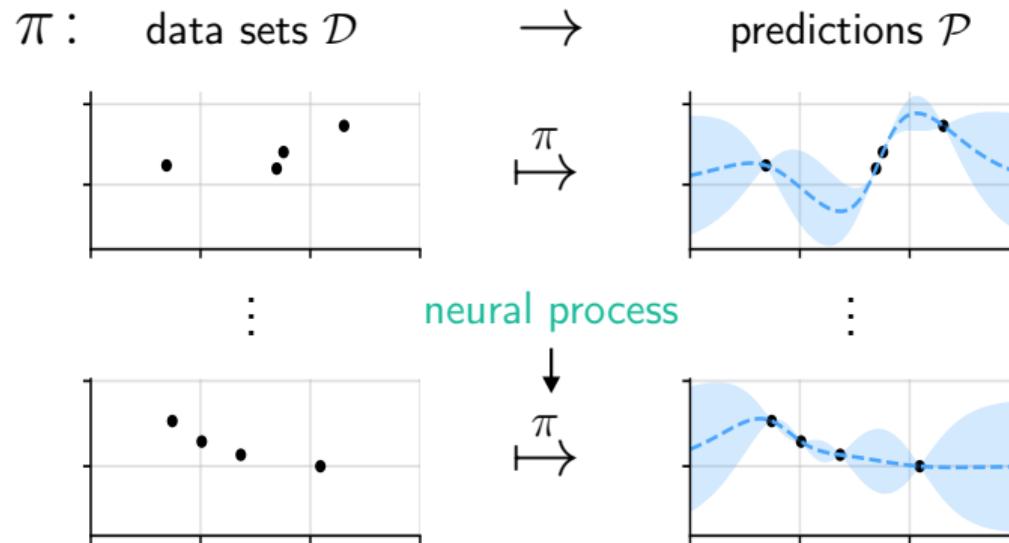


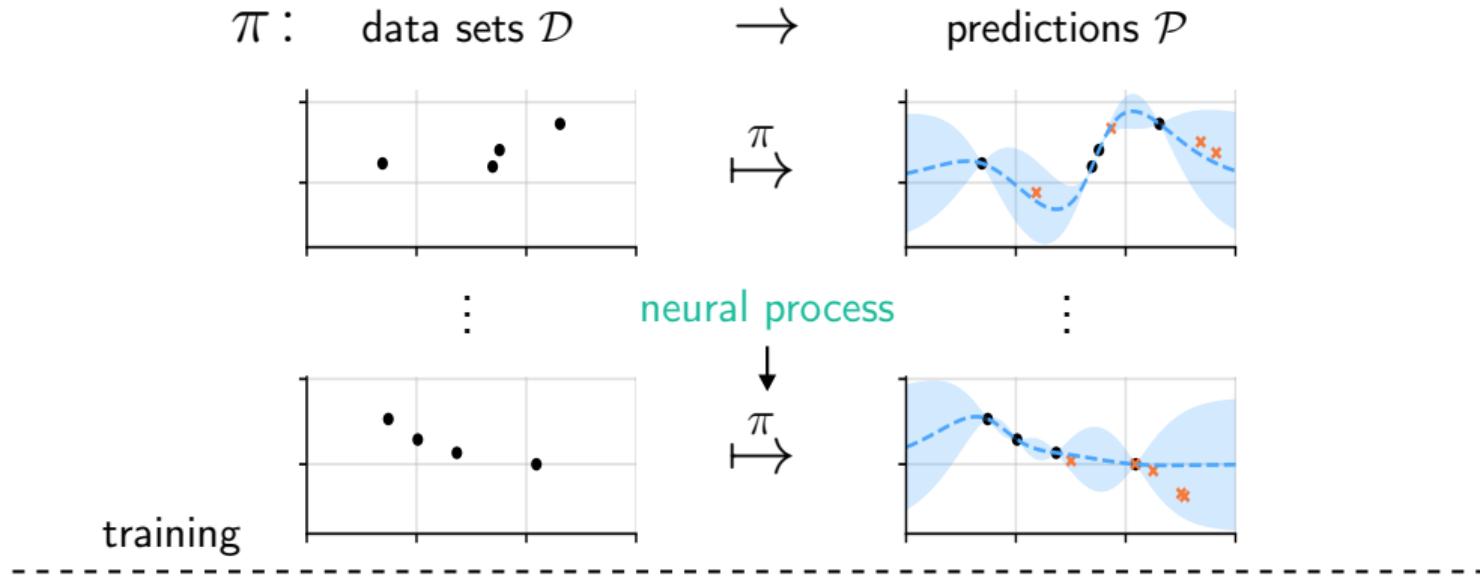
$\pi \rightarrow$

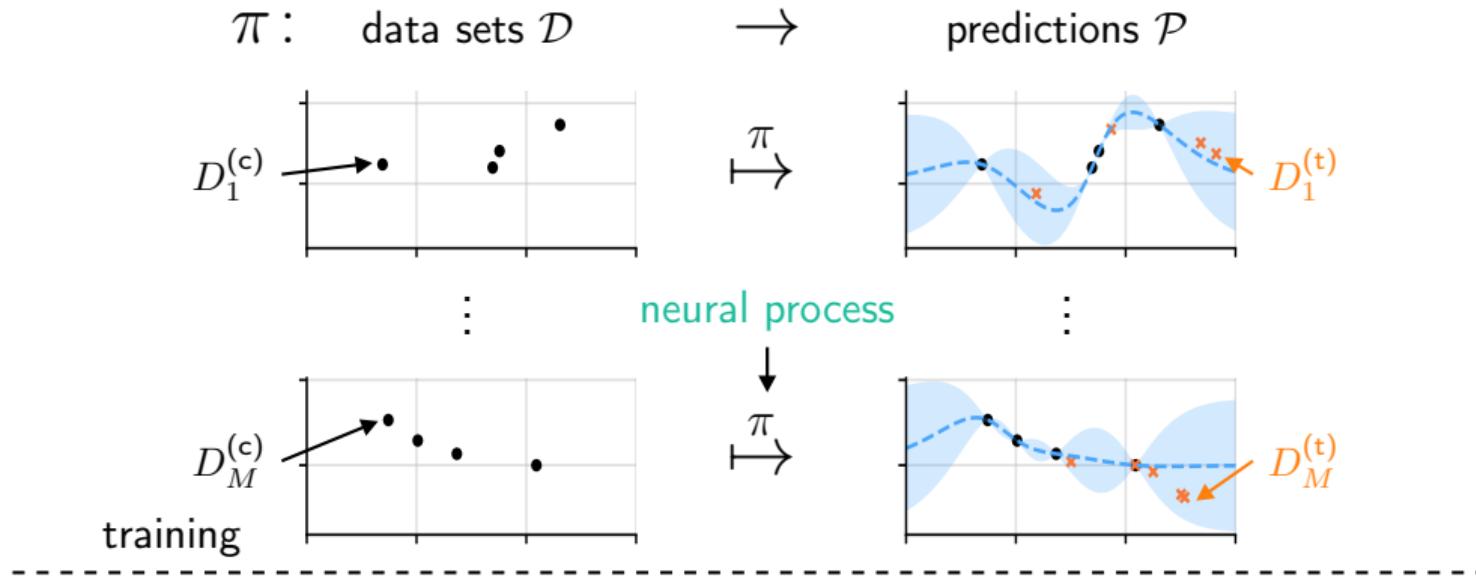


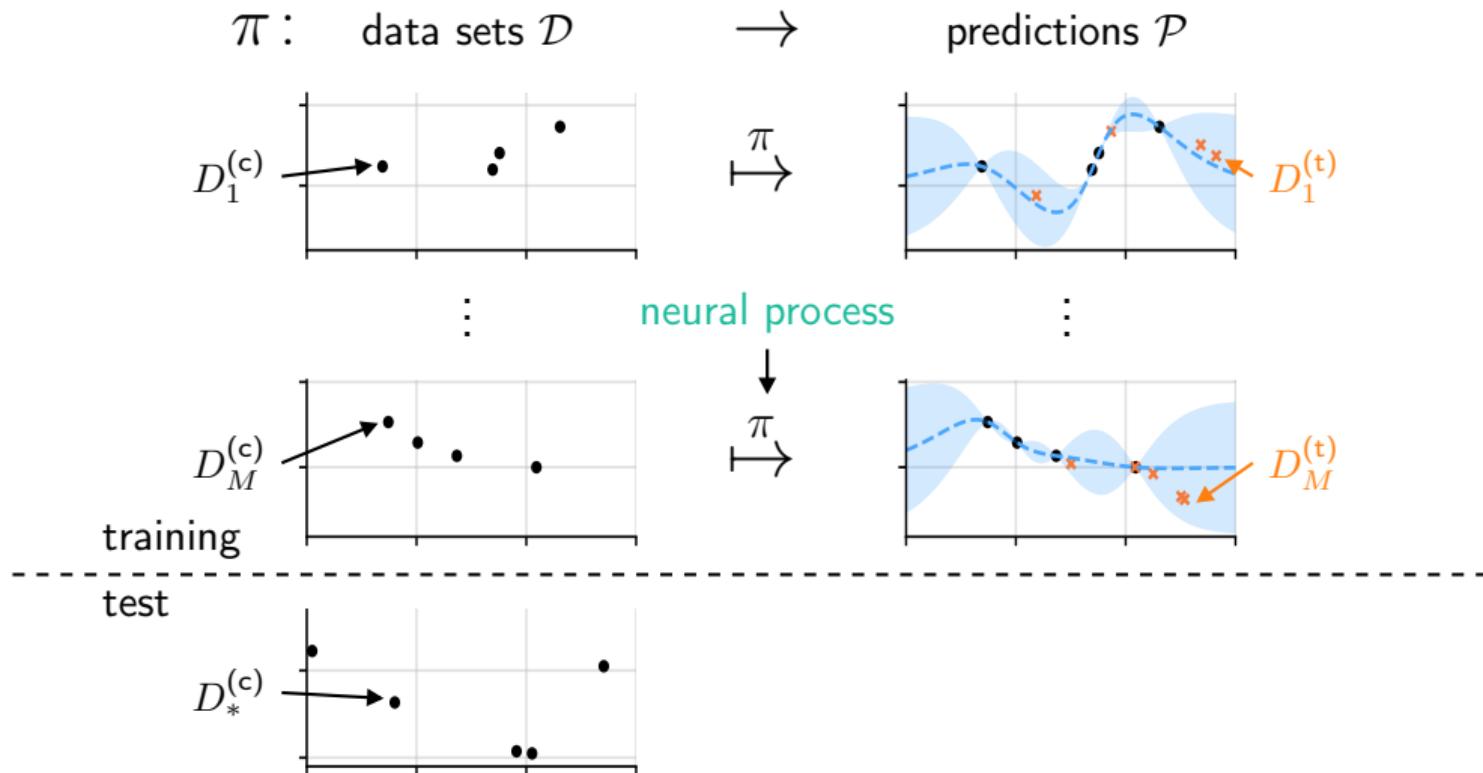
Meta-Learning and Neural Processes: Learning to Predict

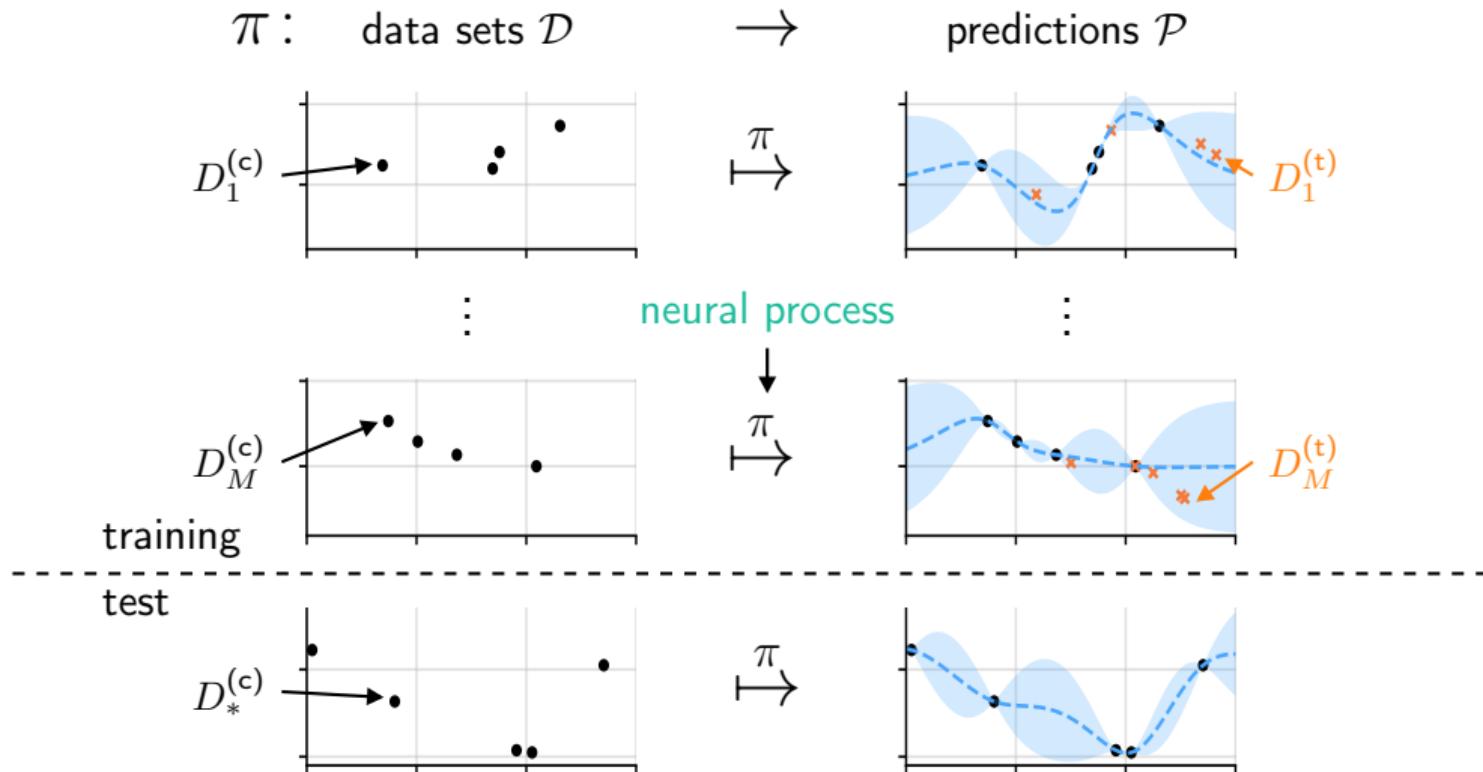
3/25











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

Definitions and Notation

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

4/25

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

Definitions and Notation

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

4/25

- A **neural process** is a function $\pi_\theta: \mathcal{D} \rightarrow \mathcal{P}$ parametrised with neural networks.
- $q_\theta(\mathbf{y} | \mathbf{x}, D)$: 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

4/25

- A **neural process** is a function $\pi_\theta: \mathcal{D} \rightarrow \mathcal{P}$ parametrised with neural networks.
- $q_\theta(\mathbf{y} | \mathbf{x}, D)$: the density of $\pi_\theta(D)$ at \mathbf{x} .
- NPs operate in setting of meta-learning, with **meta-data sets**:

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

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

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



- A **neural process** is a function $\pi_\theta: \mathcal{D} \rightarrow \mathcal{P}$ parametrised with neural networks.
- $q_\theta(\mathbf{y} | \mathbf{x}, D)$: the density of $\pi_\theta(D)$ at \mathbf{x} .
- NPs operate in setting of meta-learning, with **meta-data sets**:

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

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

- Training with MLE:

$$\hat{\theta} \in \arg \max_{\theta \in \Theta} \sum_{m=1}^M \log q_\theta(\mathbf{y}_m^{(t)} | \mathbf{x}_m^{(t)}, D_m^{(c)}).$$

Definitions and Notation

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

4/25

- A **neural process** is a function $\pi_\theta: \mathcal{D} \rightarrow \mathcal{P}$ parametrised with neural networks.
- $q_\theta(\mathbf{y} | \mathbf{x}, D)$: the density of $\pi_\theta(D)$ at \mathbf{x} .
- NPs operate in setting of meta-learning, with **meta-data sets**:

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

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

- Training with MLE:

$$\hat{\theta} \in \arg \max_{\theta \in \Theta} \sum_{m=1}^M \log q_\theta(\mathbf{y}_m^{(t)} | \mathbf{x}_m^{(t)}, D_m^{(c)}).$$

will omit when
clear from context

The Appeal of Neural Processes

5/25

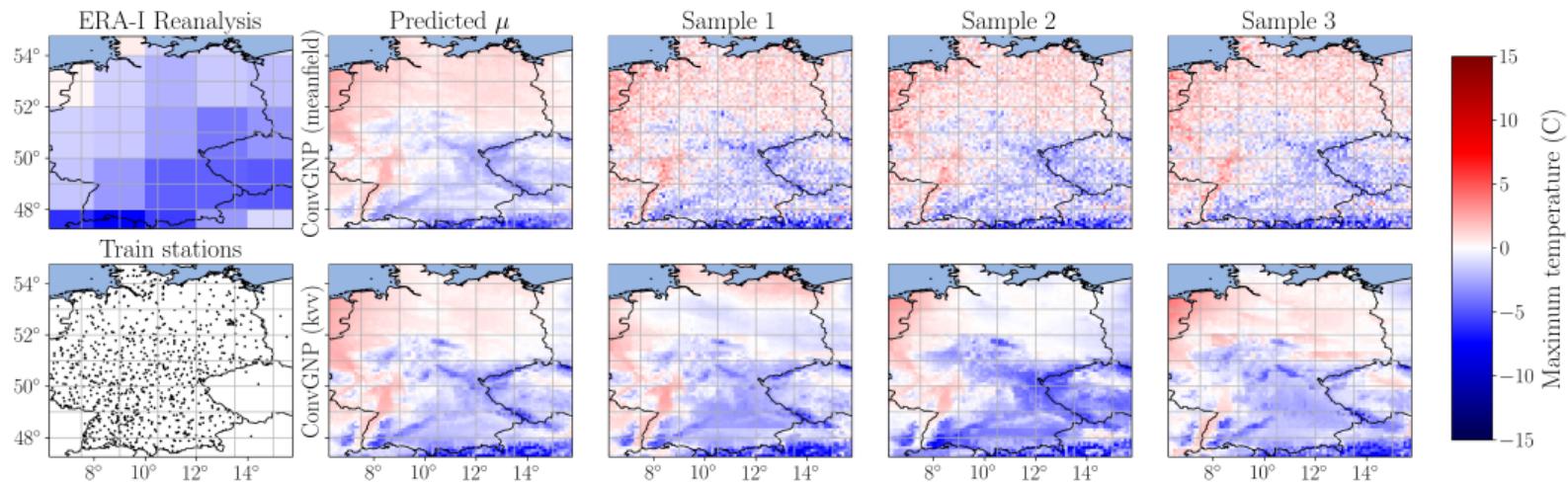
- ✓ Extremely versatile and flexible
- ✓ Fast, probabilistic predictions
- ✓ Simple to train
- ✓ Work very well in practice

The Appeal of Neural Processes

5/25

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

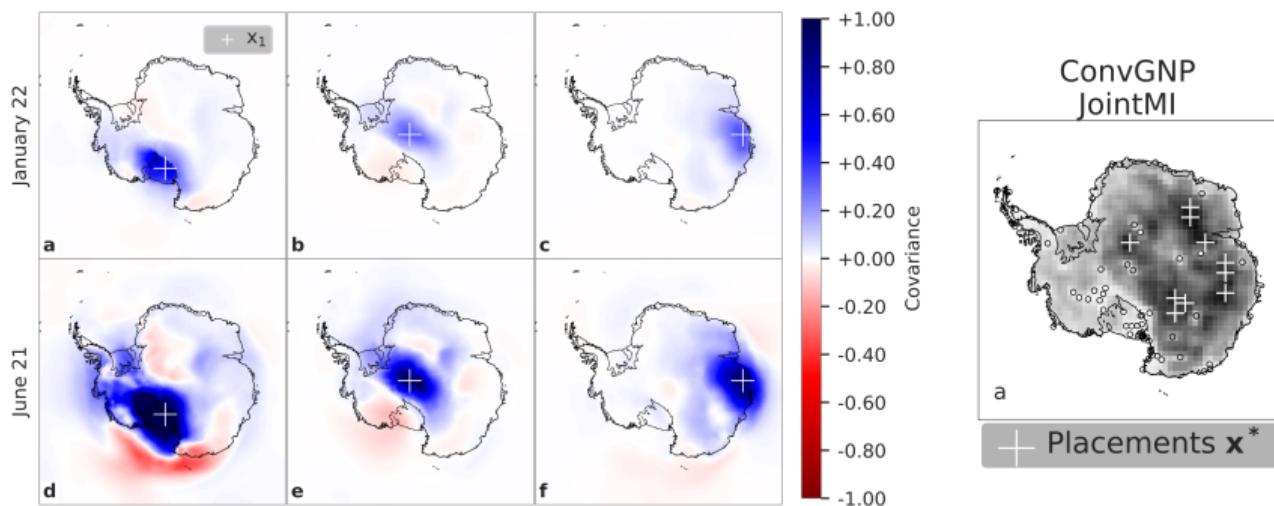
- Climate model downscaling (Markou et al., 2022):



The Appeal of Neural Processes (2)

6/25

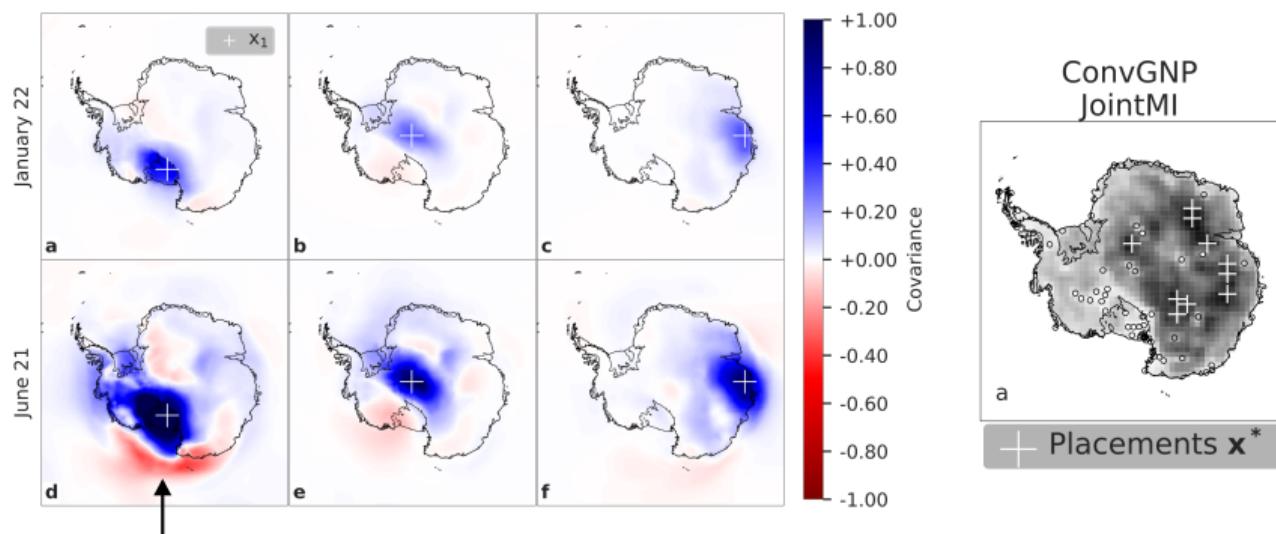
- Environmental sensor placement in Antarctica (Andersson et al., 2023):



The Appeal of Neural Processes (2)

6/25

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



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

“prediction map”
↓

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

“prediction map”
↓

- Starting point: want to parametrise $\pi_\theta: \mathcal{D} \rightarrow \mathcal{P}$.
- ① Choose the form of the predictions.

↙ “prediction map”

- Starting point: want to parametrise $\pi_\theta: \mathcal{D} \rightarrow \mathcal{P}$.
- ① Choose the form of the predictions.
- For example, $q(y | x, D) = \mathcal{N}(y | \mu_\theta(x, D), \sigma_\theta^2(x, D))$.

Two Axes of Neural Process Design

7/25

- Starting point: want to parametrise $\pi_\theta: \mathcal{D} \rightarrow \mathcal{P}$.
- ① Choose the form of the predictions.
- For example, $q(y | x, D) = \mathcal{N}(y | \mu_\theta(x, D), \sigma_\theta^2(x, D))$.

$$\begin{array}{ccc} & \xrightarrow{\hspace{1cm}} & \\ \mu_\theta: \mathcal{X} \times \mathcal{D} \rightarrow \mathbb{R} & & \sigma_\theta^2: \mathcal{X} \times \mathcal{D} \rightarrow [0, \infty) \\ \text{"mean function"} & & \text{"variance function"} \end{array}$$

“prediction map”

Two Axes of Neural Process Design

7/25

“prediction map”
↓

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

① Choose the form of the predictions.

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

$$\begin{array}{ccc} & \nearrow & \swarrow \\ \mu_\theta: \mathcal{X} \times \mathcal{D} \rightarrow \mathbb{R} & & \sigma_\theta^2: \mathcal{X} \times \mathcal{D} \rightarrow [0, \infty) \\ \text{“mean function”} & & \text{“variance function”} \end{array}$$

② Parametrise these parameter functions with a neural network architecture.

Two Axes of Neural Process Design

7/25

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

① Choose the form of the predictions.

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

$$\begin{array}{ccc} & \nearrow & \swarrow \\ \mu_\theta: \mathcal{X} \times \mathcal{D} \rightarrow \mathbb{R} & & \sigma_\theta^2: \mathcal{X} \times \mathcal{D} \rightarrow [0, \infty) \\ \text{"mean function"} & & \text{"variance function"} \end{array}$$

② Parametrise these parameter functions with a neural network architecture.

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

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

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

$\overbrace{\hspace{10em}}^{\sigma^2(x_2, D)}$

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

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

$\uparrow \sigma^2(x_2, D)$

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

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

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

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

$\uparrow \sigma^2(x_2, D)$

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

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

- Gaussian neural processes (**GNP**s; Markou et al., 2022):

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

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

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

$\uparrow \sigma^2(x_2, D)$

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

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

- Gaussian neural processes (**GNP**s; Markou et al., 2022):

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

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

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

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

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

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

General parametrisation of f_θ :

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

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

General parametrisation of f_θ :

- Deep set¹: CNP², NP³.
- Transformer⁴: ANP⁵, TNP⁶, LBANP⁷.

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

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

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

General parametrisation of f_θ :

- Deep set¹: CNP², NP³.
- Transformer⁴: ANP⁵, TNP⁶, LBANP⁷.

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

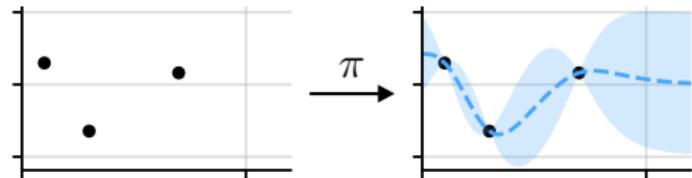
¹Zaheer et al. (2017) and Edwards et al. (2017); ²Garnelo, Rosenbaum, et al. (2018); ³Garnelo, Schwarz, et al. (2018); ⁴Vaswani et al. (2017); ⁵Kim et al. (2019); ⁶Nguyen and Grover (2022); ⁷Feng et al. (2023);

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

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

General parametrisation of f_θ :

- Deep set¹: CNP², NP³.
- Transformer⁴: ANP⁵, TNP⁶, LBANP⁷.



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

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

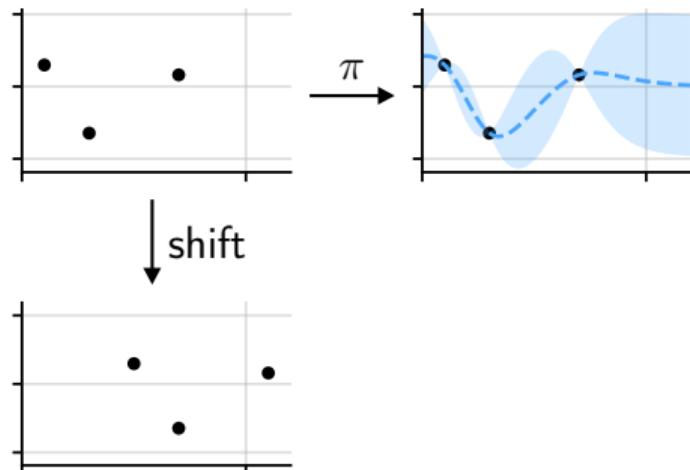
► e.g., $\mu_\theta: \mathcal{X} \times \mathcal{D} \rightarrow \mathbb{R}$

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

General parametrisation of f_θ :

- Deep set¹: CNP², NP³.
- Transformer⁴: ANP⁵, TNP⁶, LBANP⁷.

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



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

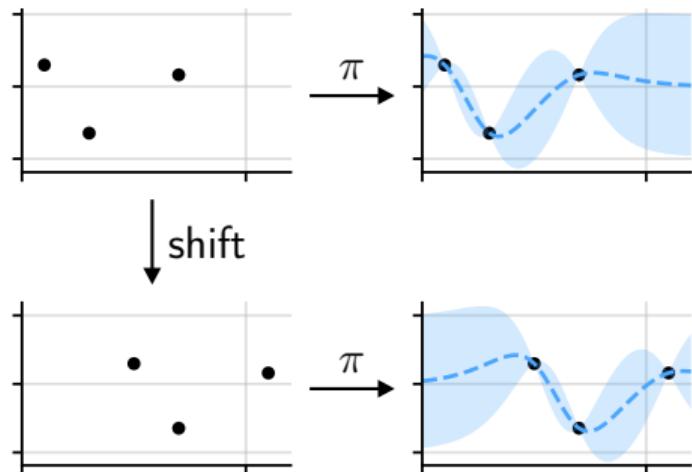
► e.g., $\mu_\theta: \mathcal{X} \times \mathcal{D} \rightarrow \mathbb{R}$

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

General parametrisation of f_θ :

- Deep set¹: CNP², NP³.
- Transformer⁴: ANP⁵, TNP⁶, LBANP⁷.

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



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

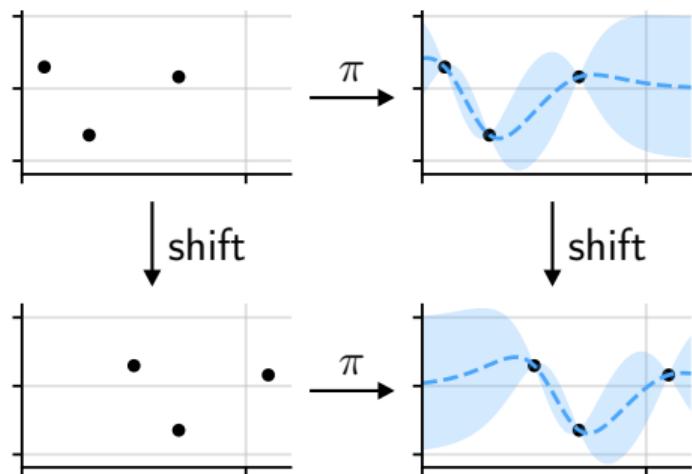
► e.g., $\mu_\theta: \mathcal{X} \times \mathcal{D} \rightarrow \mathbb{R}$

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

General parametrisation of f_θ :

- Deep set¹: CNP², NP³.
- Transformer⁴: ANP⁵, TNP⁶, LBANP⁷.

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



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

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

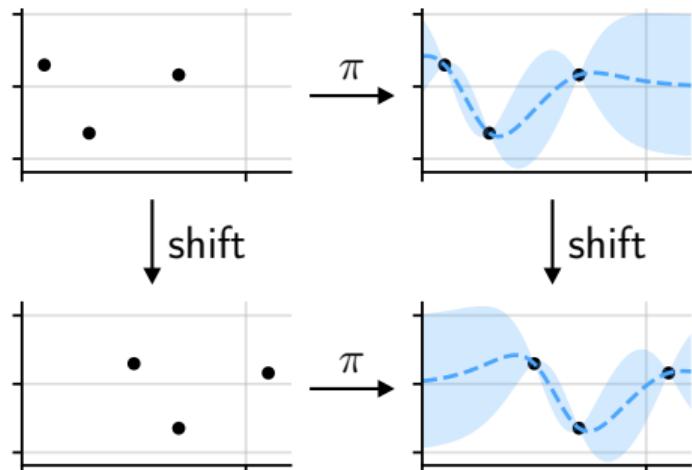
- Parametrise **parameter functions** of the form $f_\theta: \mathcal{X} \times \mathcal{D} \rightarrow Z$.

General parametrisation of f_θ :

- Deep set¹: CNP², NP³.
- Transformer⁴: ANP⁵, TNP⁶, LBANP⁷.

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

- Equivariance w.r.t. context data D : ConvCNP⁸, EquivCNP⁹, RCNP¹⁰.
- Equivariance w.r.t. input x : SteerCNP¹¹.



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

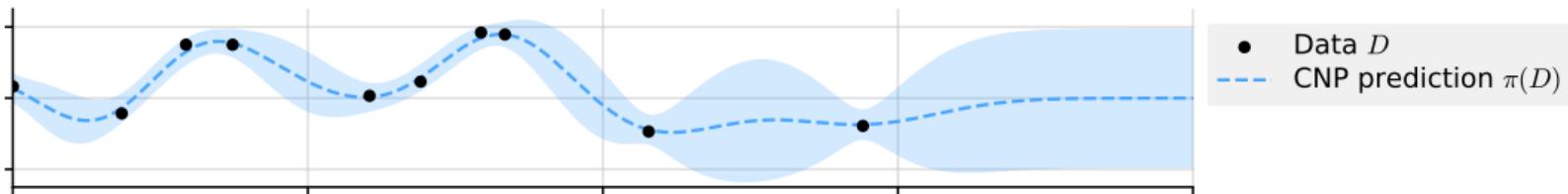
Autoregressive Conditional Neural Processes

- Prediction by a Conditional Neural Process (CNP):



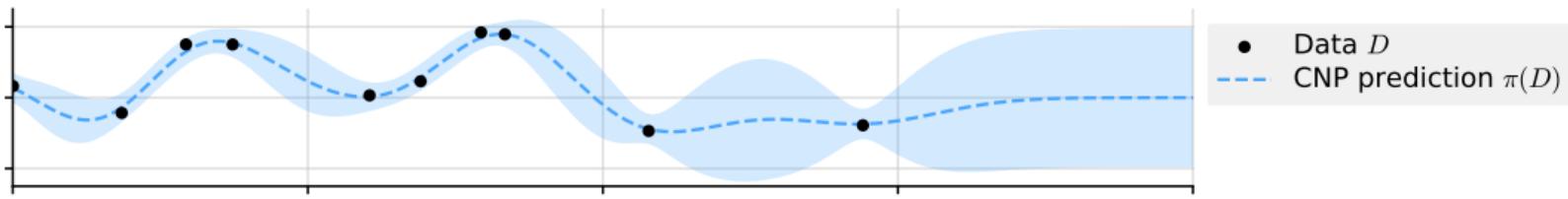
CNPs

- Prediction by a Conditional Neural Process (CNP):



CNPs

- Prediction by a Conditional Neural Process (CNP):

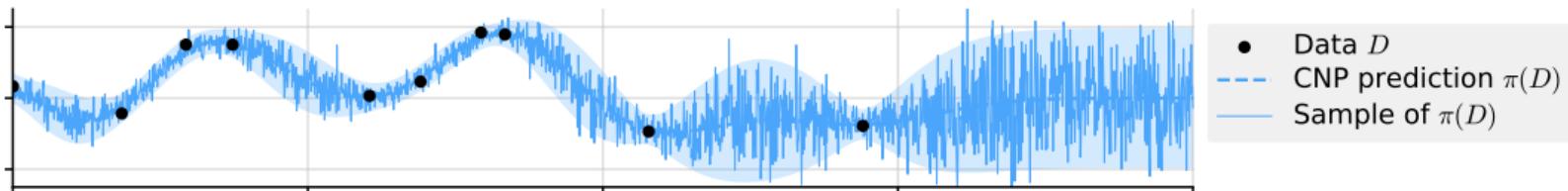


Non-Gaussian
predictions Exact
training

CNPs

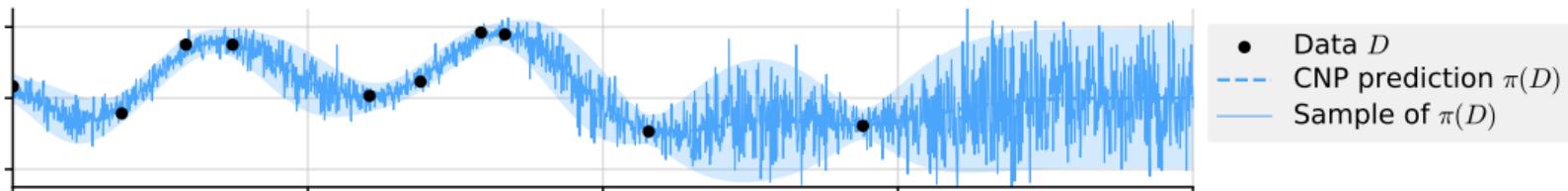


- Prediction by a Conditional Neural Process (CNP):



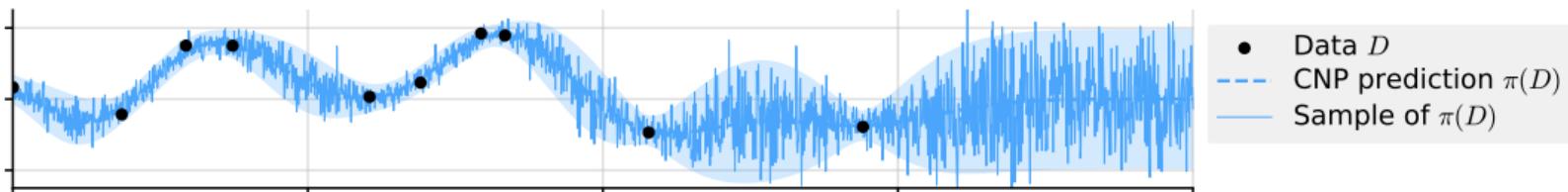
	Non-Gaussian predictions	Exact training
CNPs	✓	✓

- Prediction by a Conditional Neural Process (CNP):



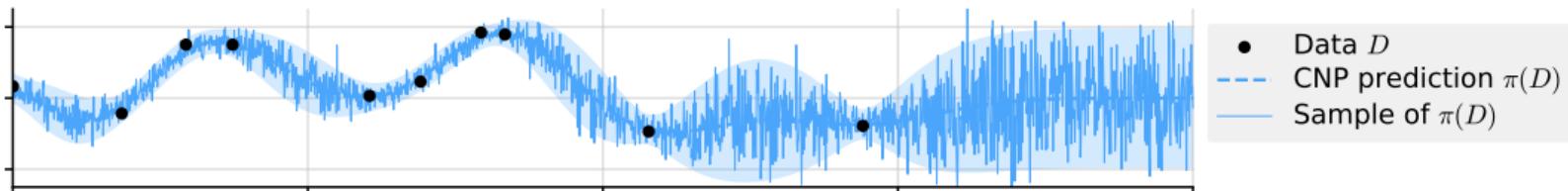
	Correlated predictions	Non-Gaussian predictions	Exact training
CNPs	✗	✓	✓

- Prediction by a Conditional Neural Process (CNP):



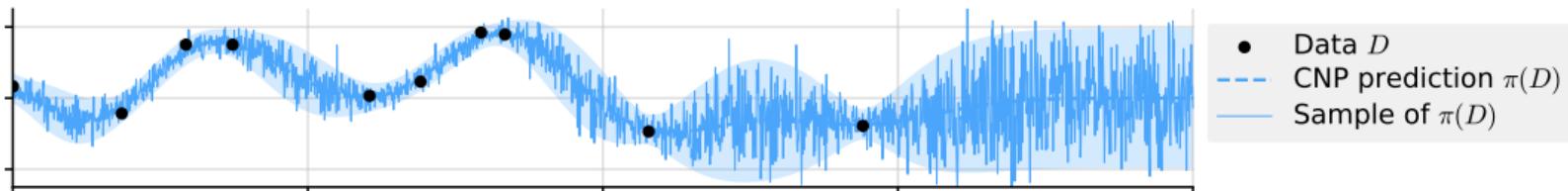
	Correlated predictions	Non-Gaussian predictions	Exact training
CNPs	✗	✓	✓
Latent-variable NPs	✓	✓	✗

- Prediction by a Conditional Neural Process (CNP):



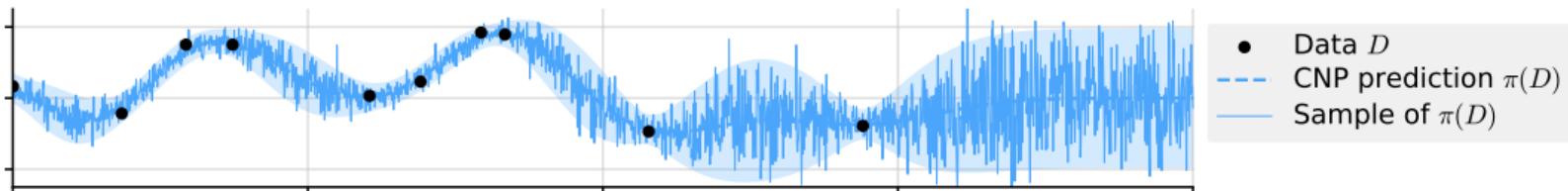
	Correlated predictions	Non-Gaussian predictions	Exact training
CNPs	✗	✓	✓
Gaussian NPs	✓	✗	✓
Latent-variable NPs	✓	✓	✗

- Prediction by a Conditional Neural Process (CNP):



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

- 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)	✓	✓	✓	✗

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

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

- 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.

- 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.
- ✓ Correlated and non-Gaussian predictions!

- 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.
- ✓ Correlated and non-Gaussian predictions!
- ✓ No modifications to model or training procedure!

- 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.
- ✓ Correlated and non-Gaussian predictions!
- ✓ No modifications to model or training procedure!
- ✗ Predictions depend on number and order of data (predictions no longer consistent)

- 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.
- ✓ 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

- 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.
- ✓ 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

- 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.
- ✓ 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

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

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

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

AR CNPs: all target points
uncountably many!

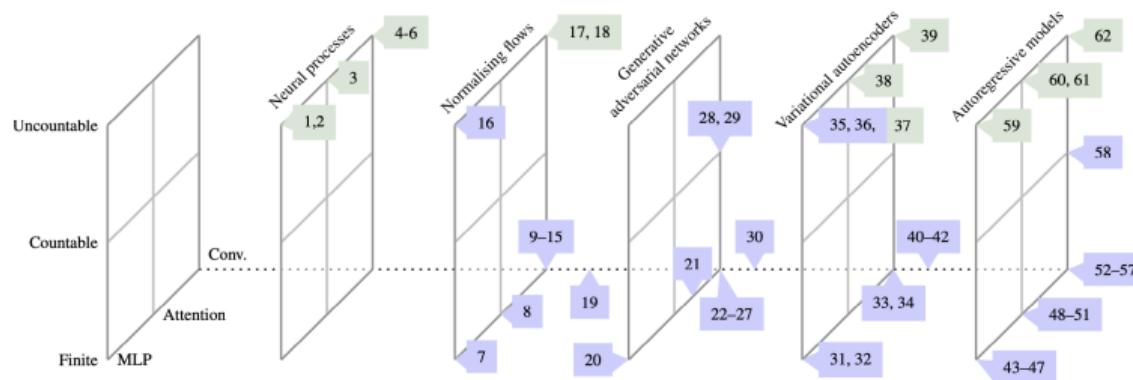
AR CNPs as a Neural Density Estimator

12/25

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

AR CNPs: all target points in any order.
uncountably many!

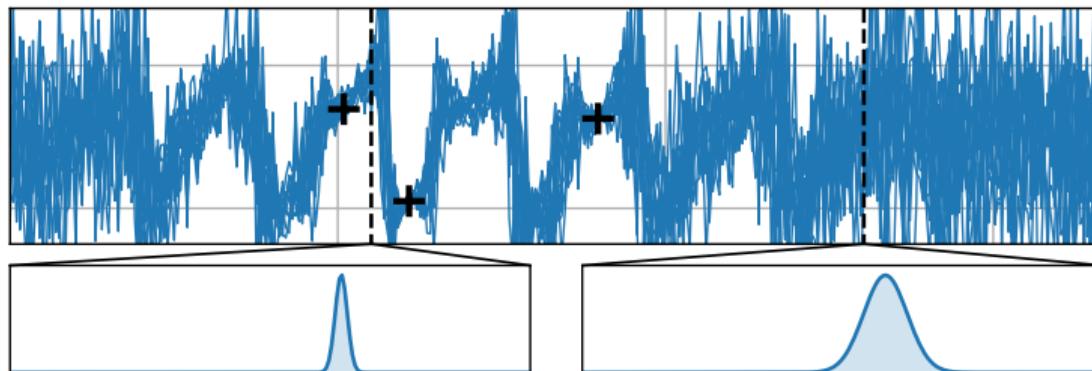
- A slightly insane diagram in the paper:



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

13/25

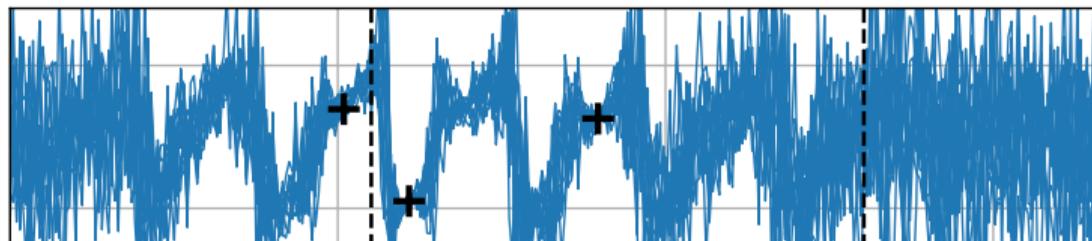
ConvCNP:



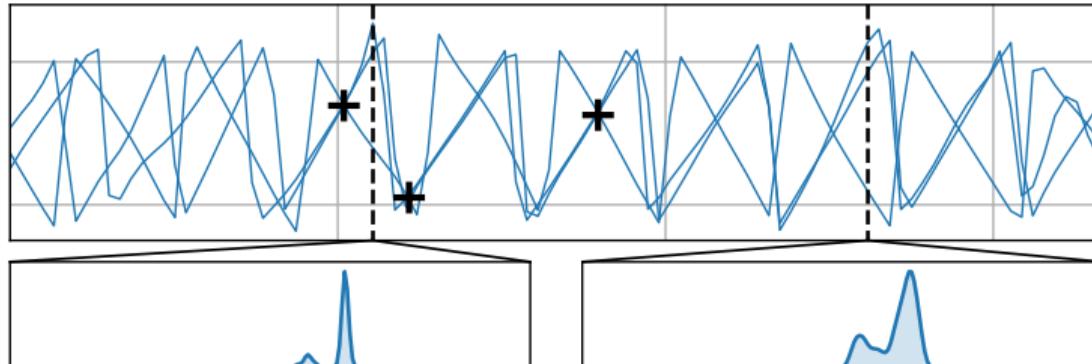
Example: ConvCNP (Gordon et al., 2020) on Sawtooth Data

13/25

ConvCNP:

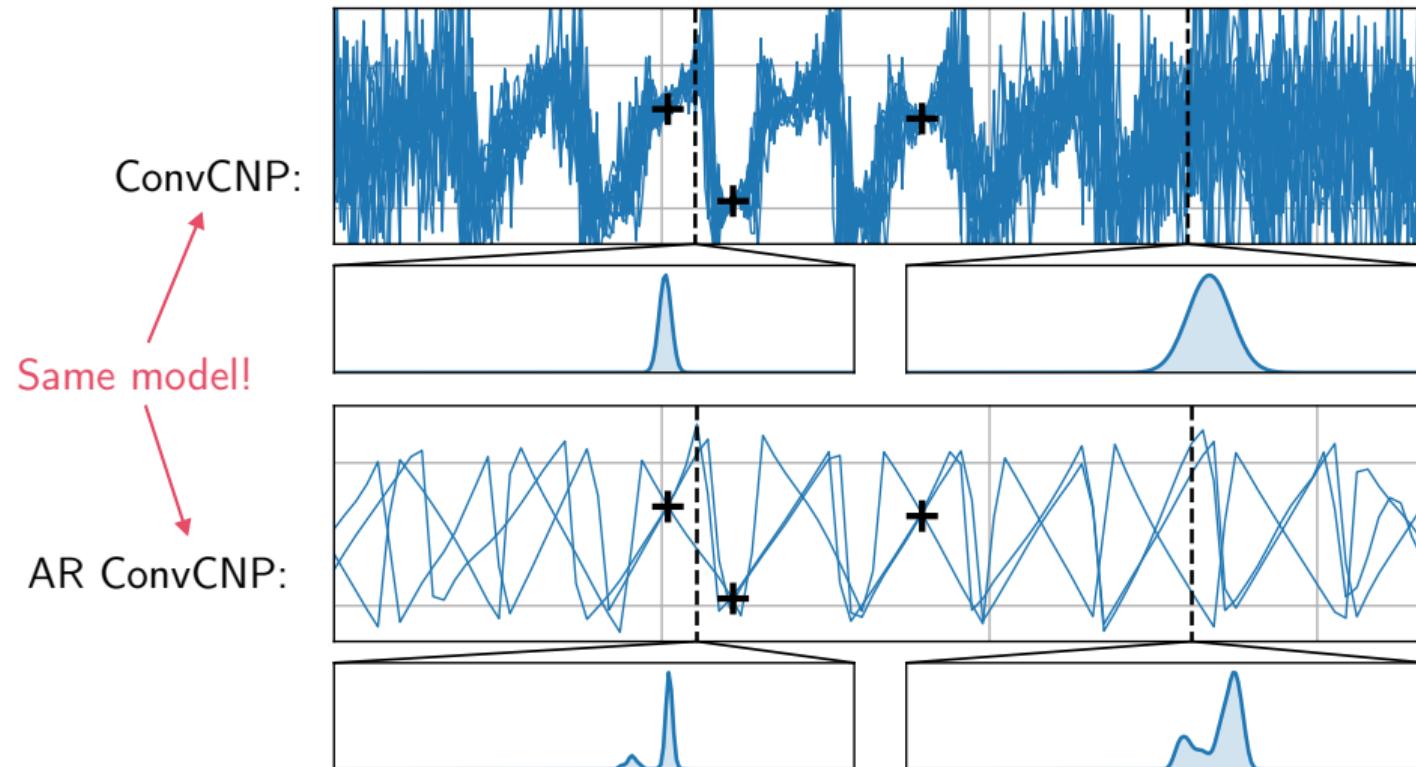


AR ConvCNP:



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

13/25



Big Surprise: AR ConvCNP Performs Really Well

14/25

	EQ		Sawtooth		Mixture	
	Norm. KL to truth (↓ better) $d_x, d_y = 1$	Norm. KL to truth (↓ better) $d_x, d_y = 2$	Norm. log-lik. (↑ better) $d_x, d_y = 1$	Norm. log-lik. (↑ better) $d_x, d_y = 2$	Norm. log-lik. (↑ better) $d_x, d_y = 1$	Norm. log-lik. (↑ better) $d_x, d_y = 2$
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
<i>Diagonal GP</i>	0.40 ± 0.01	0.40 ± 0.00				
<i>Trivial</i>	1.19 ± 0.00	0.79 ± 0.00	-0.18 ± 0.00	-0.32 ± 0.00	-1.32 ± 0.00	-1.46 ± 0.00

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

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

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

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

- $q^{(\text{CNP})}(y_1 \mid D)$ likely a poor approximation.
- **Insight:** when conditioned on many observations, the true data becomes Gaussian:
 $p(f \mid \text{many observations})$ is approximately Gaussian. (\approx Bernstein–von Mises)

Gaussian approx. becomes more accurate →

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

- $q^{(\text{CNP})}(y_1 | D)$ likely a poor approximation.
- **Insight:** when conditioned on many observations, the true data becomes Gaussian:
 $p(f | \text{many observations})$ is approximately Gaussian. (\approx Bernstein–von Mises)
⇒ $q^{(\text{CNP})}(y_i | \mathbf{y}_{1:(i-1)}, D)$ more accurate as i increases!

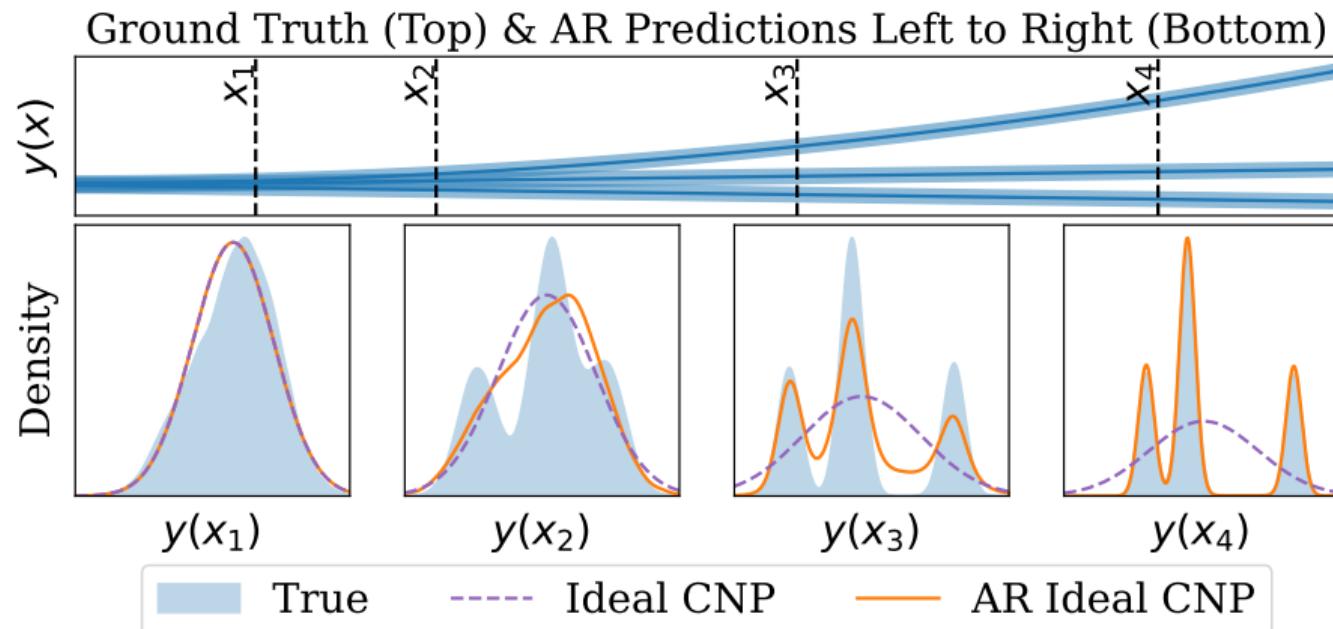
$$\xrightarrow{\text{Gaussian approx. becomes more accurate}} q^{(\text{AR CNP})}(\mathbf{y}_{1:100} | D) = q^{(\text{CNP})}(y_1 | D)q^{(\text{CNP})}(y_2 | y_1, D) \cdots q^{(\text{CNP})}(y_{100} | \mathbf{y}_{1:99}, D).$$

- $q^{(\text{CNP})}(y_1 | D)$ likely a poor approximation.
- **Insight:** when conditioned on many observations, the true data becomes Gaussian:
 $p(f | \text{many observations})$ is approximately Gaussian. (\approx Bernstein–von Mises)
⇒ $q^{(\text{CNP})}(y_i | \mathbf{y}_{1:(i-1)}, D)$ more accurate as i increases!
- First few AR steps poor, then become more accurate.

Gaussian approx. becomes more accurate →

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

- $q^{(\text{CNP})}(y_1 | D)$ likely a poor approximation.
- **Insight:** when conditioned on many observations, the true data becomes Gaussian:
 $p(f | \text{many observations})$ is approximately Gaussian. (\approx Bernstein–von Mises)
⇒ $q^{(\text{CNP})}(y_i | \mathbf{y}_{1:(i-1)}, D)$ 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.



Naively AR sampling 100 target points:

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

Naively AR sampling 100 target points:

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

- ✓ Prediction over $\mathbf{y}_{1:100}$ correlated!
- ✗ Requires 100 model forwards

Naively AR sampling 100 target points:

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

- ✓ Prediction over $\mathbf{y}_{1:100}$ correlated!
- ✗ Requires 100 model forwards

Sample in blocks of 10 points:

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

Naively AR sampling 100 target points:

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

- ✓ Prediction over $\mathbf{y}_{1:100}$ correlated!
- ✗ Requires 100 model forwards

Sample in blocks of 10 points:

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

- ✗ No correlations within a block
- ✓ Only 10 model forwards!

Naively AR sampling 100 target points:

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

- ✓ Prediction over $\mathbf{y}_{1:100}$ correlated!
- ✗ Requires 100 model forwards

Sample in blocks of 10 points:

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

- ✗ No correlations within a block
- ✓ Only 10 model forwards!

Run a GNP in AR mode:

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

Naively AR sampling 100 target points:

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

- ✓ Prediction over $\mathbf{y}_{1:100}$ correlated!
- ✗ Requires 100 model forwards

Sample in blocks of 10 points:

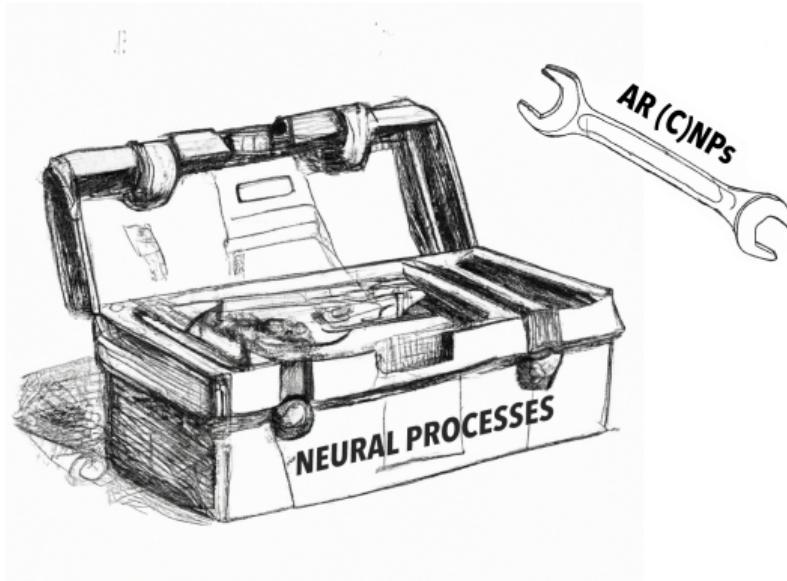
$$q^{(\text{AR CNP})}(\mathbf{y}_{1:100} | D) = q^{(\text{CNP})}(\mathbf{y}_{1:10} | D)q^{(\text{CNP})}(\mathbf{y}_{11:20} | \mathbf{y}_{1:10}, D) \cdots q^{(\text{CNP})}(\mathbf{y}_{91:100} | \mathbf{y}_{1:90}, D).$$

- ✗ No correlations within a block
- ✓ Only 10 model forwards!

Run a GNP in AR mode:

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

- ✓ Correlations within a block!
- ✓ Only 10 model forwards!



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.

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

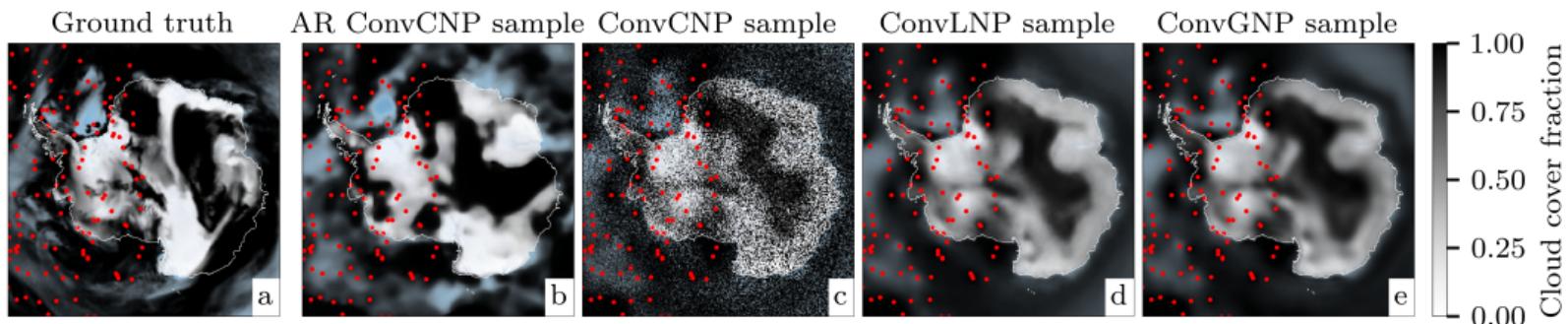
$$q(\mathbf{y} \mid \mathbf{x}, D) = \prod_{i=1}^{|\mathbf{y}|} \begin{cases} p_{0,\theta} & \text{if } y_i = 0, \\ p_{1,\theta} & \text{if } y_i = 1, \\ (1 - p_{0,\theta} - p_{1,\theta}) \text{Beta}(y_i; \alpha_\theta, \beta_\theta) & \text{if } y_i \in (0, 1). \end{cases}$$

$\uparrow \quad \beta_\theta = \beta_\theta(x_i, D), \text{ et cetera}$

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

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

$\uparrow \quad \beta_\theta = \beta_\theta(x_i, D), \text{ et cetera}$



Prediction Map Approximation: A Theoretical Analysis

AR CNPs:

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

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?

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?

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?

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?

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>

- Prediction map: $\pi: \mathcal{D} \rightarrow \mathcal{Q}$.
 - e.g., CNPs choose all GPs with independent predictions
- Posit a ground truth stochastic process f , possibly non-Gaussian.
- Posterior prediction map: $\pi_f: \mathcal{D} \rightarrow \mathcal{P}$, $\pi_f(D) = p(f | D)$.
- Approximate π_f with a neural process $\pi_\theta: \mathcal{D} \rightarrow \mathcal{Q}$.
 - $f(\mathbf{x}) + \boldsymbol{\varepsilon} \sim P_{\mathbf{x}}^\sigma \mu$ with $f \sim \mu$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$
- Do this by minimising the neural process objective \mathcal{L}_{NP} :
 - $\hat{\theta} \in \arg \min \mathcal{L}_{\text{NP}}(\pi_f, \pi_\theta)$, $\mathcal{L}_{\text{NP}}(\pi_f, \pi_\theta) = \mathbb{E}_{p(\mathbf{x})p(D)}[\text{KL}(P_{\mathbf{x}}^{\sigma_f} \pi_f(D), P_{\mathbf{x}}^\sigma \pi_\theta(D))]$.
- Study minimisers for CNPs and GNPs.
 - convergence to minimiser (consistency); compare minimisers for CNPs and GNPs

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

- 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}_{NP} for CNPs and GNPs.

- 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}_{NP} for CNPs and GNPs.
⇒ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.

- 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}_{NP} for CNPs and GNPs.
 - ⇒ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.
 - ⇒ CNPs cannot disentangle epistemic and aleatoric uncertainty, but GNPs can!

- 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}_{NP} for CNPs and GNPs.
 - ⇒ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.
 - ⇒ CNPs cannot disentangle epistemic and aleatoric uncertainty, but GNPs can!
- Props 3.34 and 3.35: Precise conditions for consistency of CNPs and GNPs.

- **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}_{NP} for CNPs and GNPs.
 - ⇒ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.
 - ⇒ 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.

- **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}_{NP} for CNPs and GNPs.
 - ⇒ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.
 - ⇒ 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.

- 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}_{NP} for CNPs and GNPs.
 - ⇒ CNPs need target set size of at least one, and GNPs need two. LNPs may need infinite.
 - ⇒ 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!

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

- 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).

- 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 $1/\sqrt{M}$.

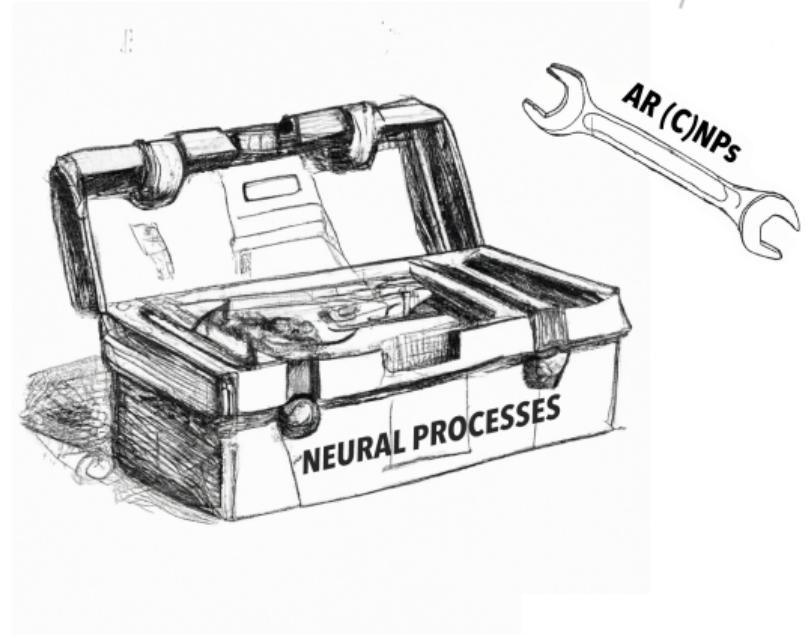
- Do predictions of AR CNPs converge to the ground truth in some sense?
- Some theoretical wrinkles for ConvDeepSets (Gordon et al., 2020) to iron out.
- Unclarity around the representation capacity of ConvGNPs (Markou et al., 2022).
- Rate of convergence w.r.t. meta-data set size M ? Suspect $1/\sqrt{M}$.
- Analysis in setting of infinitely wide neural networks. Finite widths?

- 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

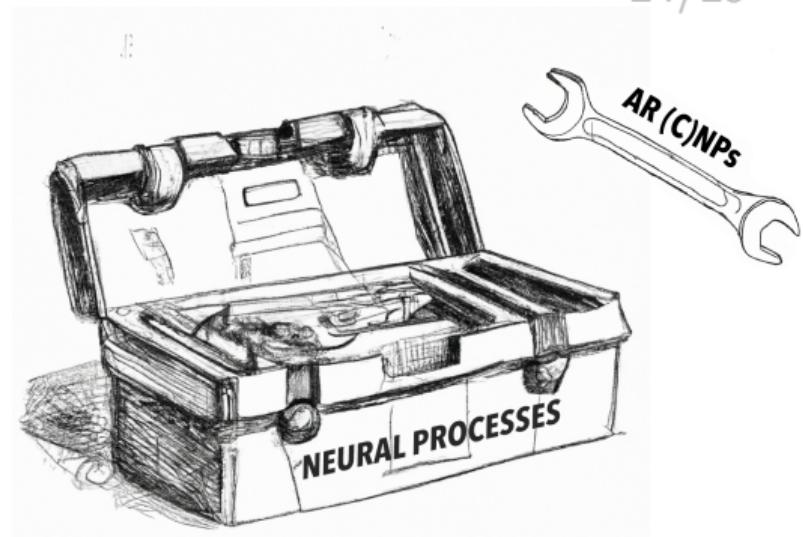
Neural processes:

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



Neural processes:

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



Paper:
wessel.ai/pdf/arcnps

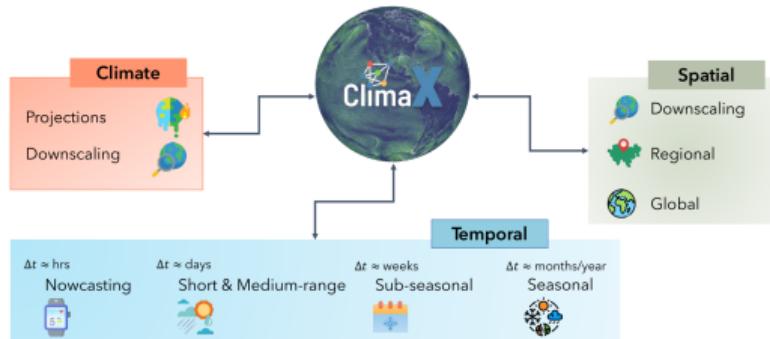
This presentation:
wessel.ai/pdf/arcnps-mlls

Code:
github.com/wesselb/neuralprocesses

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



- 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>.