Neural Processes

ICML 2018, ICML 2018 workshop

Introduction

Introduction

A neural network (NN) is a parameterized function that approximates a dataset with high accuracy, but **can't easily switch to an unknown function**.

Gaussian process (GP) has the flexibility to infer the shape of new functions at test time based on **prior knowledge**, but GP is **computationally intensive**.

Introduction

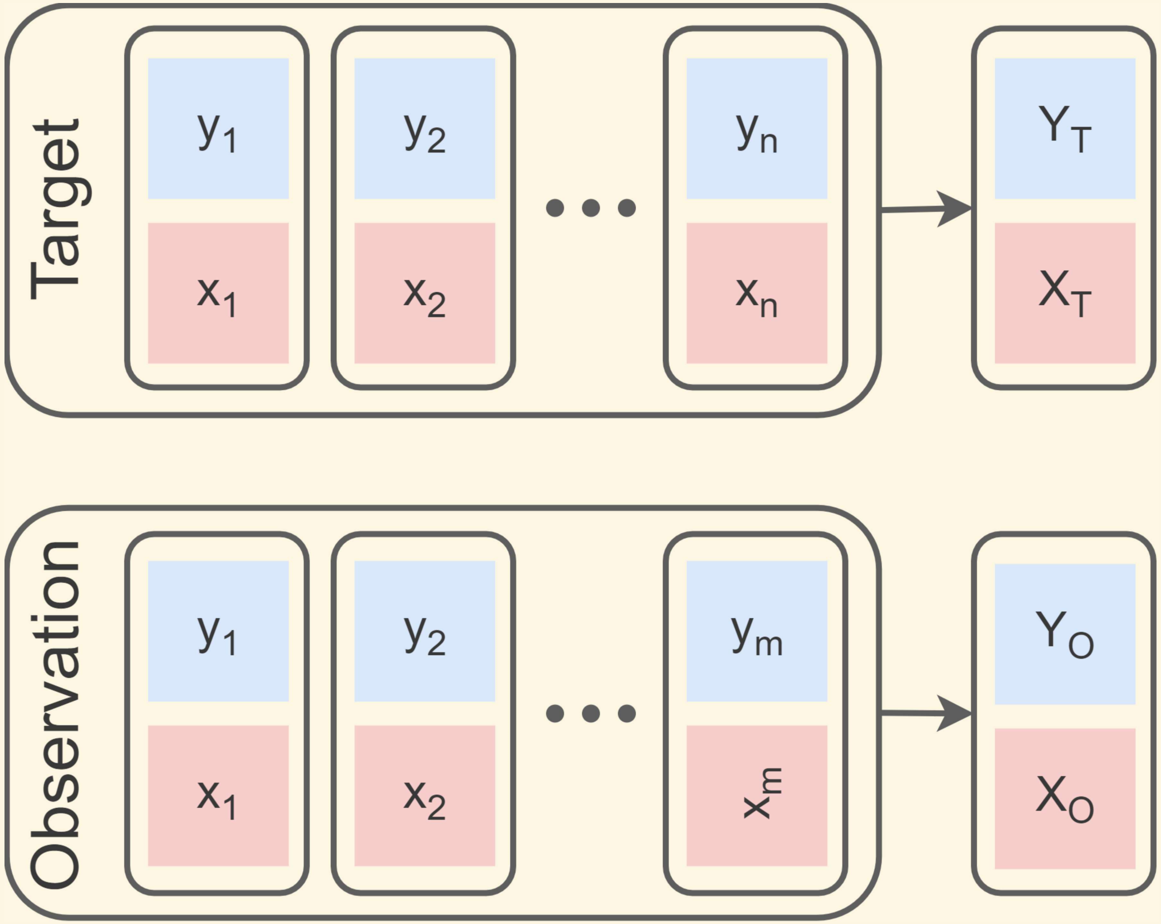
Therefore, this study proposes a new meta learning method called Neural Process, which combines the advantages of both.

This method is not only as **computationally efficient** as NNs during training and evaluation, but also as GPs can **effectively utilize prior knowledge** to quickly adapt to newly observed functions.

Conditional/Latent

Neural Processes

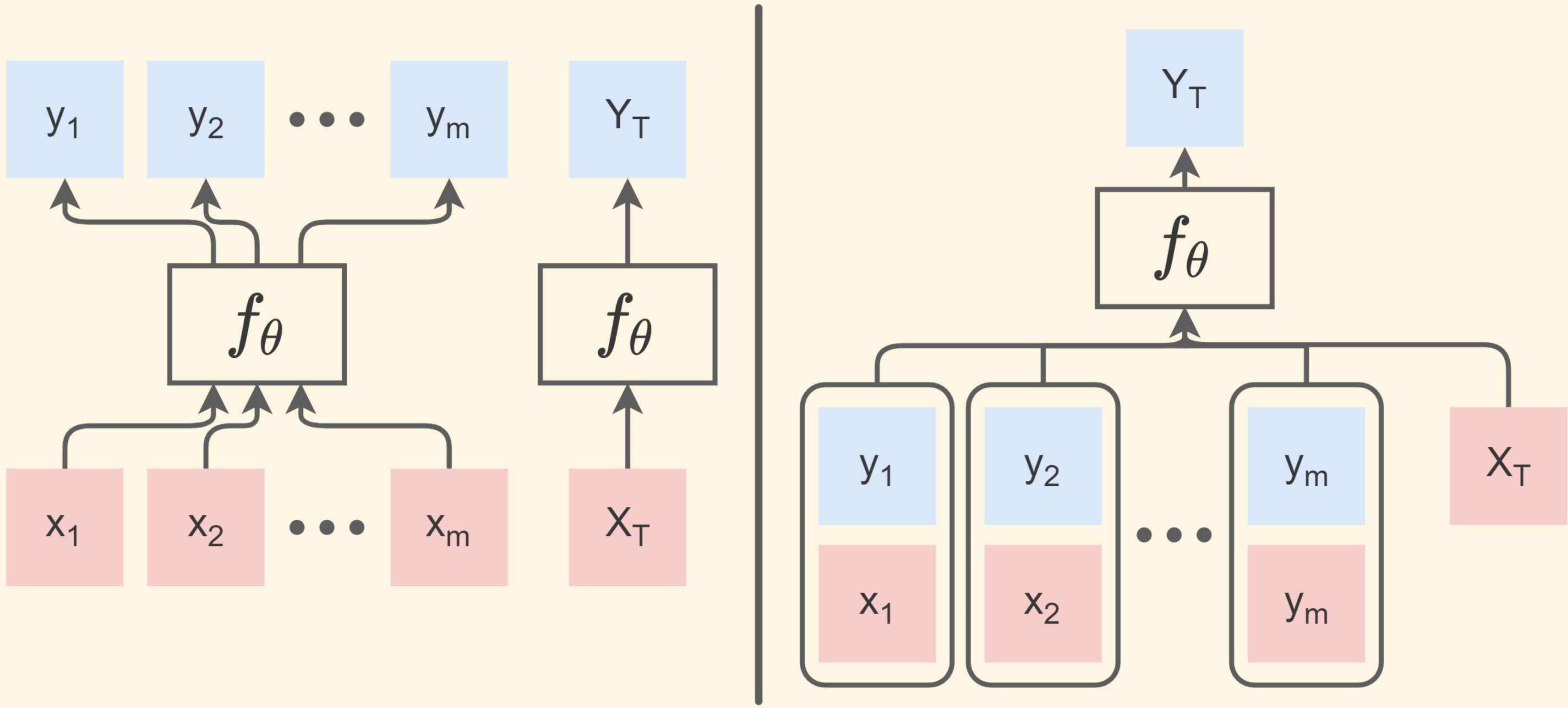
Observation and Target Dataset



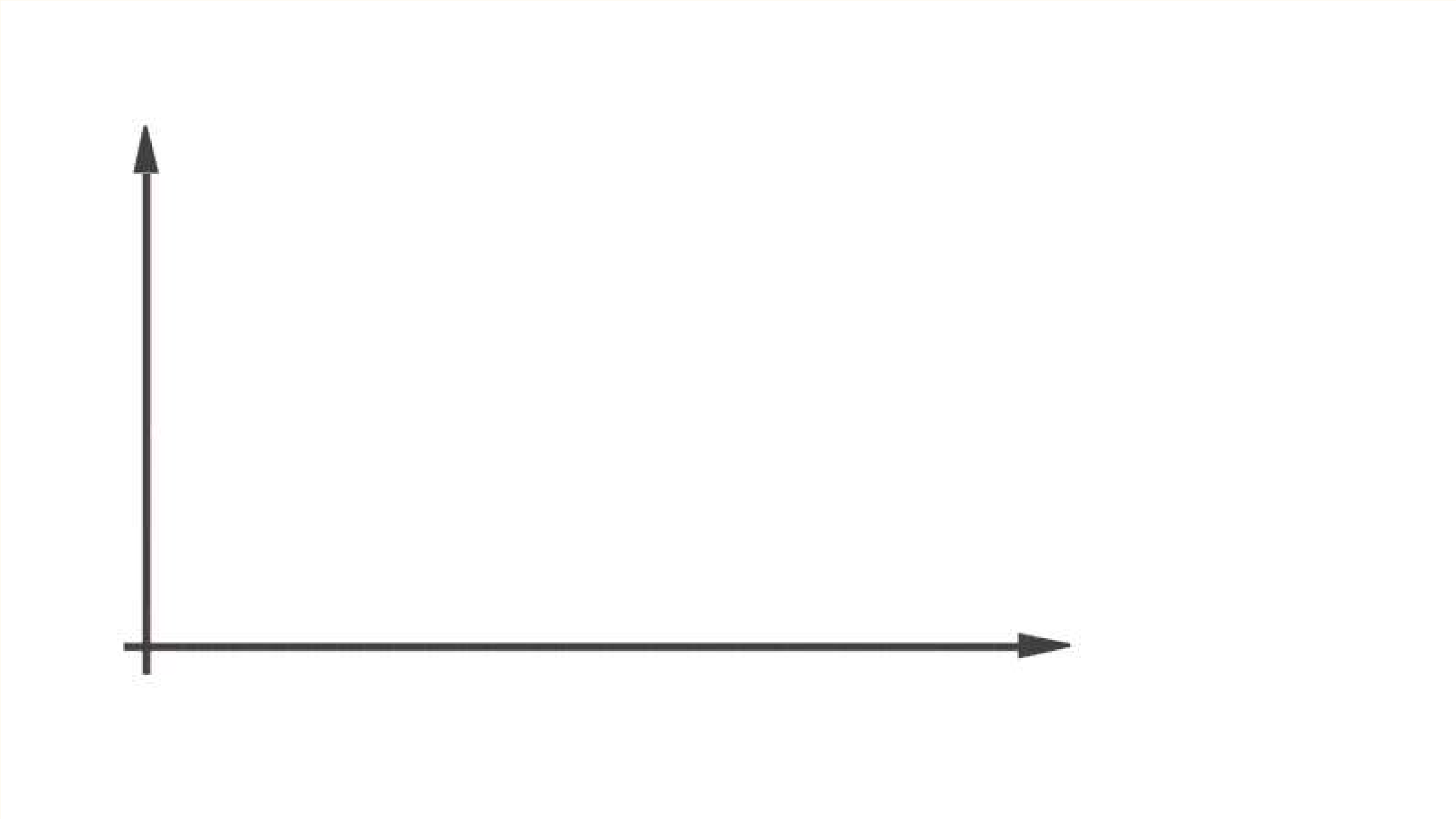
Supervised Learning

vs.

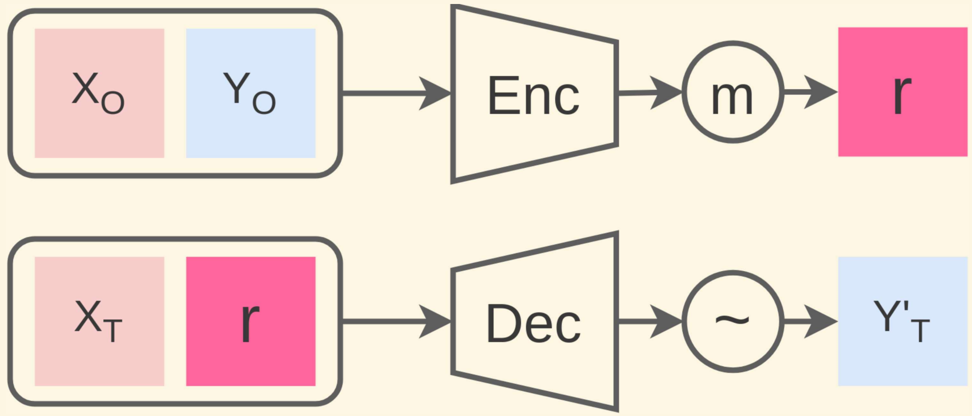
Neural Processes



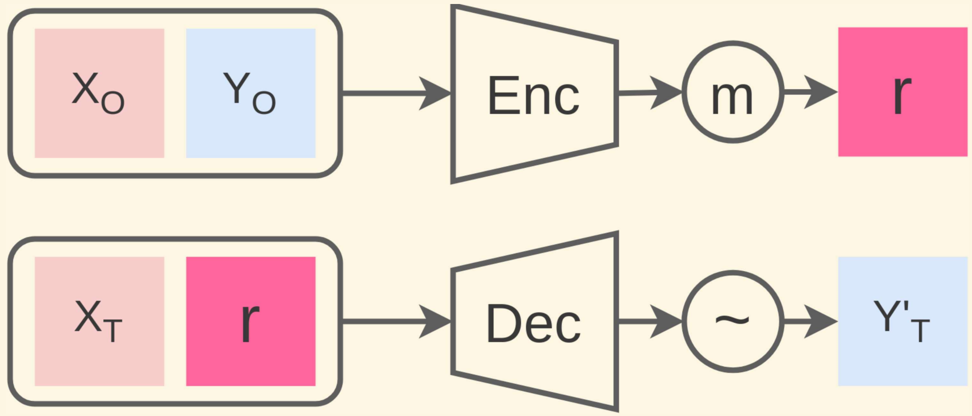
CNP Forward Pass



Conditional Neural Processes



Conditional Neural Processes

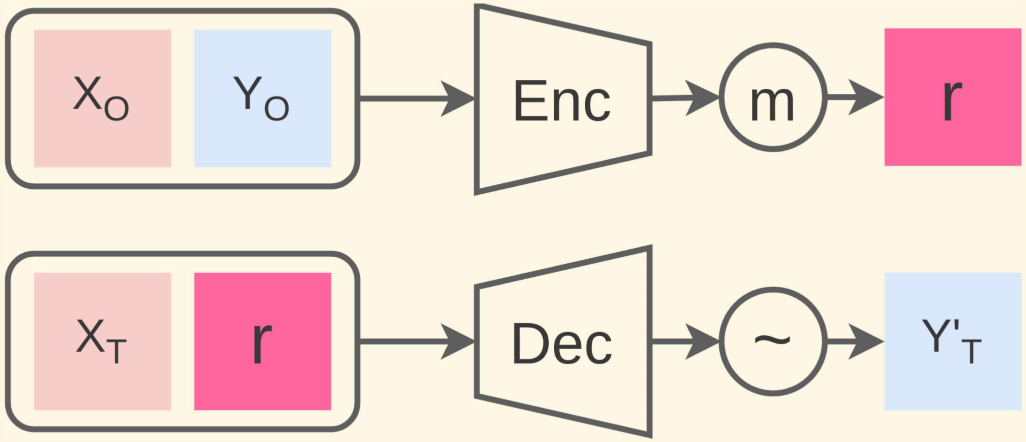


$$\mu_{Y_T}, \sigma_{Y_T} \leftarrow Dec(X_T, r)$$

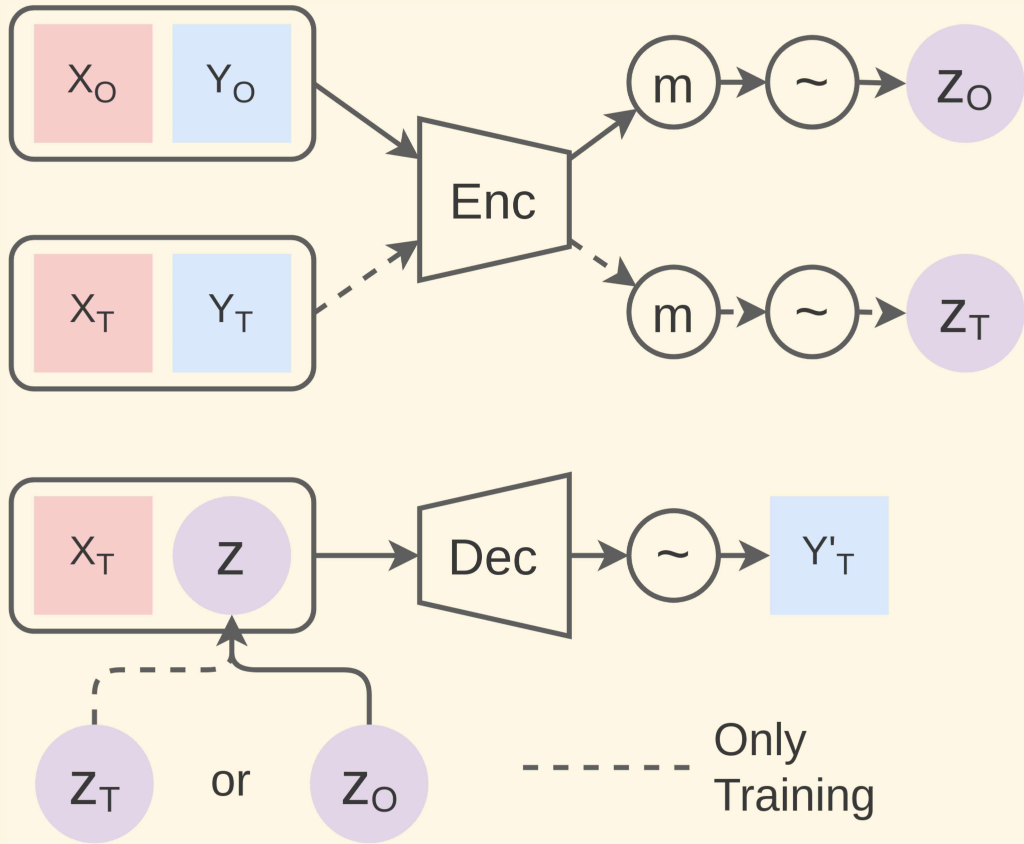
$$\tilde{Y}_T \sim \mathrm{N}(\mu_{Y_T}, \sigma_{Y_T})$$

Conditional Neural Processes

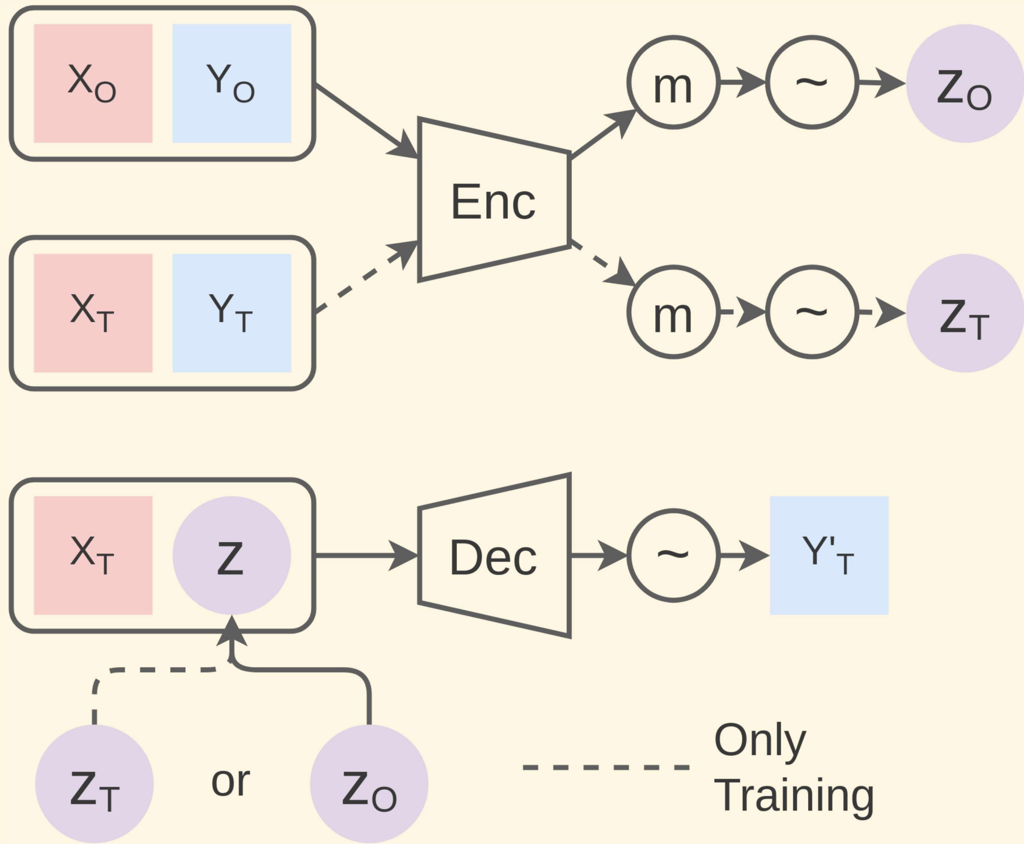
$$log\, p_\theta(Y_T | \{X_O, Y_O\}, X_T)$$



Latent Neural Processes



Latent Neural Processes

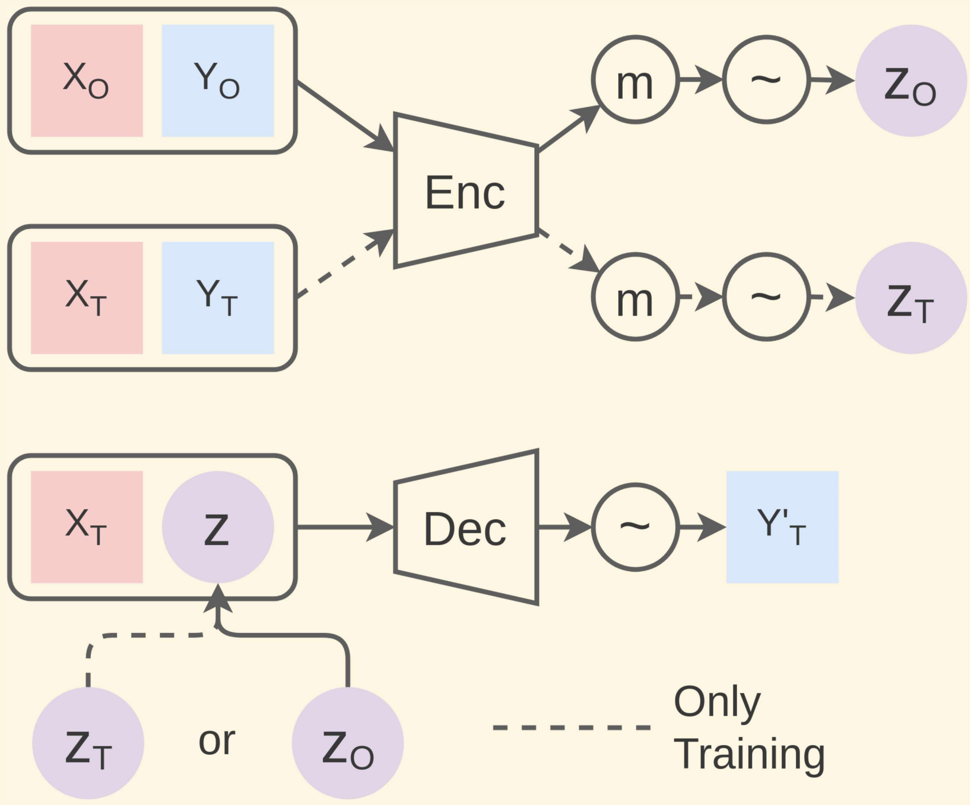


$$\mu_O, \sigma_O \leftarrow (MLP \circ mean \circ Enc)(X_O, Y_O)$$

$$z_T \sim \mathrm{N}(\mu_O, \sigma_O)$$

Latent Neural Processes

$$log \int p_\theta(Y_T|z, X_T)q_\omega(z|\{X_O, Y_O\})dz$$



Latent Neural Processes

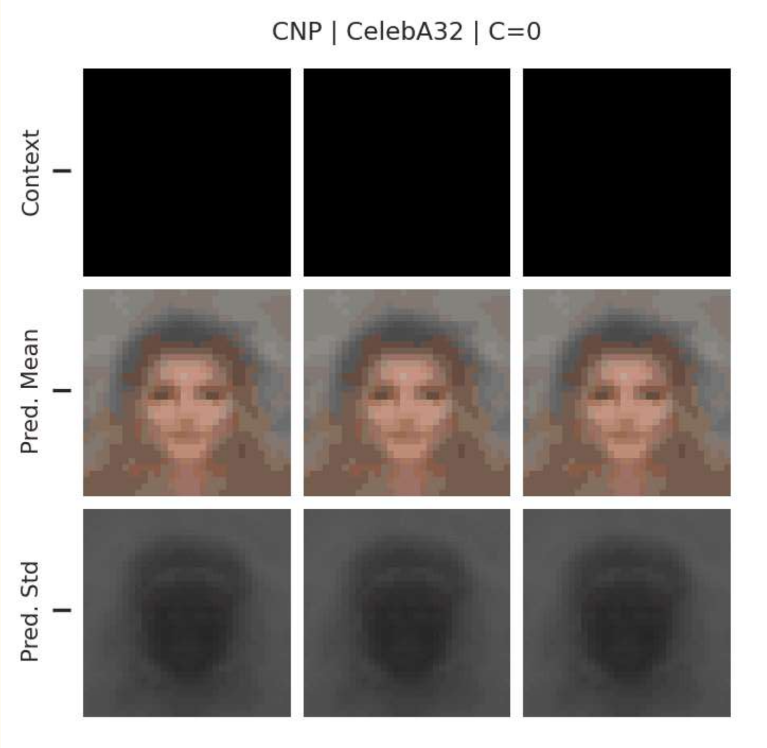
$$-log\, p_\theta(Y_T|z, X_T) + log\, \frac{q_\omega(z)}{q_\omega(z|\{X_O,Y_O\})}$$

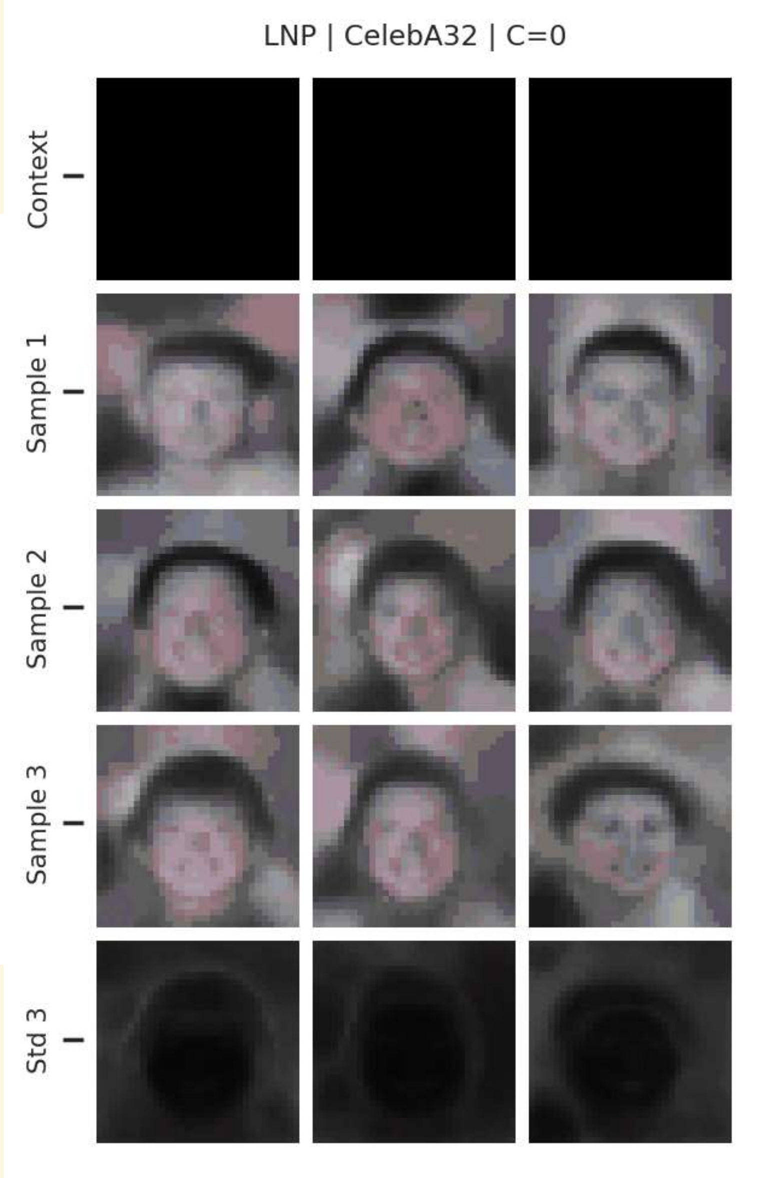
Latent Neural Processes

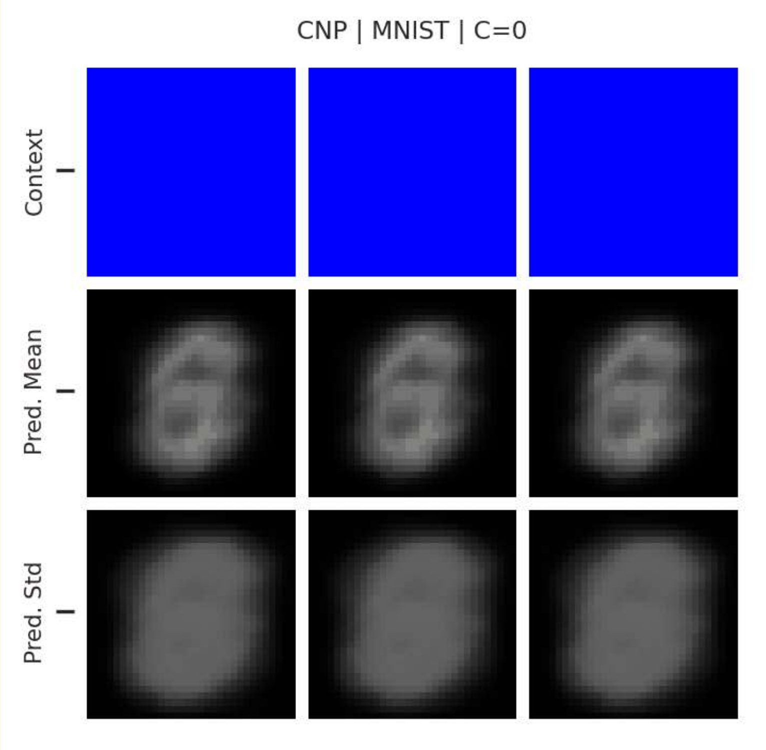
$$-log\, p_\theta(Y_T | z, X_T) + log \frac{q_\omega(z)}{q_\omega(z | \{X_O, Y_O\})}$$

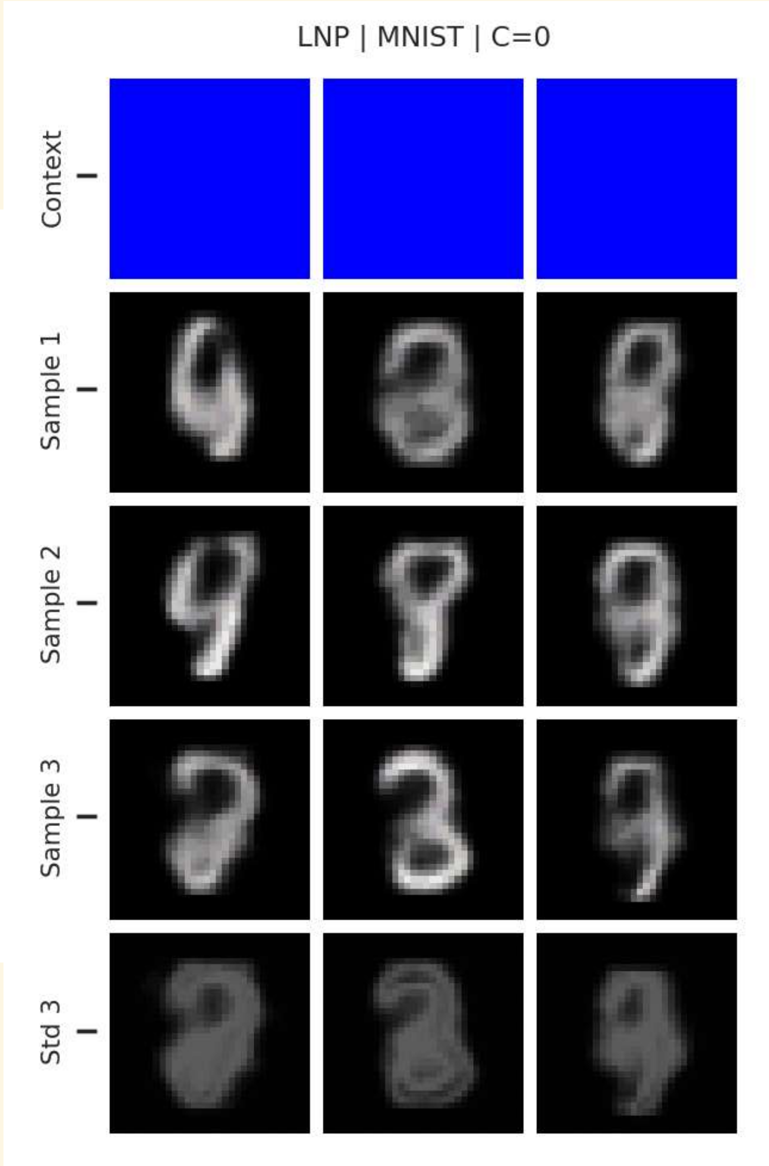
$$\begin{aligned} \mathrm{L} = & - log\, p_\theta(Y_T | z, X_T) \\ & + KLD(\mathrm{N}(\mu_T, \sigma_T), \mathrm{N}(\mu_O, \sigma_O)) \end{aligned}$$

Experiments


CNP | CelebA32 | C=0
Context
Pred. Mean
Pred. Std


LNP | CelebA32 | C=0
Context
Sample 1
Sample 2
Sample 3
Std 3


CNP | MNIST | C=0
Context
Pred. Mean
Pred. Std

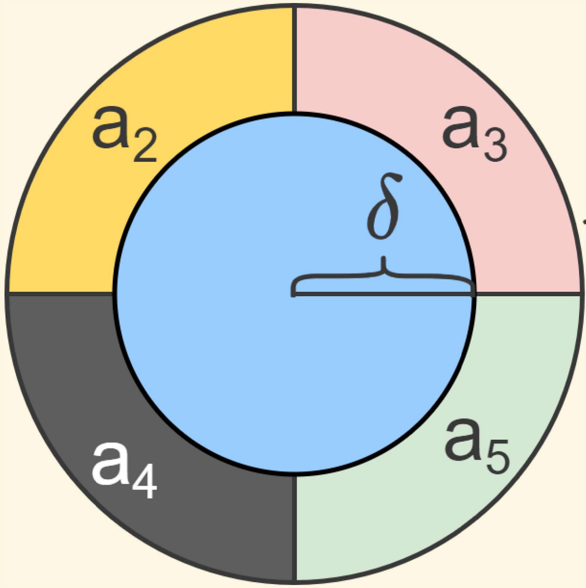

LNP | MNIST | C=0
Context
Sample 1
Sample 2
Sample 3
Std 3

#	Random Context			Ordered Context		
	10	100	1000	10	100	1000
kNN	0.215	0.052	0.007	0.370	0.273	0.007
GP	0.247	0.137	**0.001**	0.257	0.220	**0.002**
CNP	**0.039**	**0.016**	0.009	**0.057**	**0.047**	0.021

Pixel-wise mean squared error of CNP on the CelebA.

	5-way Acc		20-way Acc		Runtime
	1-shot	5-shot	1-shot	5-shot	
MANN	82.8%	94.9%	-	-	$\mathcal{O}(nm)$
MN	**98.1%**	**98.9%**	**93.8%**	**98.5%**	$\mathcal{O}(nm)$
CNP	95.3%	98.5%	89.9%	96.8%	$\mathcal{O}(n+m)$

Classification results of CNP on Omniglot.



$$\begin{cases} r \sim \mathcal{N}(\mu_1, sigma^2), \text{ a1} \\ r \sim \mathcal{N}(\mu_2, sigma^2), \text{ Sub-optimal Action} \\ r \sim \mathcal{N}(\mu_3, sigma^2), \text{ Optimal Action} \end{cases}$$

$$\mu_2 < \mu_1 << \mu_3$$

The wheel bandit problem. Optimal action for yellow, red, black and green regions, are actions 2, 3, 4 and 5, respectively.

δ	**0.5**	**0.7**	**0.9**	**0.95**	**0.99**
Cumulative regret					
Uniform	100.00 $\pm$ 0.08	100.00 $\pm$ 0.09	100.00 $\pm$ 0.25	100.00 $\pm$ 0.37	100.00 $\pm$ 0.78
LinGreedy ($\epsilon = 0.0$)	65.89 $\pm$ 4.90	71.71 $\pm$ 4.31	108.86 $\pm$ 3.10	102.80 $\pm$ 3.06	104.80 $\pm$ 0.91
Dropout	7.89 $\pm$ 1.51	9.03 $\pm$ 2.58	36.58 $\pm$ 3.62	63.12 $\pm$ 4.26	98.68 $\pm$ 1.59
LinGreedy ($\epsilon = 0.05$)	7.86 $\pm$ 0.27	9.58 $\pm$ 0.35	19.42 $\pm$ 0.78	33.06 $\pm$ 2.06	74.17 $\pm$ 1.63
Bayes by Backprob (Blundell et al., 2015)	1.37 $\pm$ 0.07	3.32 $\pm$ 0.80	34.42 $\pm$ 5.50	59.04 $\pm$ 5.59	97.38 $\pm$ 2.66
NeuralLinear	**0.95** $\pm$ 0.02	**1.60** $\pm$ 0.03	4.65 $\pm$ 0.18	9.56 $\pm$ 0.36	49.63 $\pm$ 2.41
MAML (Finn et al., 2017)	2.95 $\pm$ 0.12	3.11 $\pm$ 0.16	4.84 $\pm$ 0.22	7.01 $\pm$ 0.33	22.93 $\pm$ 1.57
Neural Processes	1.60 $\pm$ 0.06	1.75 $\pm$ 0.05	**3.31** $\pm$ 0.10	**5.71** $\pm$ 0.24	**22.13** $\pm$ 1.23
Simple regret					
Uniform	100.00 $\pm$ 0.45	100.00 $\pm$ 0.78	100.00 $\pm$ 1.18	100.00 $\pm$ 2.21	100.00 $\pm$ 4.21
LinGreedy ($\epsilon = 0.0$)	66.59 $\pm$ 5.02	73.06 $\pm$ 4.55	108.56 $\pm$ 3.65	105.01 $\pm$ 3.59	105.19 $\pm$ 4.14
Dropout	6.57 $\pm$ 1.48	6.37 $\pm$ 2.53	35.02 $\pm$ 3.94	59.45 $\pm$ 4.74	102.12 $\pm$ 4.76
LinGreedy ($\epsilon = 0.05$)	5.53 $\pm$ 0.19	6.07 $\pm$ 0.24	8.49 $\pm$ 0.47	12.65 $\pm$ 1.12	57.62 $\pm$ 3.57
Bayes by Backprob (Blundell et al., 2015)	0.60 $\pm$ 0.09	1.45 $\pm$ 0.61	27.03 $\pm$ 6.19	56.64 $\pm$ 6.36	102.96 $\pm$ 5.93
NeuralLinear	**0.33** $\pm$ 0.04	**0.79** $\pm$ 0.07	**2.17** $\pm$ 0.14	**4.08** $\pm$ 0.20	35.89 $\pm$ 2.98
MAML (Finn et al., 2017)	2.49 $\pm$ 0.12	3.00 $\pm$ 0.35	4.75 $\pm$ 0.48	7.10 $\pm$ 0.77	22.89 $\pm$ 1.41
Neural Processes	1.04 $\pm$ 0.06	1.26 $\pm$ 0.21	2.90 $\pm$ 0.35	5.45 $\pm$ 0.47	**21.45** $\pm$ 1.3

Results of LNP on the wheel bandit problem.

Conclusion

Neural Processes combines the computational efficiency of neural networks with the flexibility of stochastic processes.

- Efficiently extract prior knowledge from $\{X_O, Y_O\}$ (neural networks)
- Change strategies based on prior knowledge (stochastic processes)

Thanks for your attention.

Q&A