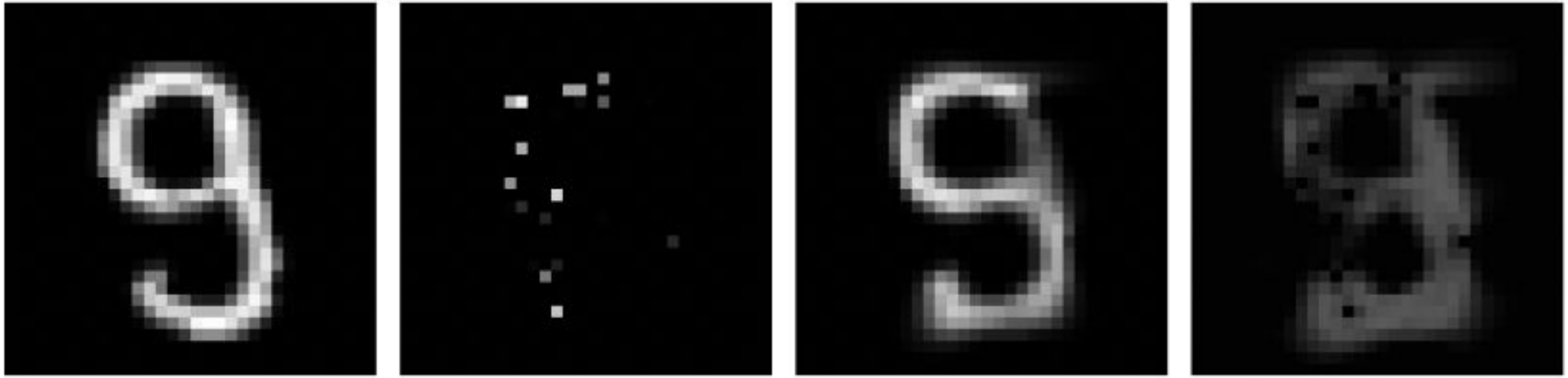
The Neural Process Family

Survey, Applications and Perspectives

Saurav Jha, Dong Gong, Xuesong Wang, Richard Turner, Lina Yao

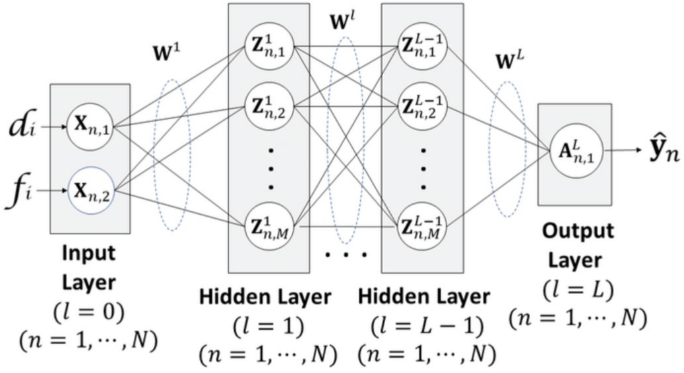


Original Corrupted Reconstructed Uncertainty

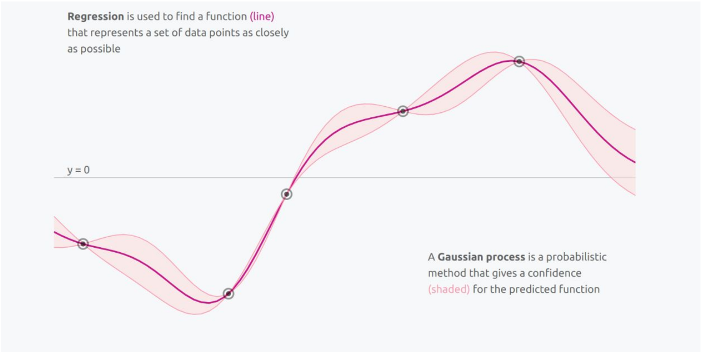
Introduction

- Neural Processes vs Gaussian Processes
- How do Neural Processes work?
- Deep sets and the underlying theory
- The Neural Process Family
- Applications

What are Neural Processes



+

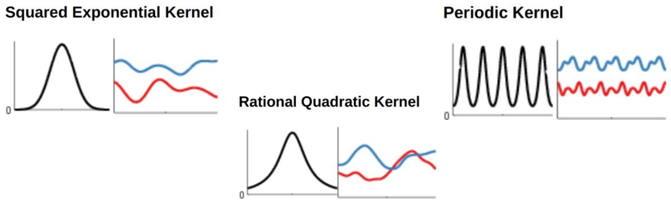


Source: Park et al., 2019
Gortler et al., 2019

In Comparison of GPs

<u>Prior</u>

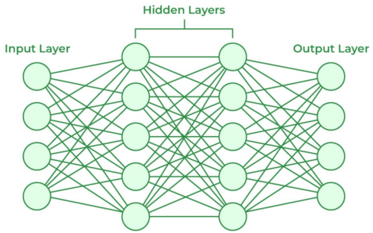
GPs



You pick

=> **GP**

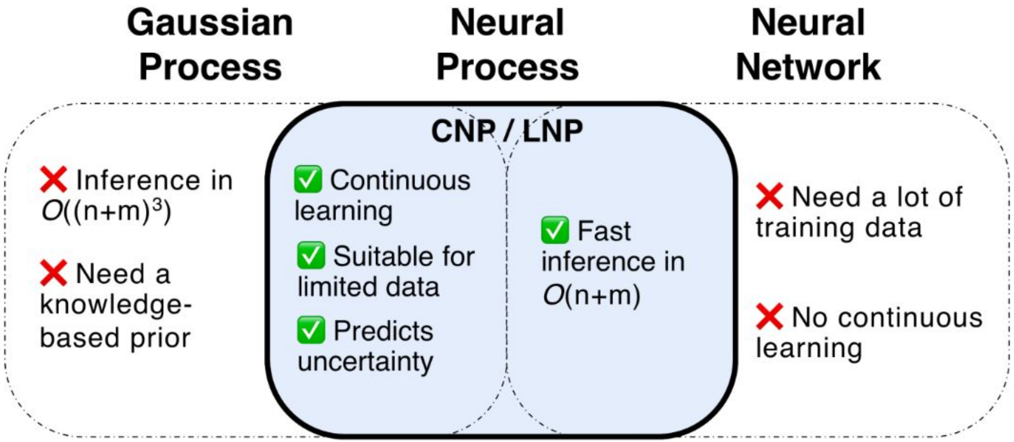
NPs



=> **Learned prior** => **NP**

You learn

Source: Duvenaud, Kernel Cookbook



Source: Baker et al., 2022

From GPs to NPs

Some Notation and Recap

$$C = \{(x_i, y_i)\}_{i=1}^{n-1} \subset \mathcal{X} \times \mathcal{Y}$$ Context set

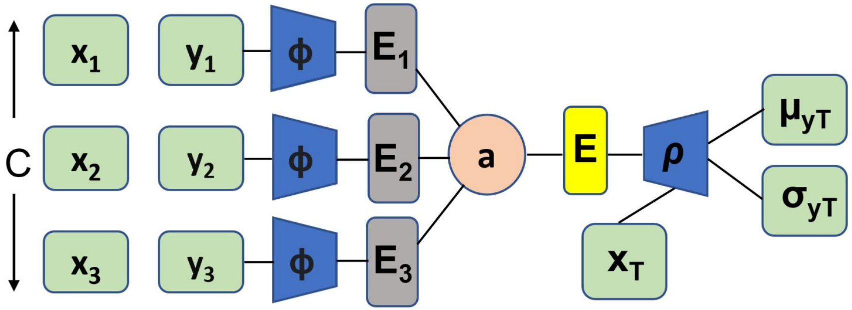
$$T = \{x_i\}_{i=n}^{n+m-1} \subset \mathcal{X}$$ Target set

In practice we train over multiple datasets of context and target sets.

$$GP \sim \mathcal{N}(\mu, \Sigma)$$ Gaussian process

$$\mathbb{P}(GP(T)|T, C) \sim \mathcal{N}(\mu_{\text{post}}, \Sigma_{\text{post}})$$ Gaussian posterior

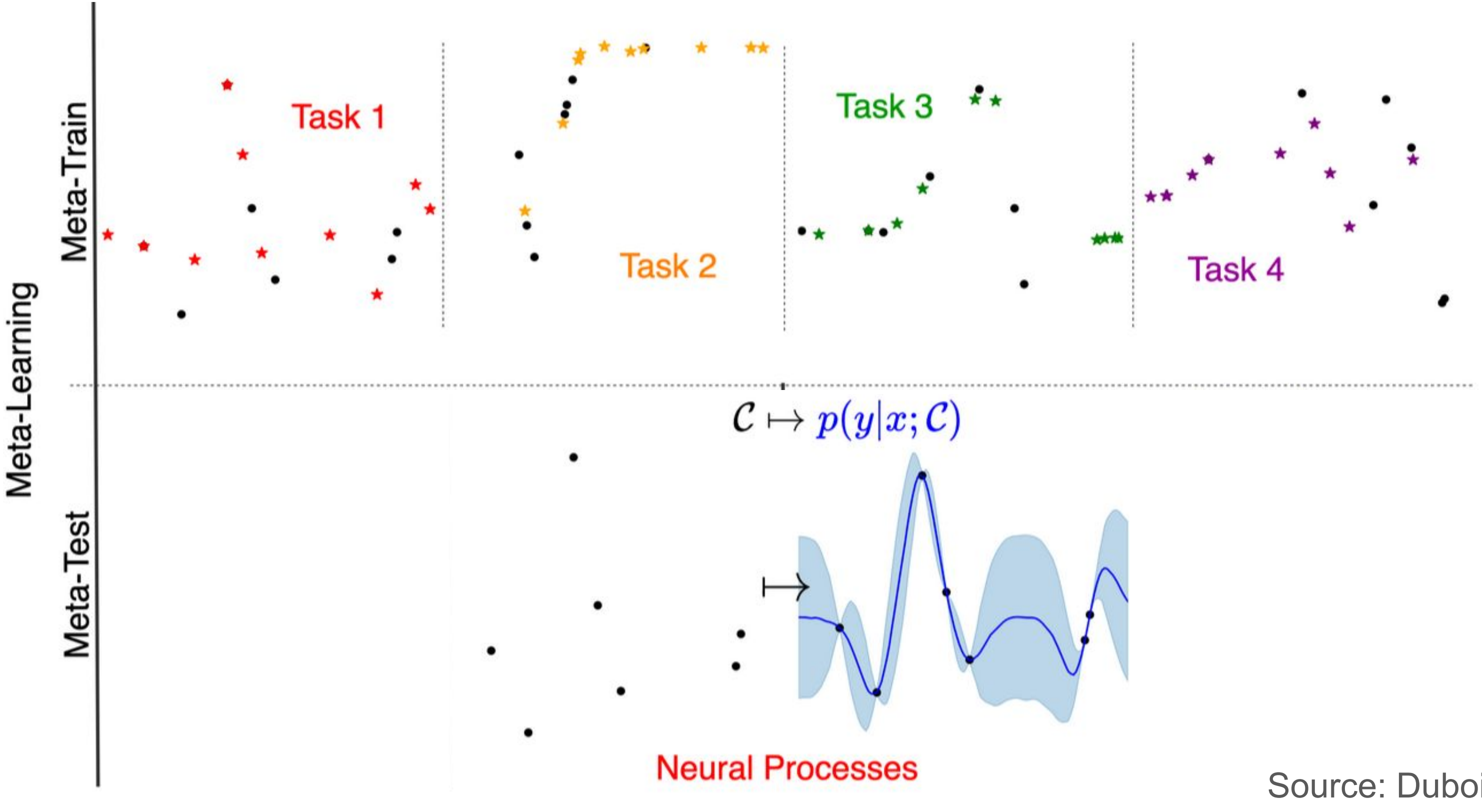
Conditional Neural Process



$$Q_\theta(f(T)|T,C) = \prod_{x\in T} Q_\theta(f(x)|x,C)$$
$$= \prod_{x\in T} Q(f(x)|\rho(E,x))$$

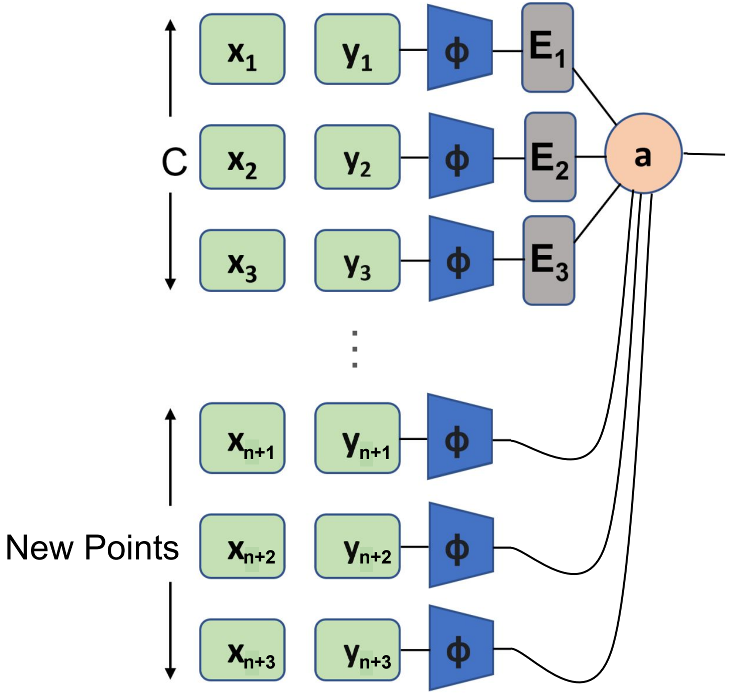
Source: Jha et al., 2023

Meta-Learning



Source: Dubois et al., 2020

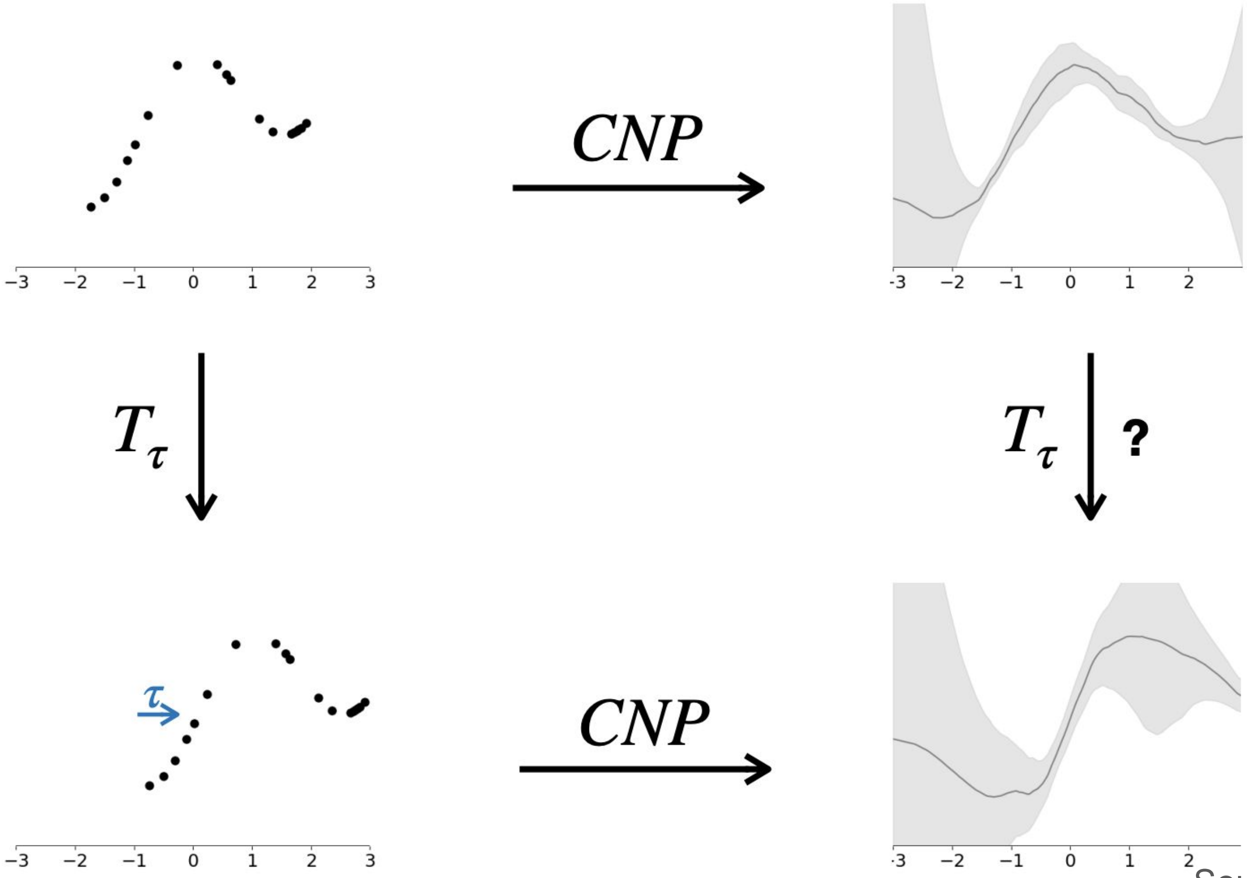
Continual Learning



Source: Jha et al., 2023

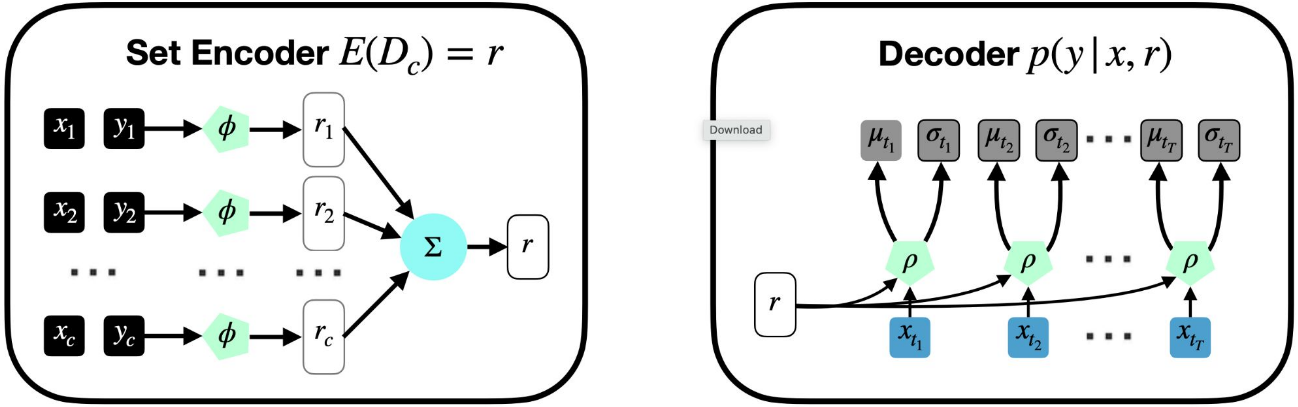
Deep Sets: Extending NPs

Translational Equivariance (Where CNP fails)



Source: Turner, 2021

Deep Sets: Take a look inside the NPF …

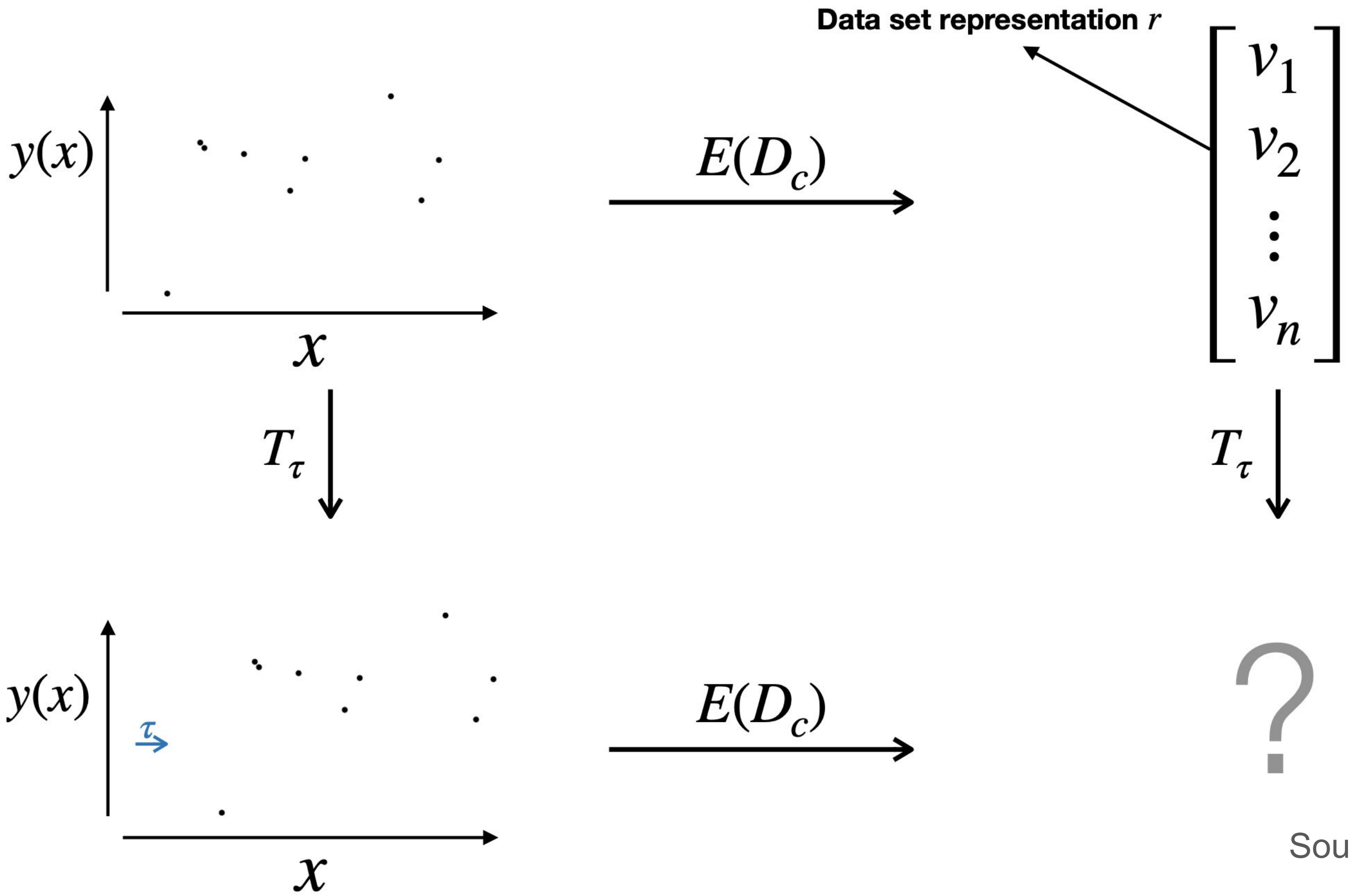


Zaheer et al. (2017): A function operating on a set $f(S)$ is a valid set function iff. It can be decomposed in the form:

$$f(S) = \rho\left(\sum_{s \in S} \phi(s)\right)$$

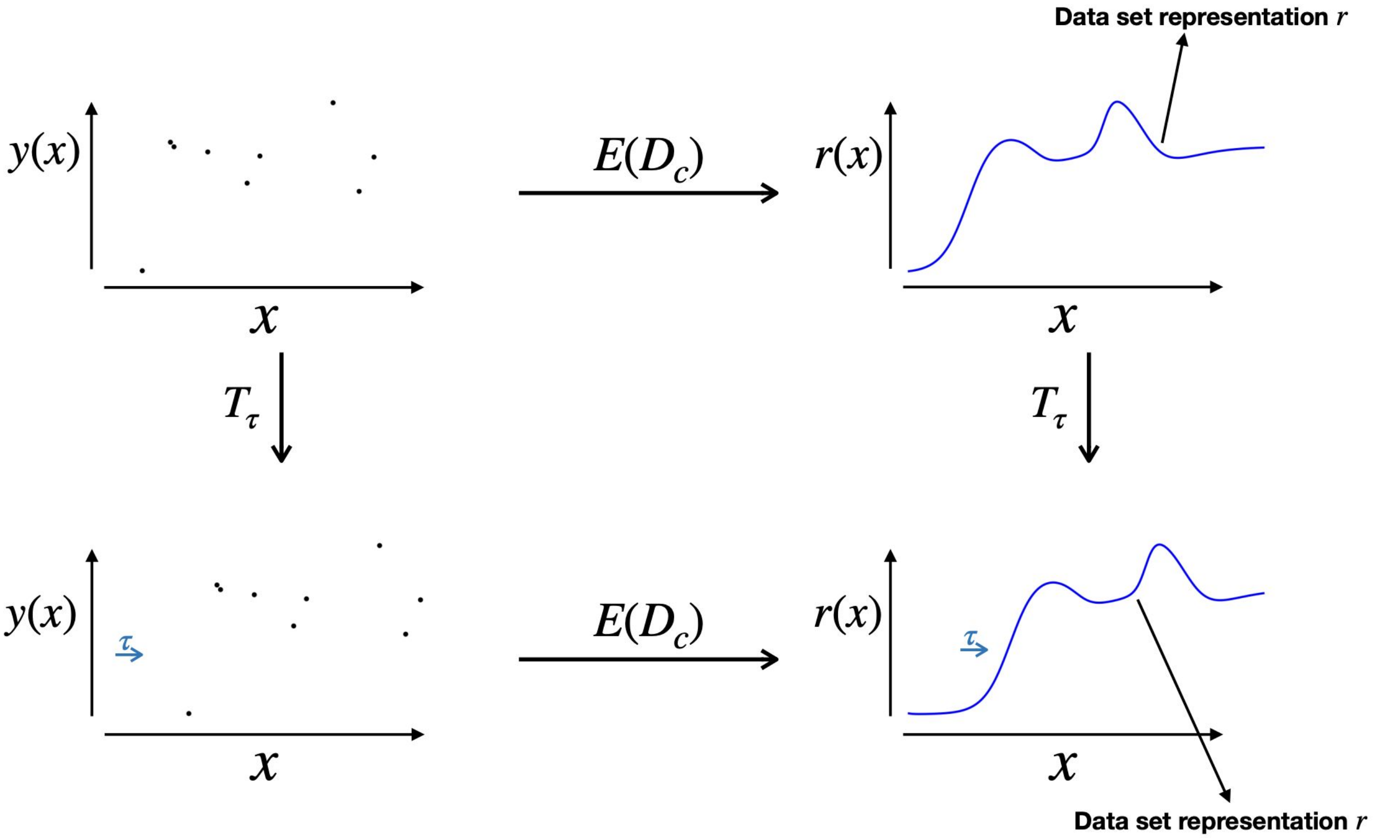
Image Source: Turner, 2023

Translational Equivariance: Vector Space to Functional Space



Source: Turner, 2021

Translational Equivariance: Vector Space to Functional Space



Source: Turner, 2021

Translational Equivariance: DeepSets to ConvDeepSets

DeepSets, Zaheer et al. (2017)

$$f(D_c) = \mathbf{MLP}\left(E(\{x_n, y_n\}_{n=1}^N)\right)$$

$$E(D_c) = \sum_{(x,y)\in D_c} \phi_{xy}([x;y])$$

$$\Longleftrightarrow$$

Here $f(D_c)$ is **permutation invariant**

ConvDeepSets Gordon et al. (2020)

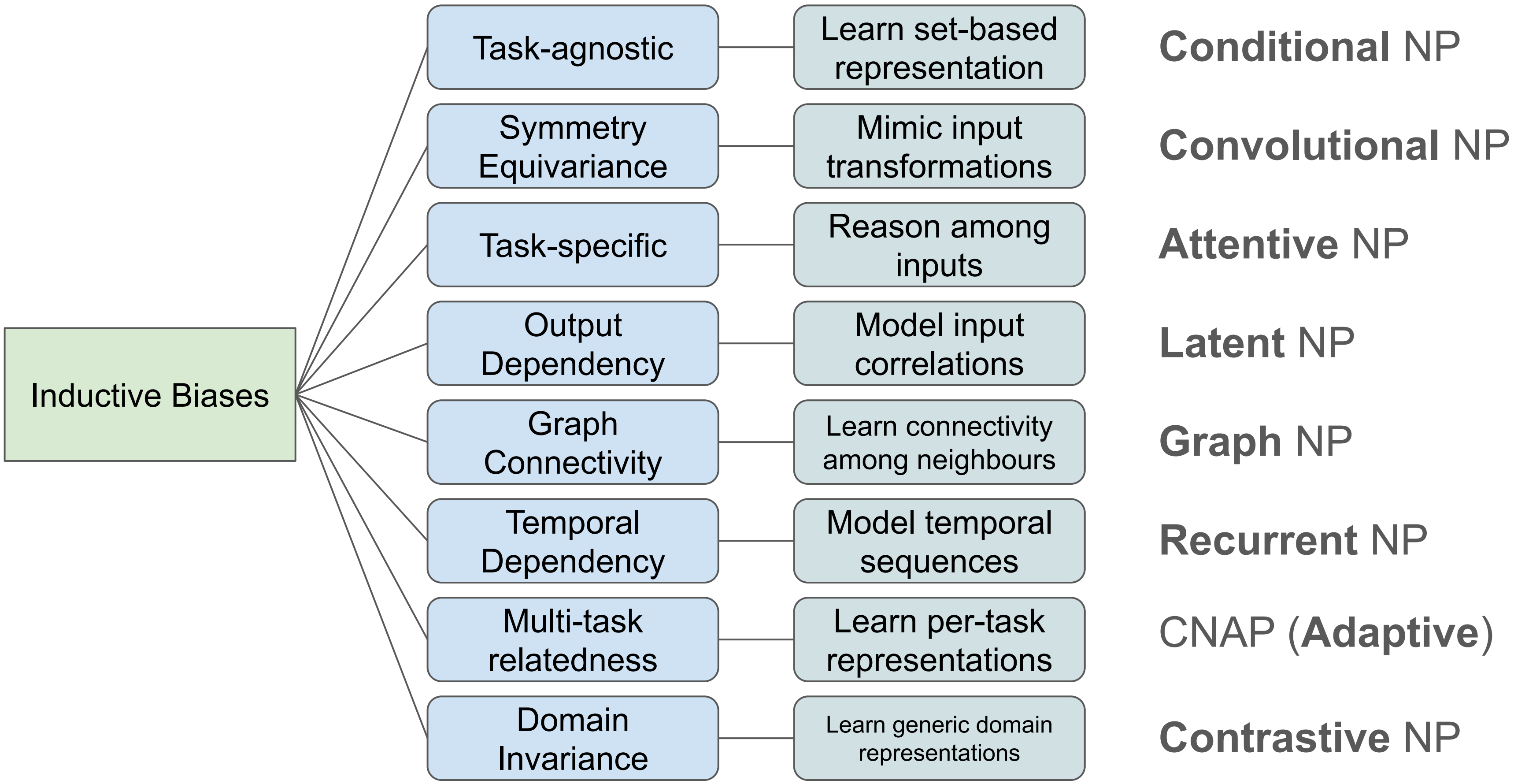
$$f(D_c) = \mathbf{CNN}\left(E(\{x_n, y_n\}_{n=1}^N)\right)$$

$$E(D_c)(x') = \sum_{(x,y)\in D_c} \phi_y(y)\psi(x - x')$$

$$\Longleftrightarrow$$

Here $f(D_c)$ is **permutation invariant** as well as **translation equivariant**

The Neural Process Family



Source: Jha et al., 2023

Research Applications of NPs

Strengths

- Cheap continual learning
- Few-shot learning
- Meta-learning
- Uncertainty estimation

Applications

- Space Science
- Recommenders
- Robotics
- Hyperparameter Optimisation
- Neuroscience
- Physics-Informed Modeling
- Weather Forecasting

Link to demo(s)

ANP:

https://github.com/edluyuan/neural-processes/blob/master/attentive_neural_process.ipynb

CNP:

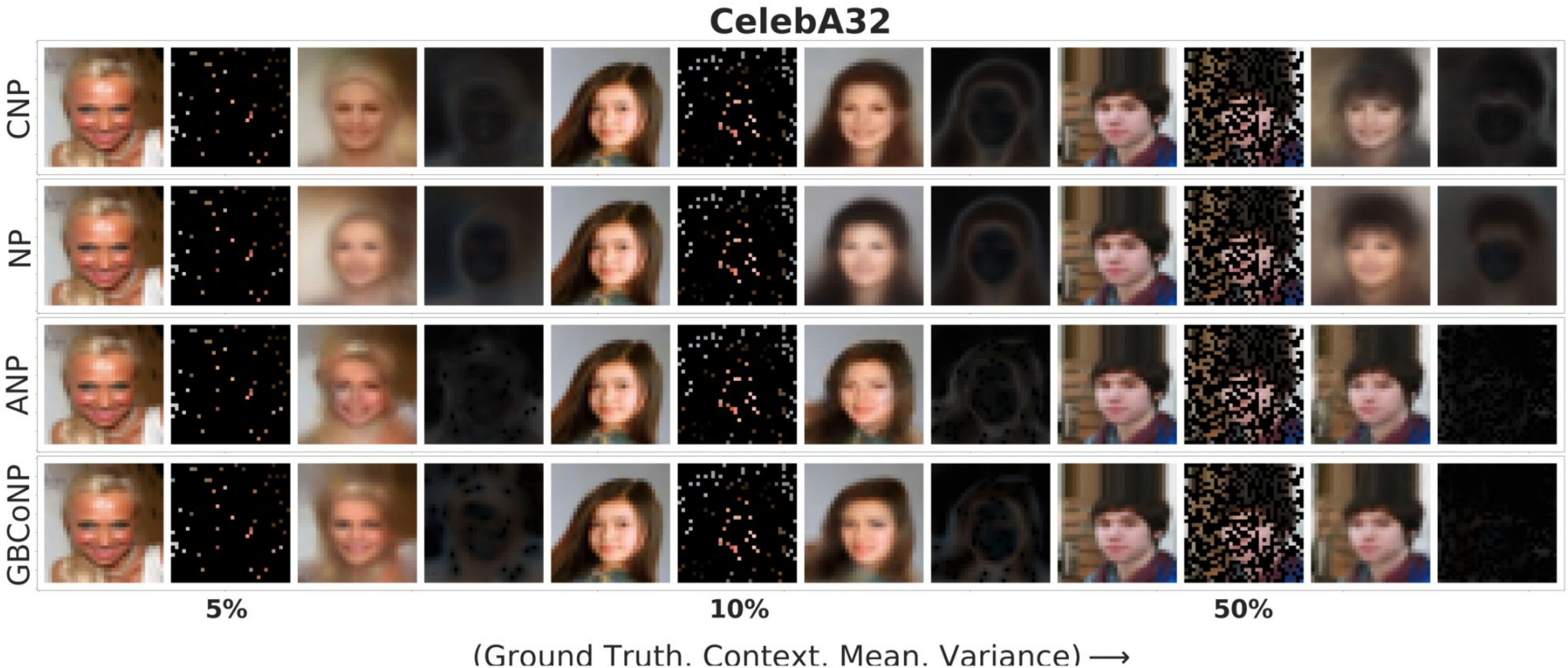
https://github.com/edluyuan/neural-processes/blob/master/conditional_neural_process.ipynb

Questions?

The Neural Process Family

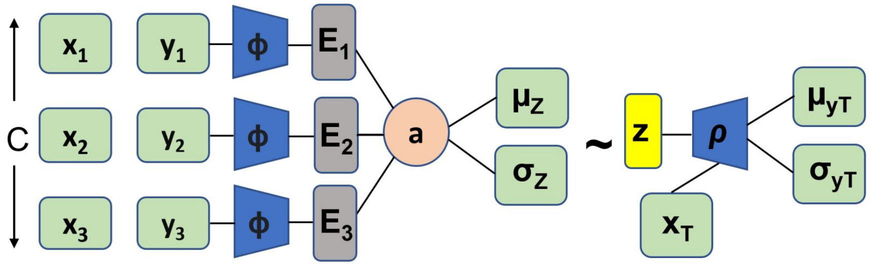
Appendices

Image Inpainting



Source: Jha et al., 2023

(latent) Neural Process

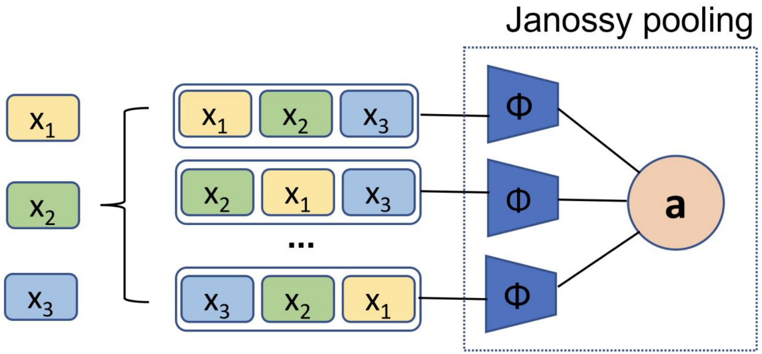


$$z \sim \mathcal{N}(\mu_z, \sigma_z^2)$$

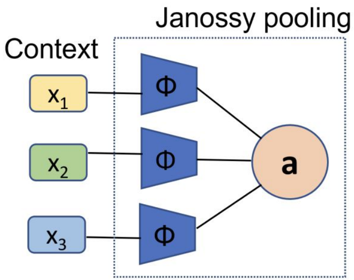
$$Q_\theta(f(T)|T, C) = \int p(z) \prod_{x \in T} Q(f(x)|\rho(z, x))dz$$

Source: Jha et al., 2023

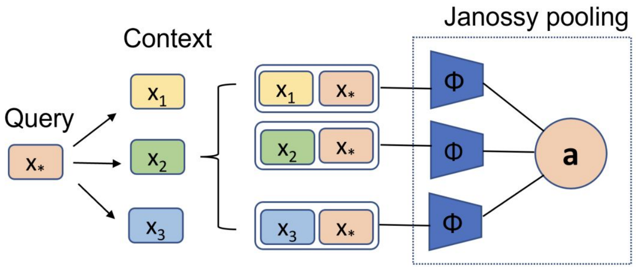
Janossy Pooling (Murphy et al., 2019)



(a) Exhaustive Janossy pooling (k=N)

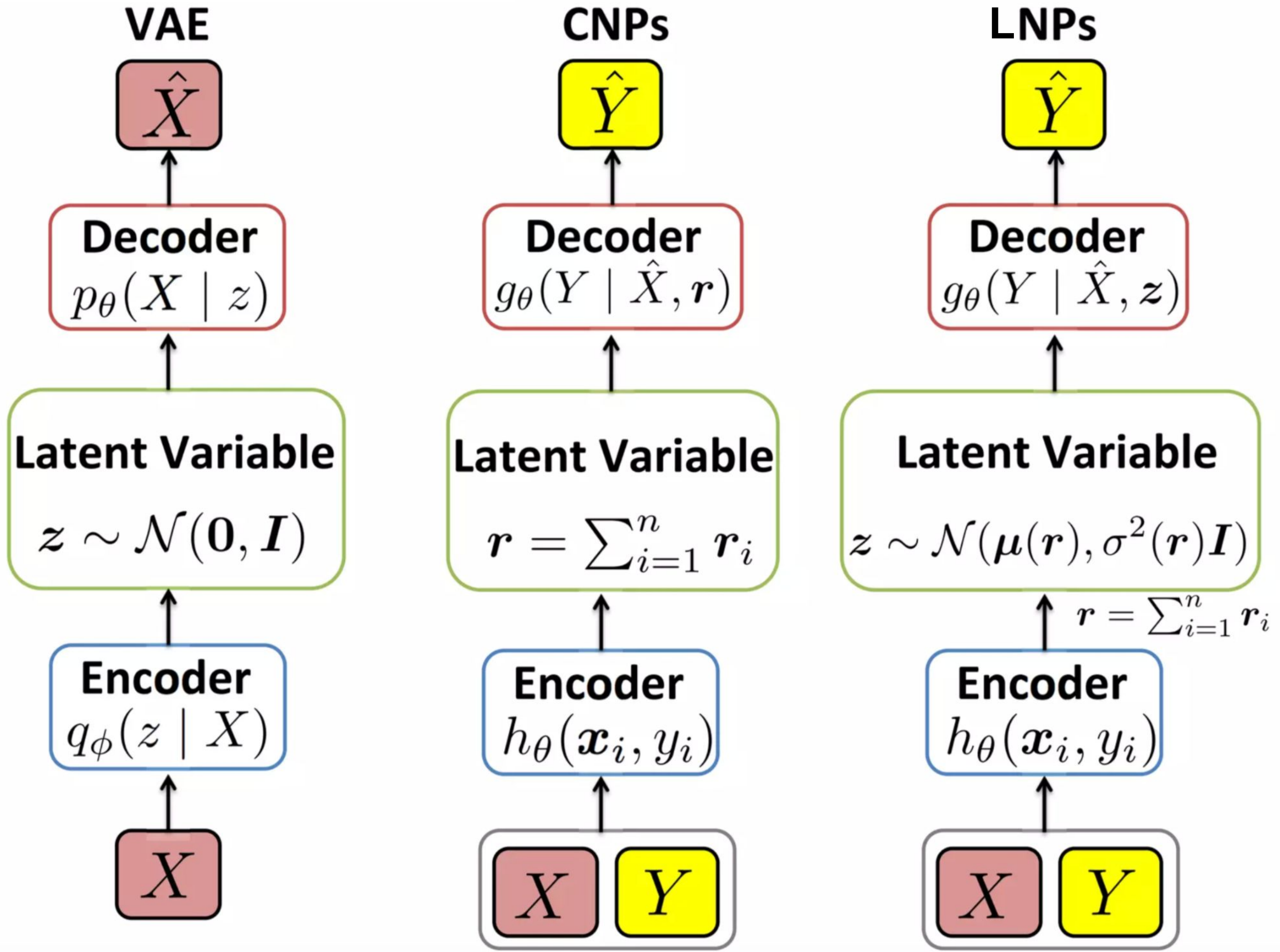


(b) NPs/CNPs (k=1)



(c) ANPs (k=2)

Source: Jha et al., 2023

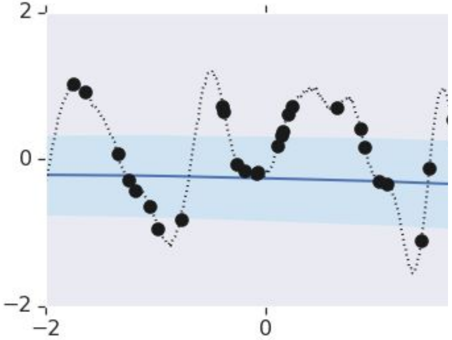

VAE
CNPs
LNPs
$\hat{X}$
$\hat{Y}$
$\hat{Y}$
Decoder
$p_\theta(X \mid z)$
Decoder
$g_\theta(Y \mid \hat{X}, \boldsymbol{r})$
Decoder
$g_\theta(Y \mid \hat{X}, \boldsymbol{z})$
Latent Variable
$\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
Latent Variable
$\boldsymbol{r} = \sum_{i=1}^{n} \boldsymbol{r}_i$
Latent Variable
$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{r}), \sigma^2(\boldsymbol{r})\boldsymbol{I})$
$\boldsymbol{r} = \sum_{i=1}^{n} \boldsymbol{r}_i$
Encoder
$q_\phi(z \mid X)$
Encoder
$h_\theta(\boldsymbol{x}_i, y_i)$
Encoder
$h_\theta(\boldsymbol{x}_i, y_i)$
X
X
Y
X
Y

Example from demo

Training ANP

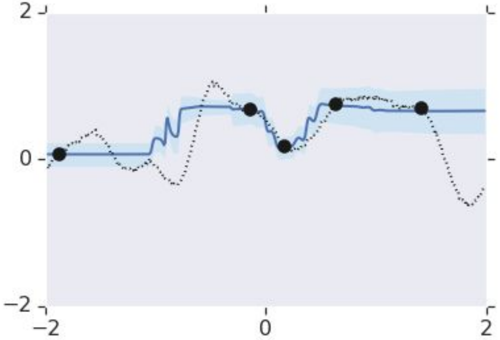
Iter 0

Loss: 1.43



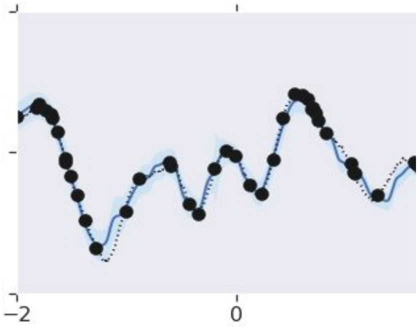
Iter 10000

Loss: 1.30



Iter 20000

Loss: 0.43



Iter 3000

Loss -0.56