Neural Process Family

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

Contents

- Conditional Neural Processes (CNP)
- Neural Processes (NP)
- Attentive Neural Processes (ANP)

Stochastic Process

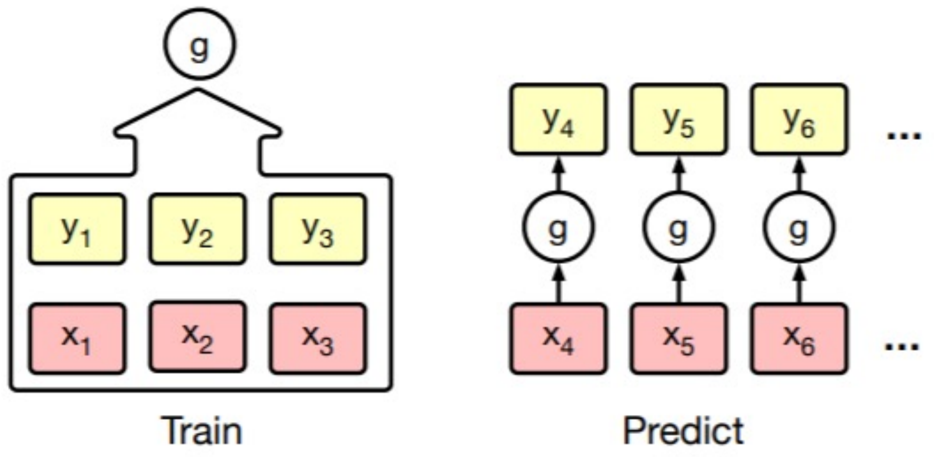
- Generative Query Networks (GQN)
- Consistent Generative Query Networks (JUMP)

Visual Scene representation
Video prediction

- Sequential Neural Process (SNP)

Temporal dynamics

Family of Neural Processes



- Motivation
  - Naïve Neural Network - No adaptation to test data
  - Gaussian Process - Computationally expensive for computing predictive posterior

- Aim
  - Devise a neural network based stochastic process that enables fast adaptation (a distribution D of functions $f: x \rightarrow y$ such that f ~ D)

- Problem scenario
  - divide the dataset into context set $\{C_x, C_y\}$ and target set $\{T_x, T_y\}$
  - learn a conditional distribution $\mathrm{p}\left(f(T_x) \middle| T_x, C_x, C_y\right)$

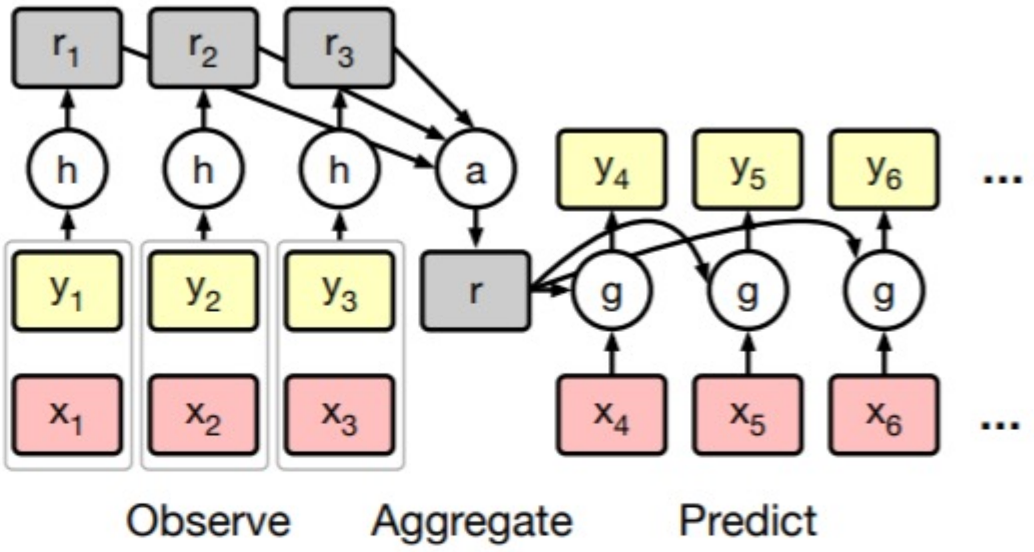
- How to enable adaptation
  - Context (C_x, C_y) is fed to encoder and extract the global representation r by aggregator
  - Target input (T_x) is fed to decoder with r for locally predicting Target output (T_y)

- Where does the stochasticity of function f come from?
  - Context and Target split → **Conditional Neural Process**

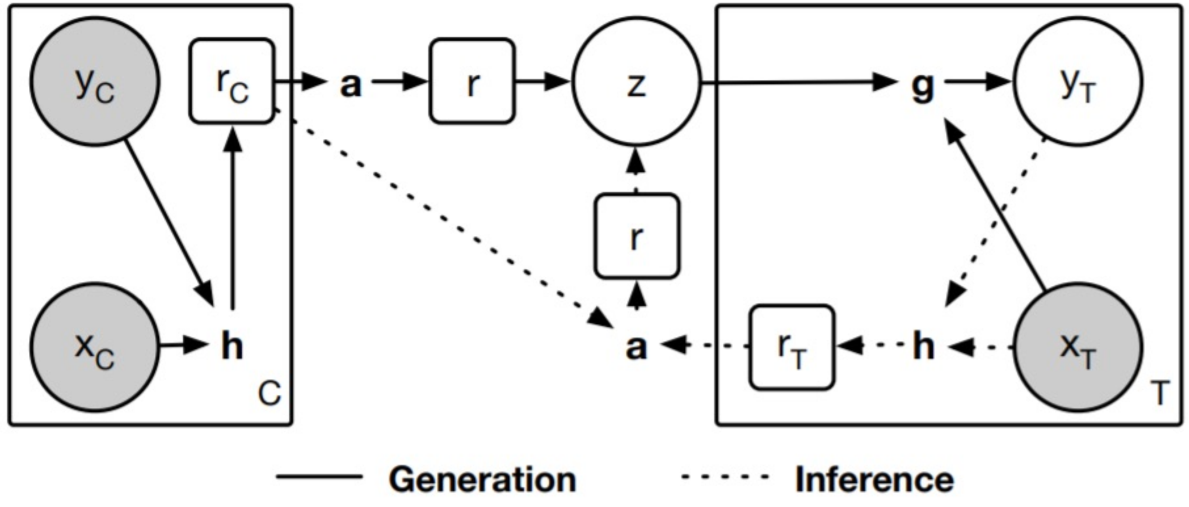
  (+ Introducing a stochastic latent variable z as in VAE → **Neural Processes**)

- Improving Expressiveness
  - Utilizing attention mechanism → **Attentive Neural Process**

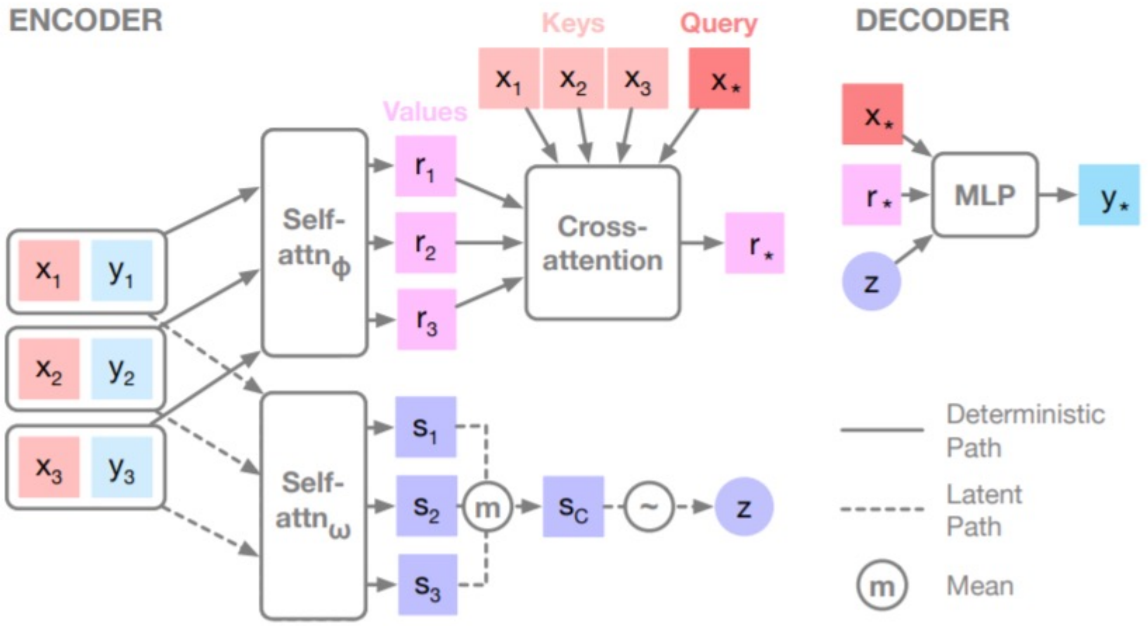
CONDITIONAL NEURAL PROCESS



NEURAL PROCESS



ATTENTIVE NEURAL PROCESS

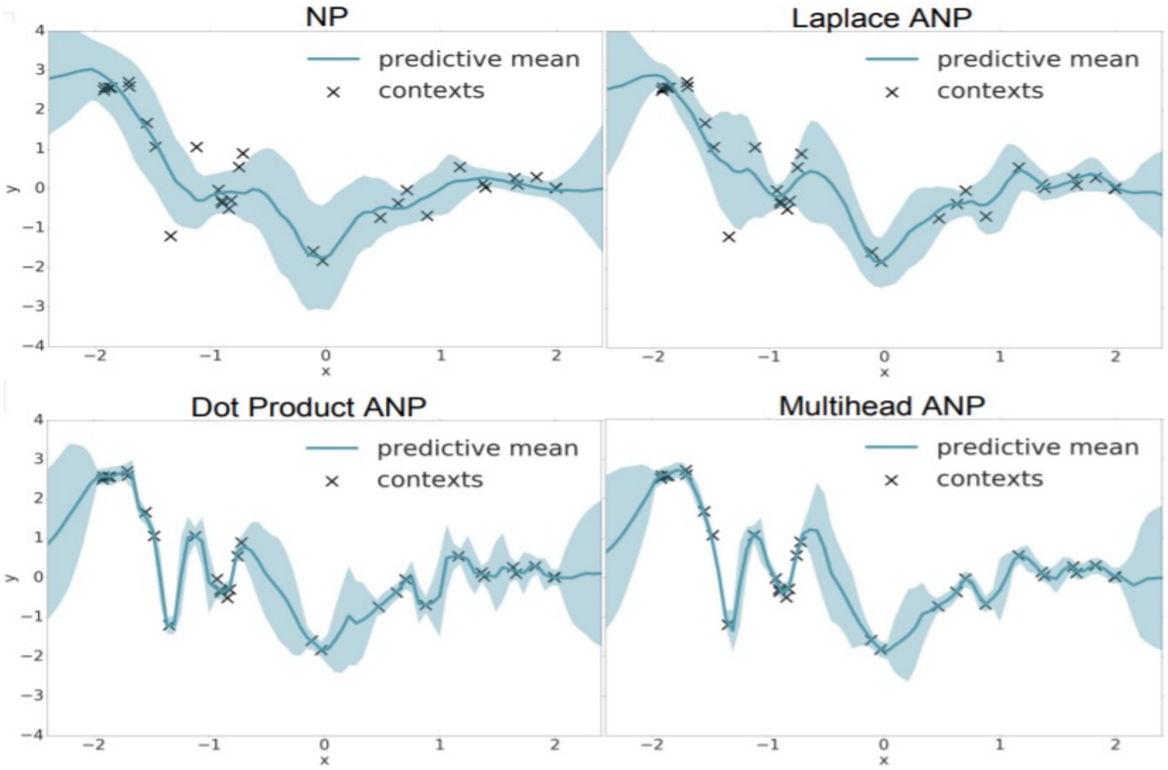


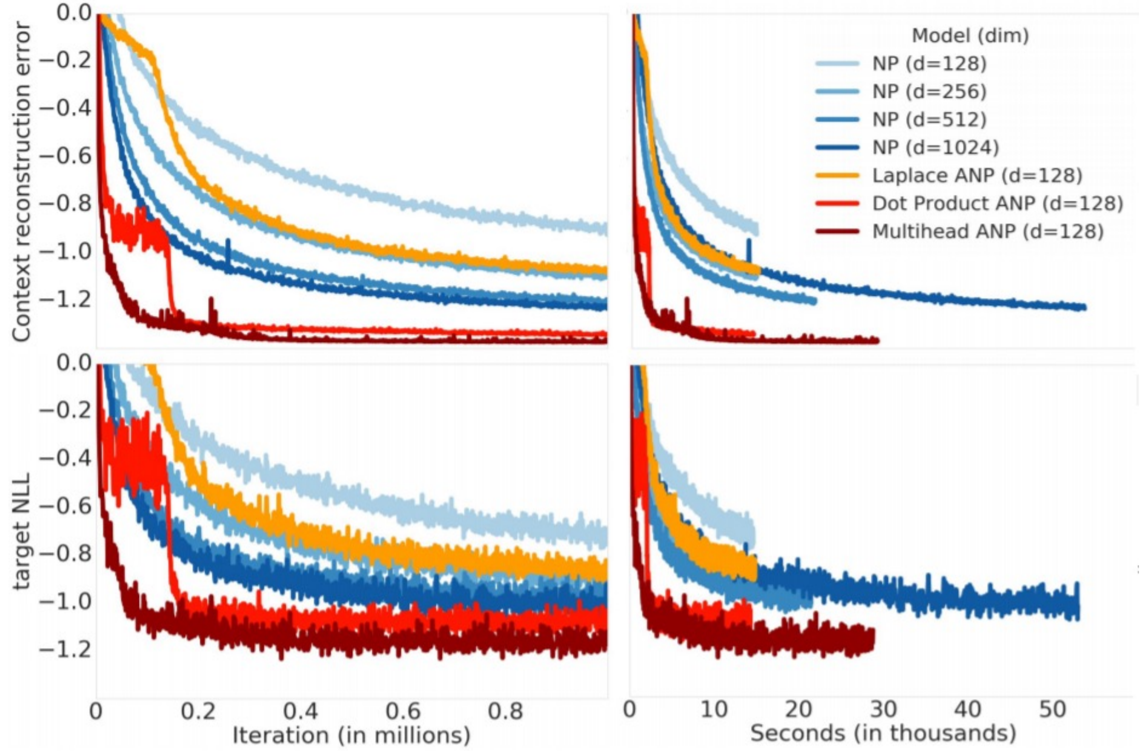
- Attention mechanism (Q: $query$, K: $keys$, V: $values$)
  - Among contexts inputs (C_x) (or among target inputs (T_x))
    - Self attention(V) : $softmax(W_2 \tanh(W_1 V))V$
  - Between context inputs (C_x) and target inputs (T_x)
    - $$\textbf{Laplace}(Q,K,V) := WV \in \mathbb{R}^{m\times d_v}, \qquad W_{i\cdot} := \text{softmax}((-\|Q_{i\cdot} - K_{j\cdot}\|_1)_{j=1}^n) \in \mathbb{R}^n$$
      $$\textbf{DotProduct}(Q,K,V) := \text{softmax}(QK^\top/\sqrt{d_k})V \in \mathbb{R}^{m\times d_v}$$
      $$\textbf{MultiHead}(Q,K,V) := \text{concat}(\text{head}_1,\ldots,\text{head}_H)W \in \mathbb{R}^{m\times d_v}$$
      $$\text{where head}_h := \text{DotProduct}(QW_h^Q, KW_h^K, VW_h^V) \in \mathbb{R}^{m\times d_v}$$

- How does it approximate Gaussian Process?
  - $$p\big(f(T_x)\big|T_x, C_x, C_y\big) = \mathcal{N}(k(T_x,C_x)(k(C_x,C_x)+\sigma^2 I)^{-1}C_y, \\ k(T_x,T_x) - k(T_x,C_x)(k(C_x,C_x)+\sigma^2 I)^{-1}k(C_x,T_x))$$

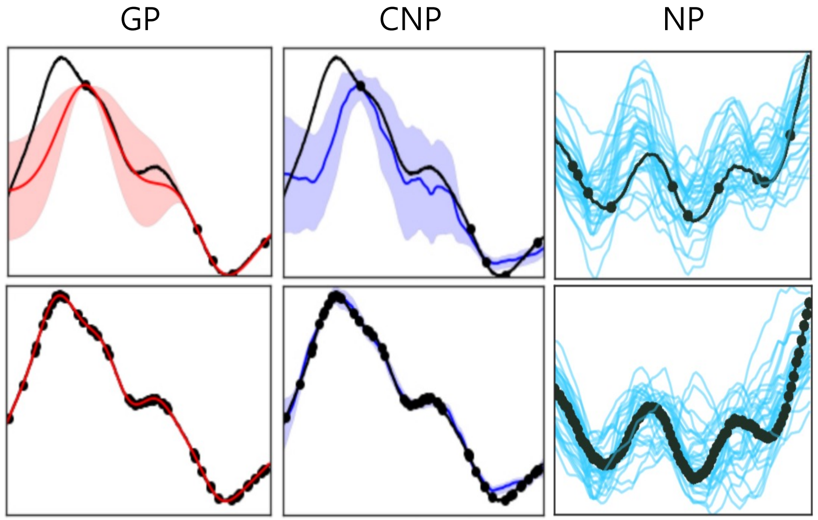
- Loss function
  - CNP : $-\log p(T_y | T_x, r)$
  - (A)NP : $-\mathbb{E}_{\mathrm{z \sim q(z|C_x, C_y, T_x, T_y)}}[\log \mathrm{p(T_y|T_x(, r), z)}] + \mathrm{KL}[\mathrm{q(z|C_x, C_y, T_x, T_y)} \| q(z|C_x, C_y)]$

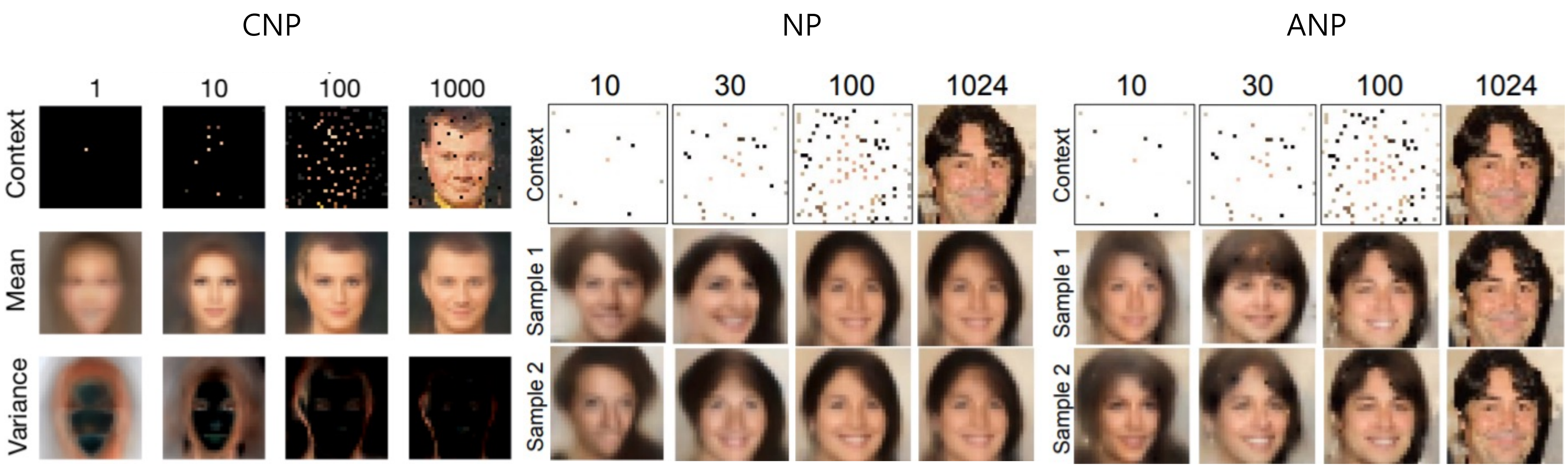
- Strength
  - **Scalability** : computation scales linearly $O(n+m)$ in CNP and NP
    (GP : $O((n+m)^3)$, ANP : $O(n(n+m))$)
  - **Flexibility** : A wide variety of family of distribution can be defined
    (arbitrary number of contexts and targets is okay)
  - **Permutation invariance** : target predictions are order invariant in the contexts
    (mean function or cross-attention)



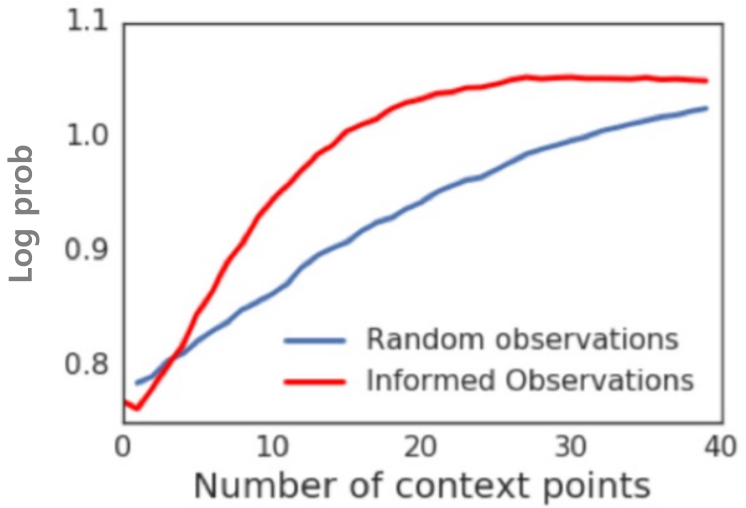


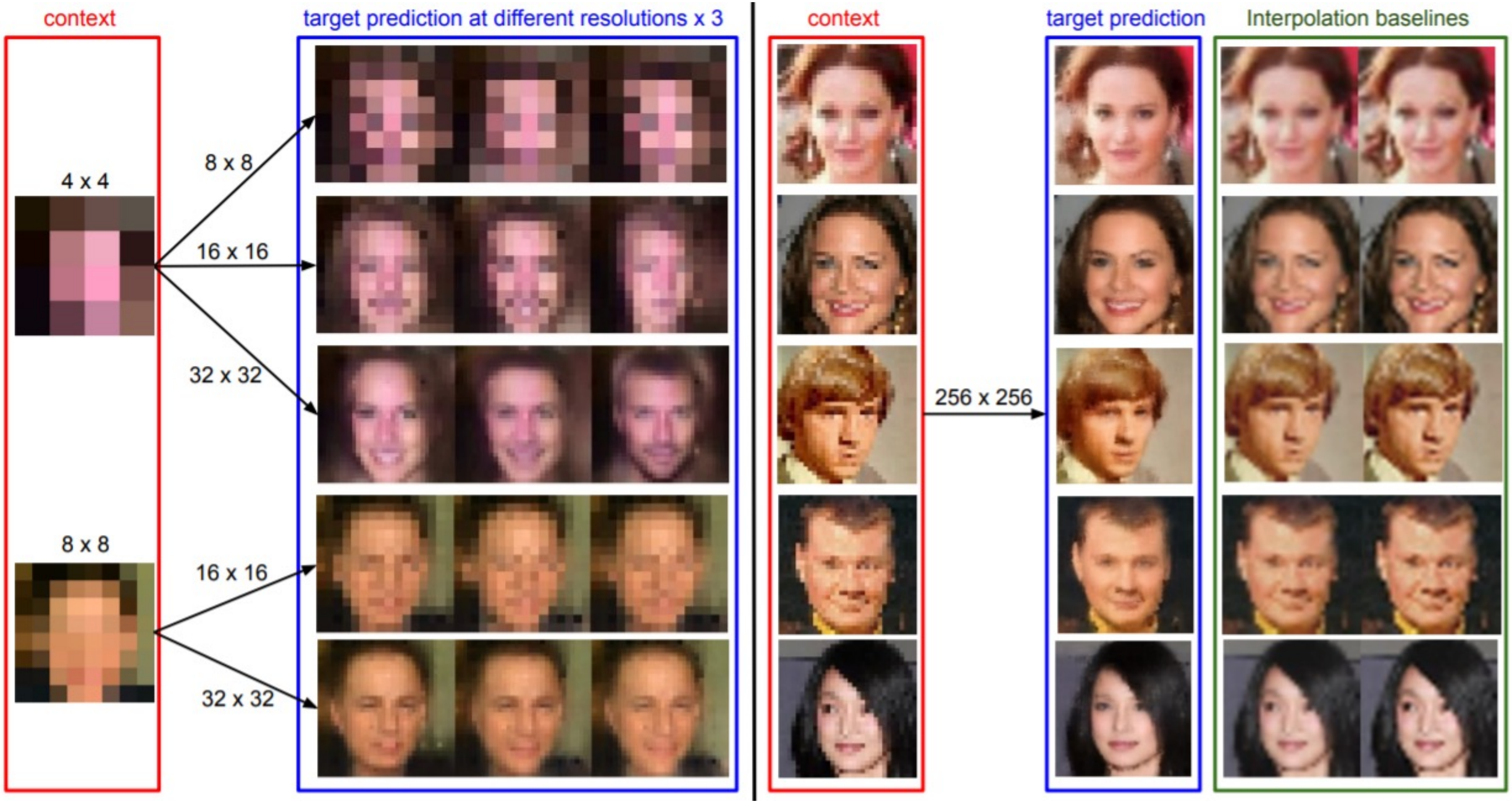
1. Variance decreases as the number of contexts increases.
2. Variance of CNP is sharper than that of GP.
3. CNP is deterministic if contexts and targets are fixed.
4. (A)NP can sample multiple functions by sampling different z.
5. NP is highly variable comparing to GP and CNP.
6. NP is underfitted even at predictions on context points.
7. ANP resolved the underfitting issue on NP.
8. ANP show nice convergence in terms of iterations and time.





1. Variance decreases as the number of contexts increases.
2. Variance of CNP is sharper than that of GP.
3. CNP is deterministic if contexts and targets are fixed.
4. (A)NP can sample multiple functions by sampling different z.
5. NP is highly variable comparing to GP and CNP.
6. NP is underfitted even at predictions on context points.
7. ANP resolved the underfitting issue on NP.
8. Choice of contexts by variance of target output boost training.

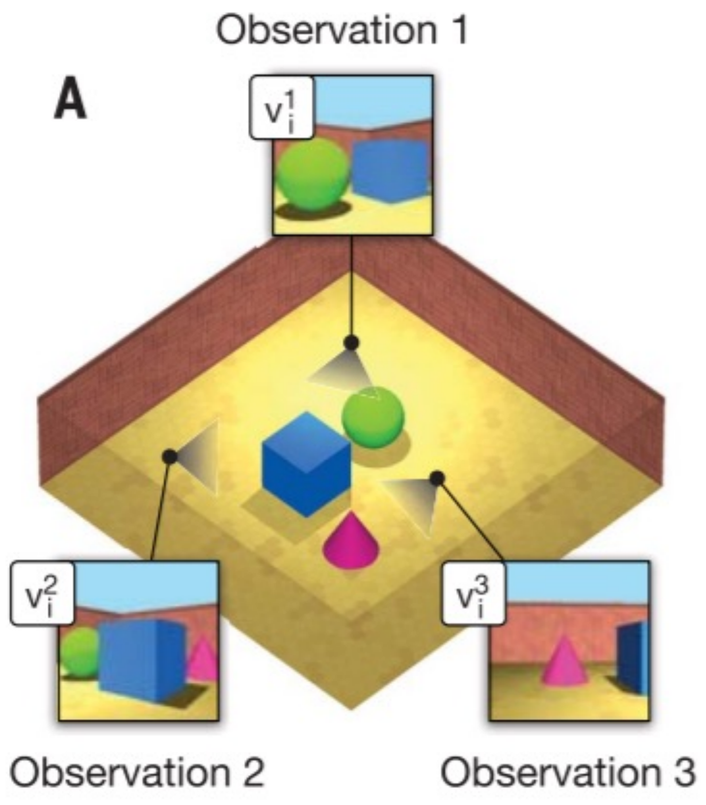




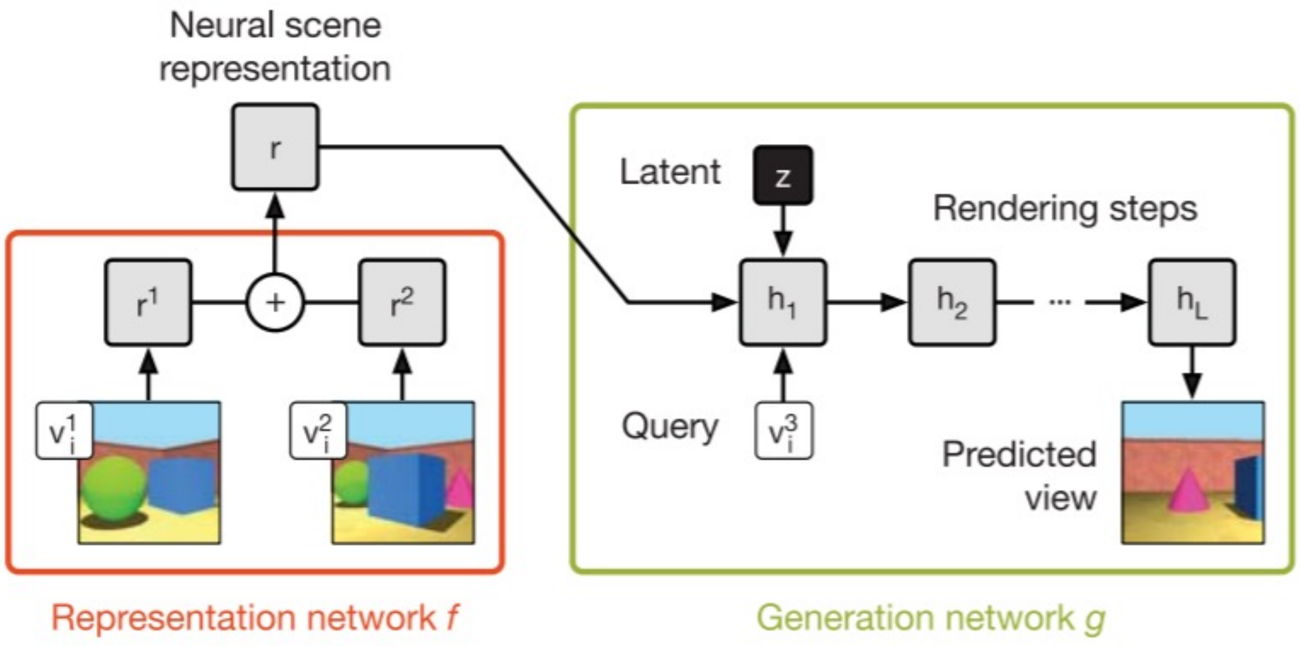
1. High resolution task cannot be done in generative models.
2. Show better performance than cubic interpolation baseline.

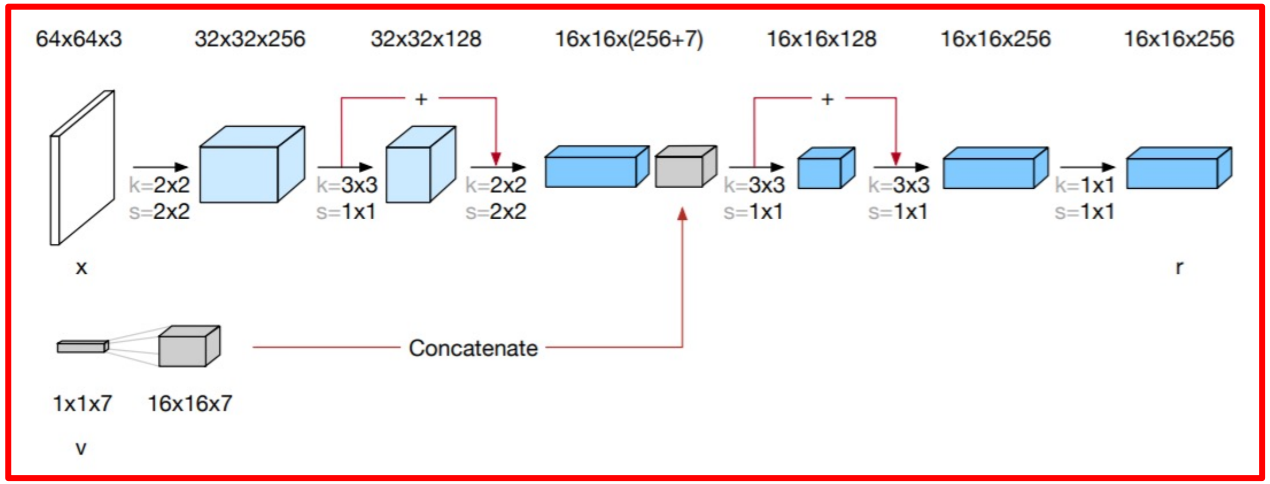
Generative Query Networks

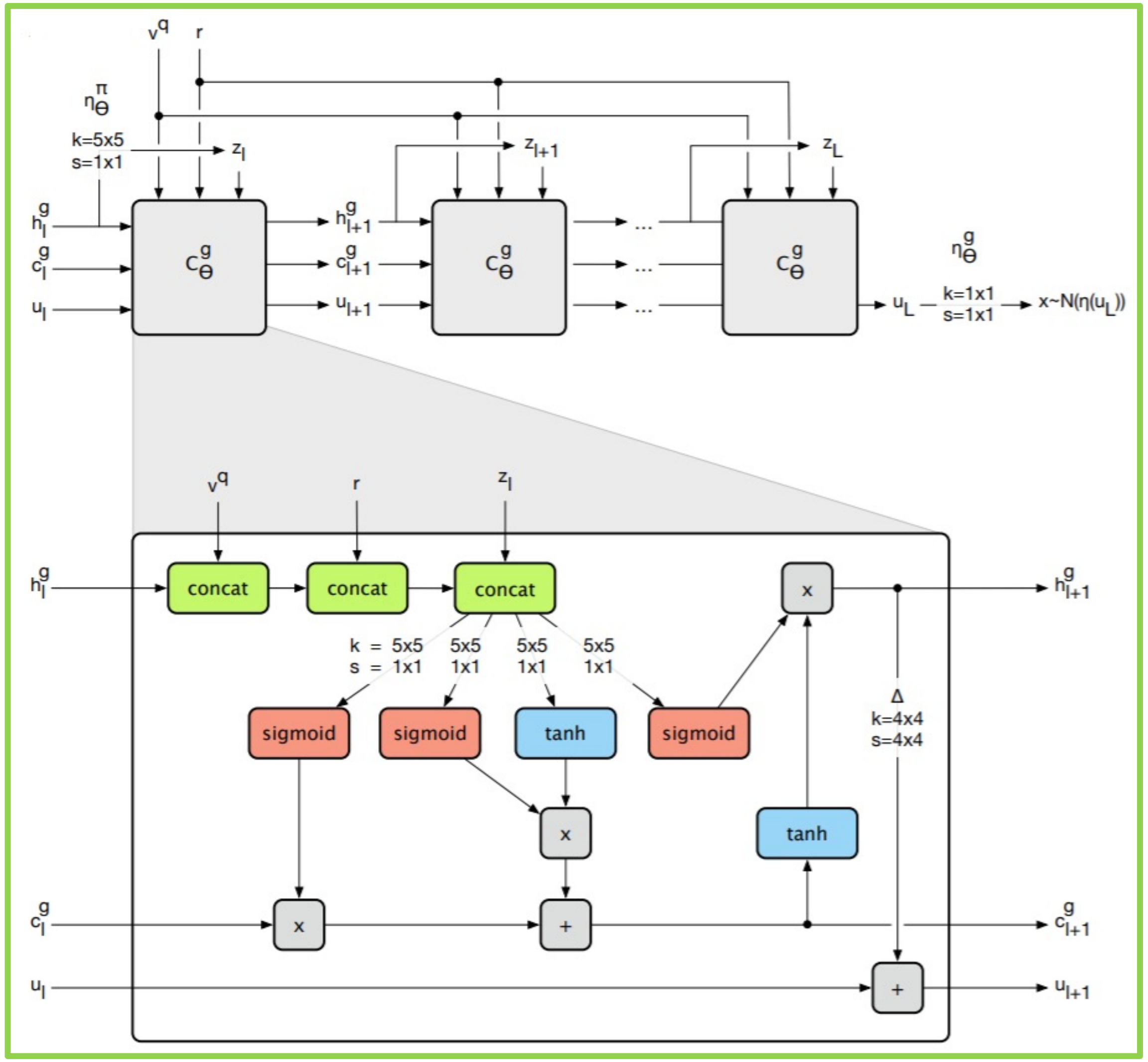
- Motivation
  - Scene representation for an intelligent agent requires the labeling by human
- Aim
  - Representation learning without human labels or domain knowledge which enables the agent to autonomously learn to understand the world
- Problem scenario
  - An agent navigates a 3D scene i and collects K images x_i^k from 2D viewpoints v_i^k
  - The network predicts the scene from an arbitrary query viewpoint v_i^q



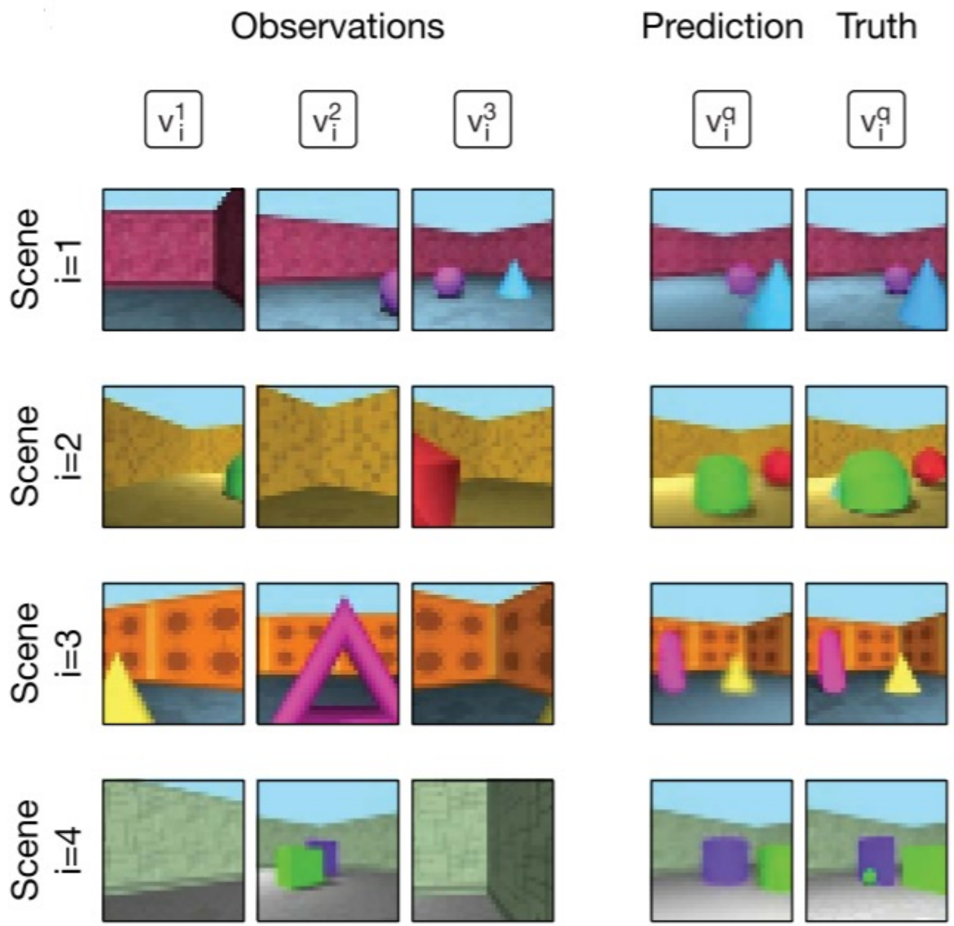
Architecture



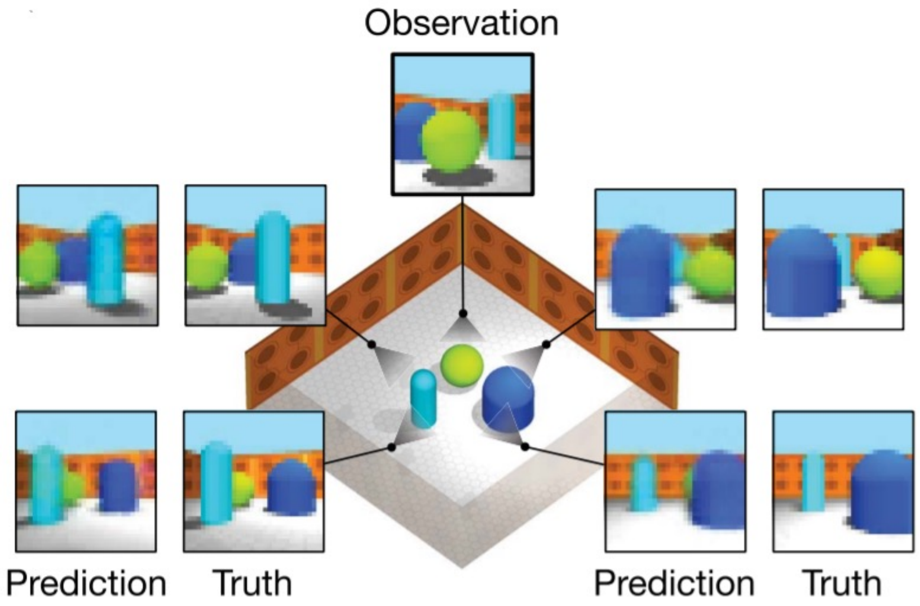




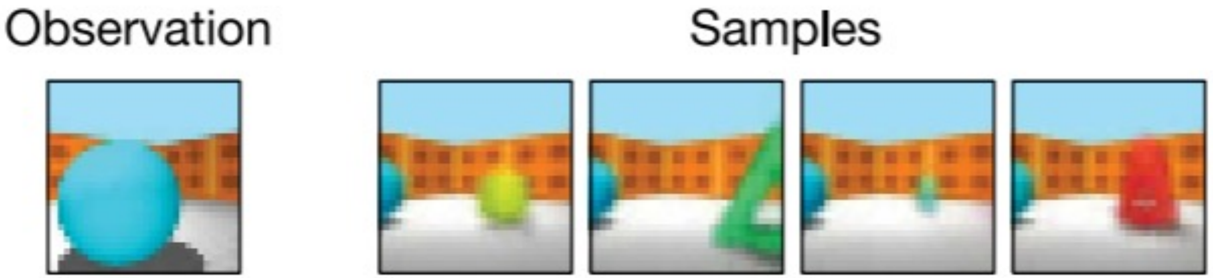
Prediction



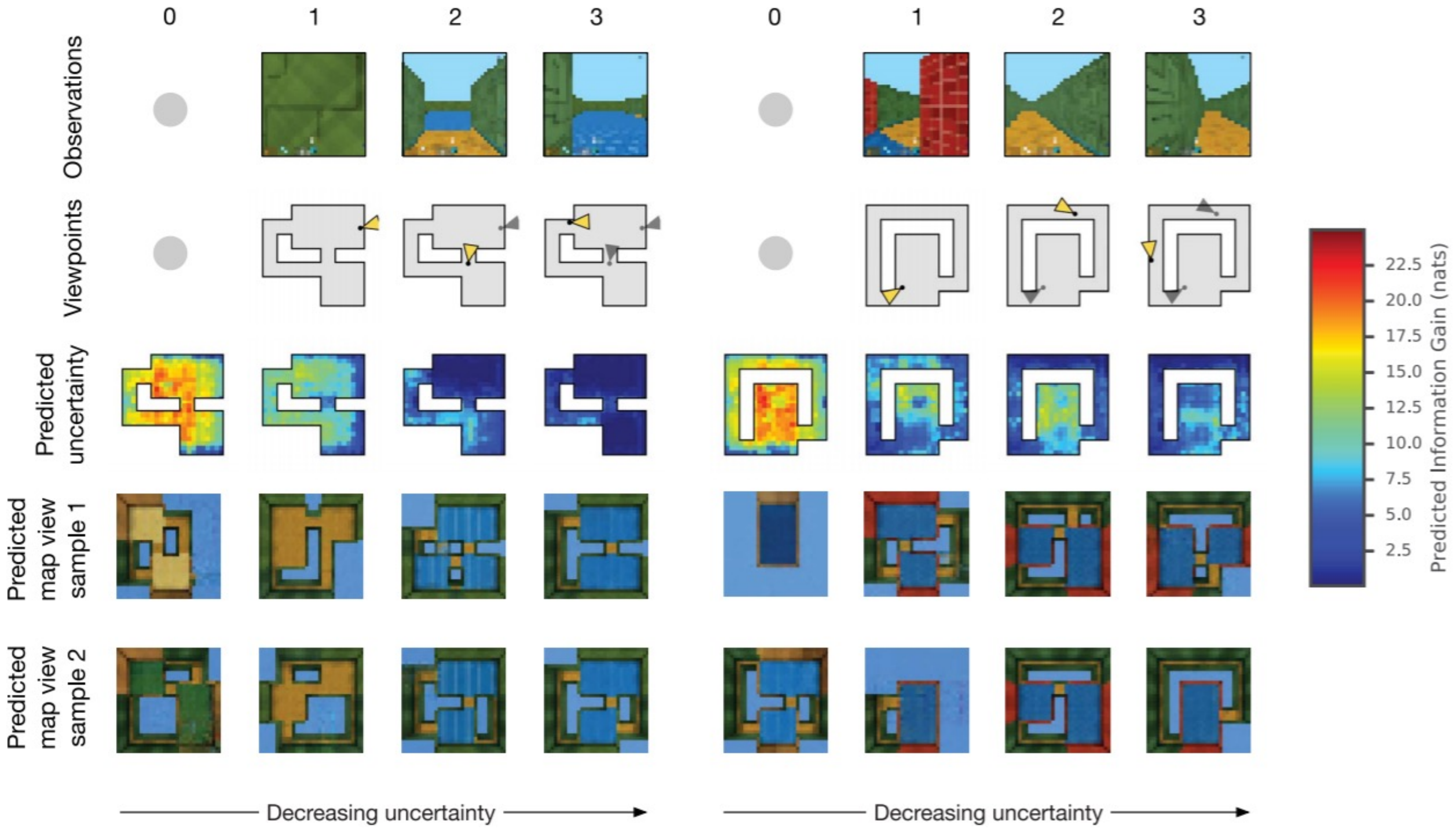
Observations
Prediction
Truth
v i 1
v i 2
v i 3
v i q
v i q
Scene i=1
Scene i=2
Scene i=3
Scene i=4



Observation
Prediction
Truth
Prediction
Truth

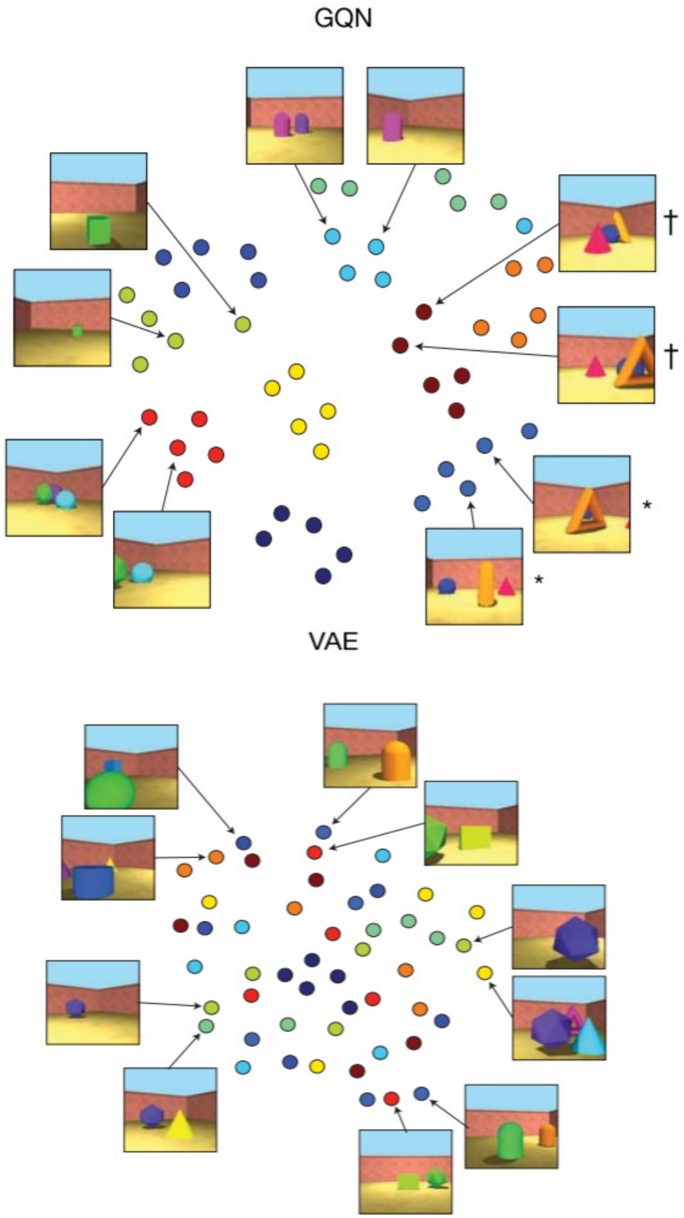


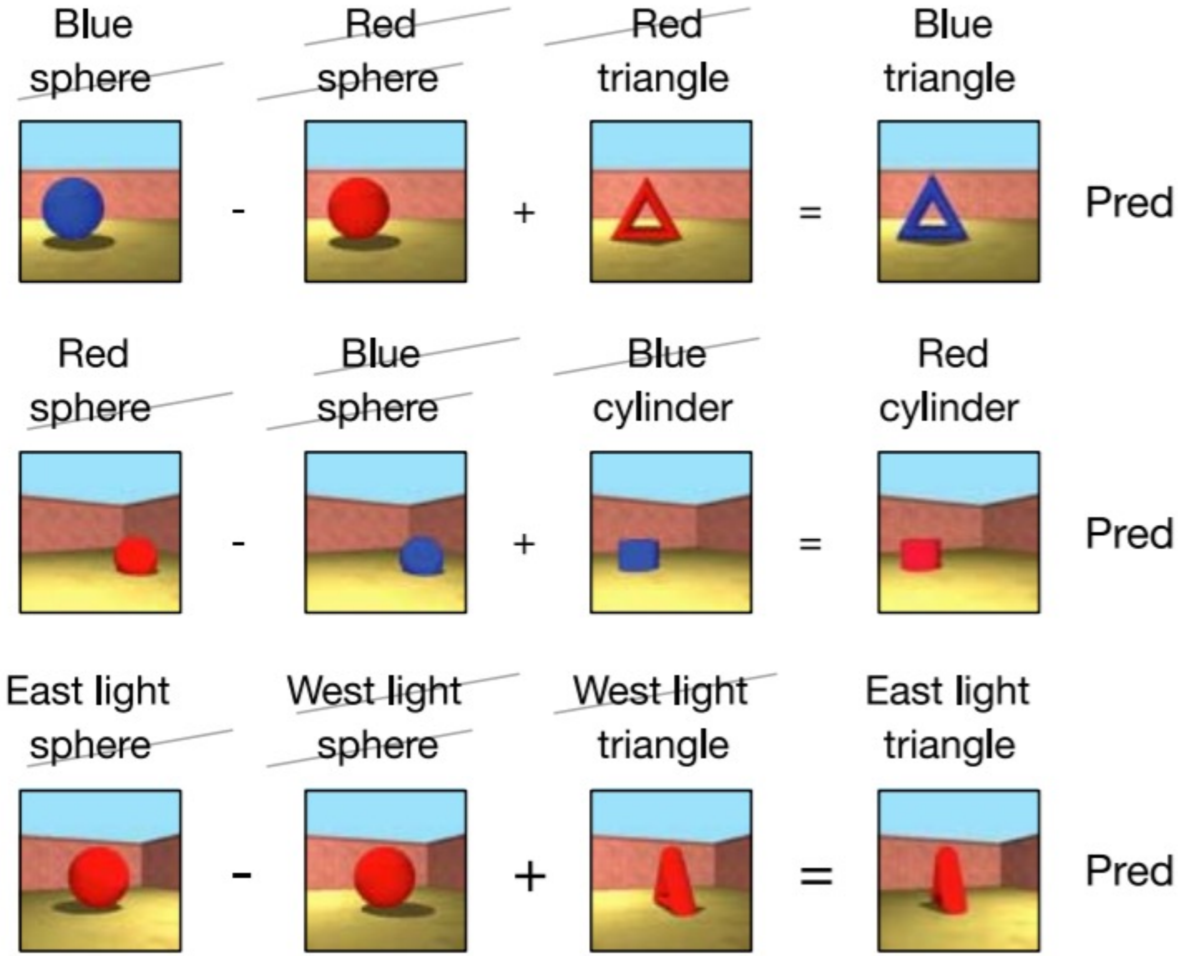
Observation
Samples



0
1
2
3
0
1
2
3
Observations
Viewpoints
Predicted uncertainty
Predicted map view sample 1
Predicted map view sample 2
Decreasing uncertainty
Decreasing uncertainty
Predicted Information Gain (nats)
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5

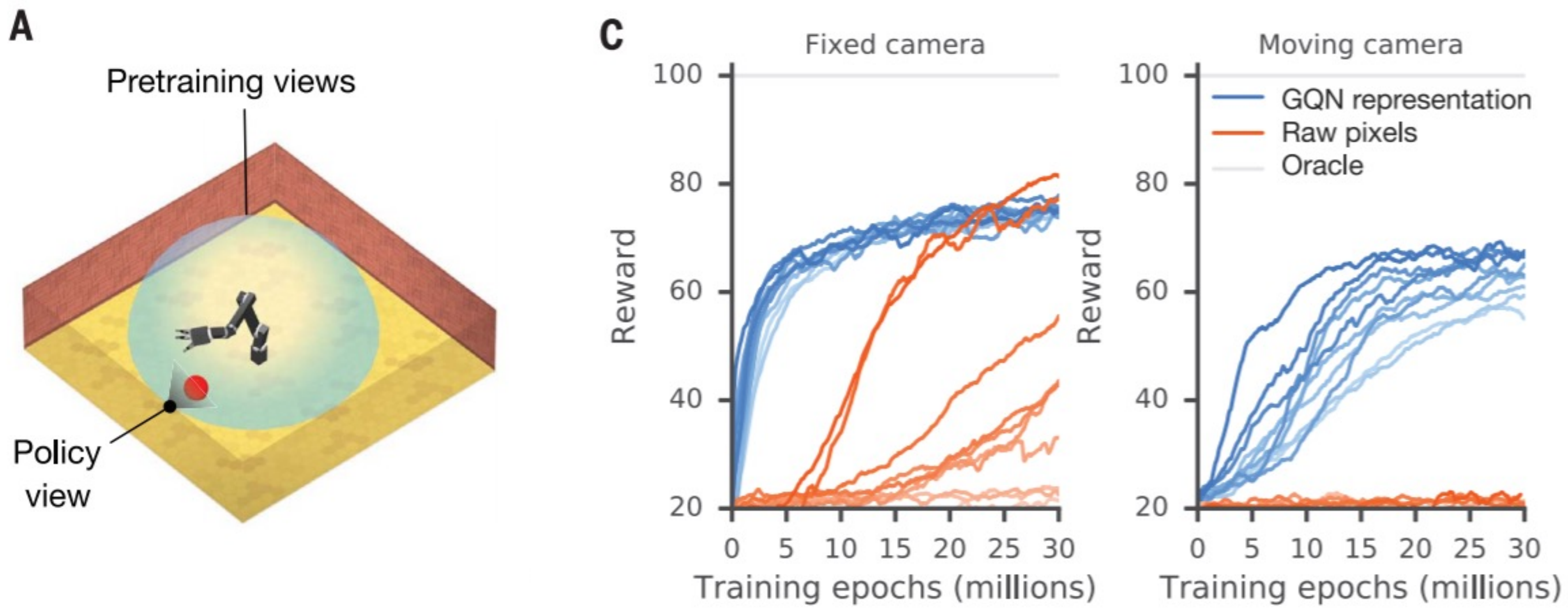
Representation





Algebra works in scene representation

Data-efficient control



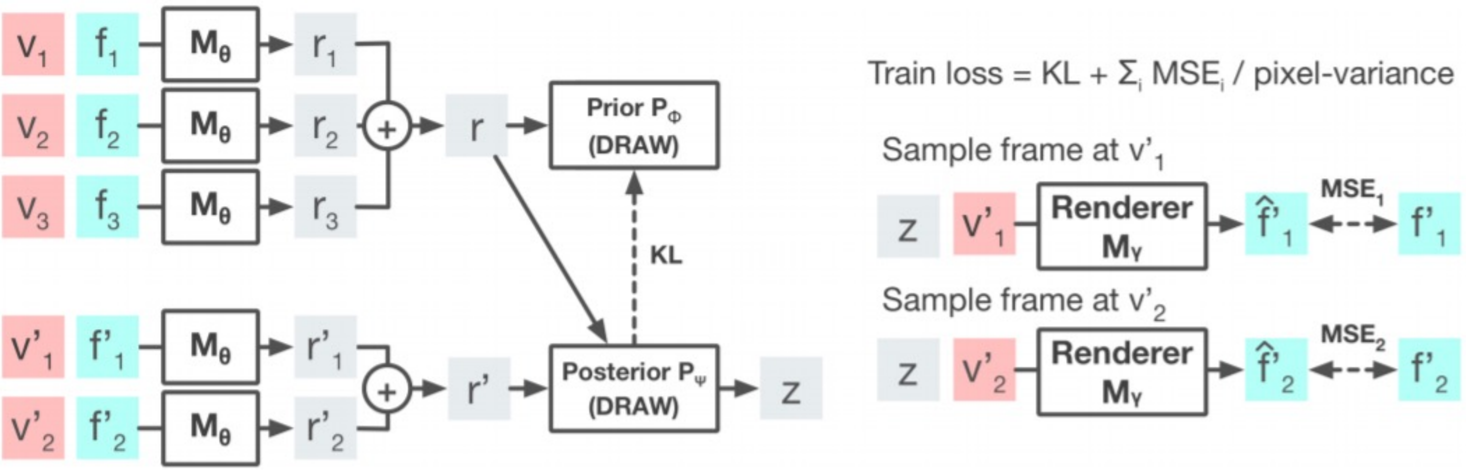
1. Goal is to learn to control a robotic arm to reach a randomly positioned colored object.
2. GQN is pretrained on randomized configurations from randomized viewpoints in the dome.
3. Controlling policy observes the scene from either fixed camera or moving camera.
4. Using GQN representation rather than raw pixel, required experiences decreases 4 times.

Generative	
Scene encoder	$\mathbf{r} = f\left(\mathbf{x}^{1,\ldots,M}, \mathbf{v}^{1,\ldots,M}\right)$
Initial state	$(\mathbf{c}_0^g, \mathbf{h}_0^g, \mathbf{u}_0) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$
Prior factor	$\pi_{\theta_l}\left(\cdot \mid \mathbf{v}^q, \mathbf{r}, \mathbf{z}_{<l}\right) = \mathcal{N}\left(\cdot \mid \eta_\theta^\pi\left(\mathbf{h}_l^g\right)\right)$
Prior sample	$\boxed{\mathbf{z}_l \sim \pi_{\theta_l}\left(\cdot \mid \mathbf{v}^q, \mathbf{r}, \mathbf{z}_{<l}\right)}$
State update	$\left(\mathbf{c}_{l+1}^g, \mathbf{h}_{l+1}^g, \mathbf{u}_{l+1}\right) = C_\theta^g\left(\mathbf{v}^q, \mathbf{r}, \mathbf{c}_l^g, \mathbf{h}_l^g, \mathbf{u}_l, \mathbf{z}_l\right)$
Observation sample	$\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}^q \mid \mu = \eta_\theta^g(\mathbf{u}_L), \sigma = \sigma_t\right)$

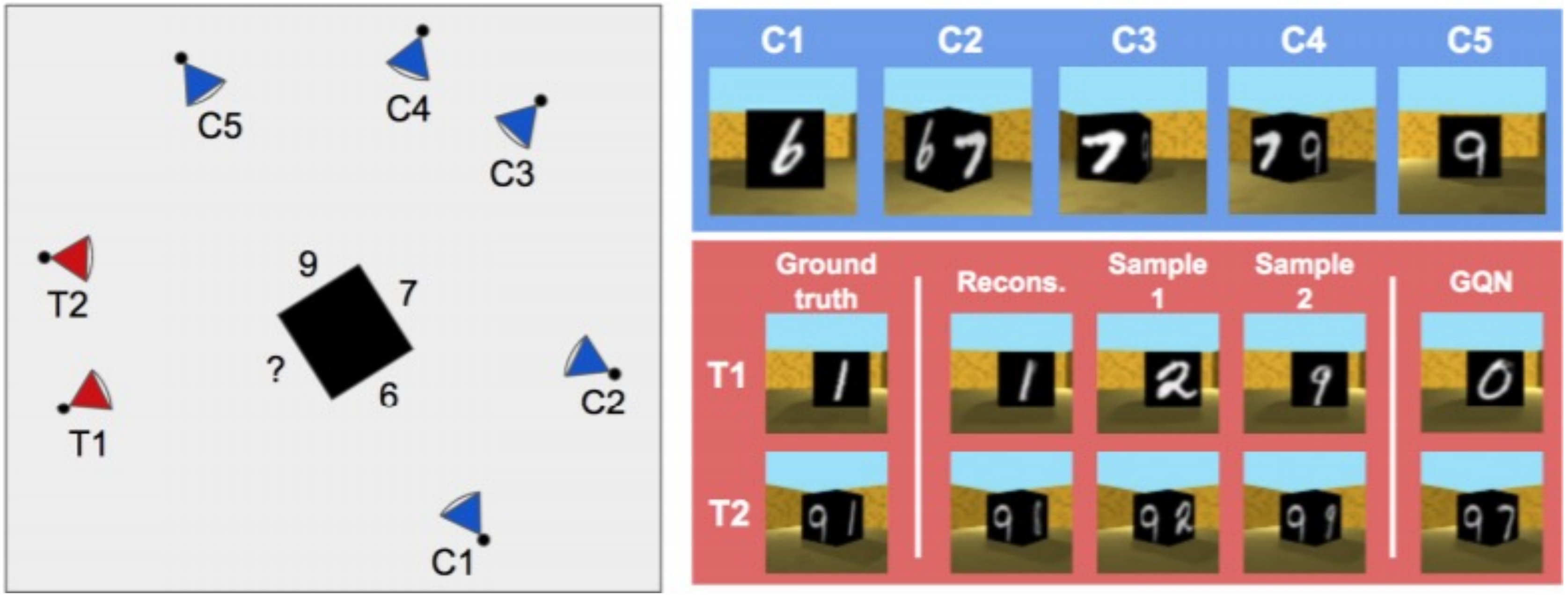
Inference	
Scene encoder	$\mathbf{r} = f\left(\mathbf{x}^{1,\ldots,M}, \mathbf{v}^{1,\ldots,M}\right)$
Generator initial state	$(\mathbf{c}_0^g, \mathbf{h}_0^g, \mathbf{u}_0) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$
Inference initial state	$(\mathbf{c}_0^e, \mathbf{h}_0^e) = (\mathbf{0}, \mathbf{0})$
Inference state update	$\left(\mathbf{c}_{l+1}^e, \mathbf{h}_{l+1}^e\right) = C_\phi^e\left(\mathbf{x}^q, \mathbf{v}^q, \mathbf{r}, \mathbf{c}_l^e, \mathbf{h}_l^e, \mathbf{h}_l^g, \mathbf{u}_l\right)$
Posterior factor	$q_{\phi_l}\left(\cdot \mid \mathbf{x}^q, \mathbf{v}^q, \mathbf{r}, \mathbf{z}_{<l}\right) = \mathcal{N}\left(\cdot \mid \eta_\phi^q\left(\mathbf{h}_l^e\right)\right)$
Posterior sample	$\boxed{\mathbf{z}_l \sim q_{\phi_l}\left(\cdot \mid \mathbf{x}^q, \mathbf{v}^q, \mathbf{r}, \mathbf{z}_{<l}\right)}$
Generator state update	$\left(\mathbf{c}_{l+1}^g, \mathbf{h}_{l+1}^g, \mathbf{u}_{l+1}\right) = C_\theta^g\left(\mathbf{v}^q, \mathbf{r}, \mathbf{c}_l^g, \mathbf{h}_l^g, \mathbf{u}_l, \mathbf{z}_l\right)$

1. Latent variable z is sampled from the distribution that is conditioned on target input v^q.
2. Different target inputs take different latent variable, which results in inconsistent target outputs.

Consistent Generative Query Networks



1. Just as Neural Processes, a sampled latent variable is shared among target inputs.
2. Pixel variance plays the same role as KL-annealing to handle posterior collapse.



Sequential Neural Process

- Motivation
  - Temporal dynamics may exist in a sequence of stochastic processes
- Aim
  - Generalization of neural process
  - Meta-transfer learning for a sequence of stochastic processes
- Problem Scenario
  - Consider a sequence of stochastic processes $P_1, P_2, \ldots, P_T$
  - P_t's are modeled in neural process framework by further considering the temporal change $P_{t-1} \rightarrow P_t$.

Model Comparison

- Neural Process
  - $p\left(D_y \middle| D_x, C_x, C_y\right) = \int p\left(D_y \middle| D_x, z\right) p\left(z \middle| C_x, C_y\right) dz$
- **Sequential Neural Process**
  - $p\left(D_y^t \middle| D_x^t, C_x^t, C_y^t\right) = \int p\left(D_y^t \middle| D_x^t, z^t\right) p(z^t | z^{<t}, C_x, C_y)\, dz^t$
- Generative Query Networks
  - $p\left(D_y \middle| D_x, C_x, C_y\right) = \int p\left(D_y \middle| D_x, z_L\right) \prod_{l=1}^{L} p\left(z_l \middle| z_{<l}, D_x, C_x, C_y\right))\, dz_{1:L}$
- Consistent Generative Query Networks
  - $p\left(D_y \middle| D_x, C_x, C_y\right) = \int p\left(D_y \middle| D_x, z_L\right) \prod_{l=1}^{L} p\left(z_l \middle| z_{<l}, C_x, C_y\right))\, dz_{1:L}$
- **Temporal Generative Query Networks**
  - $p\left(D_y^t \middle| D_x^t, C_x^t, C_y^t\right) = \int p\left(D_y^t \middle| D_x^t, z_L^t\right) \prod_{l=1}^{L} p(z_l^t | z_{<l}^t, z_l^{<t}, C_x, C_y)\, dz_{1:L}^t$

Loss function

- SNP ELBO
  - $L_1 = \sum_{t=1}^{T} \mathbb{E}_{z^t \sim q_\phi(z^t|z^{<t}, C^t, D^t)} \left[\log p_\theta \left(D_y^t \middle| D_x^t, z^t \right) \right] - KL \left(q_\phi(z^t|z^{<t}, C^t, D^t) \middle\| p_\theta(z^t|z^{<t}, C^t) \right)$

    $where \quad z^t = (z_1^t, z_2^t, \ldots, z_L^t)$

- Posterior Dropout ELBO
  - Transition collapse : Tendency to ignore the context information in the transition model
  - Restricting the latent information while maintaining reconstruction quality
  - $L_2 = \mathbb{E}_{\bar{T}} \left[\sum_{t \in \bar{T}} \mathbb{E}_{z^t} \left[\log p_\theta \left(D_y^t \middle| D_x^t, z^t \right) \right] - KL \left(q_\phi(z^t|z^{<t}, C^t, D^t) \middle\| p_\theta(z^t|z^{<t}, C^t) \right) \right]$

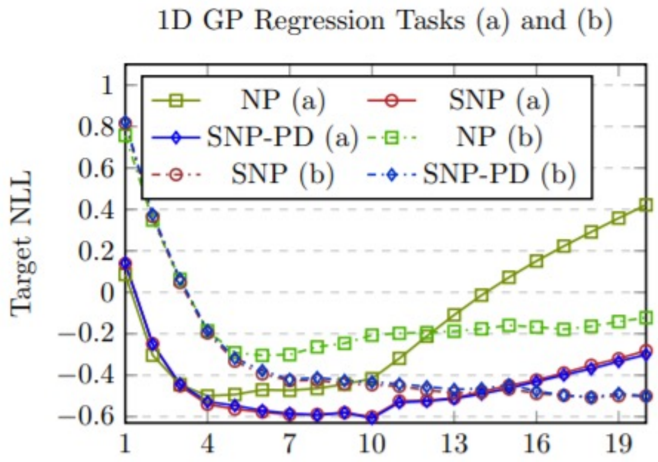
    $where \quad z^t \sim q_\phi(z^t|z^{<t}, C^t, D^t) \quad for \quad t \in \bar{T} \quad or \quad z^t \sim p_\theta(z^t|z^{<t}, C^t) \quad otherwise$

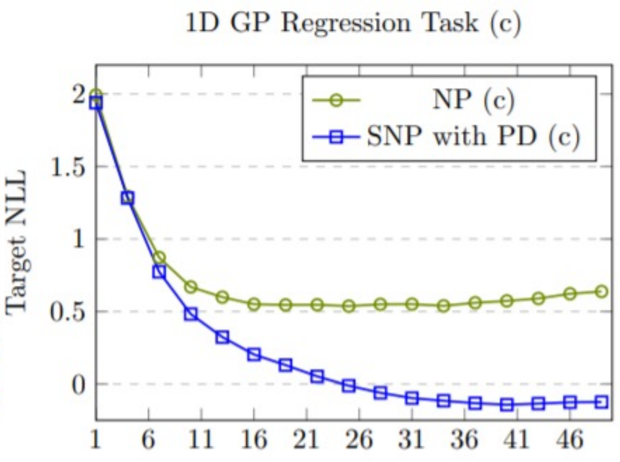
- Total Loss : $L_1 + \alpha L_2$

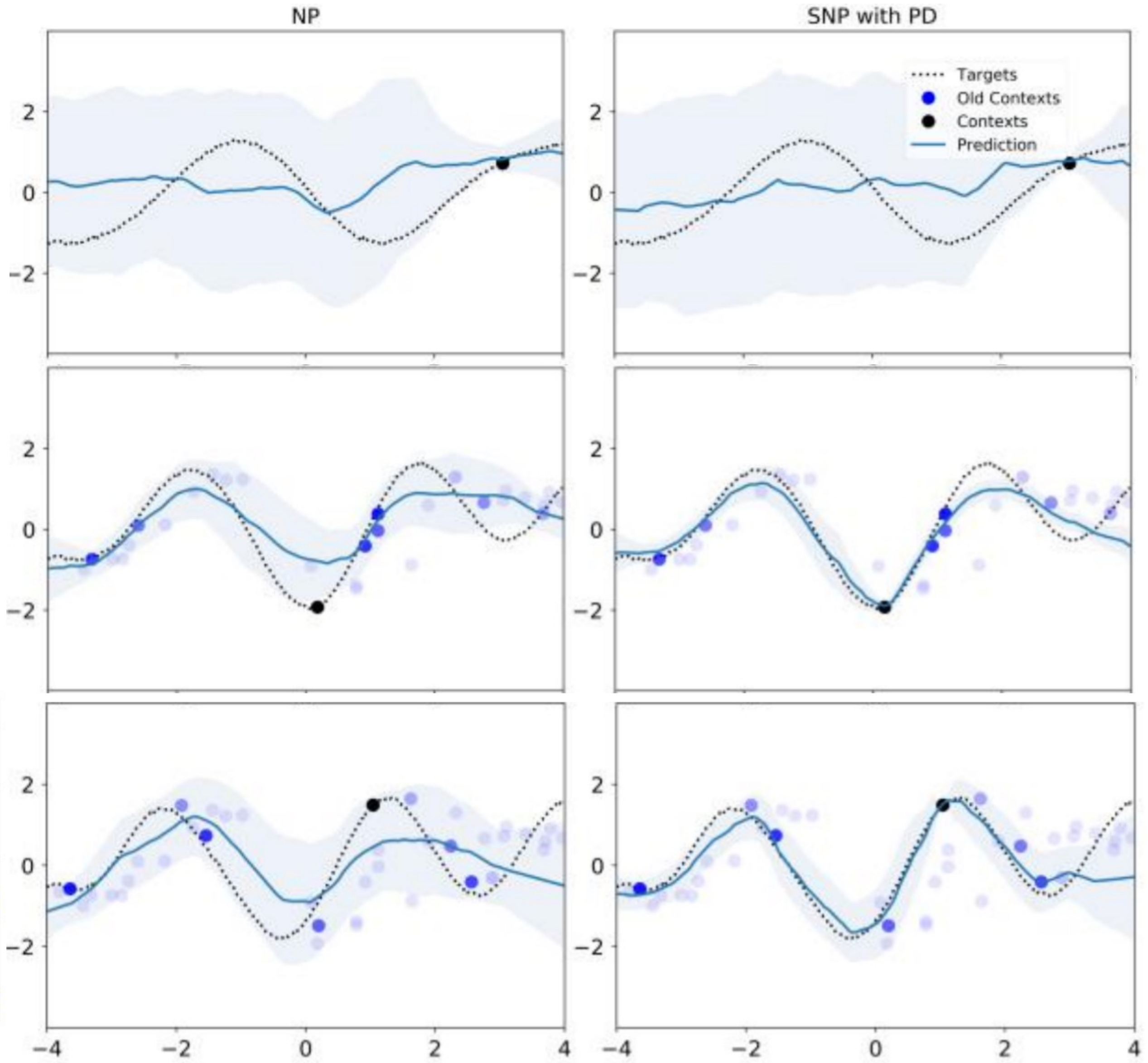
- Synthetic data from gaussian process
- $l, \sigma, \Delta l, \Delta\sigma$ are randomly drawn at t=0

Task (a) and (b)
  context ~ [5,50](50%) or 0(50%)
  target ~ [0, 50-context]

Task (c)
  context : 1(90%) or 0(10%)
  target ~ [0, 10-context]





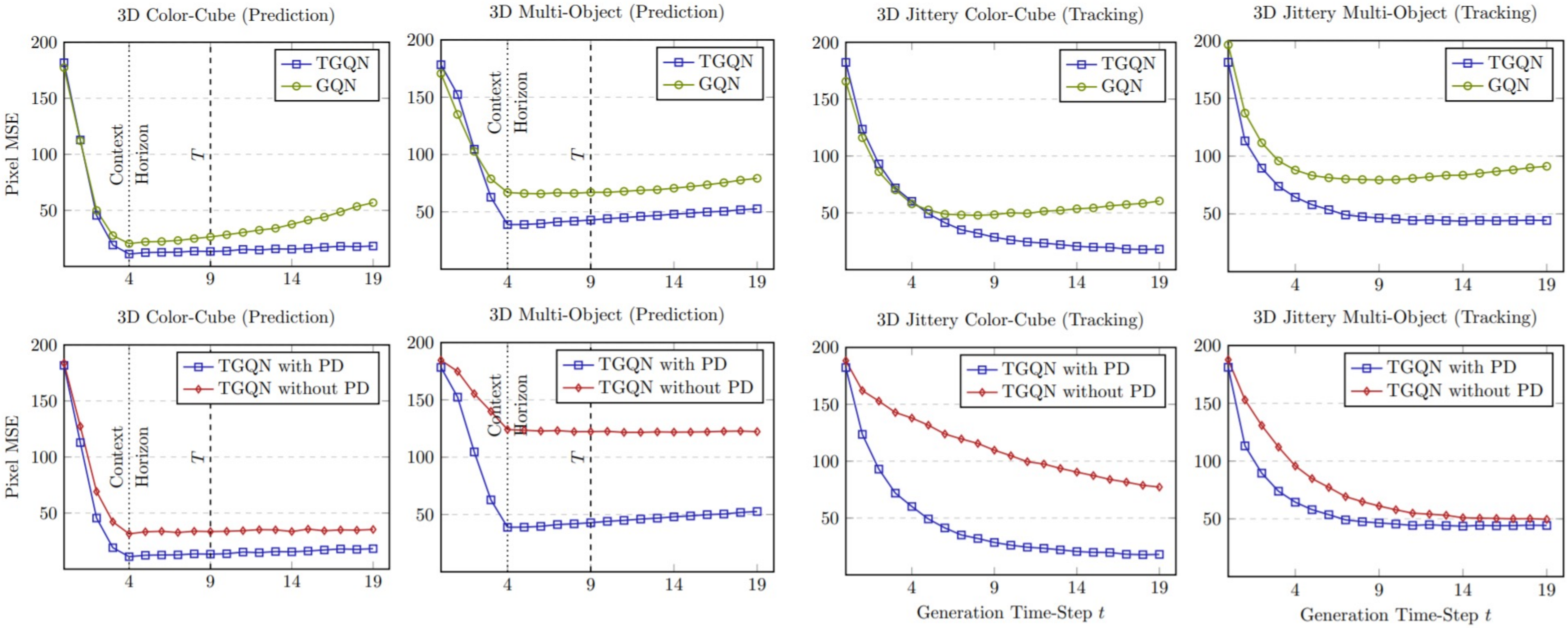


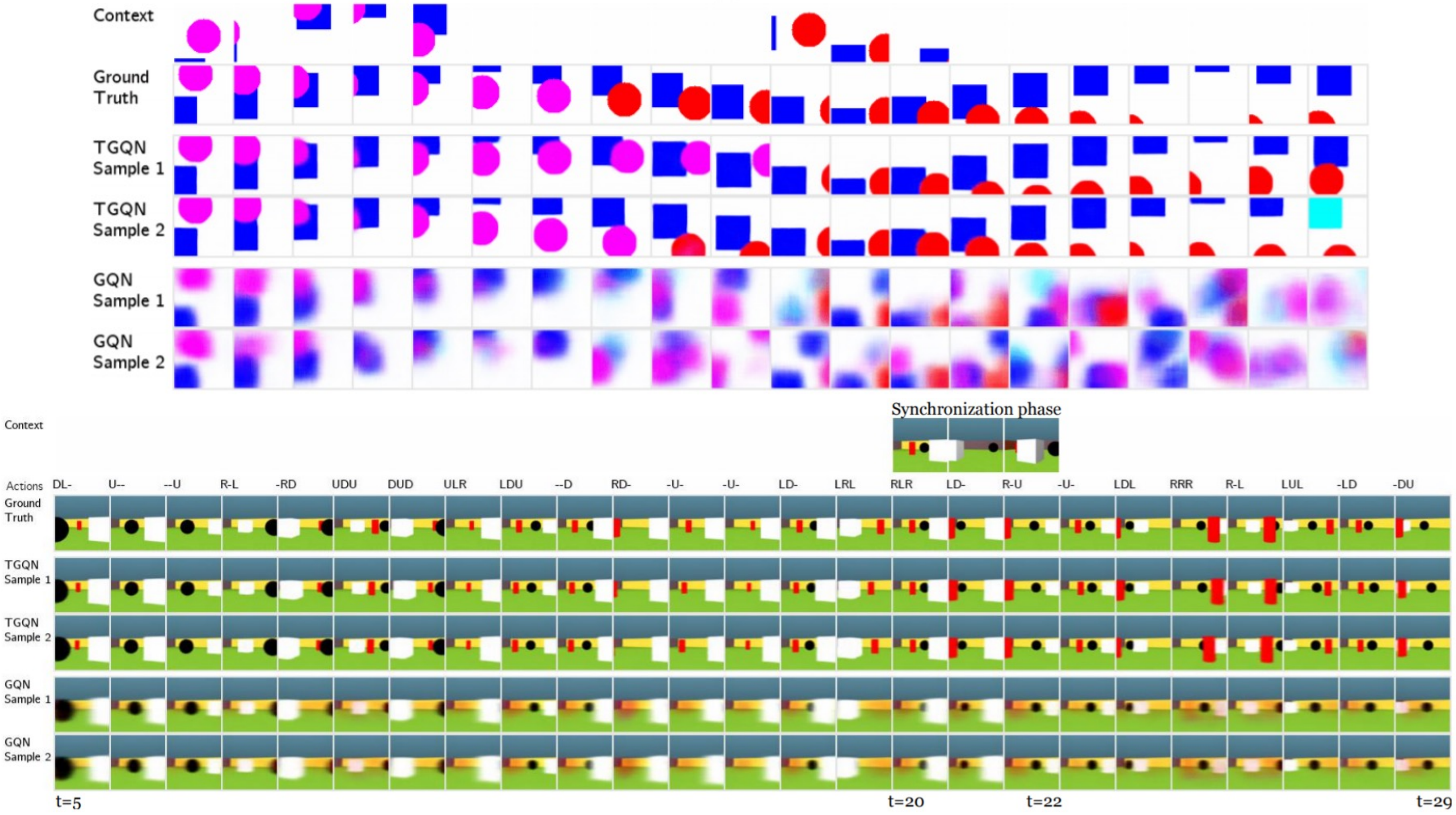
Prediction

Up to 4 observations in first 5 steps

Tracking

Up to 2 observations in every step





Context
Ground Truth
TGQN Sample 1
TGQN Sample 2
GQN Sample 1
GQN Sample 2
Synchronization phase
Context
Actions
DL- U-- --U R-L -RD UDU DUD ULR LDU --D RD- -U- -U- LD- LRL RLR LD- R-U -U- LDL RRR R-L LUL -LD -DU
Ground Truth
TGQN Sample 1
TGQN Sample 2
GQN Sample 1
GQN Sample 2
t=5
t=20
t=22
t=29