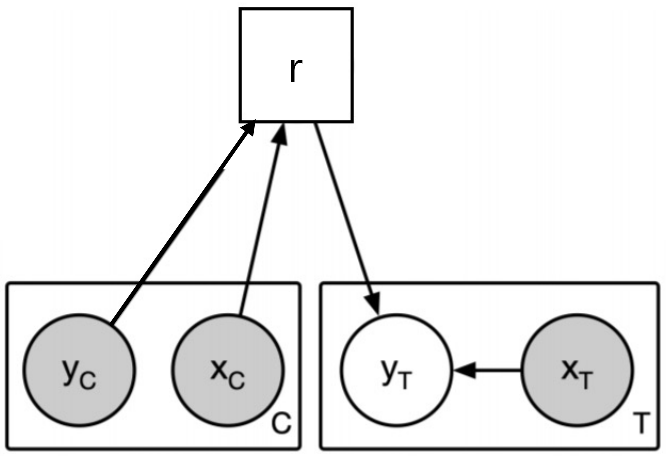
The Functional Neural Process

Accepted in Neurips 2019

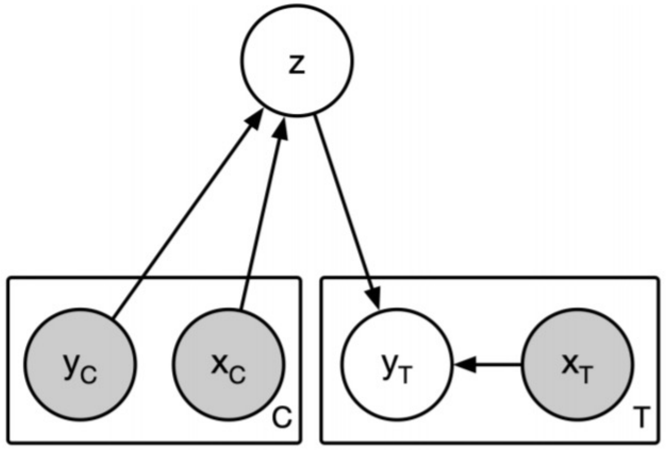
Kyeong Ryeol, Go
M.S. Candidate of OSI Lab

Family of Neural Processes

- Devise a neural network based Gaussian process
- Enables fast adaptation by the prior knowledge from the portion of the data



Conditional Neural Process



Neural Processes

- Loss function

$$\text{CNP} : -\log p(y_T | x_T, r)$$

$$\text{NP} : -\mathbb{E}_{\text{z}\sim \text{q}(\text{z}|x_C, y_C, x_T, y_T)}[\log \text{p}(y_T|x_T, \text{z})] + \text{KL}[\text{q}(\text{z}|x_C, y_C, x_T, y_T) \| q(z|x_C, y_C)]$$

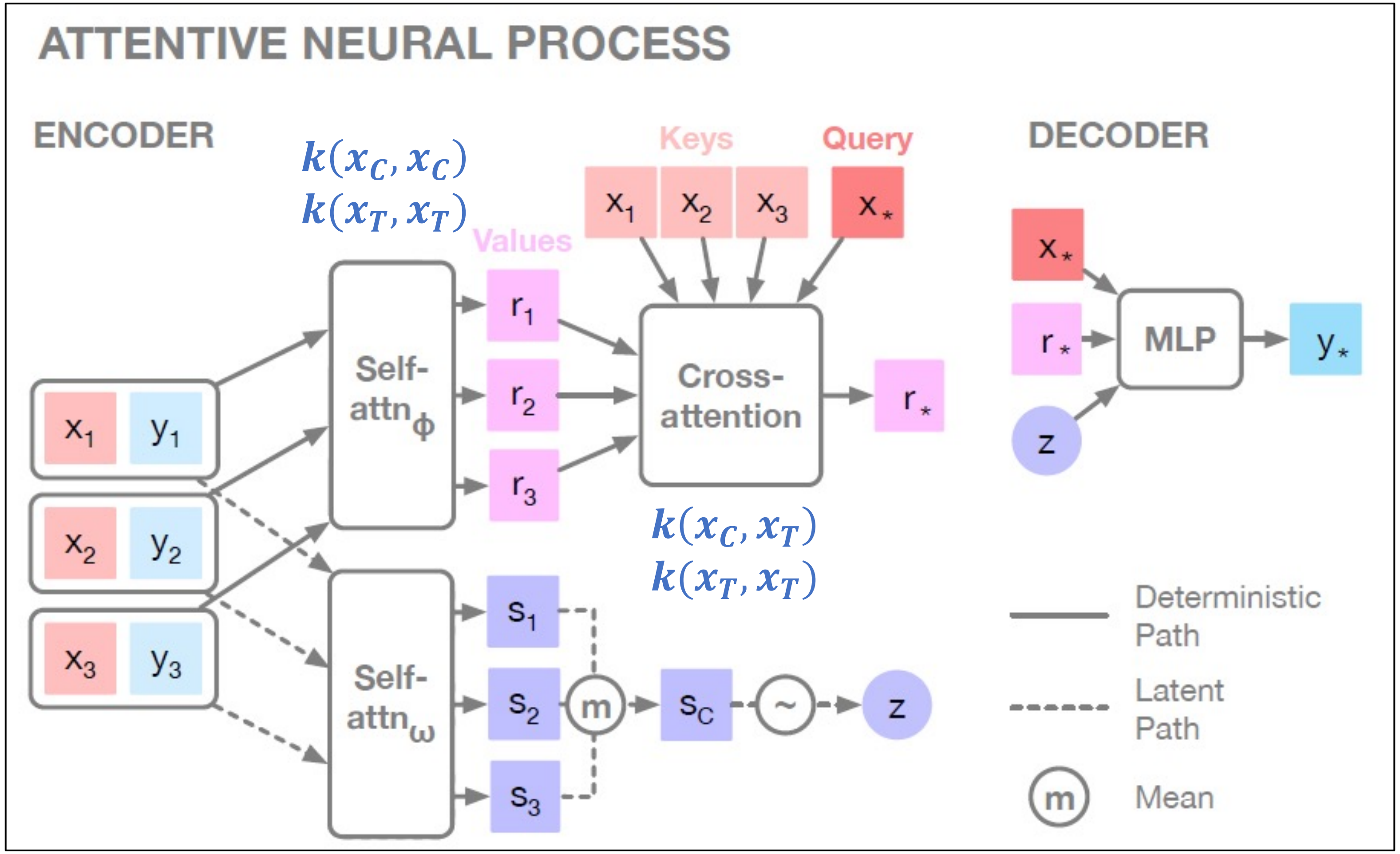
continued

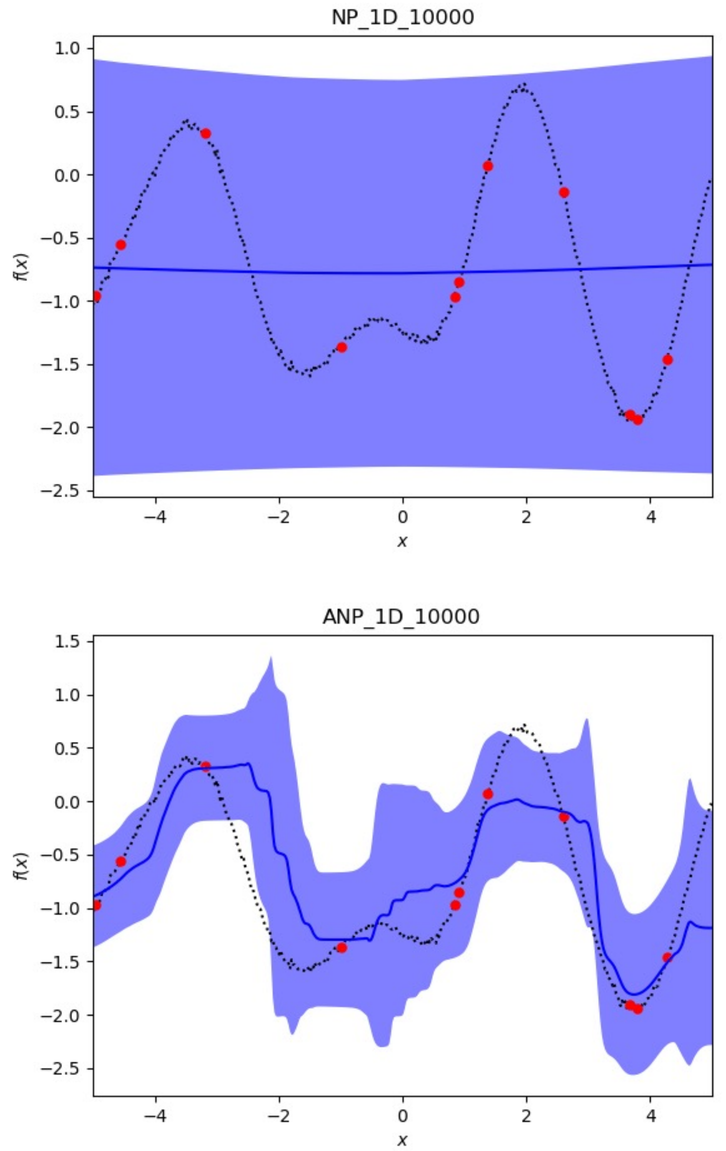
- Problem scenario
  - divide the dataset into context set $\{x_C, y_C\}$ and target set $\{x_T, y_T\}$
  - learn a conditional distribution p$(f(x_T)|x_T, x_C, y_C)$

- Gaussian Process : $p(f(x_T)|x_T, x_C, y_C) = \mathcal{N}\left(y_{mu}, y_{sigma}^2\right)$
  - $y_{mu} = k(x_T, x_C)(k(x_C, x_C) + \sigma^2 I)^{-1} y_C$
  - $y_{sigma}^2 = k(x_T, x_T) - k(x_T, x_C)(k(x_C, x_C) + \sigma^2 I)^{-1} k(x_C, x_T))$

- **Scalability** : computation scales linearly in NP
- **Flexibility** : A wide variety of family of distribution can be defined
- **Permutation invariance** : target predictions are order invariant in the contexts

continued





"Considering the dependency of the data points are significant"

The Functional Neural Process

- Devise a neural network based Gaussian process
- Enables fast adaptation by adopting the relational structure of the dataset

- Problem scenario
  - Divide the input X into reference set $\{R\}$ and remaining set $\{M\}$
  - learn a conditional distribution $\mathrm{p}(f(M)|y_R, R, M)$

- Structure
  1. Local latent variables u is computed from the dataset X
  2. The relational structure A, G is constructed by the local latent variables u
  3. Prediction of y_M is computed by the information from the reference set R, y_R and the relational structure A, G

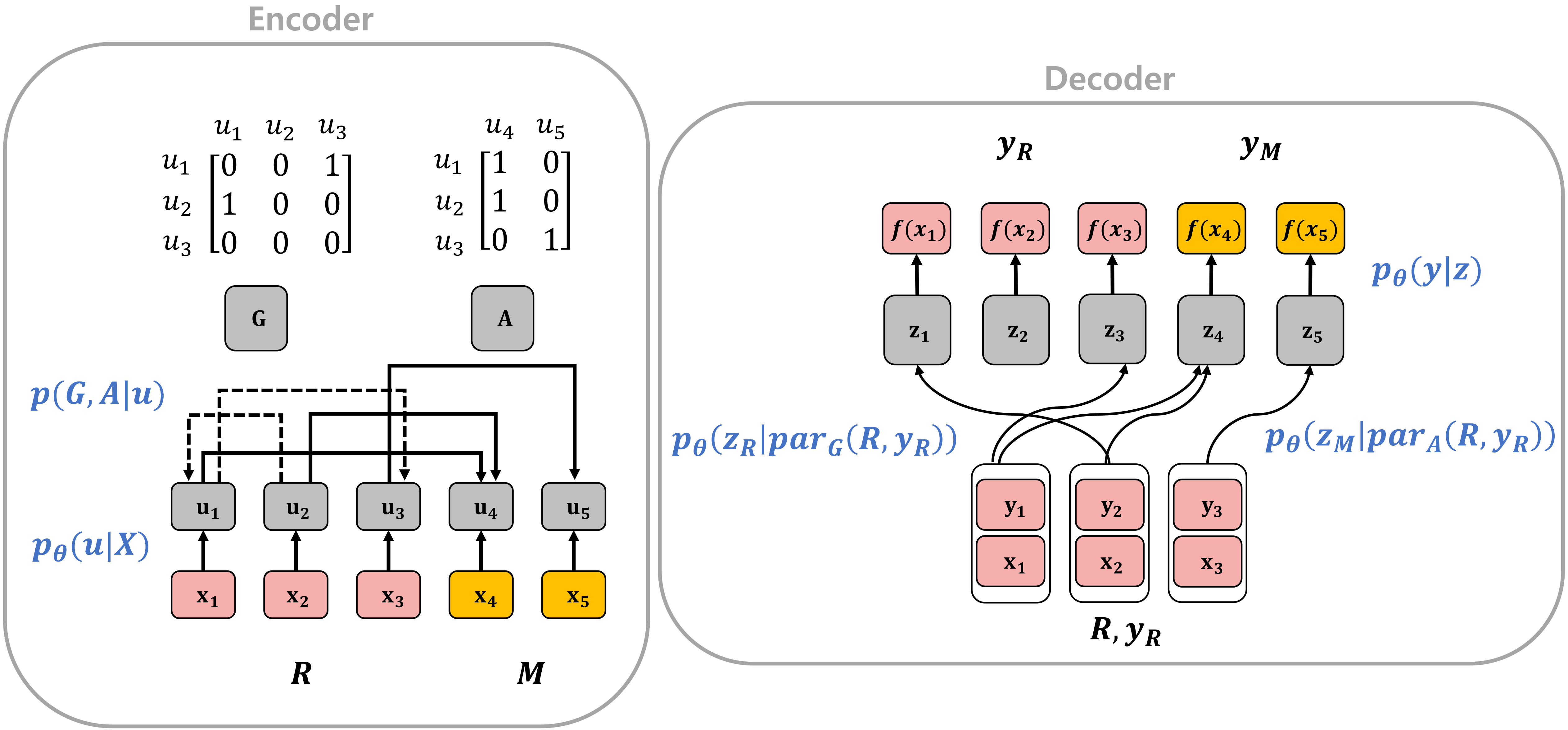
Relational structure

- Graph
  - A : from the reference set R to the remaining set M
  - G : among the reference set R

$$A : \begin{bmatrix} A_{1,1} & \cdots & A_{1,|M|} \\ \vdots & \ddots & \vdots \\ A_{|R|,1} & \cdots & A_{|R|,|M|} \end{bmatrix} \qquad G : \begin{bmatrix} G_{1,1} & \cdots & G_{1,|R|} \\ \vdots & \ddots & \vdots \\ G_{|R|,1} & \cdots & G_{|R|,|R|} \end{bmatrix}$$

- Formulation
  - $p(A|u_R, u_M) = \prod_{i\in R} \prod_{j\in M} Bernoulli\left(A_{ij} \mid g(u_i, u_j)\right)$
  - $p(G|u_R) = \prod_{i\in R} \prod_{j\in R, j\neq i} Bernoulli\left(G_{ij} \mid I\left[t(u_i) > t(u_j)\right] g(u_i, u_j)\right)$
    - $t(u_i) = \sum_k t_k(u_{ik})$ $where$ $t_k(\cdot)$ is the $\log CDF$ of $N(0,1)$
    - $g(u_i, u_j) = \exp\left(-\frac{\tau}{2}\left\|u_i - u_j\right\|^2\right)$: functional space

continued



Encoder
u_1 u_2 u_3
u_1 u_2 u_3
0 0 1
1 0 0
0 0 0
u_4 u_5
u_1 u_2 u_3
1 0
1 0
0 1
G
A
$p(G, A|u)$
u_1 u_2 u_3 u_4 u_5
$p_\theta(u|X)$
x_1 x_2 x_3 x_4 x_5
R
M
Decoder
y_R
y_M
$f(x_1)$ $f(x_2)$ $f(x_3)$ $f(x_4)$ $f(x_5)$
$p_\theta(y|z)$
z_1 z_2 z_3 z_4 z_5
$p_\theta(z_R|par_G(R, y_R))$
$p_\theta(z_M|par_A(R, y_R))$
y_1 y_2 y_3
x_1 x_2 x_3
R, y_R

Loss function

- Marginal likelihood
  - $\log p(y|X) = \log \sum_{G,A} \int p_\theta(u, G, A, z, y|X)\, du\, dz$

$$= \log \sum_{G,A} \int p_\theta(u|X) p(G, A|u) p_\theta(z_R|par_G(R, y_R)) p_\theta(z_M|par_A(R, y_R)) p_\theta(y|z)\, du\, dz$$

To be canceled

- Variational Learning
  - Variational distribution : $q_\phi(u, G, A, z|X) = p_\theta(u|X) p(G, A|u) q_\phi(z|X)$
  - $\log p(y|X)$

$$\geq E_{q_\phi(u,G,A,z|X)}[\log p_\theta(u, G, A, z, y|X) - \log q_\phi(u, G, A, z|X)]$$

$$= E_{q_\phi(u,G,A,z|X)}[\log p_\theta(z_R|par_G(R, y_R)) p_\theta(z_M|par_A(R, y_R)) p_\theta(y|z) - \log q_\phi(z|X)]$$

$$= E_{p_\theta(u_R, G|R)}\left[E_{q_\phi(z_R|R)}[\log p_\theta(z_R|par_G(R, y_R) p_\theta(y_R|z_R)) - \log q_\phi(z_R|R)]\right]$$

$$+ E_{p_\theta(u, A|X)}\left[E_{q_\phi(z_M|M)}[\log p_\theta(z_M|par_A(R, y_R) p_\theta(y_M|z_M) - \log q_\phi(z_M|M)]\right]$$

$$:= L_R + L_{M|R}$$

Minibatch optimization

- L_R cannot be decomposed to independent sums due to DAG structure
- $L_{M|R}$ can be decomposed to |M| independent sums from its i.i.d. nature

$$Loss = L_R + L_{M|R} \approx L_R + \hat{L}_{M|R}$$

* $L_{M|R}$

$$= E_{p_\theta(u,A|X)}\left[E_{q_\phi(z_M|M)}[\log p_\theta(z_M|par_A(R,y_R)p_\theta(y_M|z_M) - \log q_\phi(z_M|M)]\right]$$

$$= E_{p_\theta(u_R|R)}\left[E_{p_\theta(u_M|M)p(A|u_R,u_M)q_\phi(z_M|M)}[\log p_\theta(z_M|par_A(R,y_R)p_\theta(y_M|z_M) - \log q_\phi(z_M|M)]\right]$$

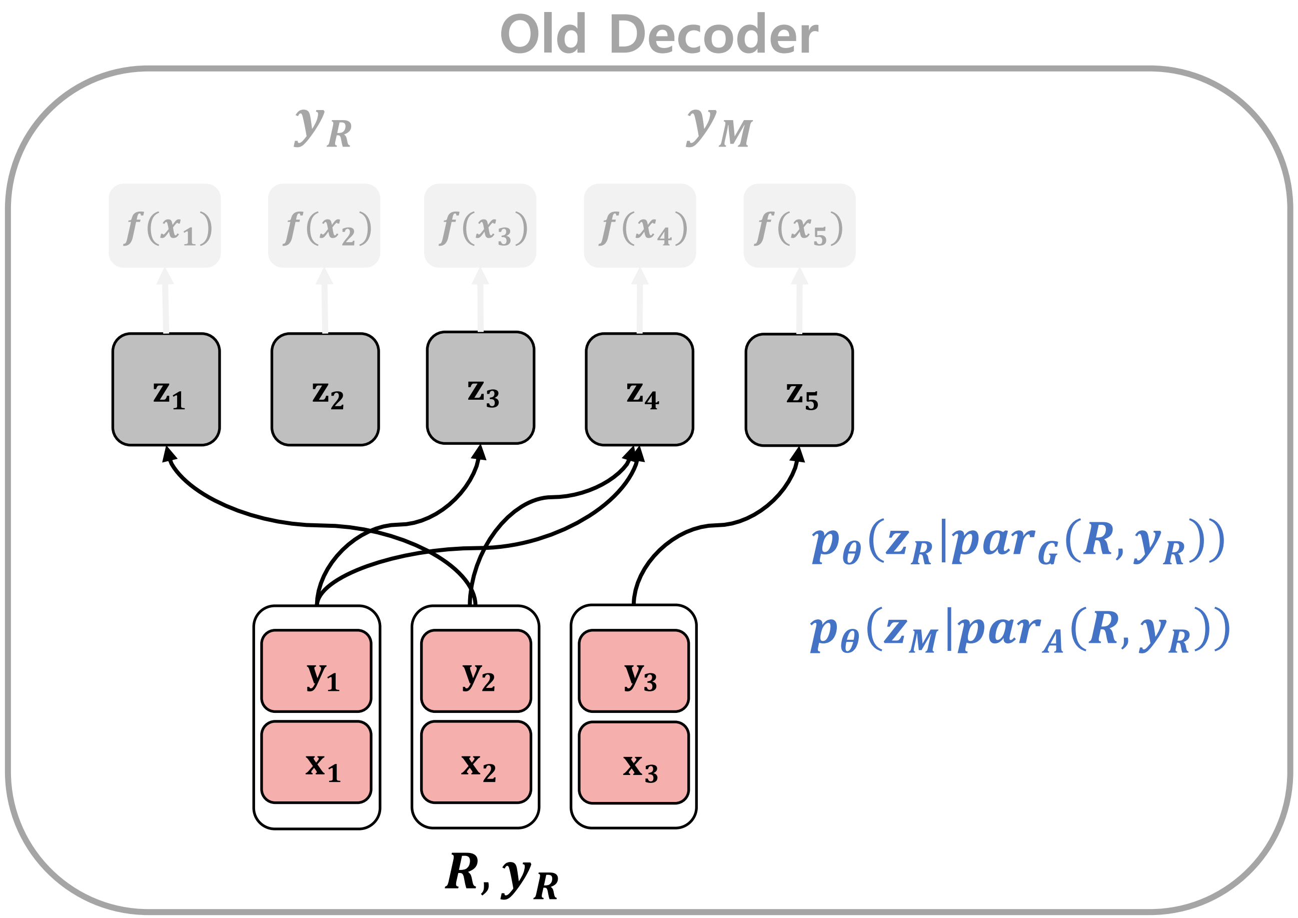
$$\approx E_{p_\theta(u_R|R)}\left[\frac{|M|}{|\hat{M}|}\sum_{i=1}^{|\hat{M}|} E_{p_\theta(u_i|x_i)p(A_i|u_R,u_i)q_\phi(z_i|x_i)}[\log p_\theta\left(z_i\,|par_{A_i}(R,y_R)p_\theta(y_i|z_i\right) - \log q_\phi(z_i|x_i)]\right]$$

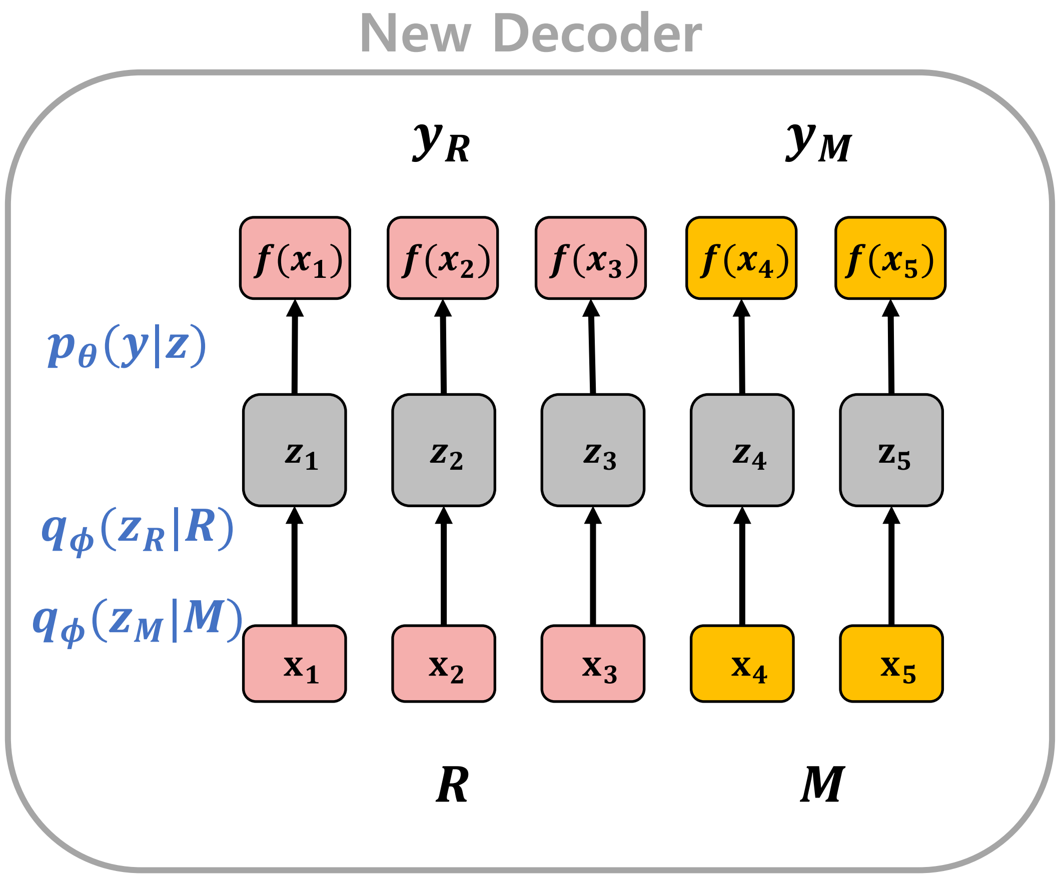
$$:= \hat{L}_{M|R}$$

Implementation strategy

$$L_R : E_{p_\theta(u_R, G|R)}\left[E_{q_\phi(z_R|R)}[\log p_\theta(y_R|z_R) + \log p_\theta(z_R|par_G(R, y_R)) - \log q_\phi(z_R|R)]\right]$$

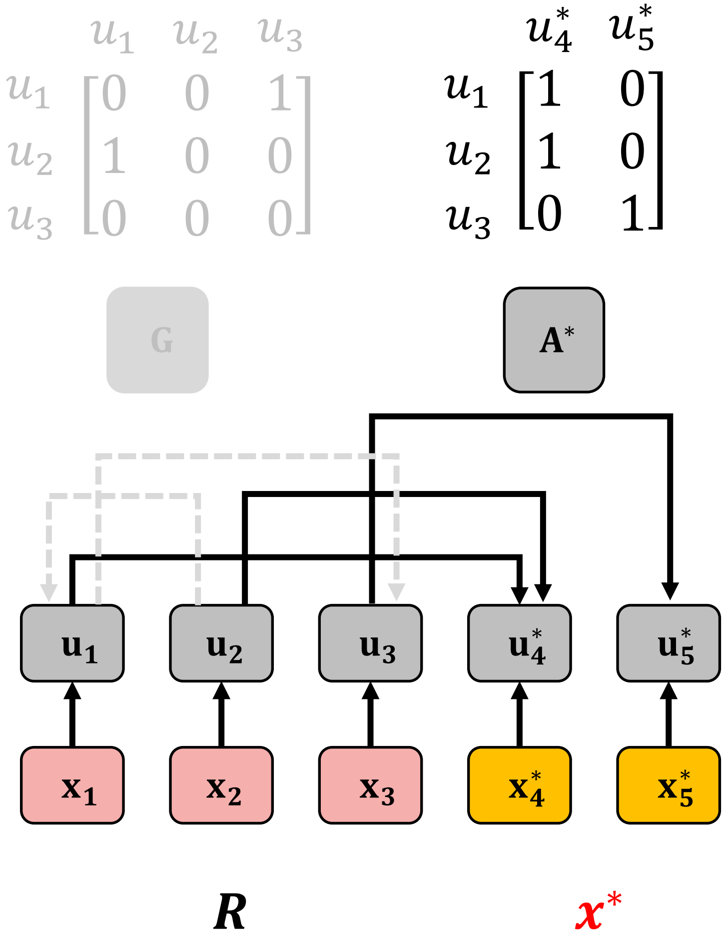
$$L_{M|R} : E_{p_\theta(u, A|X)}\left[E_{q_\phi(z_M|M)}[\log p_\theta(y_M|z_M) + \log p_\theta(z_M|par_A(R, y_R)) - \log q_\phi(z_M|M)]\right]$$

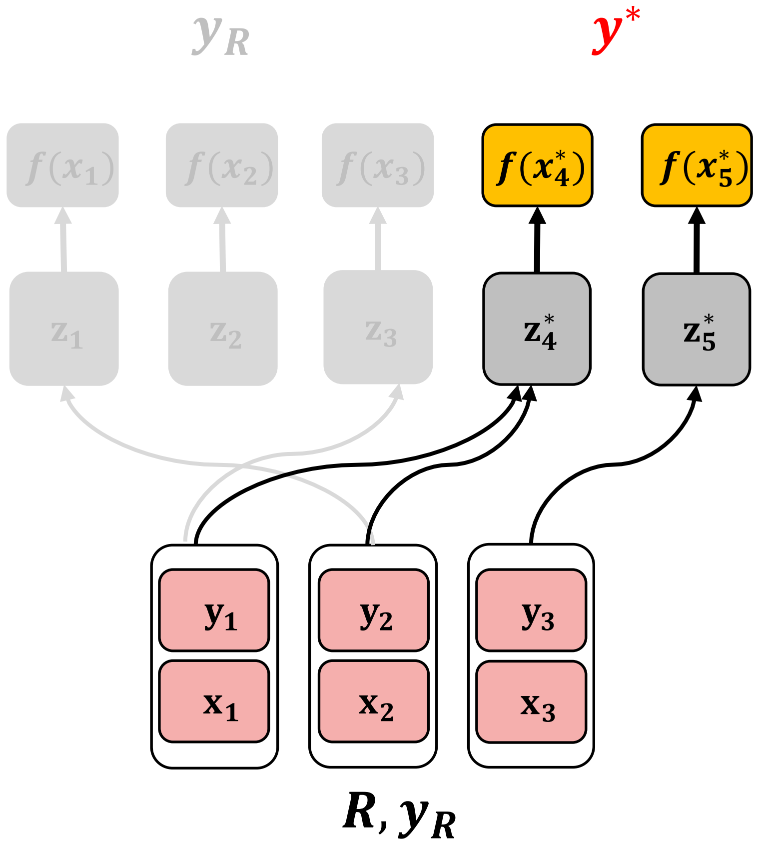




Predictive distribution

- $p_\theta(y^*|y, X, x_*) = \sum_{A^*} \int p_\theta(u_R, u^*|R, x^*) p(A^*|u_R, u^*) p_\theta(z^*, y^*|R, y_R, A^*)\, du_R du^* dz^*$





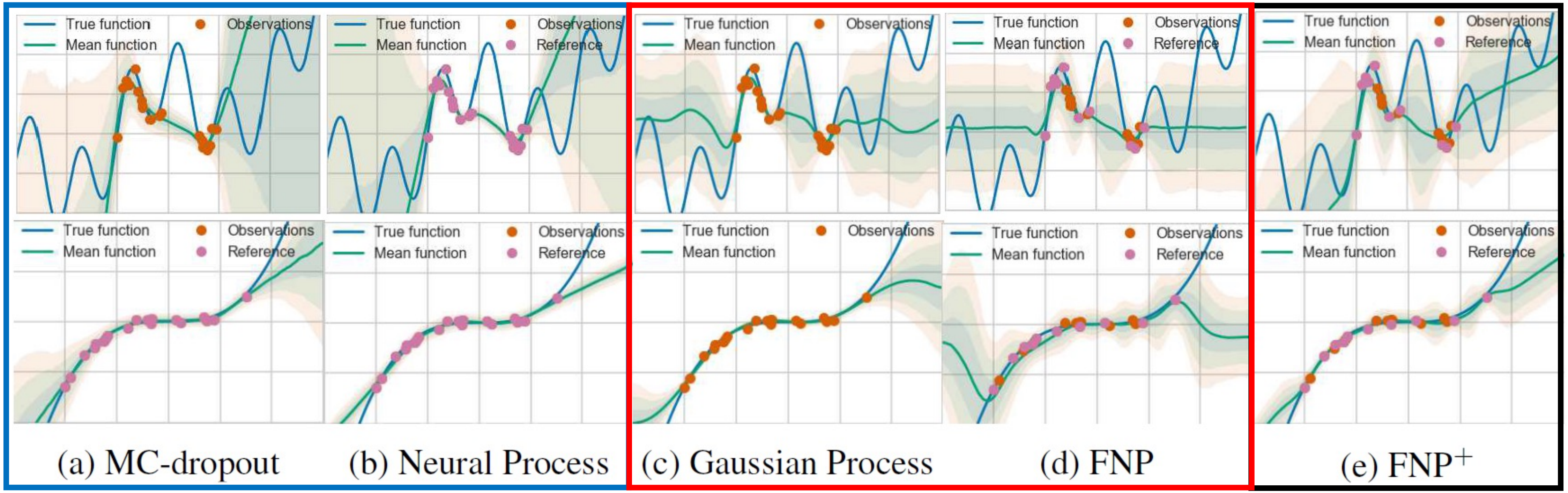
Experiment

1. Toy 1d regression

- Dataset

  1) $y_i = x_i + \epsilon + \sin(4(x_i + \epsilon)) + \sin(13(x_i + \epsilon))$ where $\epsilon \sim N(0, 0.03^2)$

  2) $y_i = x_i^3 + \epsilon$ where $\epsilon \sim N(0, 9)$



Overconfident inductive bias

Similar behavior

In-between

Experiment

2. Image classification

- Dataset
  1) In-distribution (MNIST, CIFAR10) : average predictive entropy(↓) / test error(↓)
  2) Out-of-distribution : average predictive entropy(↑) / area under the ROC curve(↑)

	NN	MC-Dropout	VI BNN	NP	FNP	FNP$^+$
MNIST	0.01 / 0.6	**0.05 / 0.5**	0.02 / 0.6	0.01 / 0.6	0.04 / 0.7	0.02 / 0.7
nMNIST	1.03 / 99.73	1.30 / 99.48	1.33 / 99.80	1.31 / 99.90	1.94 / 99.90	**1.77 / 99.96**
fMNIST	0.81 / 99.16	1.23 / 99.07	0.92 / 98.61	0.71 / 98.98	**1.85 / 99.66**	1.55 / 99.58
Omniglot	0.71 / 99.44	1.18 / 99.29	1.61 / 99.91	0.86 / 99.69	1.87 / 99.79	**1.71 / 99.92**
Gaussian	0.99 / 99.63	**2.03 / 100.0**	**1.77 / 100.0**	1.58 / 99.94	1.94 / 99.86	**2.03 / 100.0**
Uniform	0.85 / 99.65	0.65 / 97.58	1.41 / 99.87	1.46 / 99.96	2.11 / 99.98	**1.88 / 99.99**
Average	0.9±0.1 / 99.5±0.1	1.3±0.2 / 99.1±0.4	1.4±0.1 / 99.6±0.3	1.2±0.2 / 99.7±0.2	1.9±0.1 / 99.8±0.1	**1.8±0.1 / 99.9±0.1**
CIFAR10	0.05 / 6.9	0.06 / 7.0	**0.06 / 6.4**	0.06 / 7.5	0.18 / 7.2	0.08 / 7.2
SVHN	0.44 / 93.1	0.42 / 91.3	0.45 / 91.8	0.38 / 90.2	**1.09 / 94.3**	0.42 / 89.8
tImag32	0.51 / 92.7	0.59 / 93.1	0.52 / 91.9	0.45 / 89.8	**1.20 / 94.0**	0.74 / 93.8
iSUN	0.52 / 93.2	0.59 / 93.1	0.57 / 93.2	0.47 / 90.8	**1.30 / 95.1**	0.81 / 94.8
Gaussian	0.01 / 72.3	0.05 / 72.1	0.76 / 96.9	0.37 / 91.9	1.13 / 95.4	**0.96 / 97.9**
Uniform	**0.93 / 98.4**	0.08 / 77.3	0.65 / 96.1	0.17 / 87.8	0.71 / 89.7	**0.99 / 98.4**
Average	0.5±0.2 / 89.9±4.5	0.4±0.1 / 85.4±4.5	0.6±0.1 / 94±1.1	0.4±0.1 / 90.1±0.7	1.1±0.1 / 93.7±1.0	**0.8±0.1 / 94.9±1.6**

Experiment

3. NP vs FNP

- FNP still provides robust uncertainty and o.o.d detection is improved
- NP's performance is hurt as o.o.d detection is decreased

	NP	FNP$^+$
MNIST	0.01 / 0.6	0.02 / 0.7
nMNIST	1.31 / 99.90	**1.77 / 99.96**
fMNIST	0.71 / 98.98	1.55 / 99.58
Omniglot	0.86 / 99.69	**1.71 / 99.92**
Gaussian	1.58 / 99.94	**2.03 / 100.0**
Uniform	1.46 / 99.96	**1.88 / 99.99**
CIFAR10	0.06 / 7.5	0.08 / 7.2
SVHN	0.38 / 90.2	0.42 / 89.8
tImag32	0.45 / 89.8	0.74 / 93.8
iSUN	0.47 / 90.8	0.81 / 94.8
Gaussian	0.37 / 91.9	**0.96 / 97.9**
Uniform	0.17 / 87.8	**0.99 / 98.4**

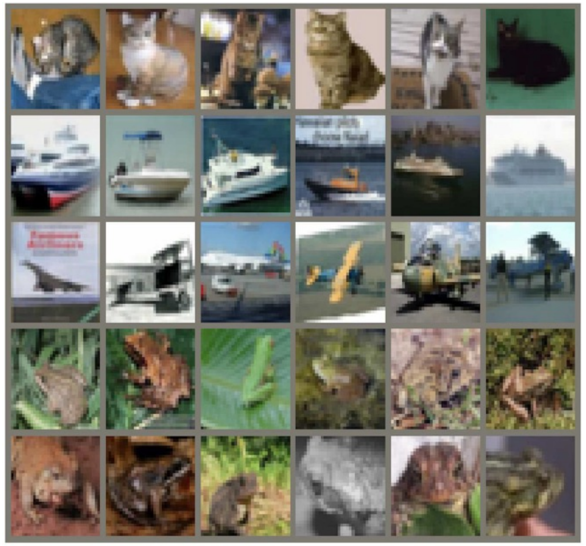
→

	NP fixed R	FNP$^+$ random R
MNIST	0.01 / 0.6	0.02 / 0.8 ↑
nMNIST	1.09 / 99.78 ↓	2.20 / 100.0 ↑
fMNIST	0.64 / 98.34 ↓	1.58 / 99.78 ↑
Omniglot	0.79 / 99.53 ↓	2.06 / 99.99 ↑
Gaussian	1.79 / 99.96 ↑	2.28 / 100.0 ↓
Uniform	1.42 / 99.93 ↓	2.23 / 100.0 ↑
CIFAR10	0.07 / 7.5	0.09 / 6.9 ↓
SVHN	0.46 / 91.5 ↑	0.56 / 91.4 ↑
tImag32	0.55 / 91.5 ↑	0.77 / 93.4 ↓
iSUN	0.60 / 92.6 ↑	0.83 / 94.0 ↓
Gaussian	0.20 / 87.2 ↓	1.23 / 99.1 ↑
Uniform	0.53 / 94.3 ↑	0.90 / 97.2 ↓

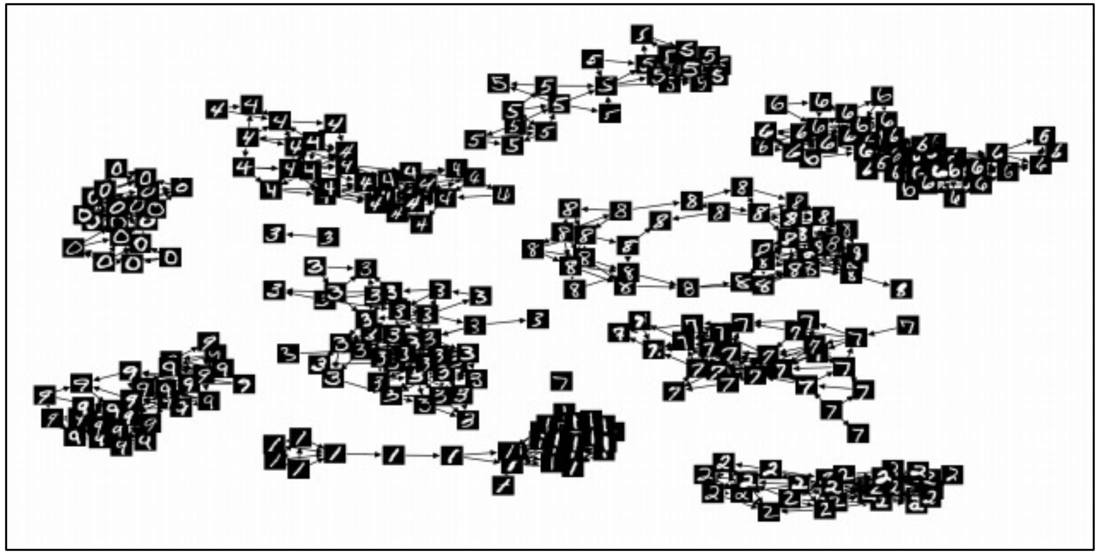
Experiment

4. Graph justification

- Semantic structure is captured both in A and G



$p(A|u_R, u_M)$



$p(G|u_R)$

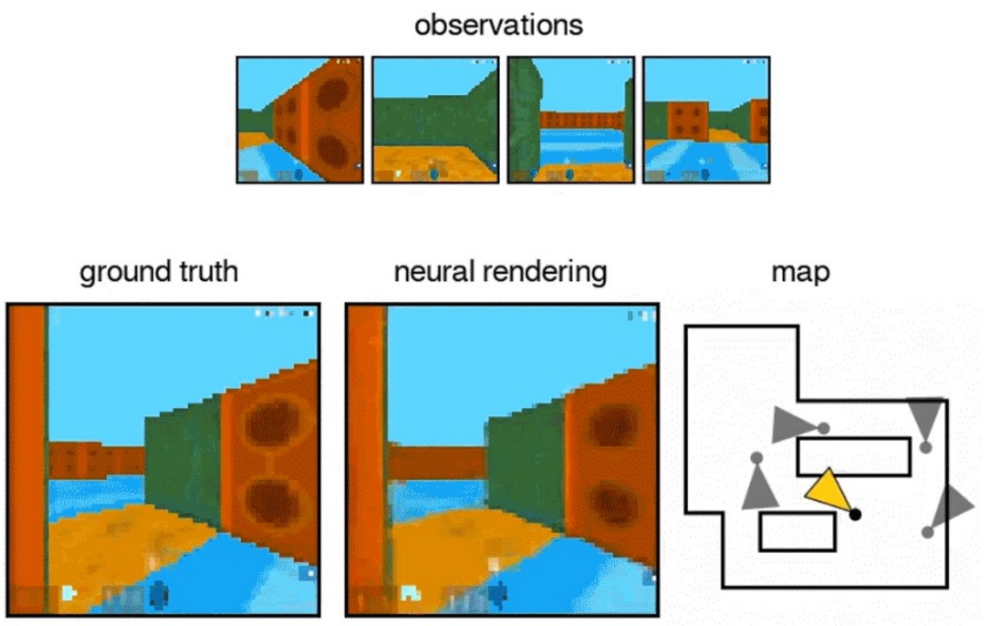
Conclusion

- Main idea
  - Build a graph of dependencies among local latent variables

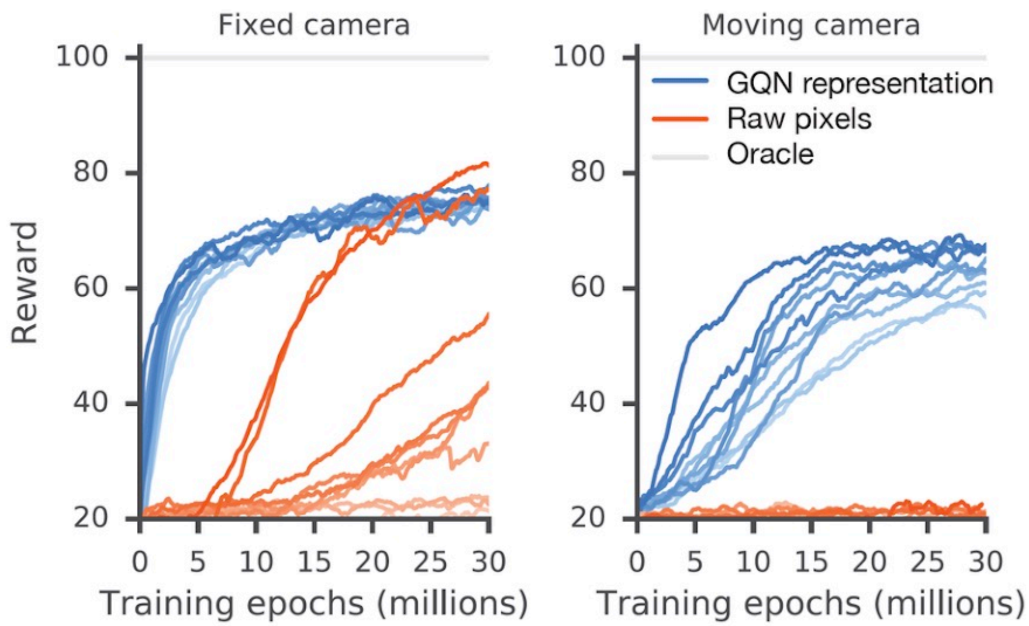
- Strength
  - No information loss due to global latent variable
  - Behave similar to Gaussian Process with RBF kernel
  - Satisfy exchangeability and consistency as a stochastic process

Weakness

- Not easily transferable comparing to the conventional neural processes
  - Boost the learning if nice representation can be learned
  - Helps exploration strategy in reinforcement learning task



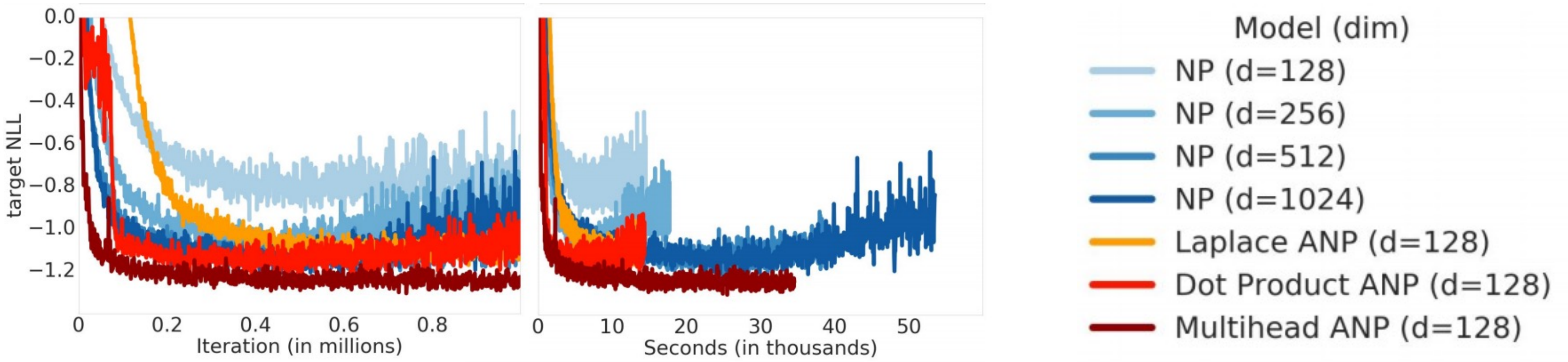
Maze exploration



Application on DQN

Weakness

- Not scalable when large reference set is required
  - Time complexity : $O(n+m) \rightarrow O(nm)$
    - What about ANP? $O(n(n+m))$



Weakness

- No reasonable or theoretical rule for choosing reference set
  - Greedy selection, Variational learning of pseudo input

  Ex. Bayesian GP-LVM, Deep gaussian process, Set-transformer

Thank you for the attention