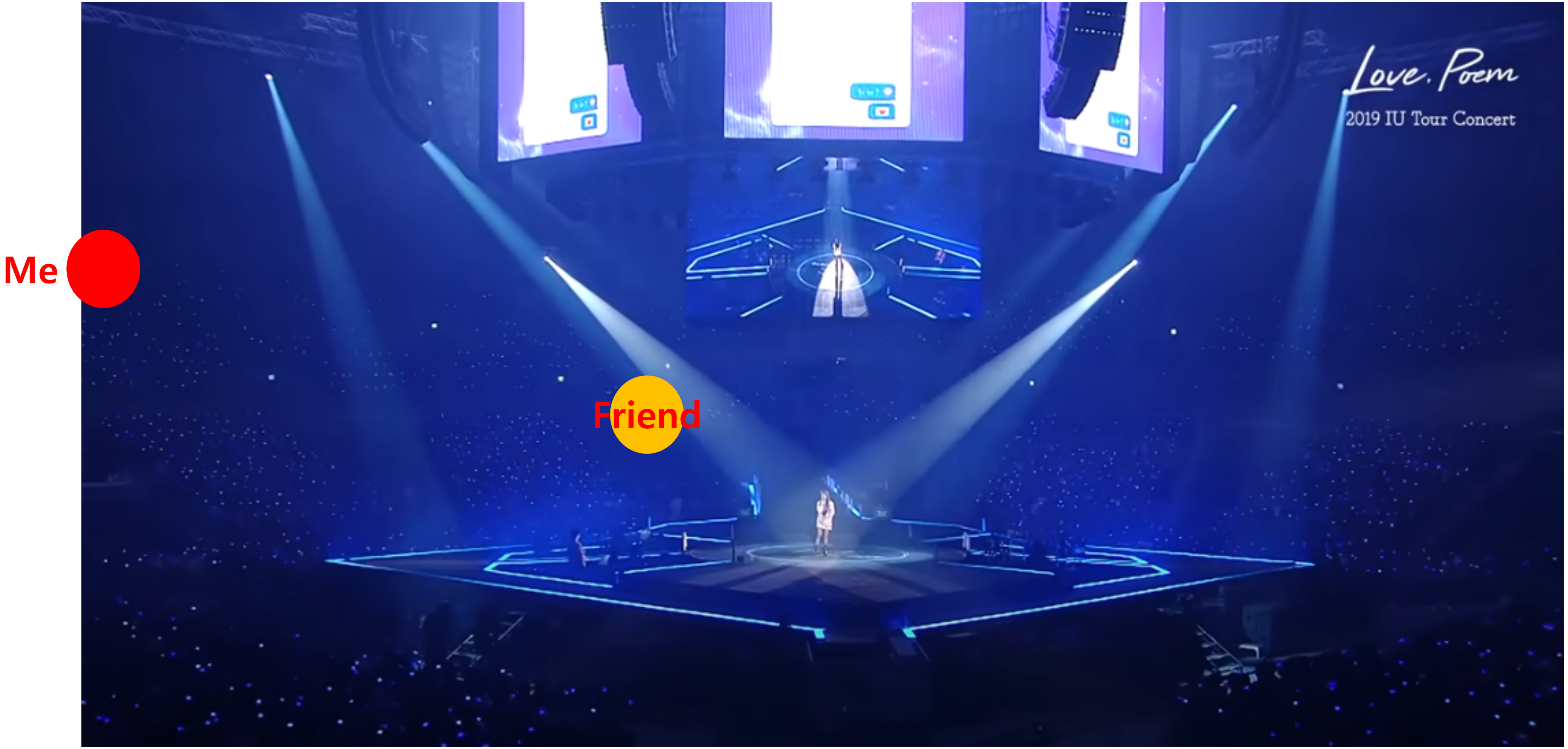
Neural process : Another way to merge NN and GP

Jaehyoung Hong

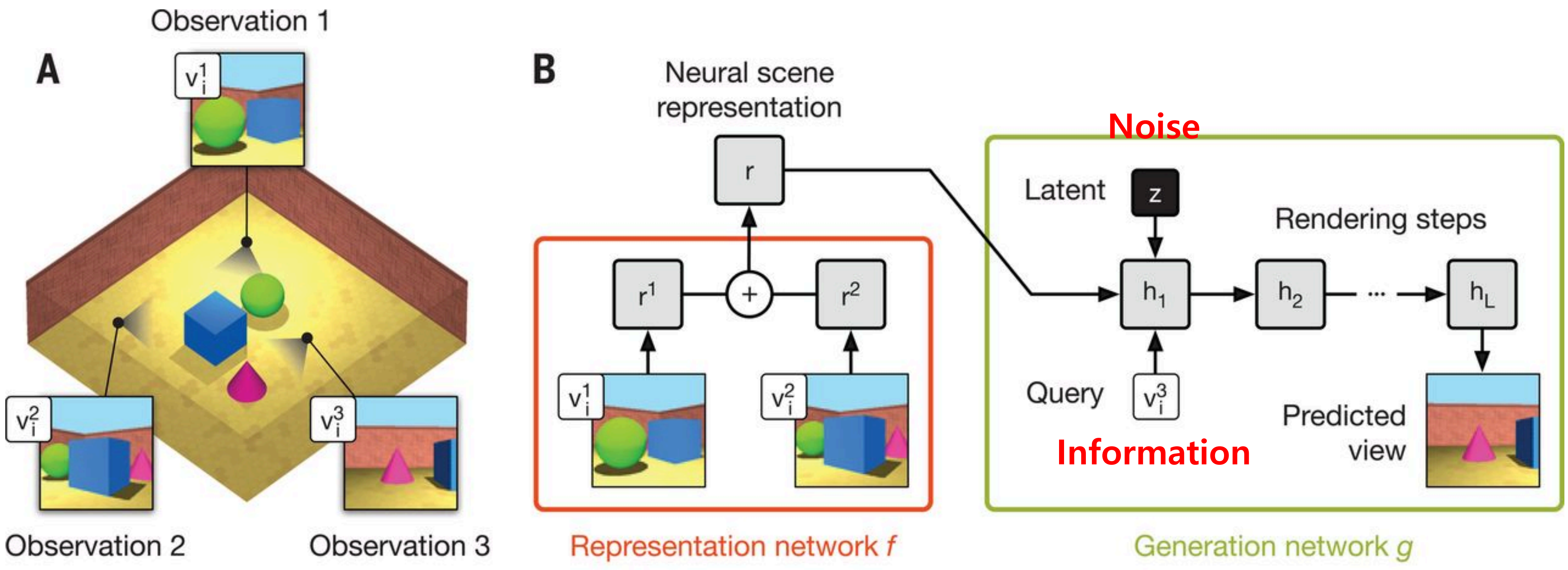
Can we predict scene of people in different positions?



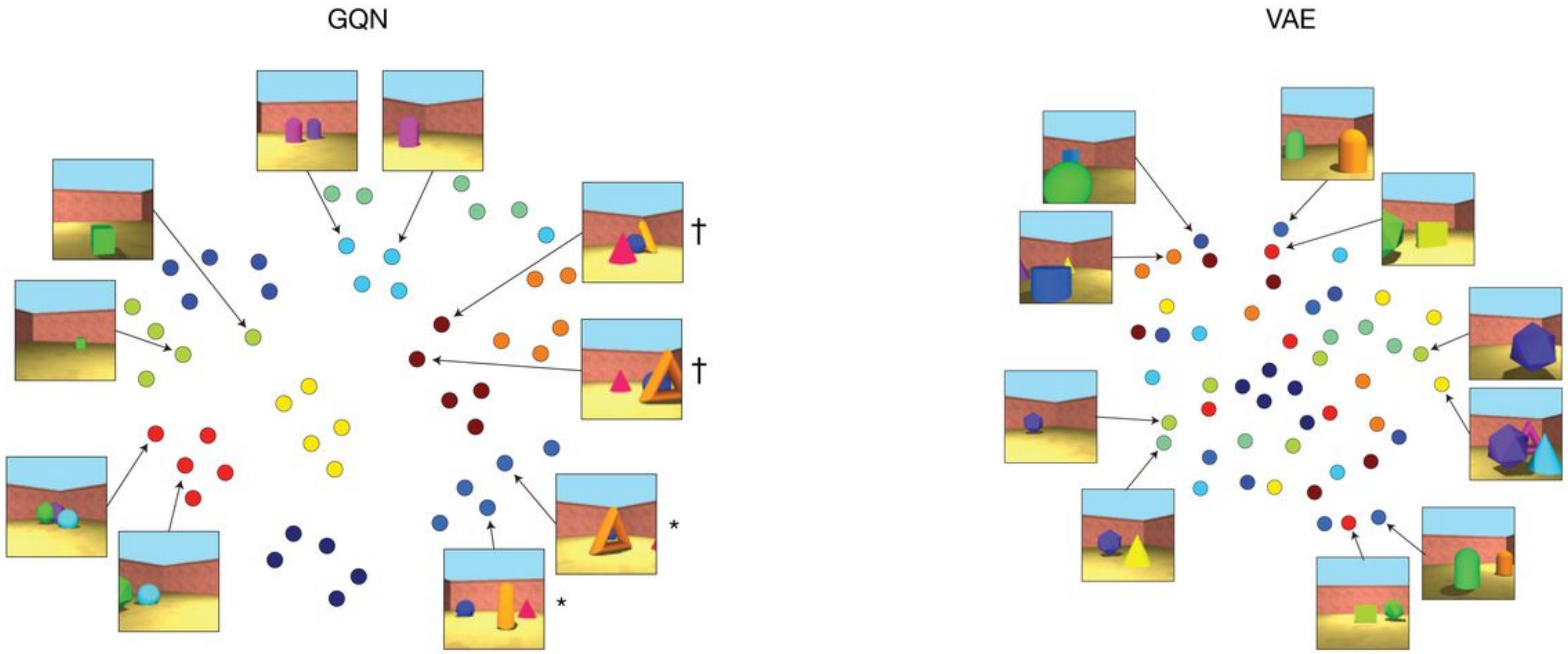
For such purpose, we need more information

For such purpose, we need more information

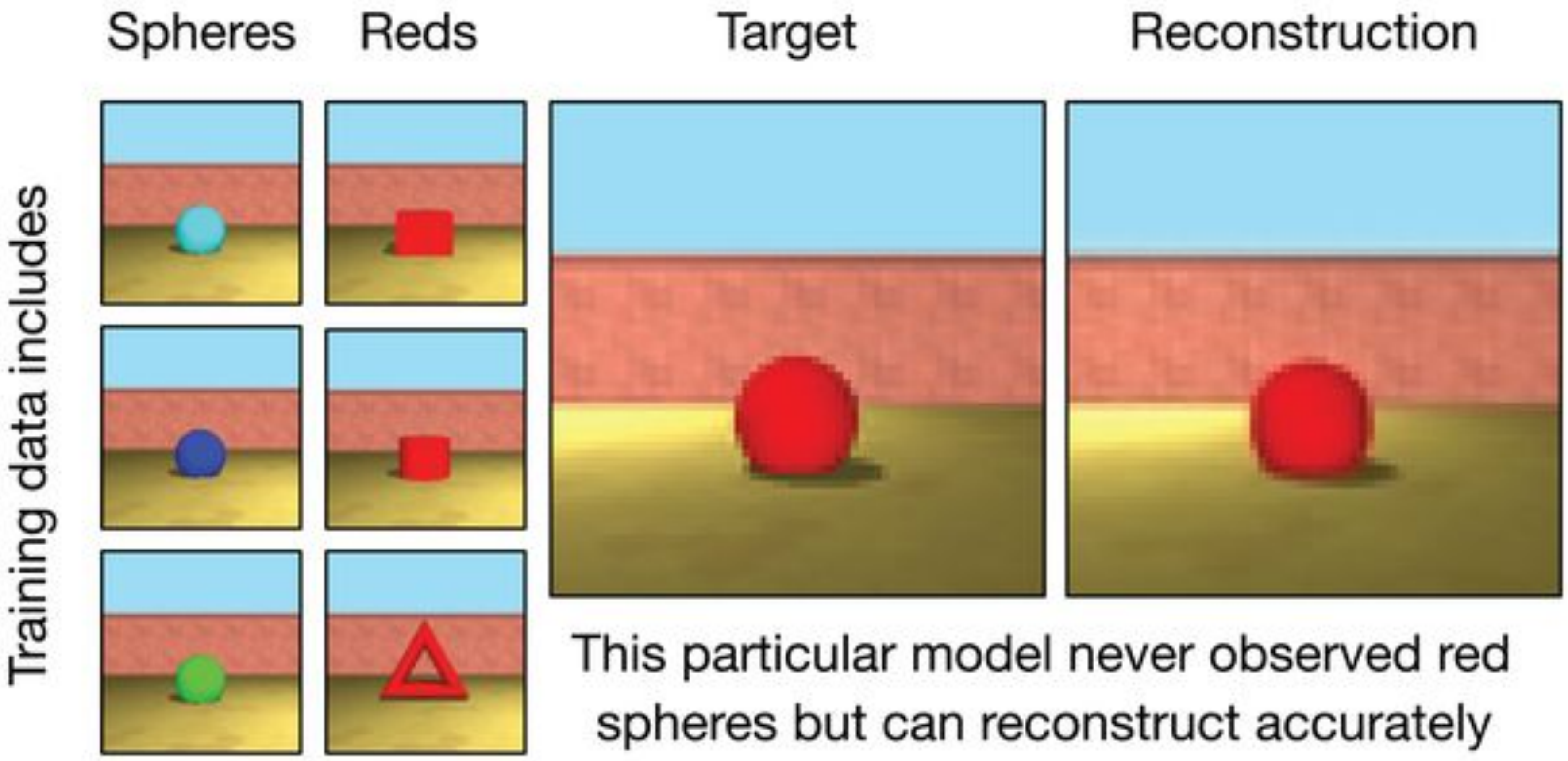
Generative Query Network (GQN) can predict different scene



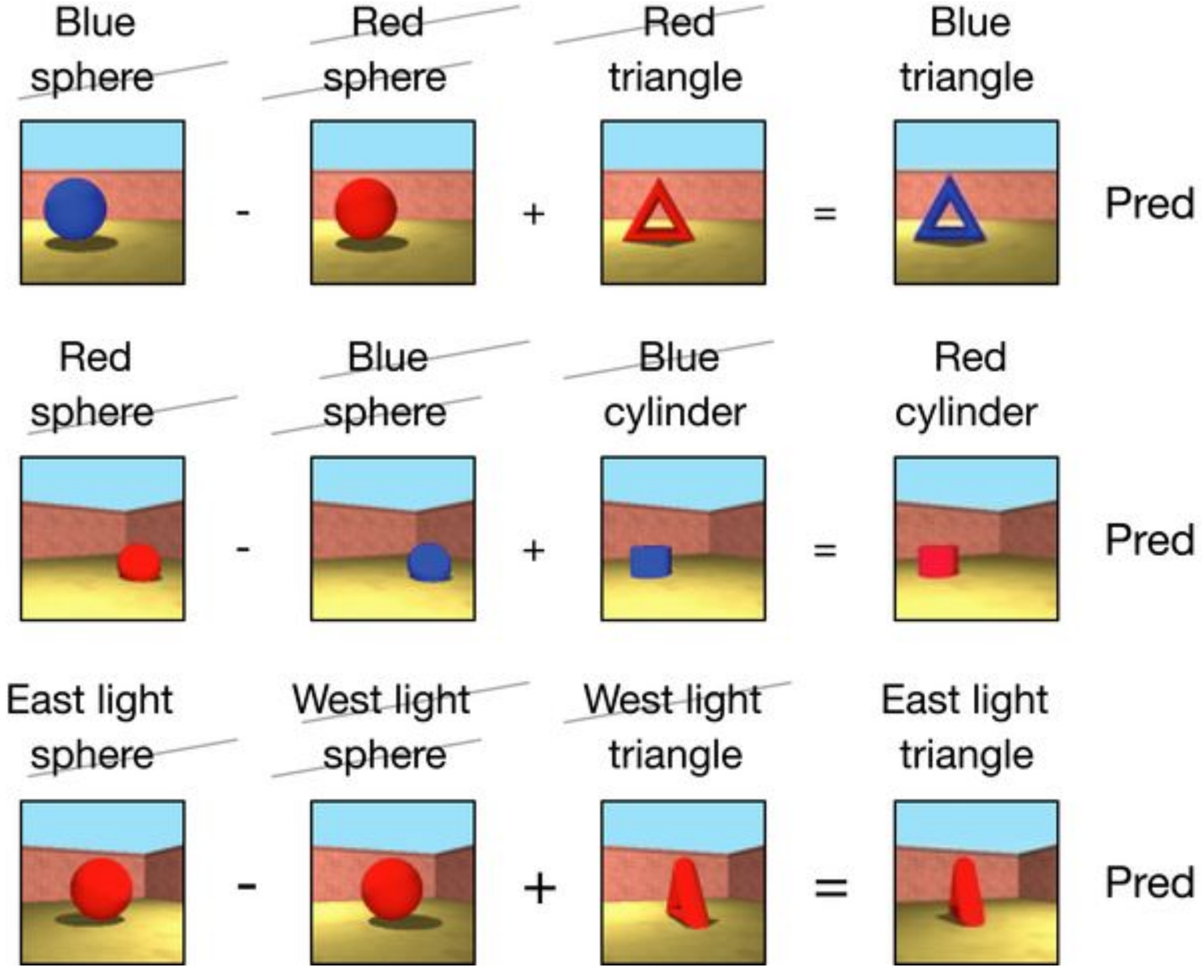
GQN learns some information (latent variables) for scene latter than pixels



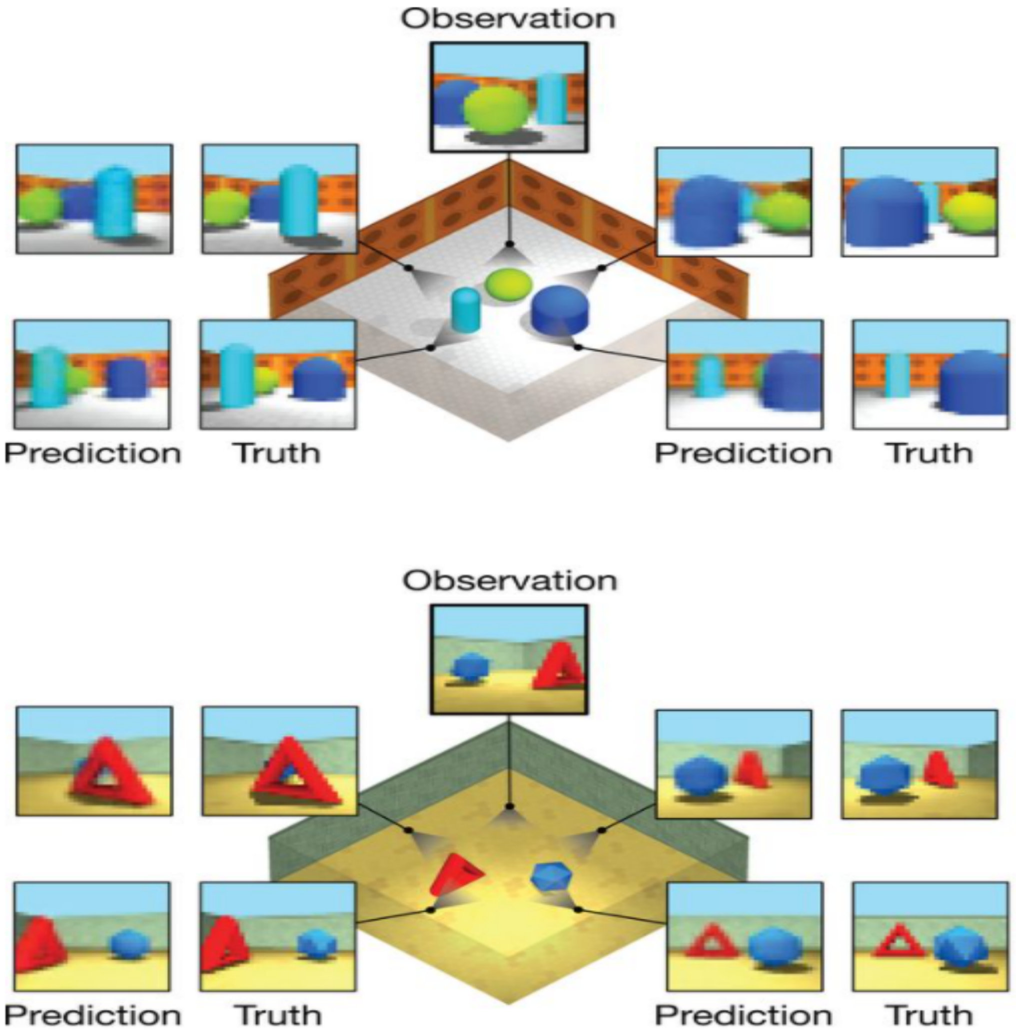
GQN learns some information (latent variables) to predict unknown target : Flexibility



GQN learns some information (latent variables) to predict unknown target : Flexibility

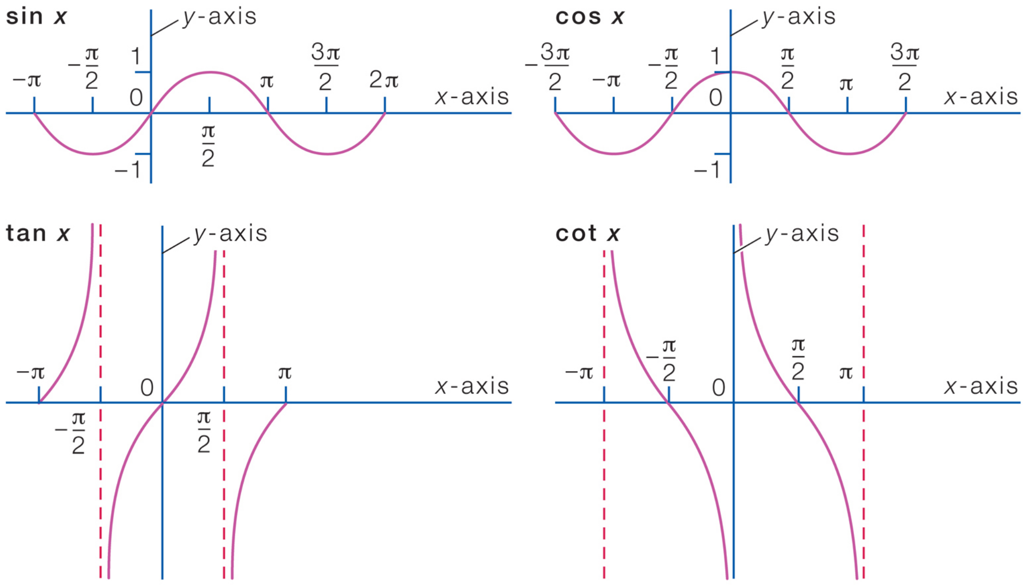


GQN for function regression gives conditional neural process (CNP)

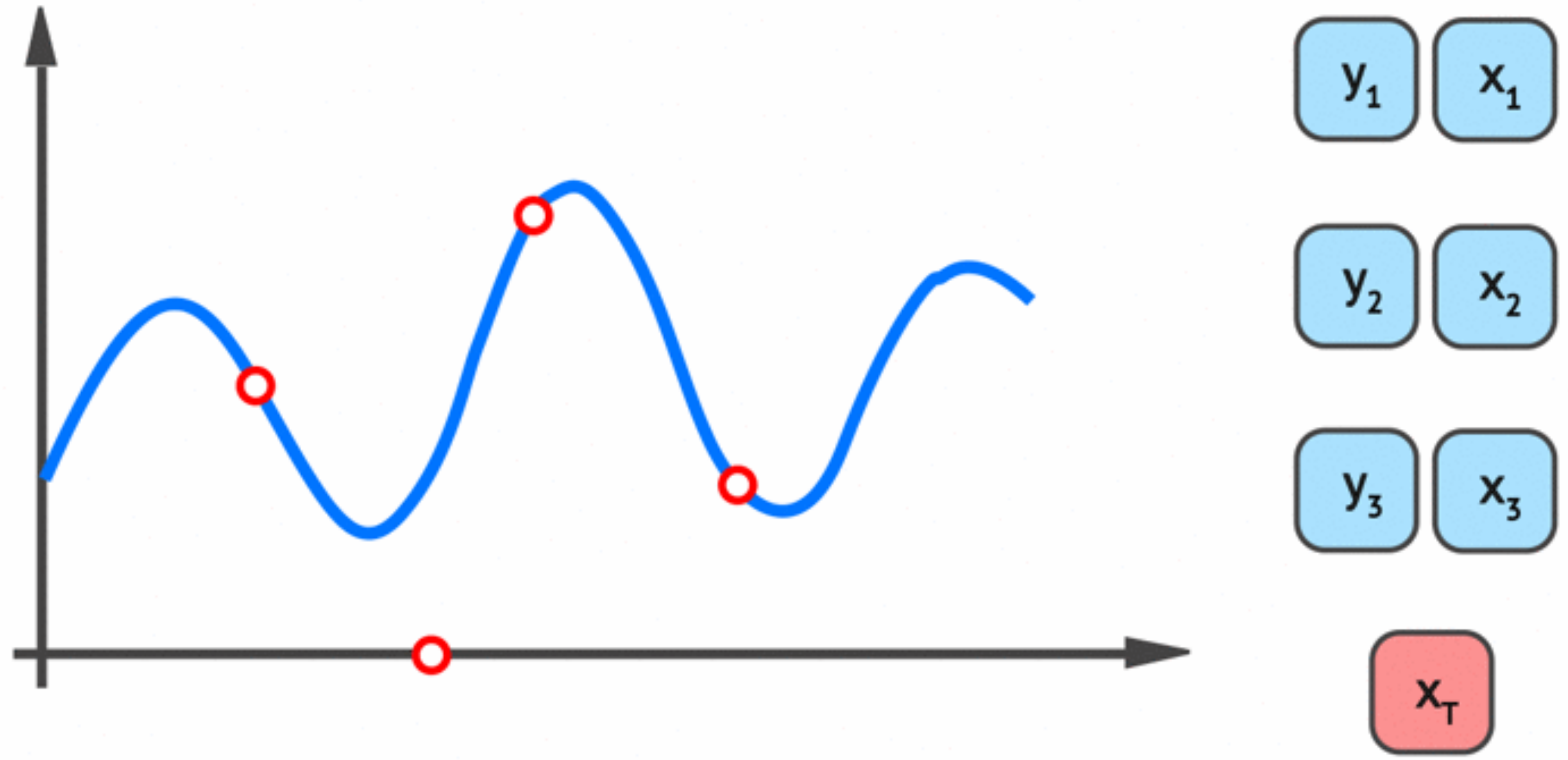


- GQN learns objects / position / light / color of the observation

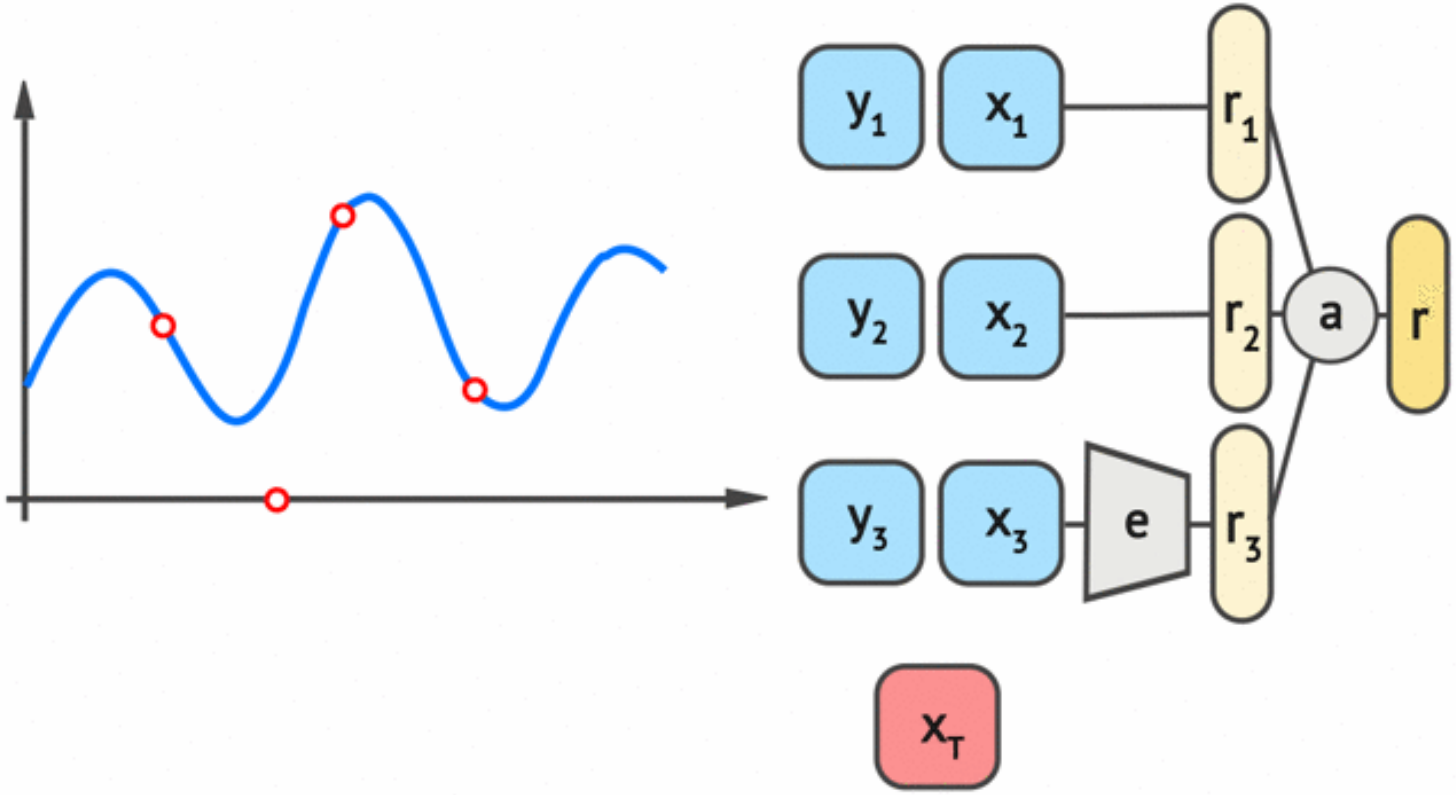
GP!!



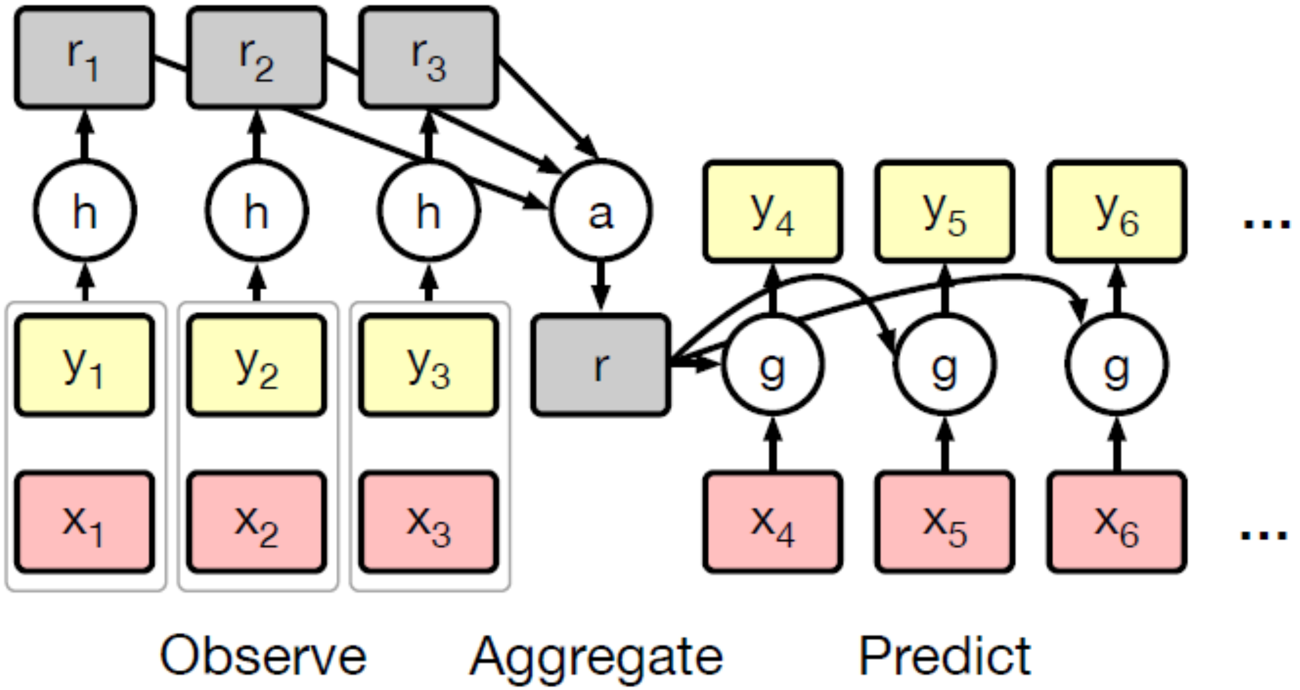
CNP encodes training data (context) with encoder (NN) to find representation



CNP decodes representation at test time as mean and variance of Gaussian distribution



Architecture of CNP ensures invariance to permutation of observation and target



Encoder (NN)

$r_i = h_\theta(x_i, y_i) \qquad \forall (x_i, y_i) \in O$

$r = r_1 \oplus r_2 \oplus \ldots r_{n-1} \oplus r_n$

Commutative operation (Mean)

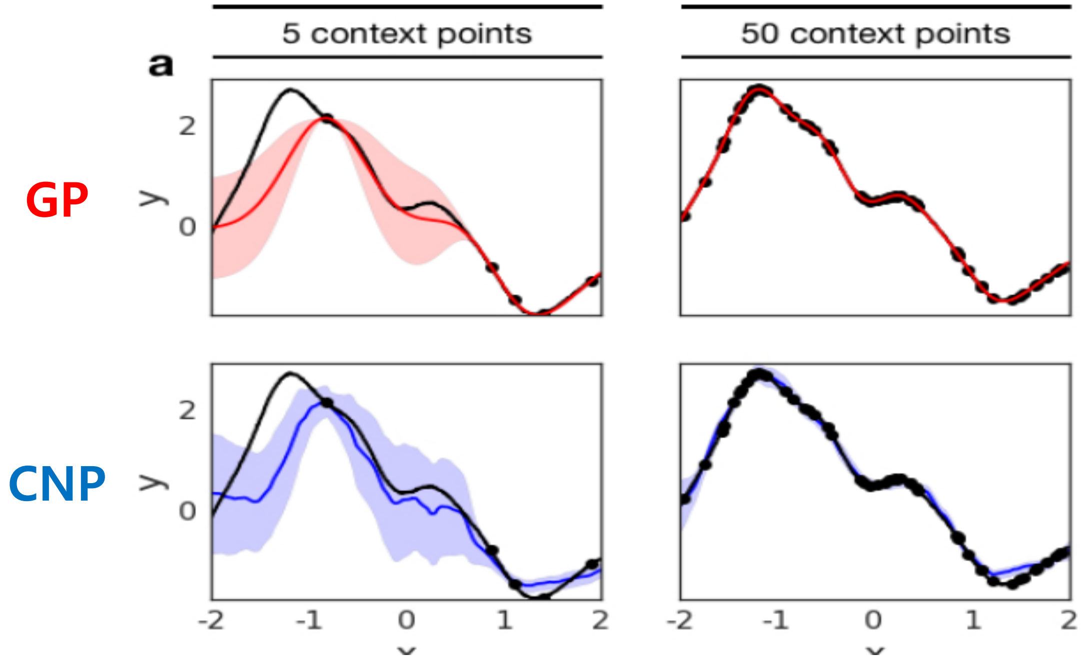
$\phi_i = g_\theta(x_i, r) \qquad \forall (x_i) \in T$

Decoder ($\phi_i = (\mu_i, \sigma_i^2)$: Gaussian)

✓ $O(n+m)\ cost : n = \#obs, m = \#target$

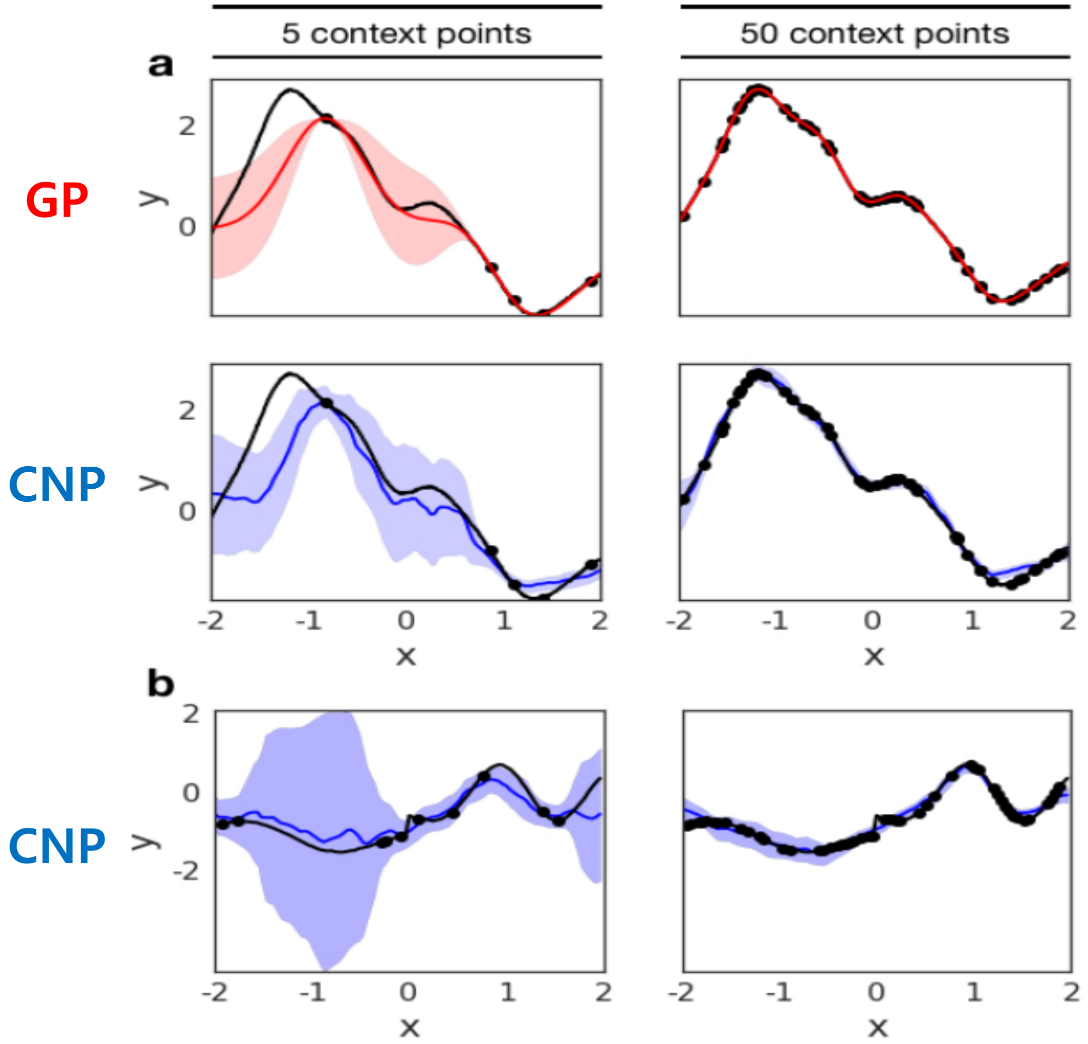
✓ Suitable for online learning ($O(1)$ for calculate r from previous)

Result for 1d-regression : Can achieve similar accuracy with GP



- Training : Sample a curve from the GP
- Context : We observe 5 / 50 points for new curve we want to fit
- GP is upper bound for accuracy

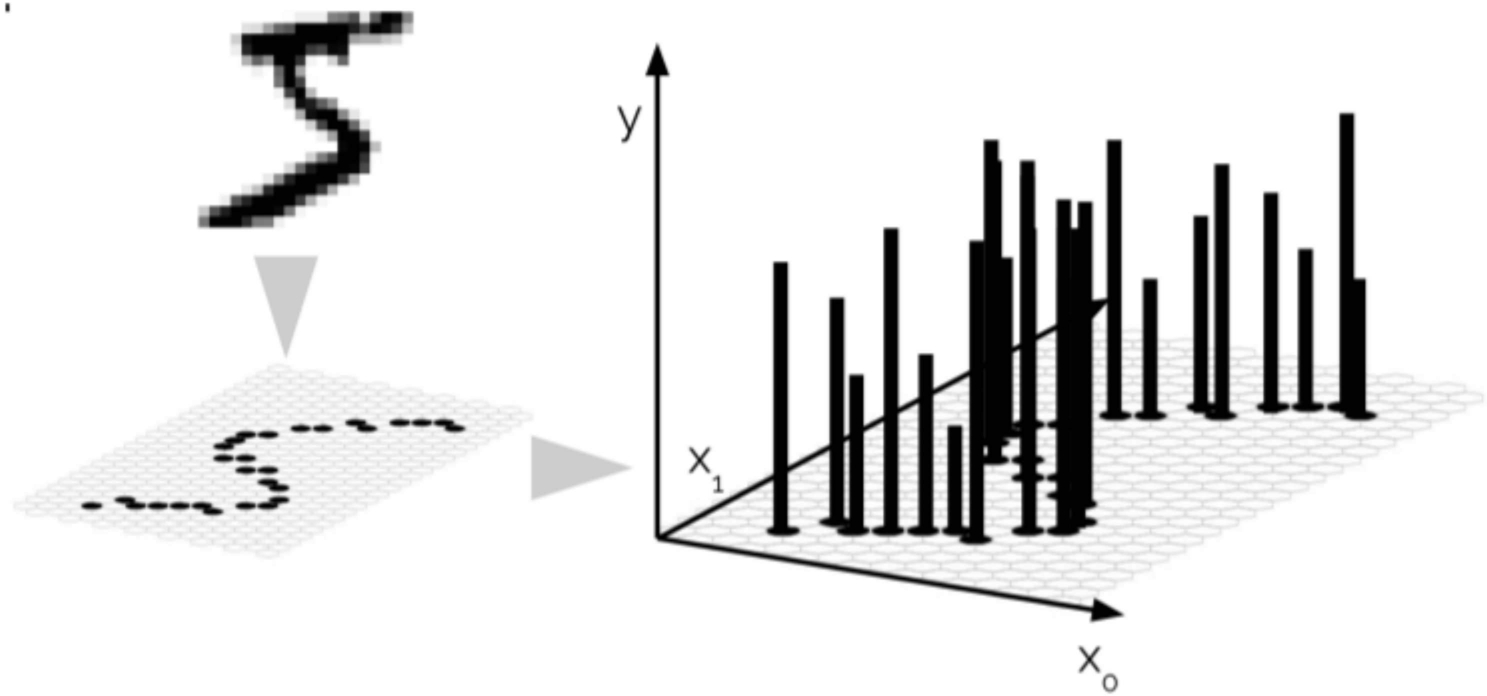
Result for 1d-regression : Can predict curve from GPs having different kernels



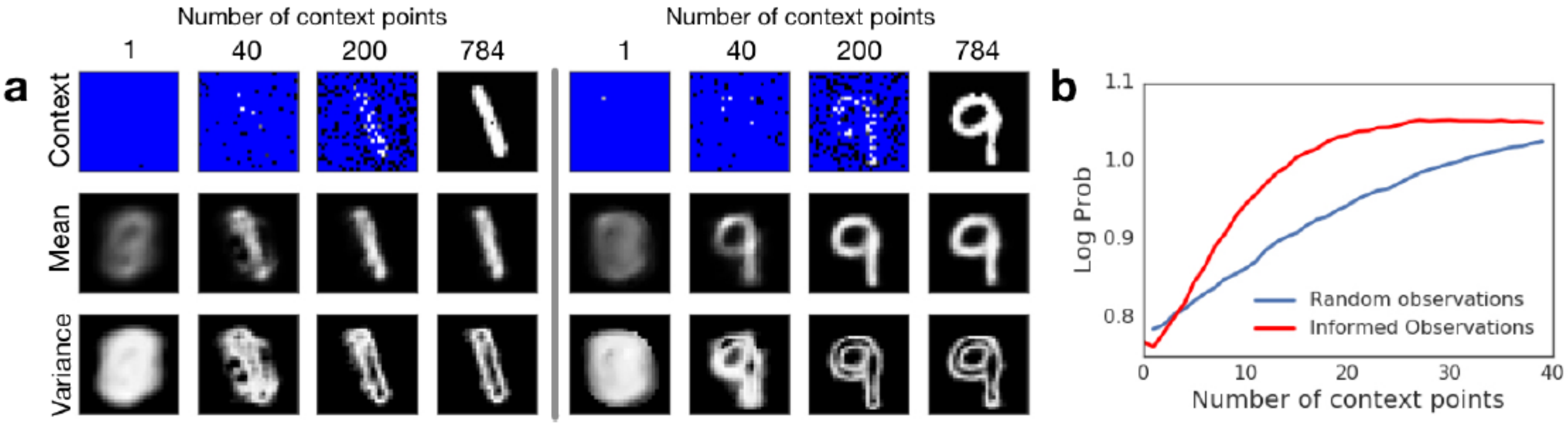
- Training : Sample a curve from the GP
- Context : We observe 5 / 50 points for new curve we want to fit
- GP is upper bound for accuracy

- Training : Sample point of curve with GPs having different kernel parameters
- Not trivial for GPs

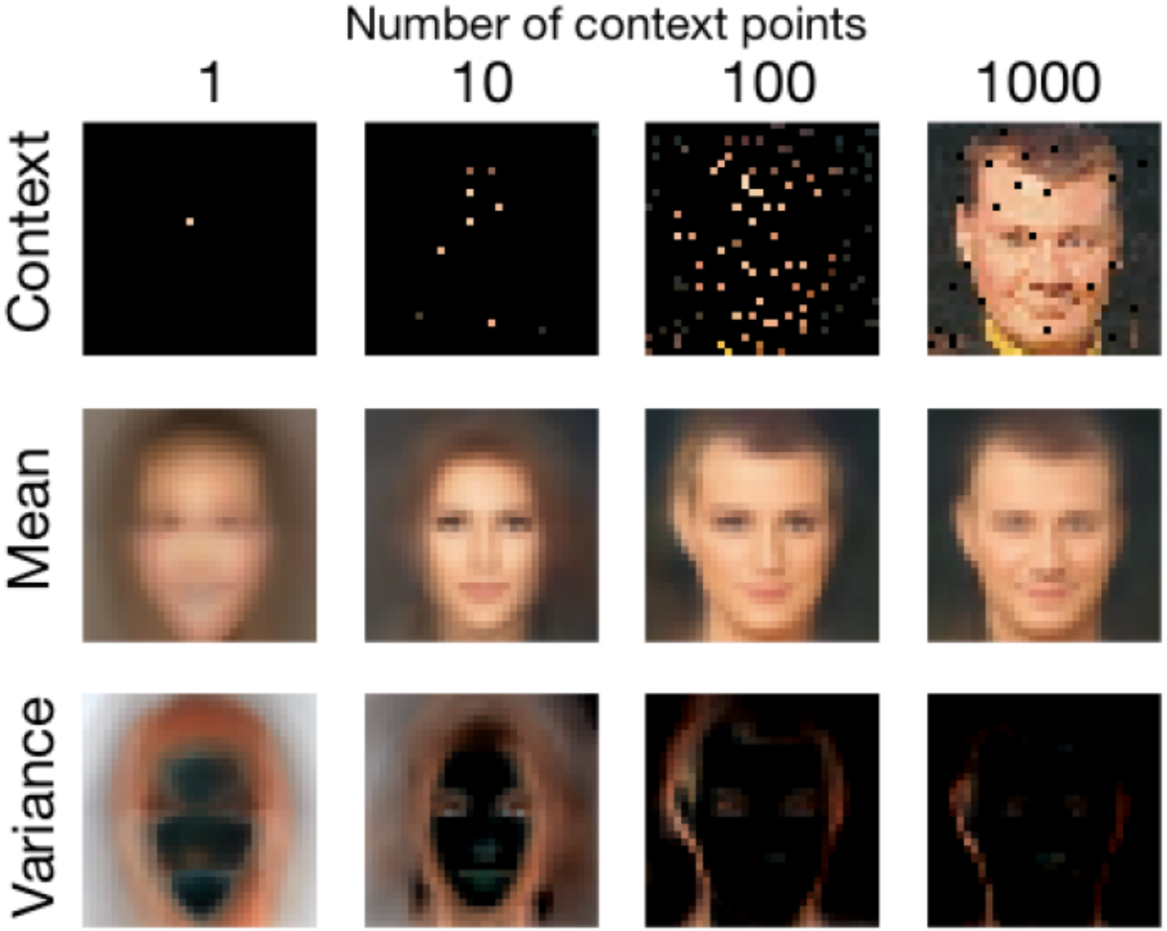
Image completion : Regression for pixel



Result for image completion
: Good prediction of mean & variance



Result for image completion
: Good prediction of mean & variance



✓ Bad points : Too smooth
(underfitting for lots of context)

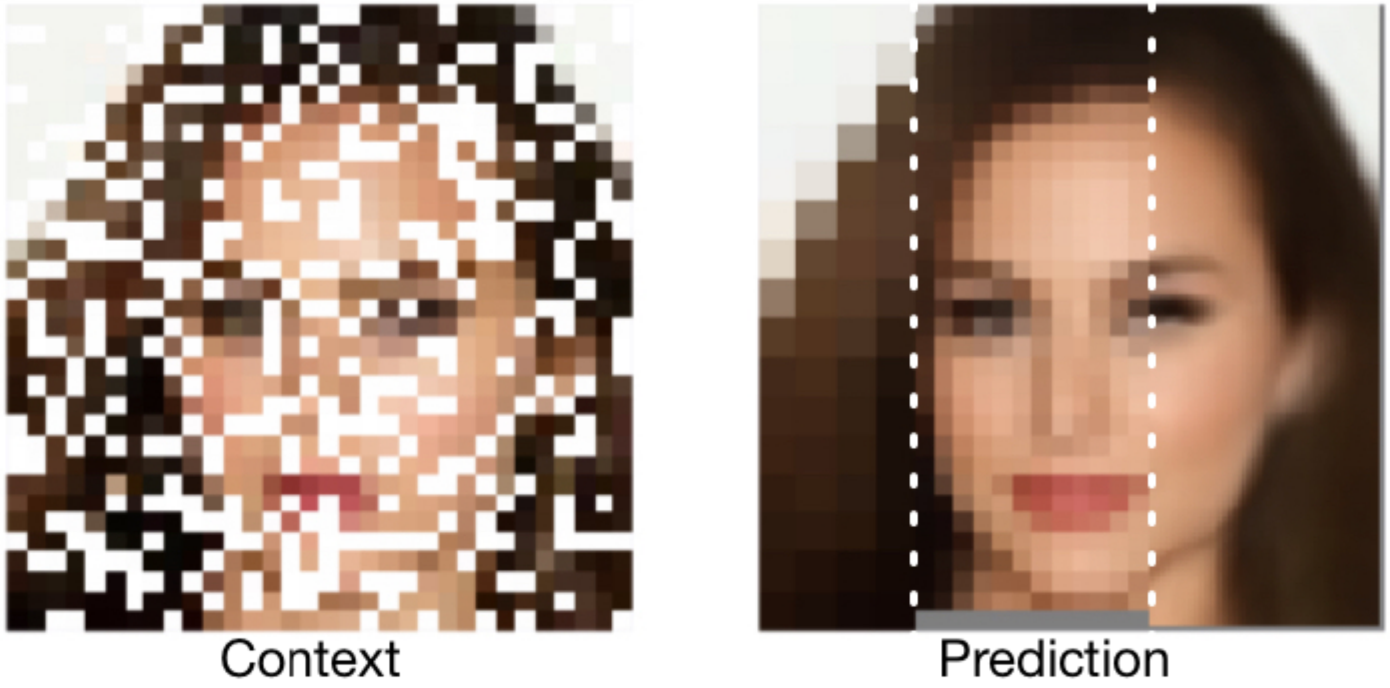
Result for image completion
: Flexible image completion (Recall permutation invariance)

✓ Any position of context are given, CNP can predict other parts

Result for image completion
: Flexible image completion (Recall permutation invariance)

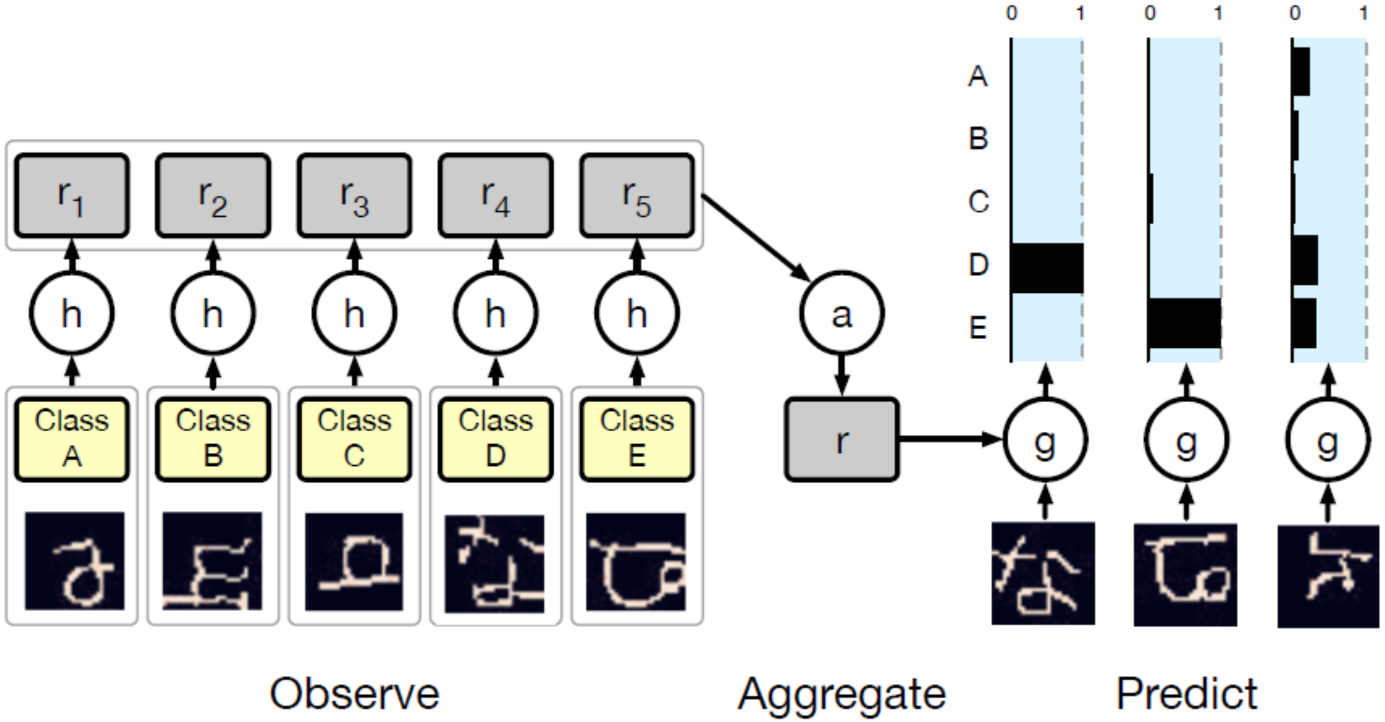
✓ Any position of context are given, CNP can predict other parts

Result for image completion
: Flexible image completion (Recall generalization)



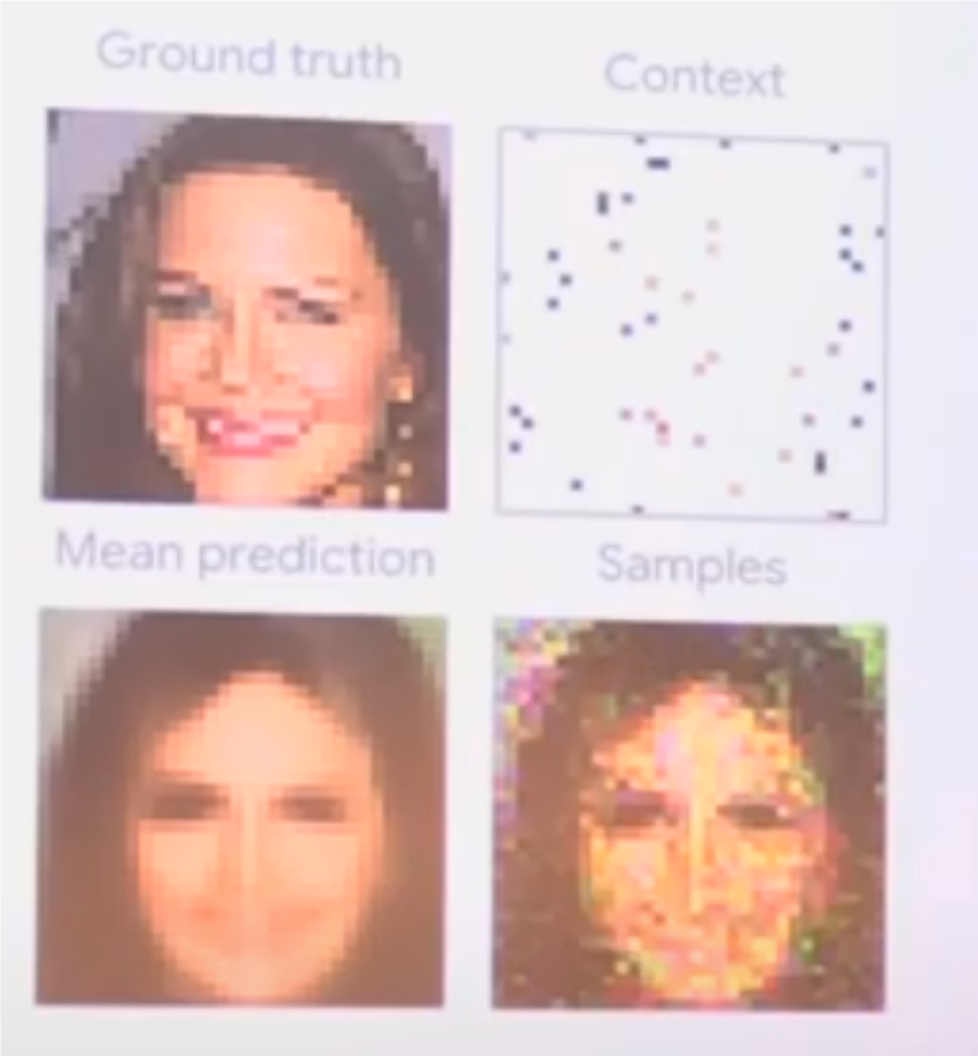
✓ Can predict different resolution which are not shown during training

Image classification : Given image and label



	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%	$\mathcal{O}(n+m)$

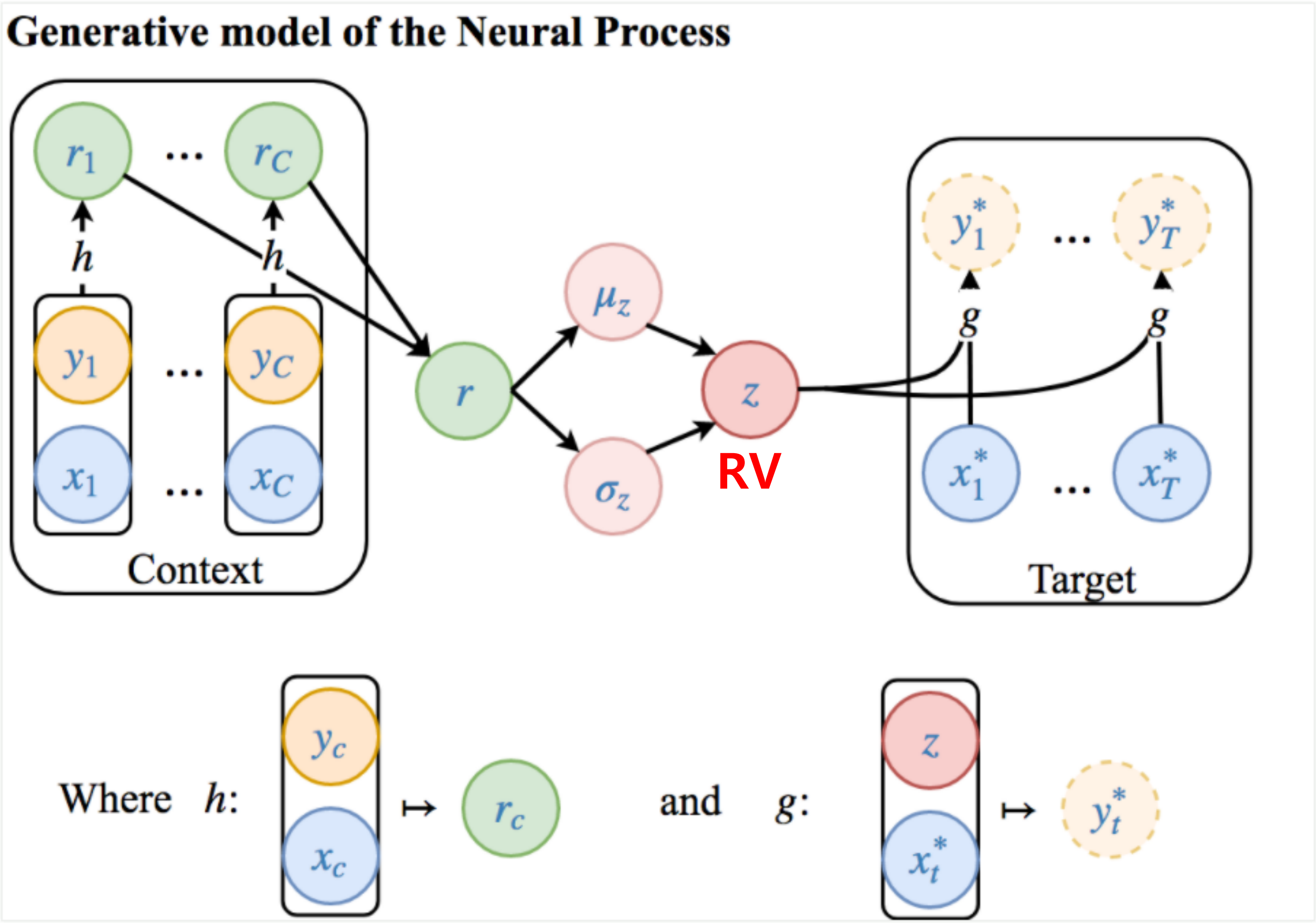
Sampling is not adequate for CNP

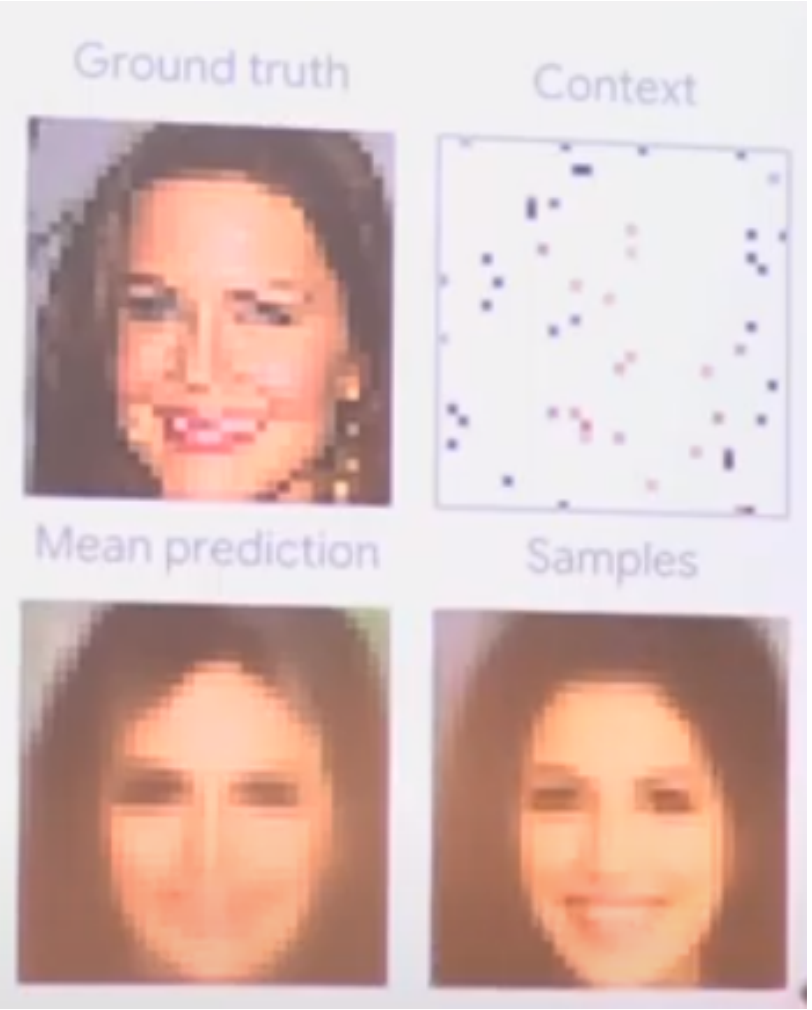


Sampling is done for each pixel
(no trend finding in whole pixels)

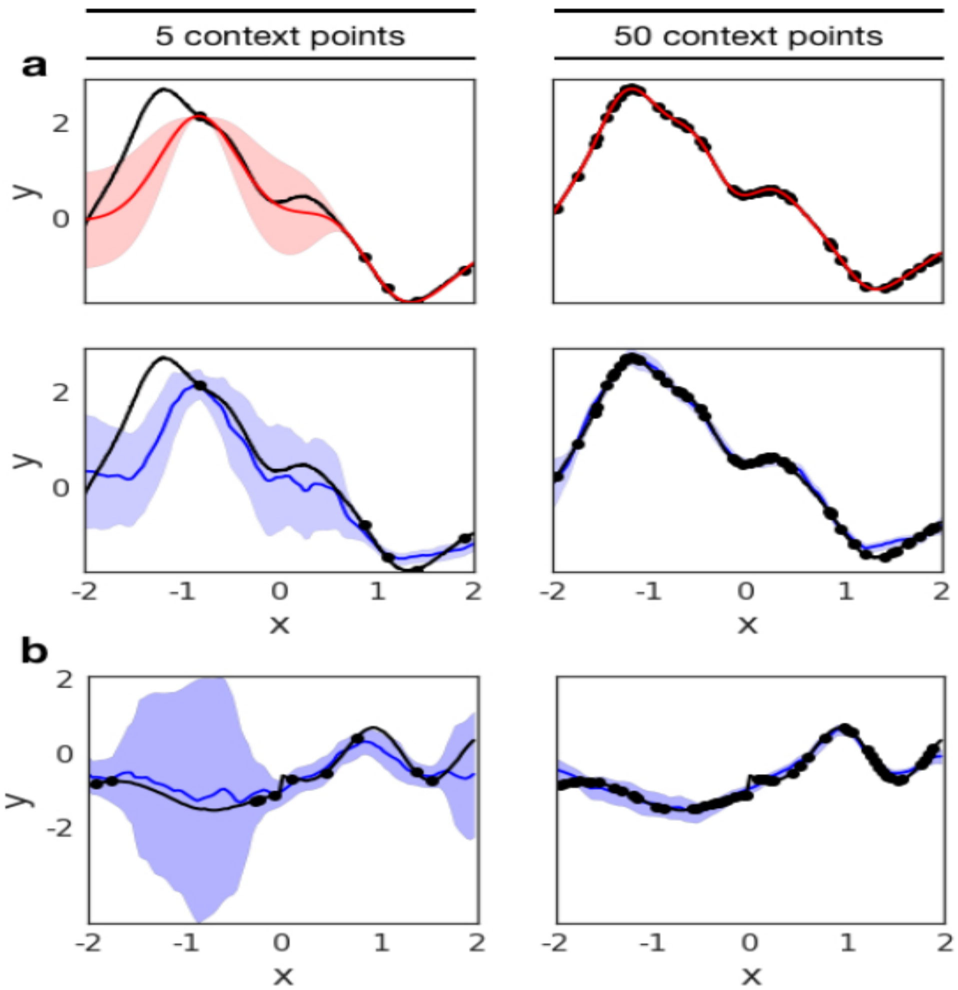
✓ Deterministic process (No random variable)

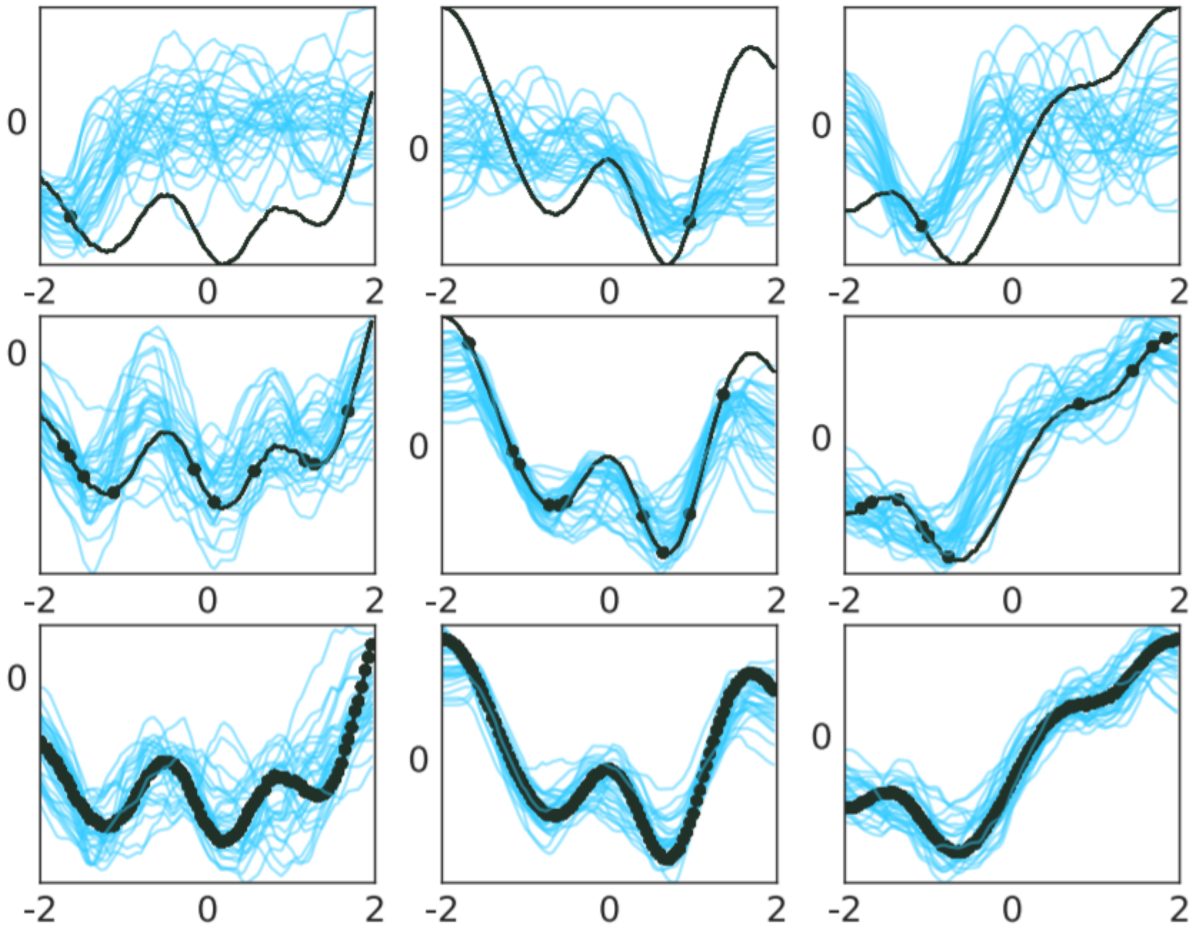
Neural process (NP) use latent variables to enable sampling and to give global consistency





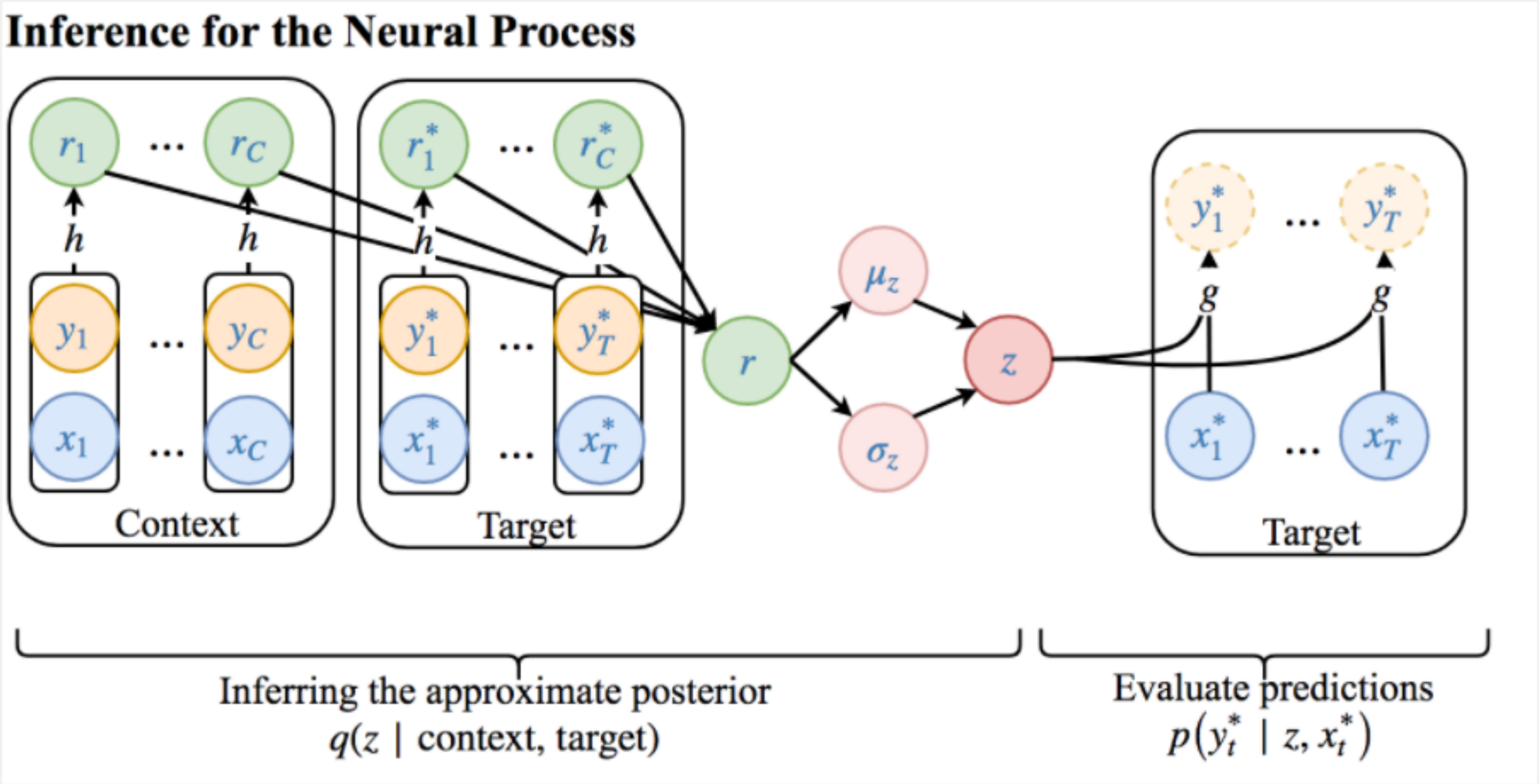
Rather than one mean & variance in CNP, NP provides samples



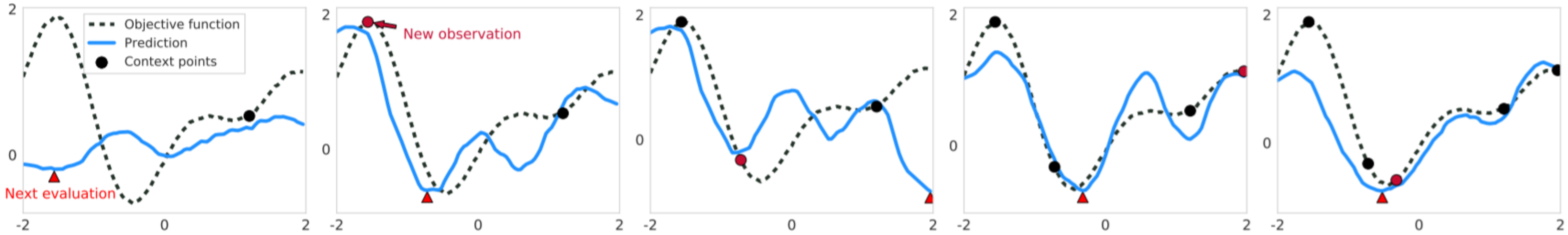


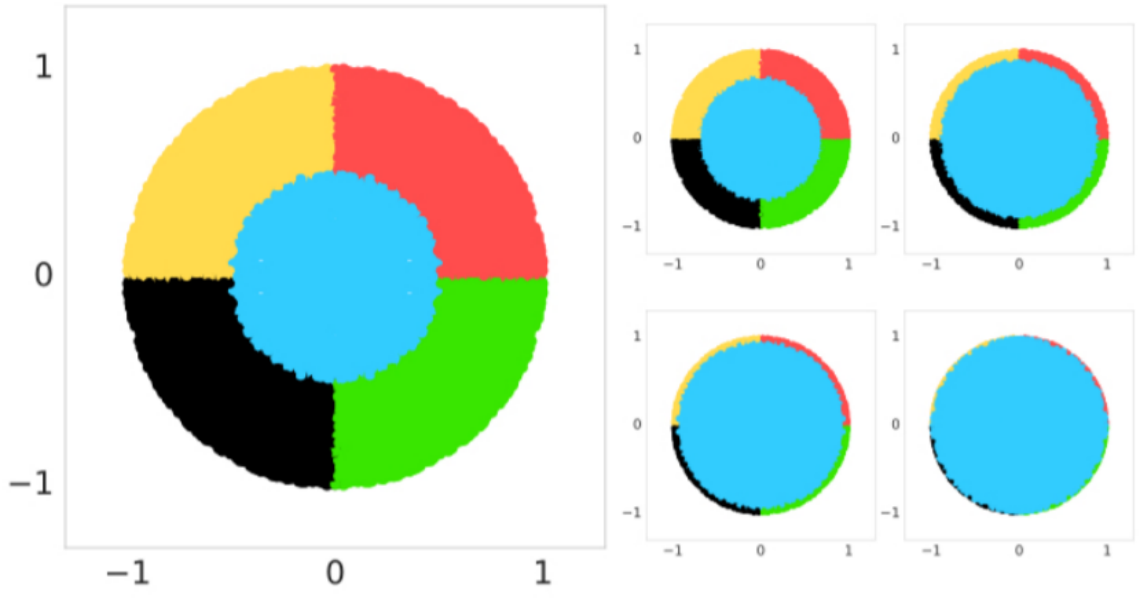
Inference of NP

- $\log(y_T|x_T, x_C, y_C) \geq E_{q(z|x_T,y_T)}[\sum_{target} \log p(y_i|z, x_i) + \log(\frac{p(z|x_T,y_T)}{q(z|x_T,y_T,x_C,y_C)})]$



Result for problem need sampling





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

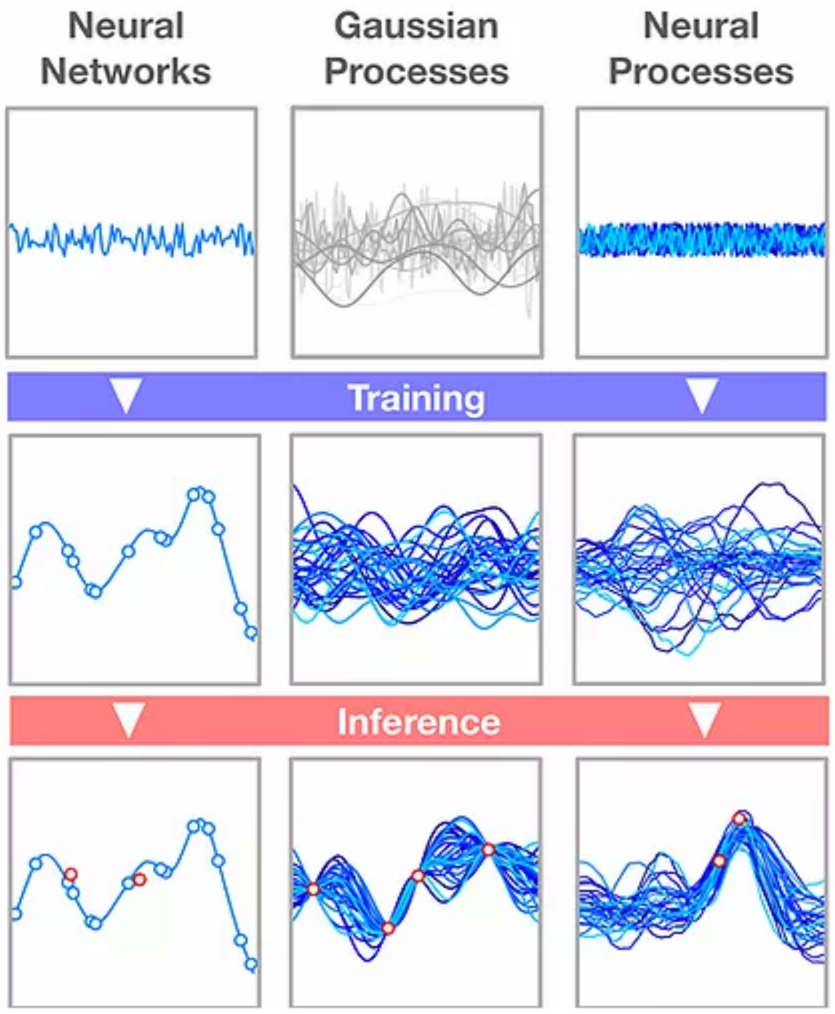
Comparison between NN, GP and NP

Neural Networks

Can learn the right parameters from data directly

Large number of parameters can regress complex functions

Quick predictions at test time



Prior : kernel Prior : Training domain

Gaussian processes

Model distribution over functions and updates belief about underlying ground truth based on observations at test time

Has a measure of the uncertainty over its predictions given the observations

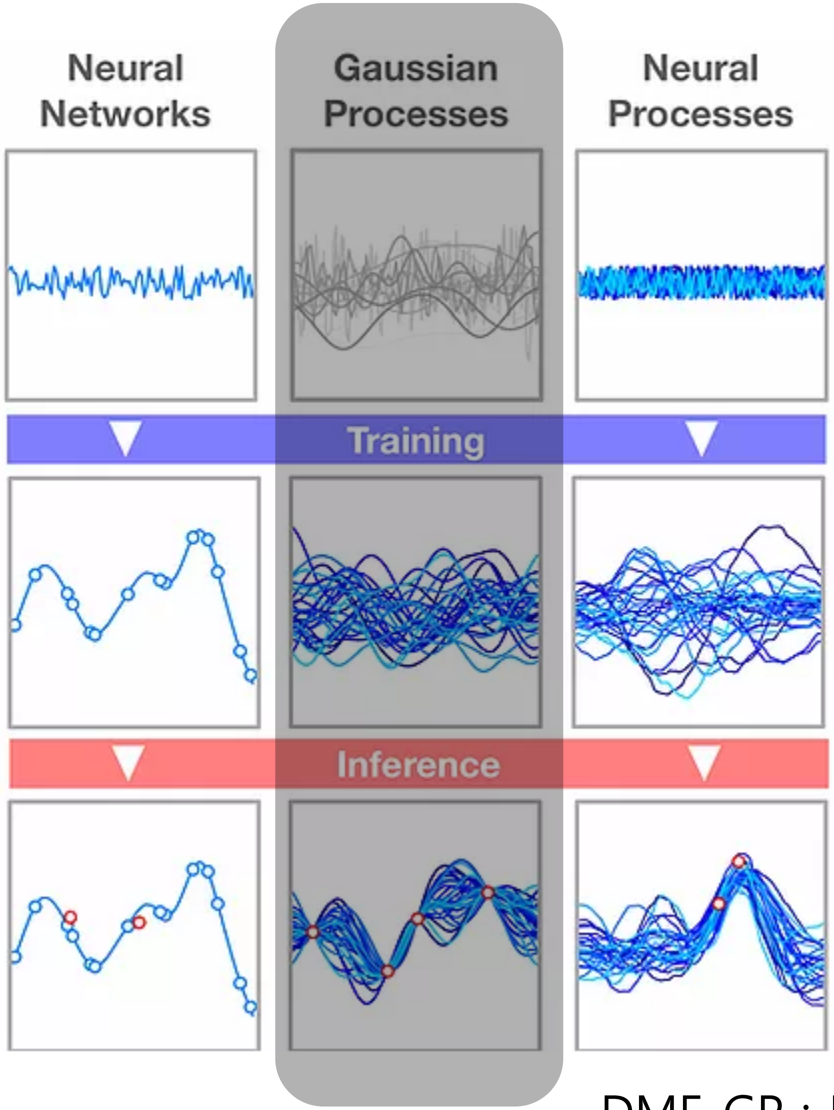
What about DME-GP?

Neural Networks

Can learn the right parameters from data directly

Large number of parameters can regress complex functions

Quick predictions at test time



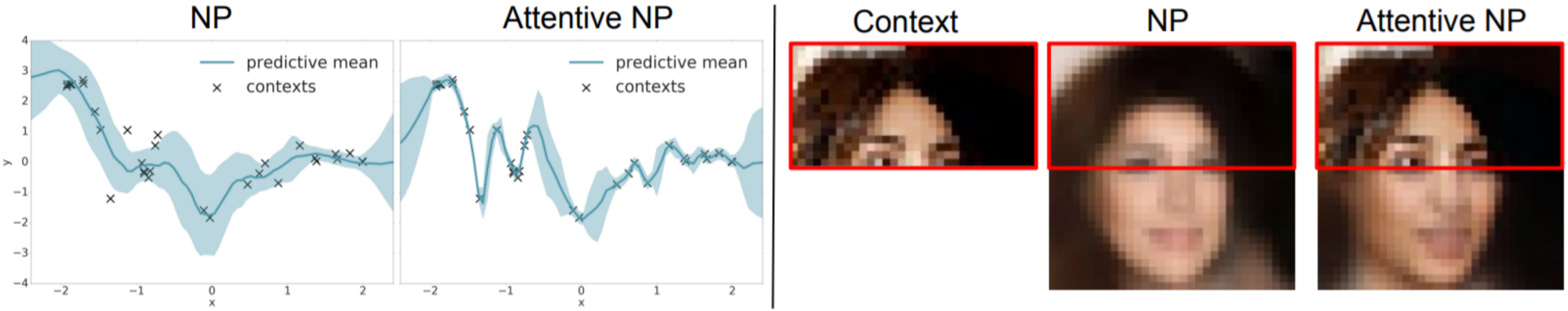
Gaussian processes

Model distribution over functions and updates belief about underlying ground truth based on observations at test time

Has a measure of the uncertainty over its predictions given the observations

DME-GP : Fast GP

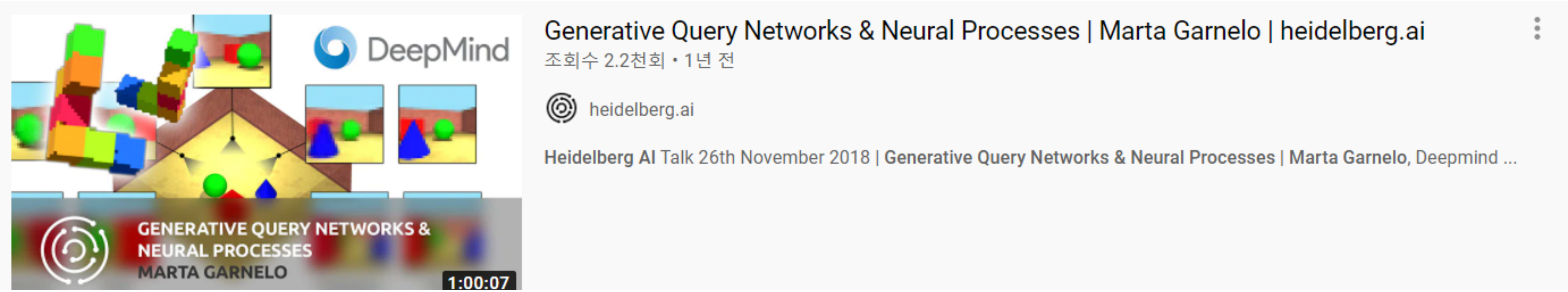
Although NP has lower cost than DME-GP, to get high accuracy for given contexts it NP also needs same cost



✓ Attentive NP have almost same cost with DME-GP

Reference

- Eslami, SM Ali, et al. "Neural scene representation and rendering." Science 360.6394 (2018): 1204-1210.
- Garnelo, Marta, et al. "Conditional neural processes." *arXiv preprint arXiv:1807.01613* (2018).
- Garnelo, Marta, et al. "Neural processes." *arXiv preprint arXiv:1807.01622* (2018).
- Kim, Hyunjik, et al. "Attentive neural processes." *arXiv preprint arXiv:1901.05761* (2019).



- https://kasparmartens.rbind.io/post/np/