Secondmind

UNIVERSITY OF CAMBRIDGE

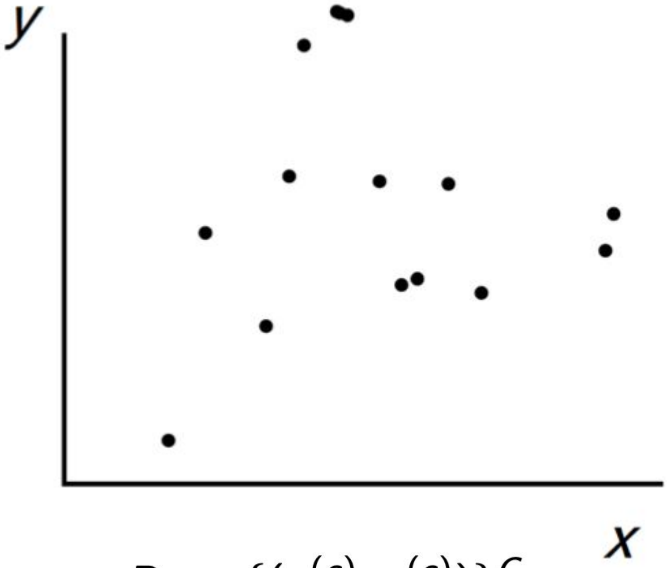
Neural Diffusion Processes

Generative Modelling and Uncertainty Quantification

Copenhagen – 2022

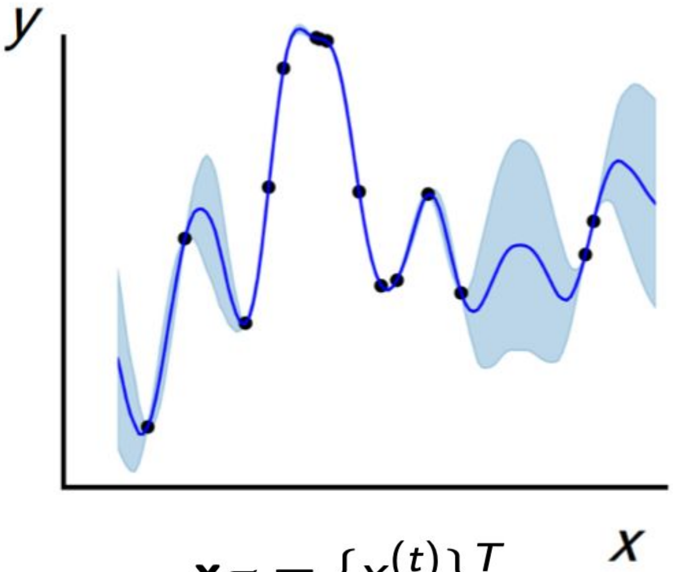
Vincent Dutordoir

Work in progress. V1 on Arxiv.



$$D_{\mathcal{C}} = \{(x^{(c)}, y^{(c)})\}_{c=1}^{C}$$

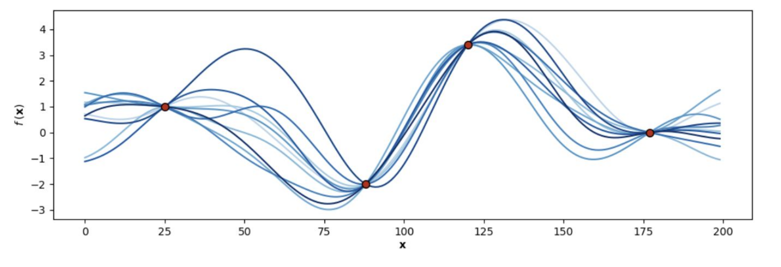
$$\mapsto$$



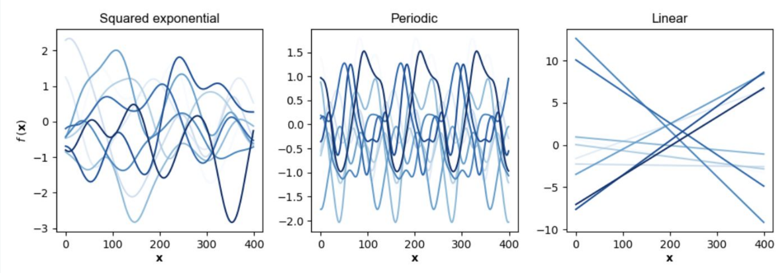
$$\mathbf{x}_{\mathcal{T}} = \{x^{(t)}\}_{t=1}^{T}$$

Illustration: Dubois et al. Neural Process Family. *Blogpost*. 2020

Gaussian Processes Regression



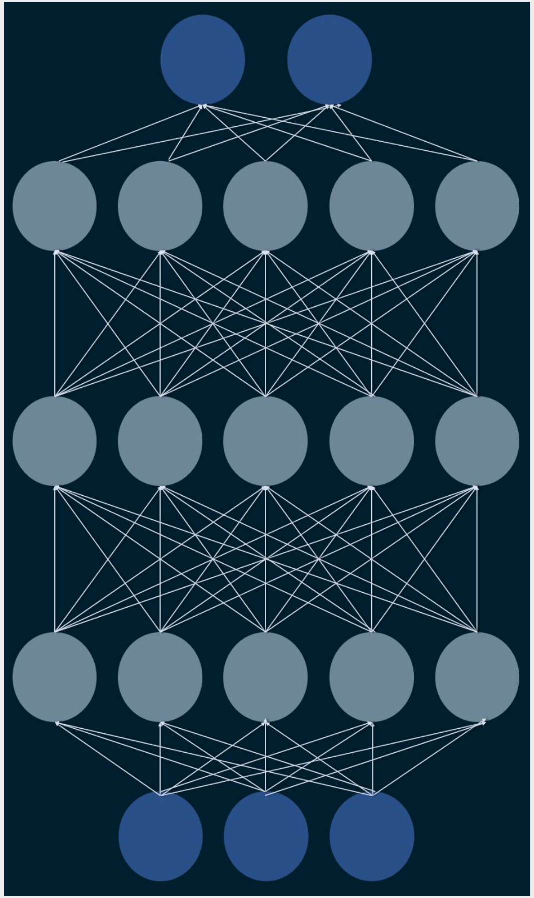
Gaussianity



A priori modelling decisions, such as kernels and, hyper-parameters.

Vision

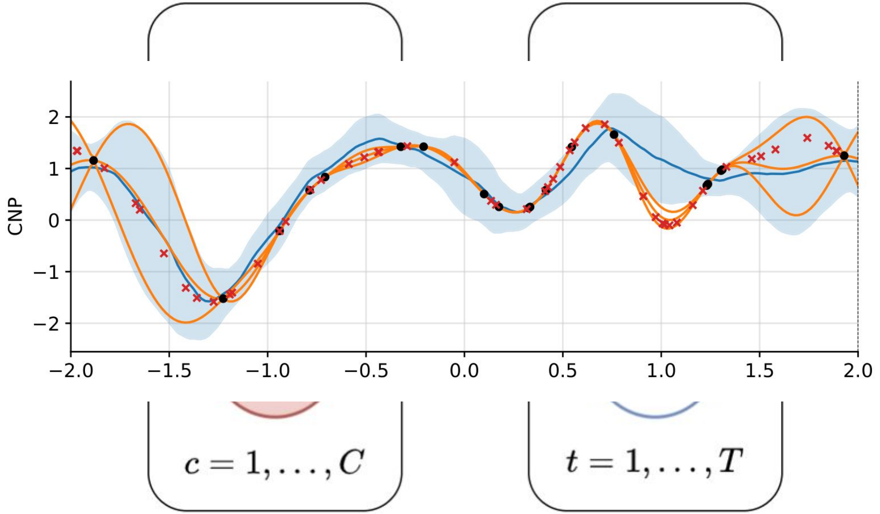
1. Amortize Bayesian inference using a large neural network.
2. The network 'eats' the entire dataset.
3. Train in a *meta-learning* fashion on many datasets. We have **infinite** amount of synthetic datasets available using GPs.
4. Potentially fine-tune on specific tasks.



$$D_{\mathcal{C}} = \{(x^{(c)}, y^{(c)})\}_{c=1}^{C}$$

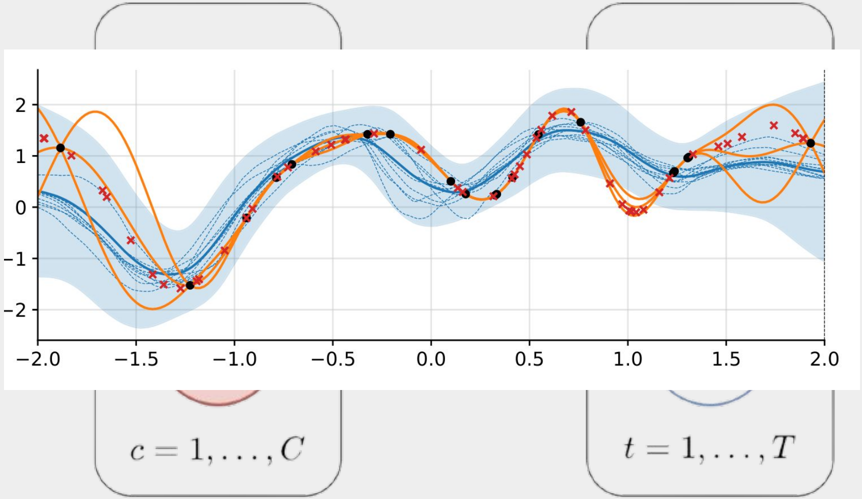
Neural Processes – Garnelo et al. 2018

Conditional Neural Processes



$$p(\mathbf{y}_{\mathcal{T}}|\mathbf{x}_{\mathcal{T}}, D_{\mathcal{C}}) = \prod_{t=1}^{T} p(y_t|x^{(t)}, R(D_{\mathcal{C}})).$$

Latent Neural Processes



$$p(\mathbf{y}_{\mathcal{T}}|\mathbf{x}_{\mathcal{T}}, D_{\mathcal{C}}) = \int \prod_{t=1}^{T} p(y_t|x^{(t)}, \mathbf{z}) p(\mathbf{z}|R(D_{\mathcal{C}}))\, \mathrm{d}\mathbf{z}.$$

Graphical models: Dubois et al. 2020

“Deep learning has landed straight in our backyard”

– Fergus Simpson

Diffusion Models

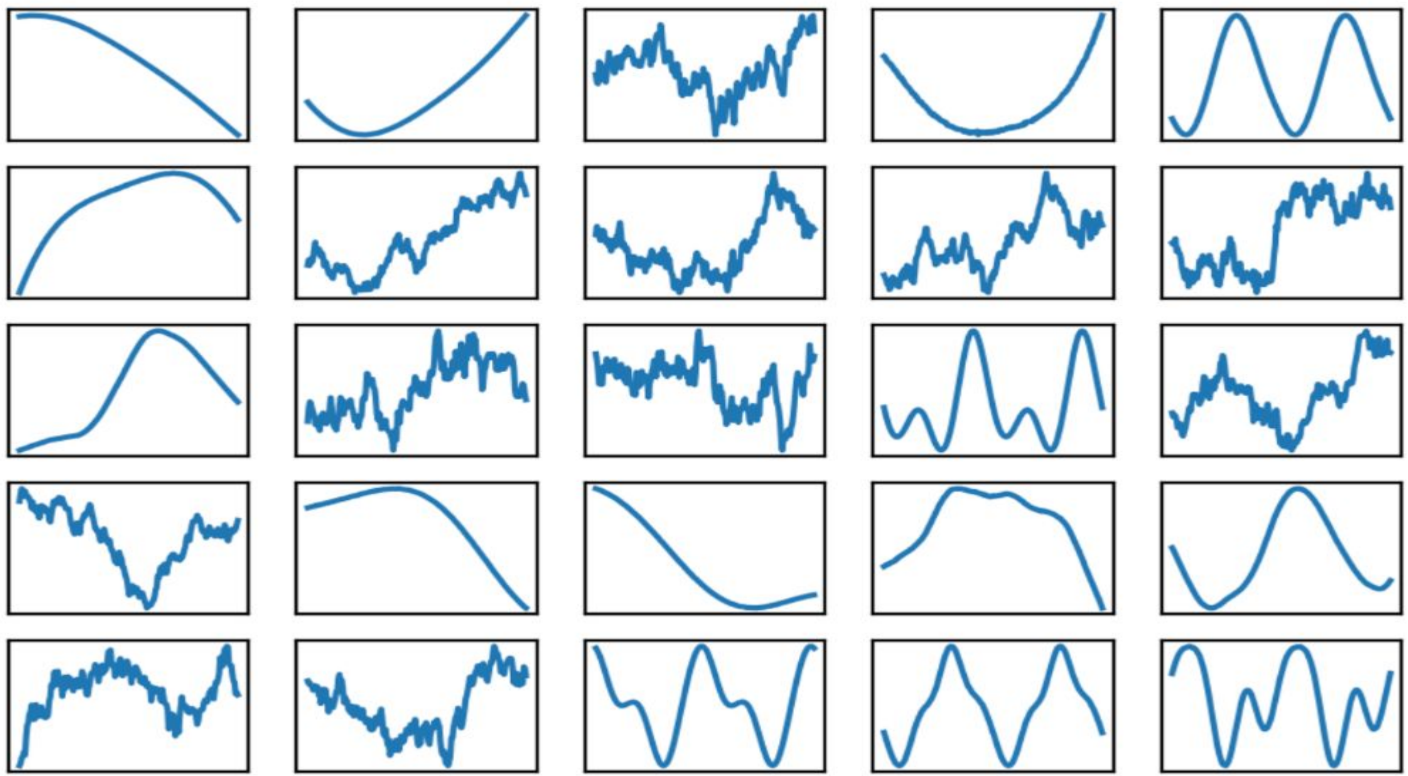
Source: Arno Solin's 3-year-old daughter using Stable Diffusion.

Diffusion Models

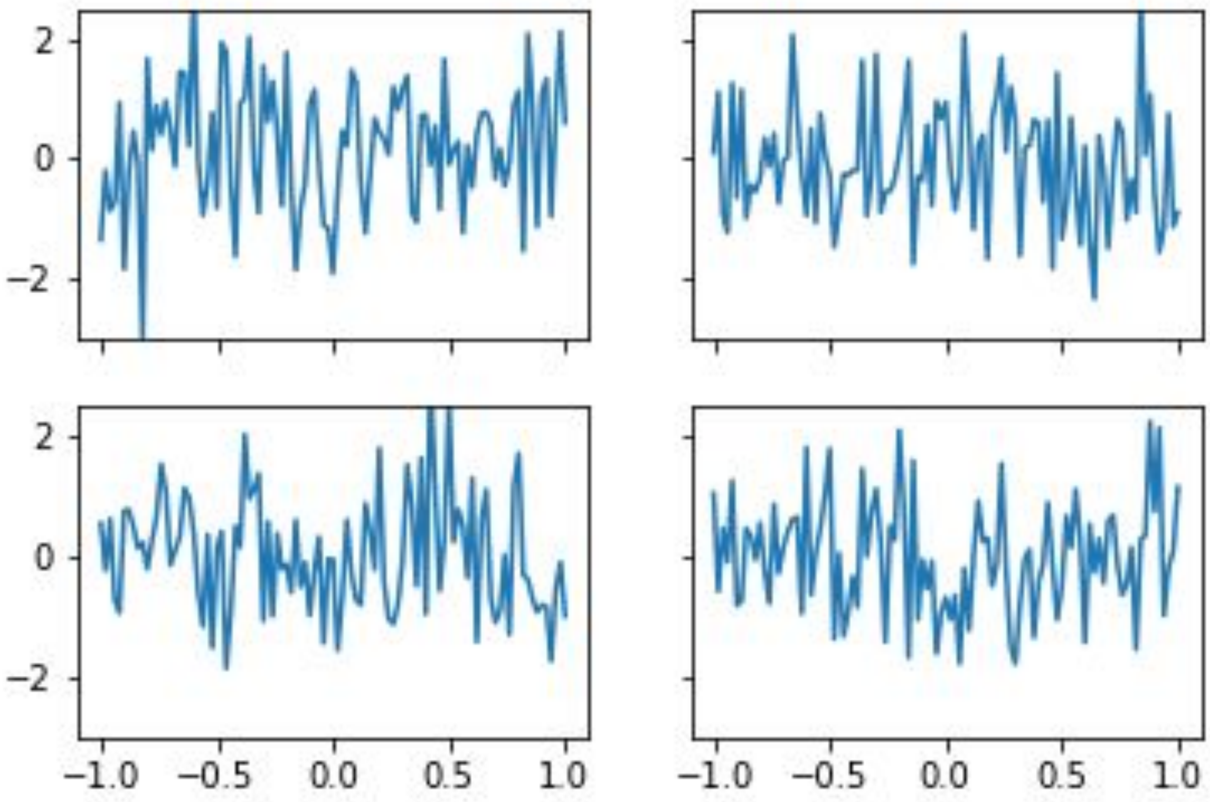
A dragon fruit wearing karate belt in the snow.

A small cactus wearing a straw hat and neon sunglasses in the Sahara desert.

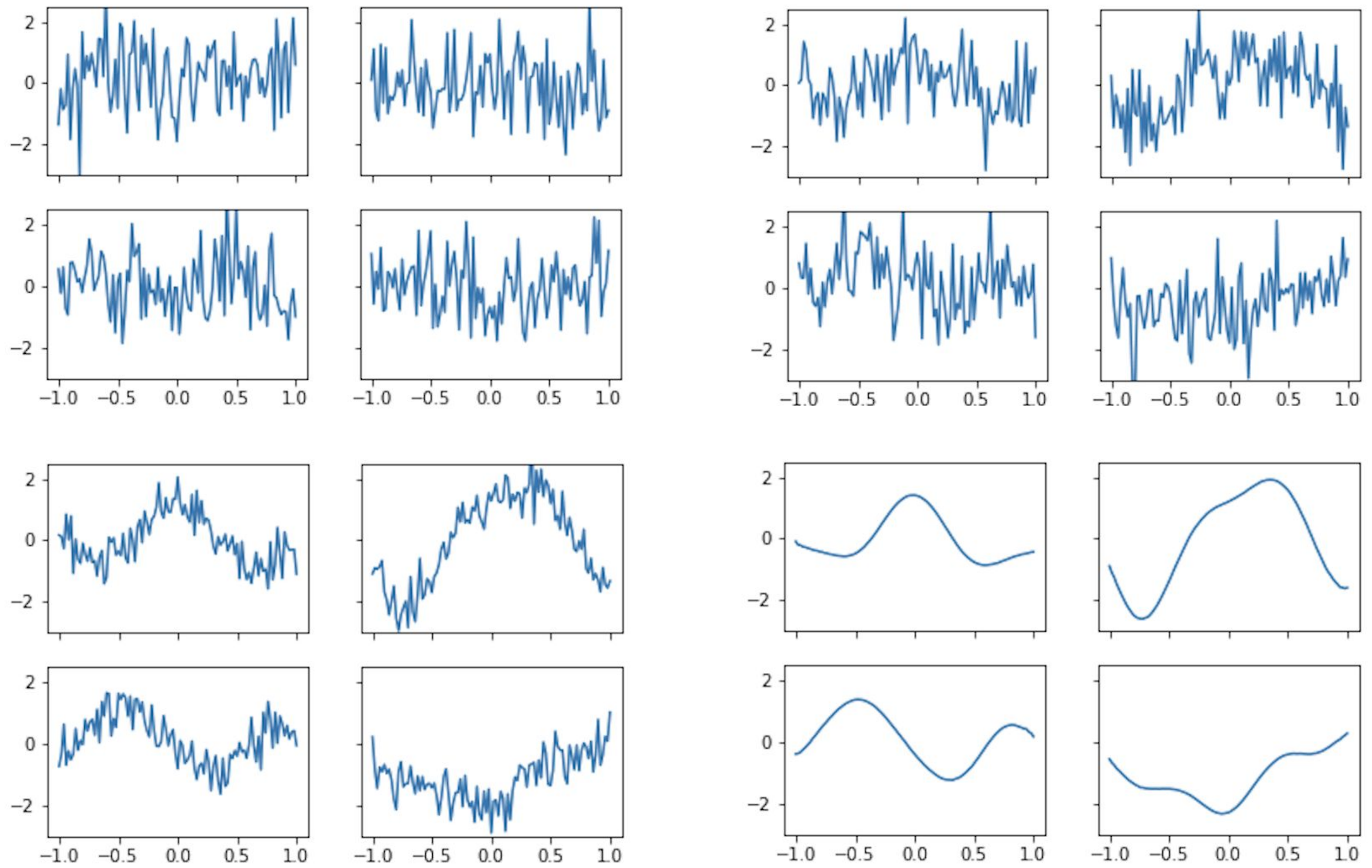
A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat.



Proof-of-Concept Experiment



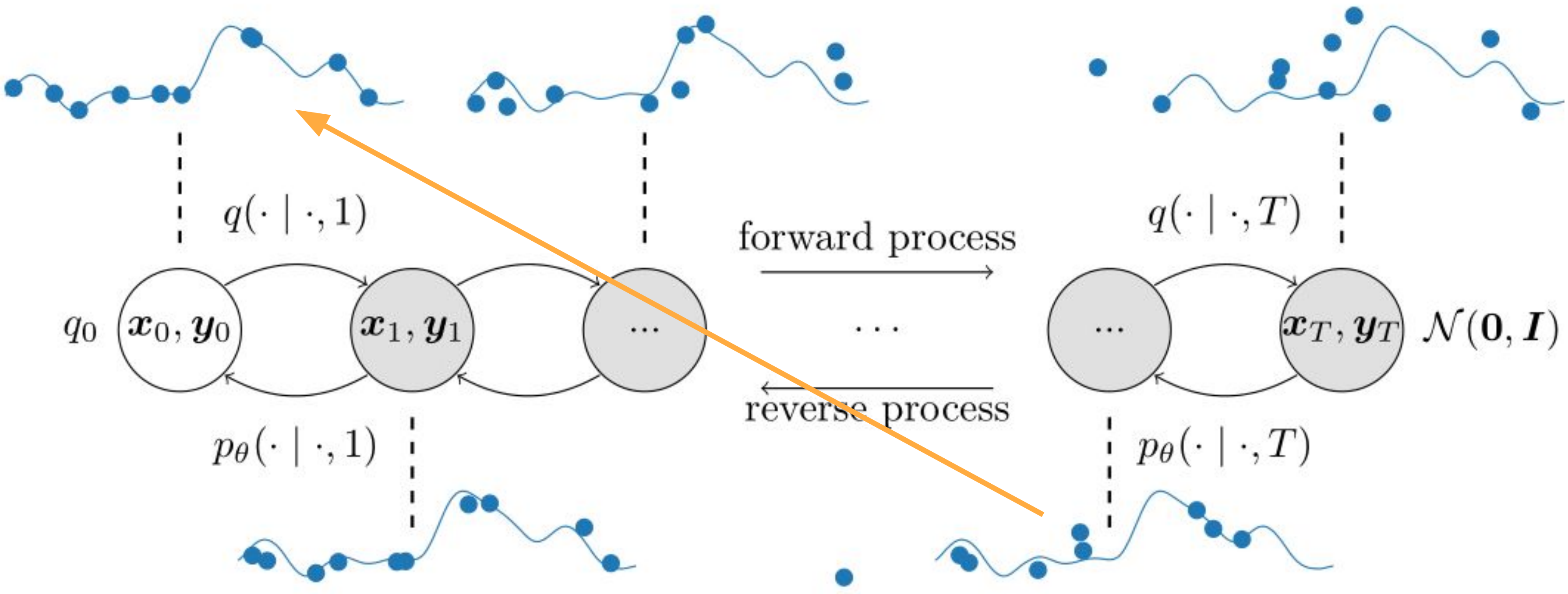
Proof-of-Concept Experiment



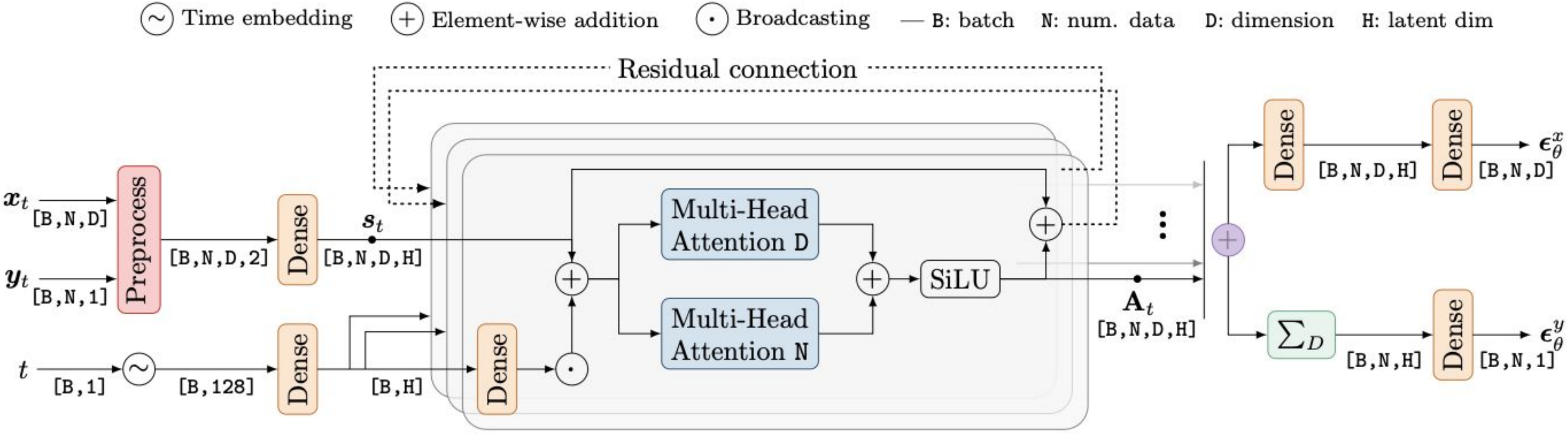
Difficulties of stochastic processes

1. We require samples that can be evaluated at arbitrary locations in the input domain.
2. **Exchangeability**: the joint probability distribution does not change when the order of function evaluations is altered.
3. (Marginal) **Consistency** $p(f_1) = \int p(f_1, f_2)\, df_2.$

Diffusion models for stochastic processes



NDP's Noise model

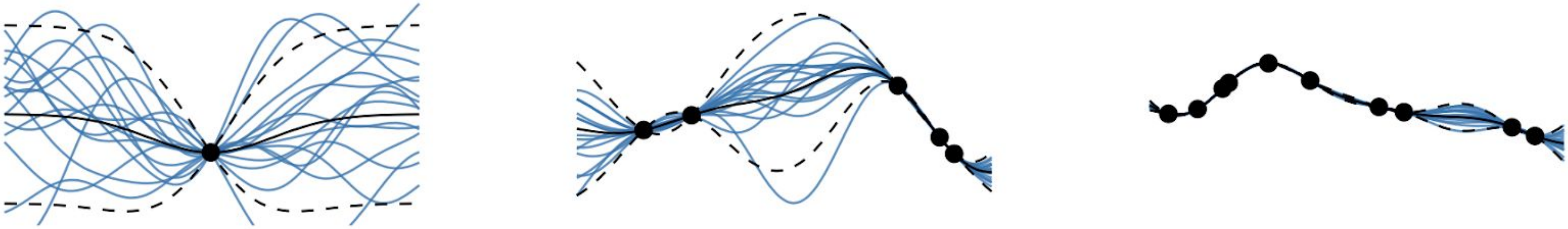


- **Equivariant** to the dataset ordering (N)
- **Invariant** to the feature ordering (D)

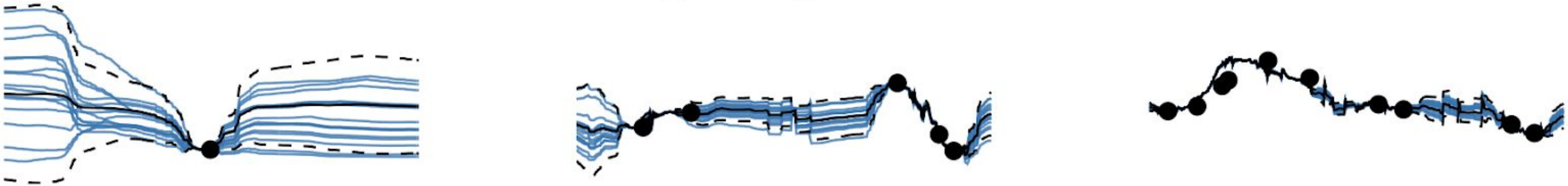
Conditional Sampling

(a) GP Regression (b) Attentive Latent NP (c) Neural Diffusion Process (ours)

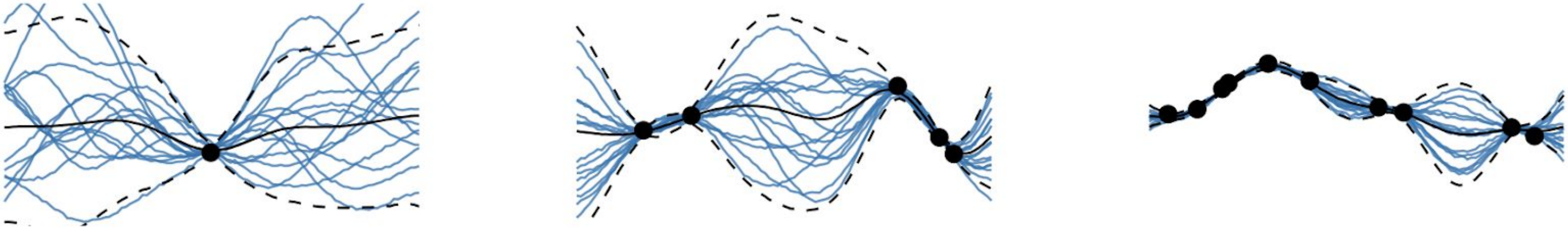
We use a technique from image inpainting to create a conditional sample which is consistent with the context dataset and coherent among itself.



(a) GP Regression

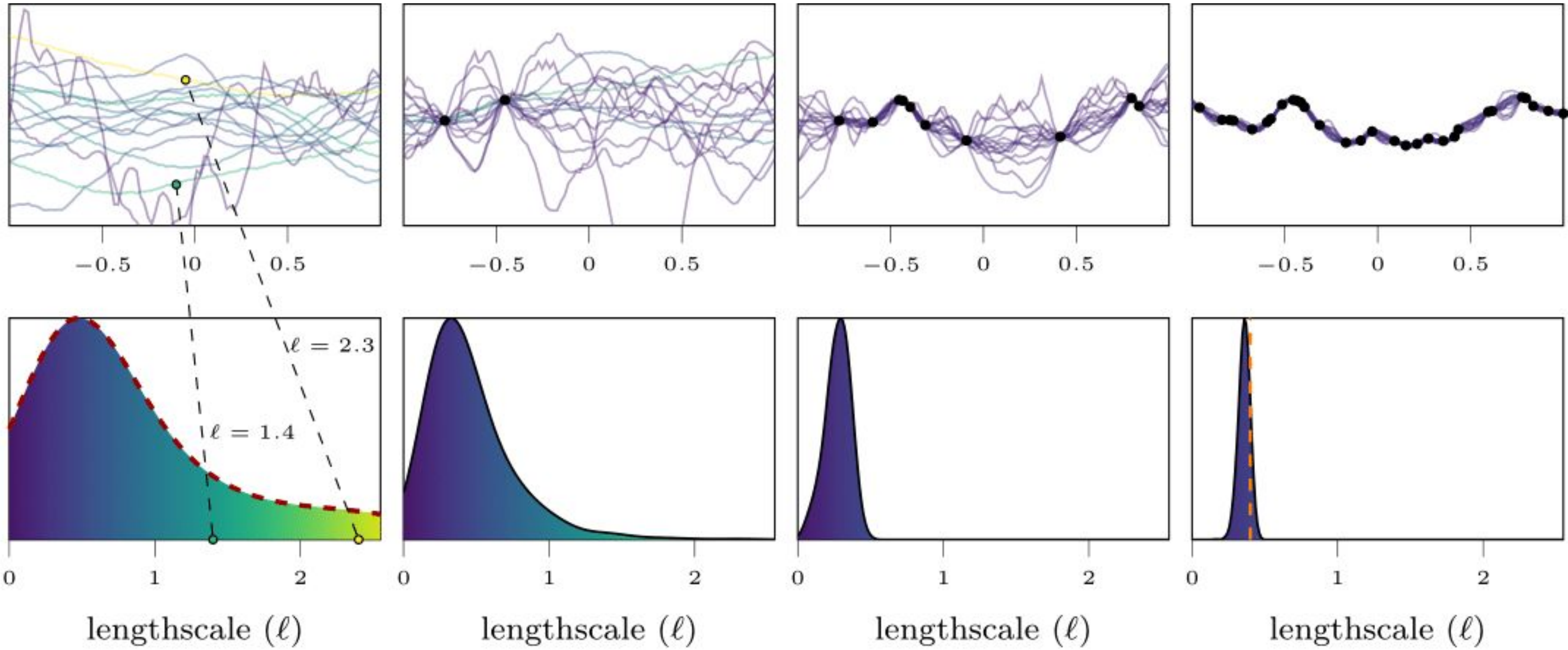


(b) Attentive Latent Neural Process

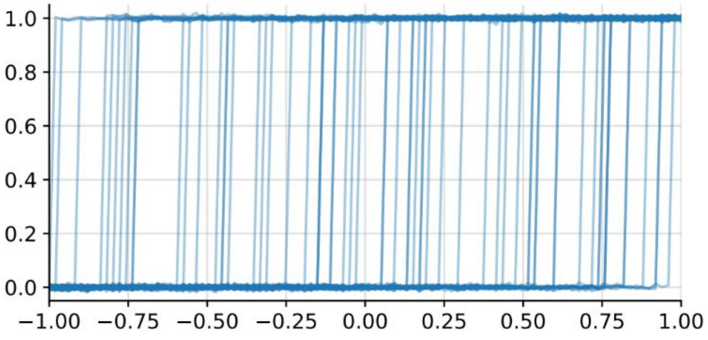


(c) Neural Diffusion Process (ours)

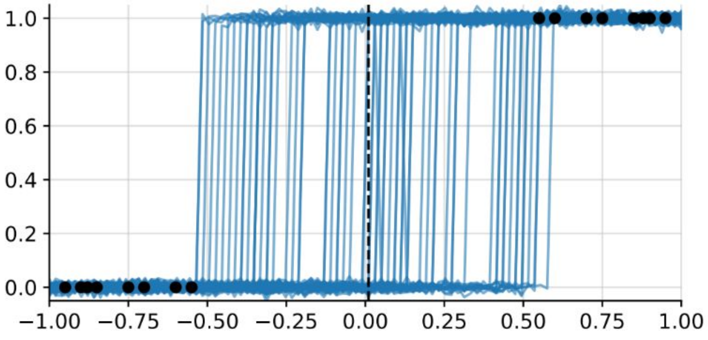
Hyperparameter Marginalization



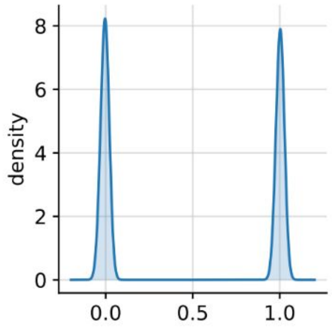
Non-Gaussian Marginals



(a) Prior



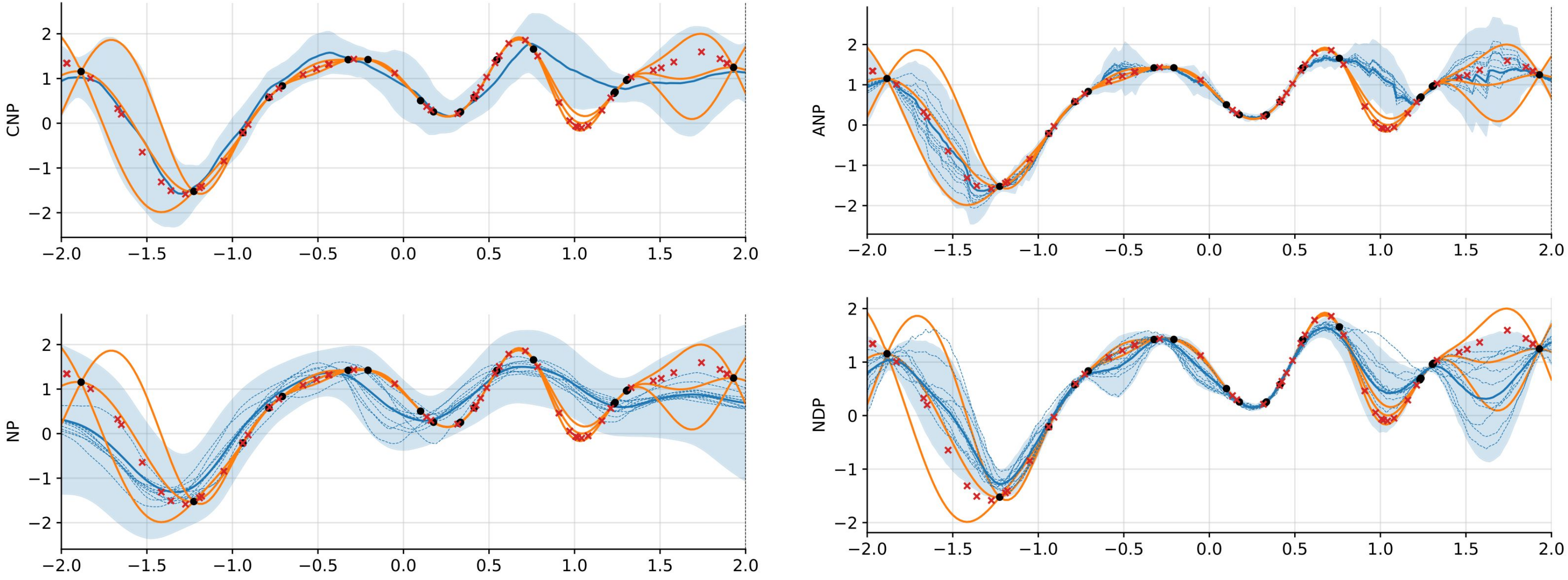
(b) Conditional



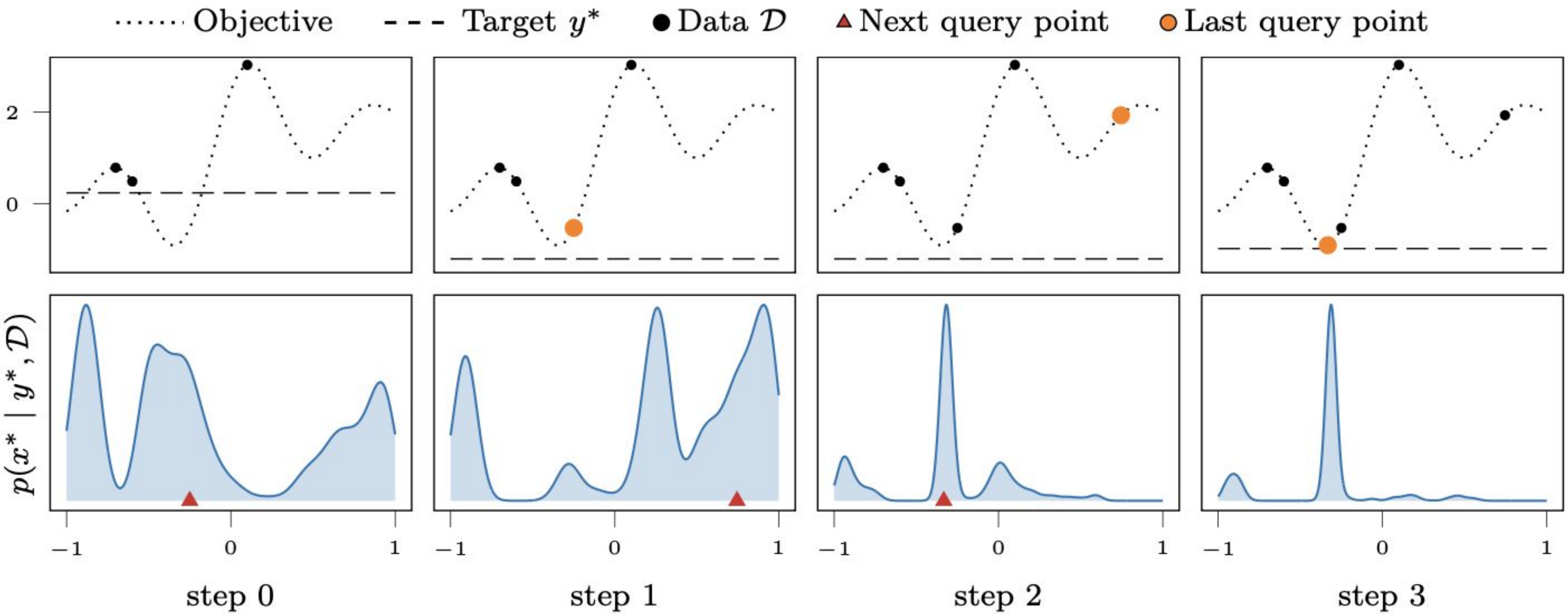
(c) $p(y \mid x = 0.0)$

Regression

	Squared Exponential			Matérn		
	$D_x = 1$	$D_x = 2$	$D_x = 3$	$D_x = 1$	$D_x = 2$	$D_x = 3$
NDP	**−0.38**±0.05	**1.01**±0.03	**1.20**±0.01	**0.13**±0.05	**1.15**±0.02	**1.19**±0.01
ANP	0.29±0.10	**1.05**±0.06	1.25±0.03	0.60±0.07	**1.14**±0.05	1.29±0.02
NP	0.67±0.06	1.23±0.04	1.35±0.02	0.84±0.04	1.26±0.03	1.36±0.01
CNP	0.77±0.09	1.26±0.05	1.35±0.02	0.91±0.07	1.30±0.04	1.37±0.02
trivial	1.41±0.03	1.42±0.02	1.45±0.02	1.43±0.02	1.43±0.02	1.45±0.02



Global Optimisation



Vision

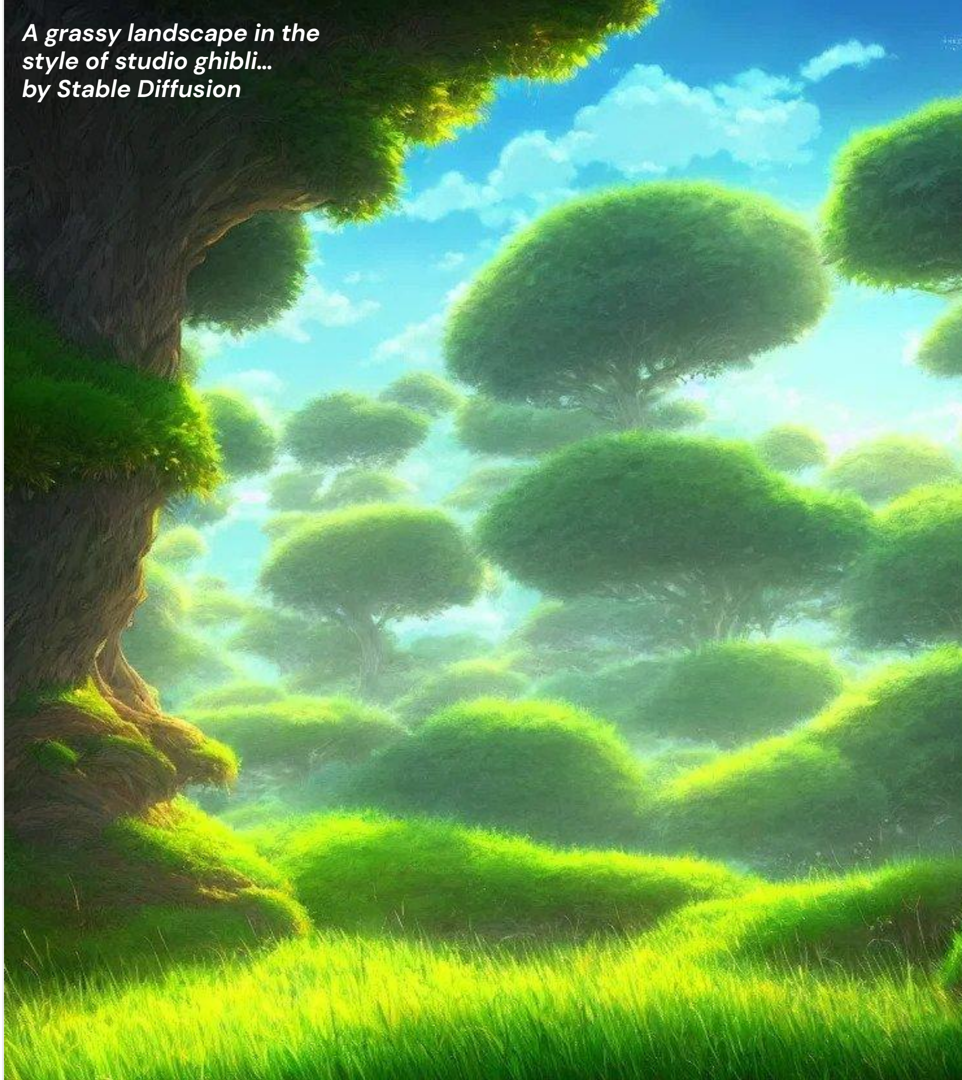
Instantaneous Bayesian inference.

- No need to train a new ‘model’ (GP or Neural Network).

- Amortized training once. Inference becomes a simple forward pass.

Next Steps

- Larger datasets. Sparse attention? Work in Hz domain?
- Faster sampling
- Accurate likelihood estimations
- Noise model? $p(y \mid f) = N(y \mid f, 1e{-6})$?



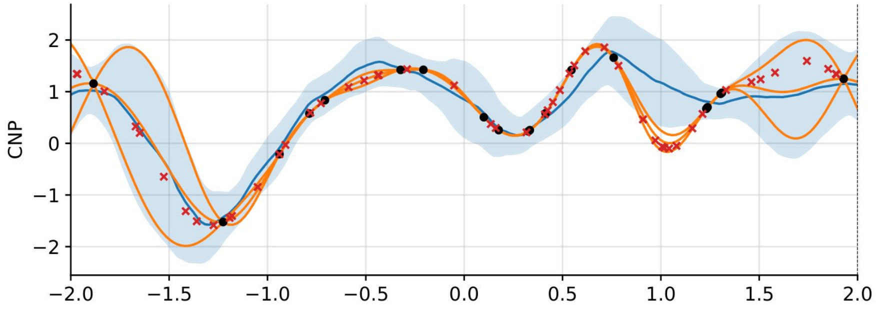
A grassy landscape in the style of studio ghibli…
by Stable Diffusion

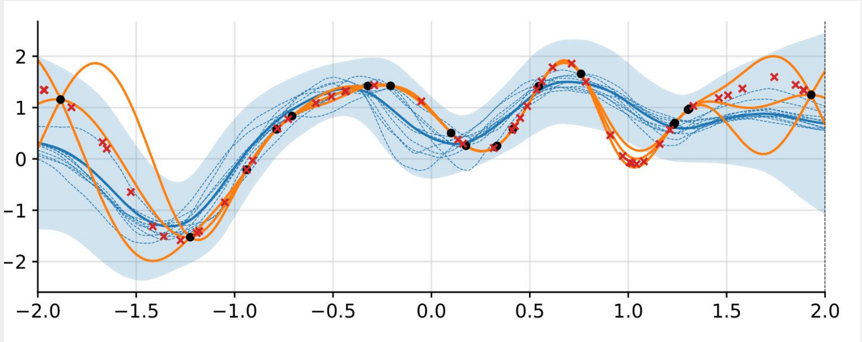
Thank you
for your attention.

Neural Processes – Garnelo et al. 2018

Conditional Neural Processes

Latent Neural Processes





$$p(\mathbf{y}_{\mathcal{T}}|\mathbf{x}_{\mathcal{T}}, D_{\mathcal{C}}) = \prod_{t=1}^{T} p(y_t|x^{(t)}, R(D_{\mathcal{C}})).$$

$$p(\mathbf{y}_{\mathcal{T}}|\mathbf{x}_{\mathcal{T}}, D_{\mathcal{C}}) = \int \prod_{t=1}^{T} p(y_t|x^{(t)}, \mathbf{z}) p(\mathbf{z}|R(D_{\mathcal{C}})) \, \mathrm{d}\mathbf{z}.$$

Graphical models: Dubois et al. 2020