View-based Programming with Reinforcement Learning for Robotic Manipulation

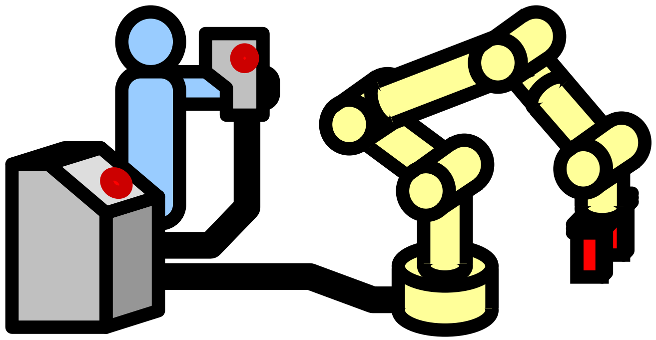
Yusuke MAEDA*, Takumi WATANABE**
and Yuki MORIYAMA*

*Yokohama National University

**Seiko Epson Corp.

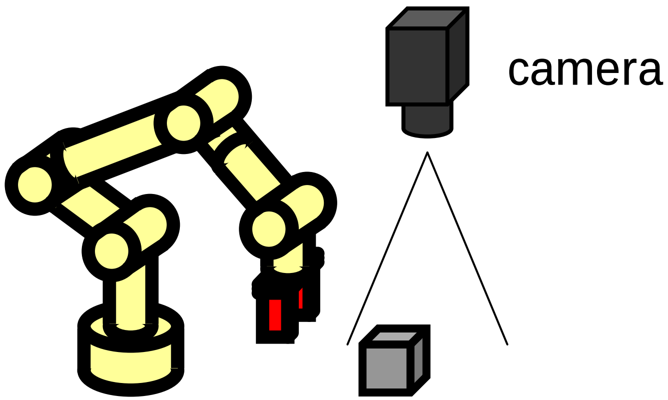
Background

- Conventional Teaching/Playback
  - still widely used
  - versatile
  - for constant task conditions
    - e.g.) initial pose of object does not change



When the initial object pose is not constant…

- Object localization with cameras
  - Model-based image processing
    - Feature extraction: edge, vertex, …
    - Pattern matching
  - Object-specific: versatility is limited



Motivation

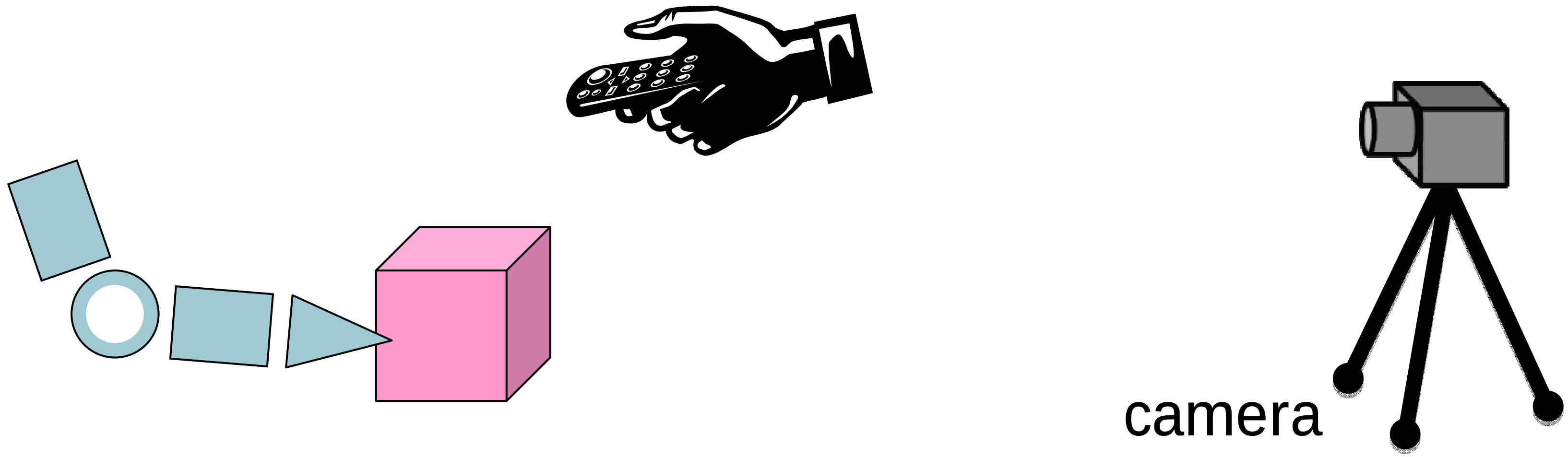
- To develop a **versatile** robot programming method that can cope with change of task conditions

“View-based teaching/playback”:
robot programming with
view-based image processing

Model-based vs. View-based

- Model-based approach
  - with object-specific models
  - accurate
- View-based (Appearance-based) approach
  - without object-specific models
  - versatile
  - no need for camera calibration
  - not so sensitive to lighting

Overview of view-based teaching/playback (1/3)

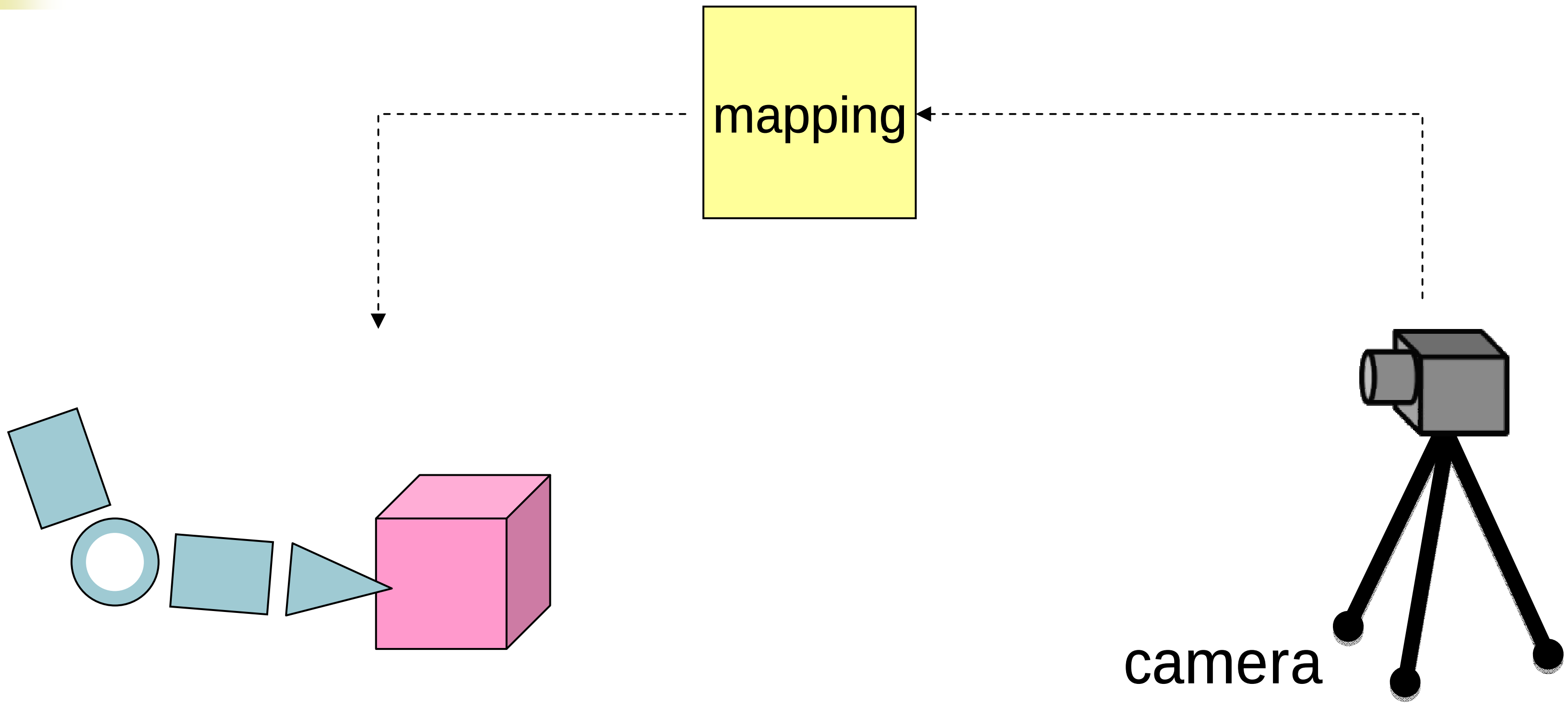


1. Human demonstration: Operator commands a robot to perform a manipulation task

Overview of view-based teaching/playback (2/3)

2. Obtain a mapping from image to motion

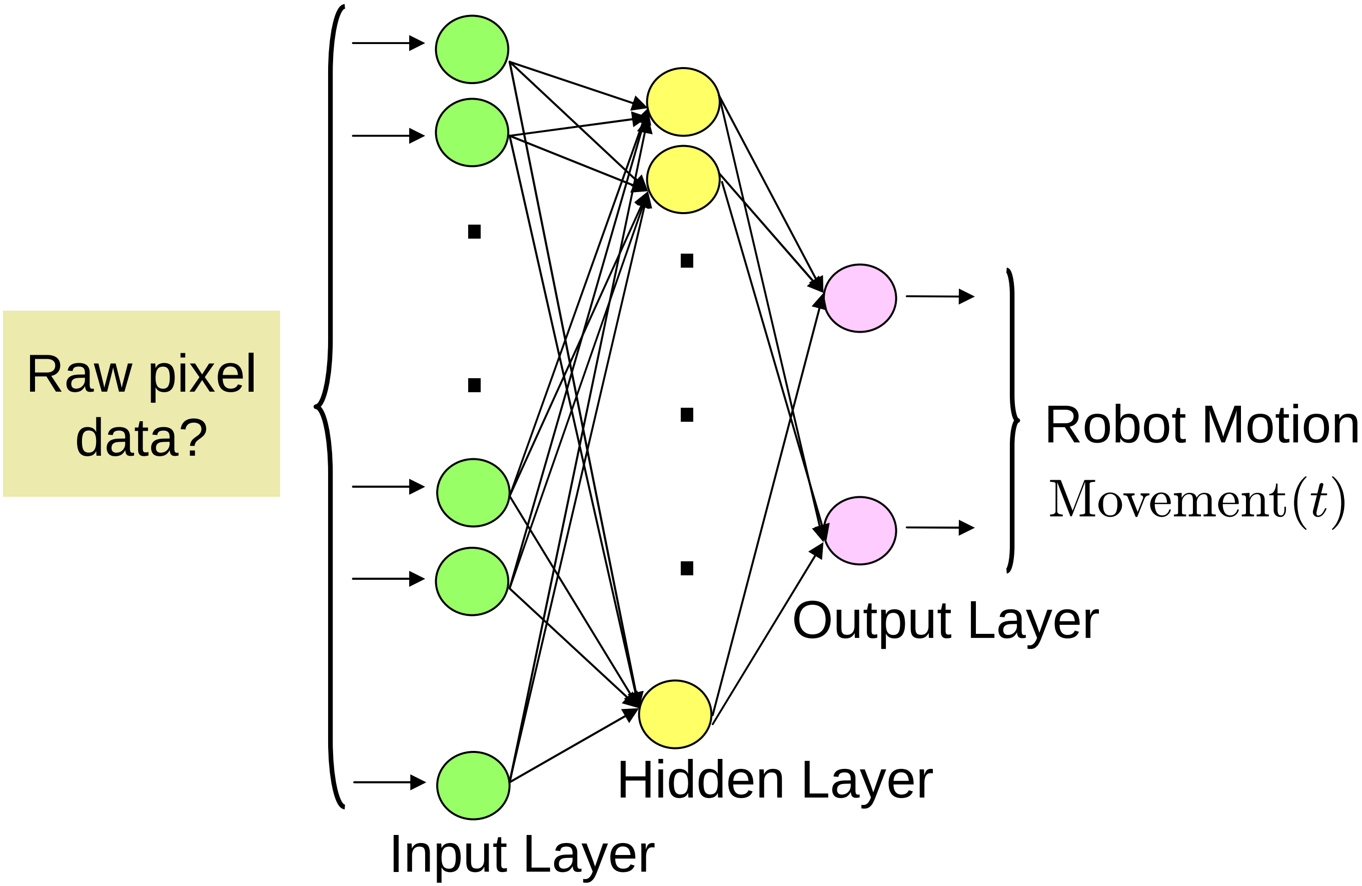
Overview of view-based teaching/playback (3/3)



3. Playback: Robot motion generation according to the mapping

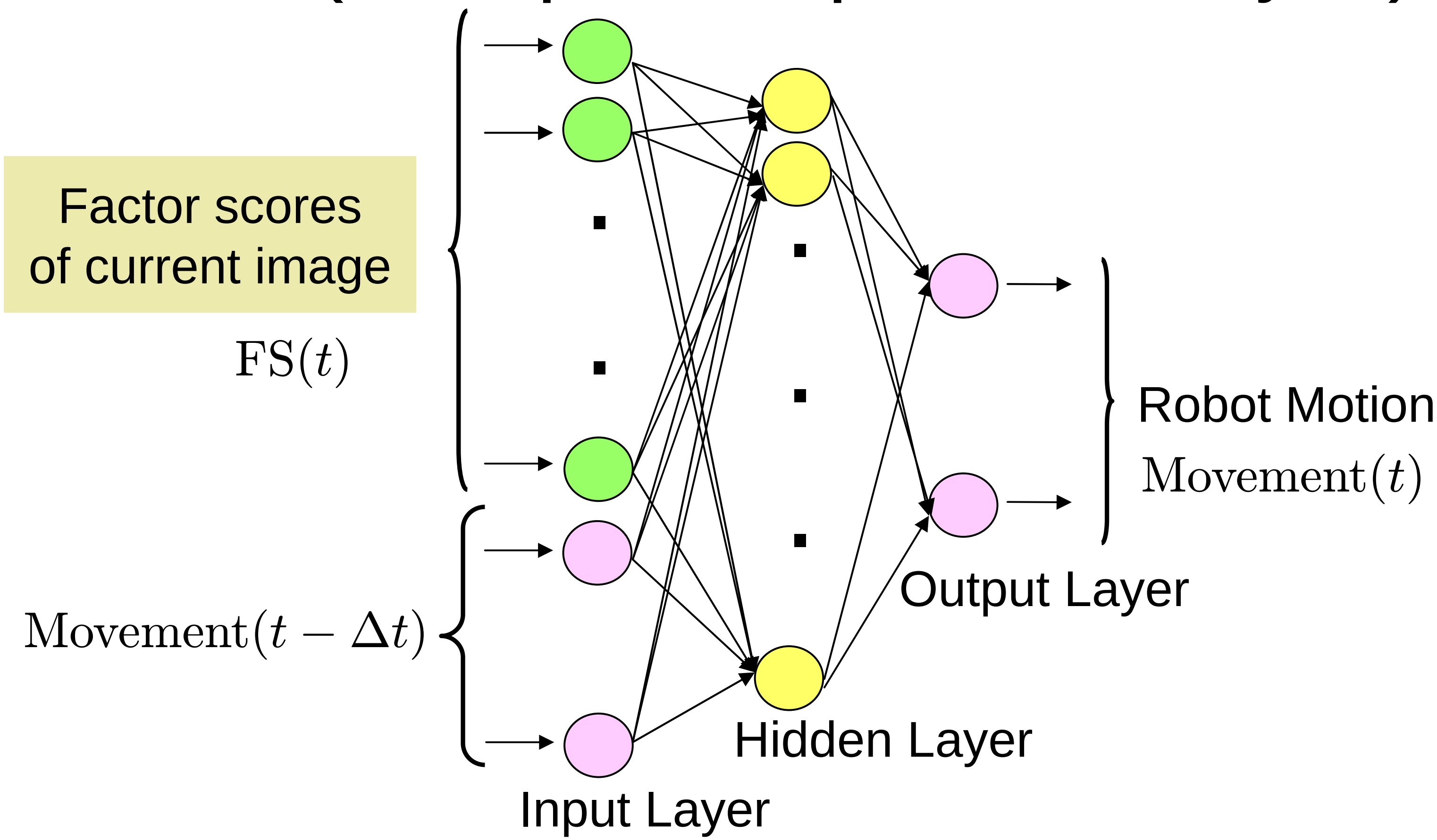
Mapping from image to motion (1/2)

- Neural Network



Mapping from image to motion (2/2)

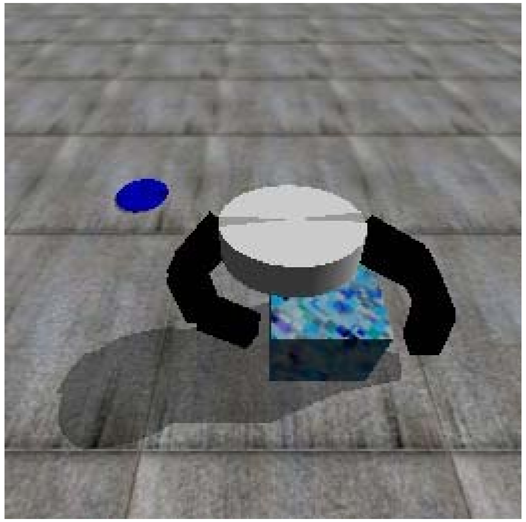
- PCA (Principal Component Analysis)



View-based teaching/playback

- View-based image processing using PCA
  - not object-specific
  - no need for camera calibration
- Adaptability to change of initial object pose using the generalization ability of neural networks
  - generalization from multiple demonstrations

Implementation of view-based teaching/playback



Pushing/Pick-and-Place by a virtual robot hand [Maeda 2010 ICAM]

Pushing by an actual industrial manipulator [Maeda 2011 ICRA]

Objective

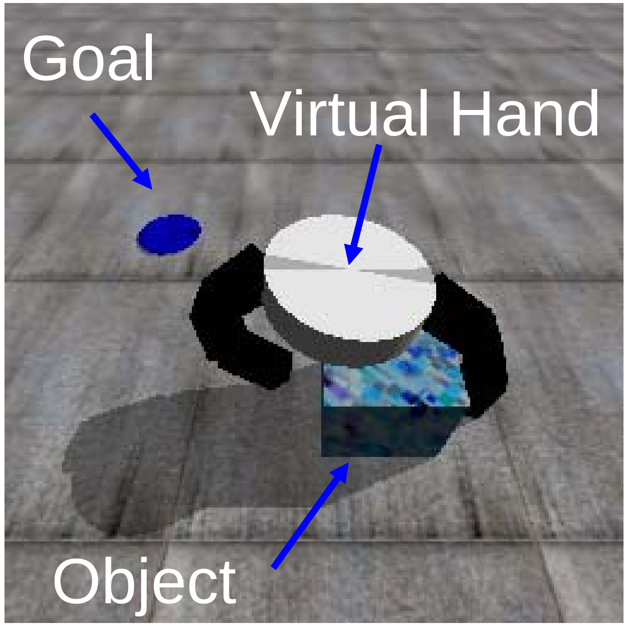
- To deal with wider change of task conditions **without** additional human demonstrations

Integration of Reinforcement Learning

- tested on virtual manipulation environment

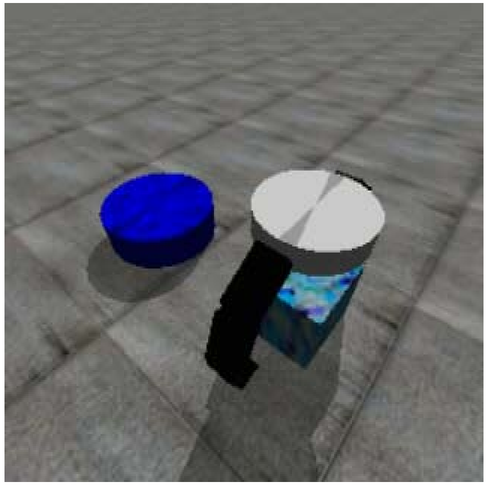
Virtual Hand

- PD-controlled in ODE (Open Dynamics Engine) according to keyboard input
- 12 DOF (at most)
  - 6 DOF for palm
  - 3 DOF for thumb
  - 3 DOF for index finger

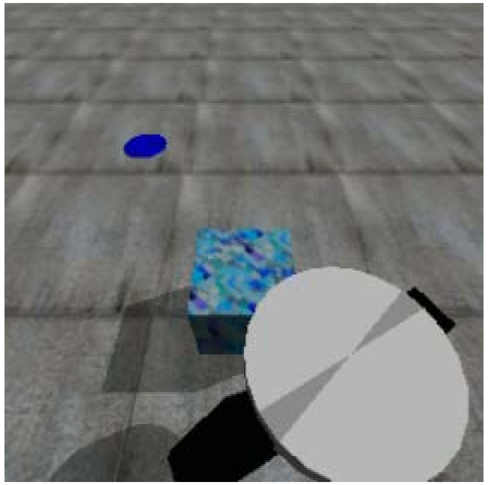


Target Manipulation

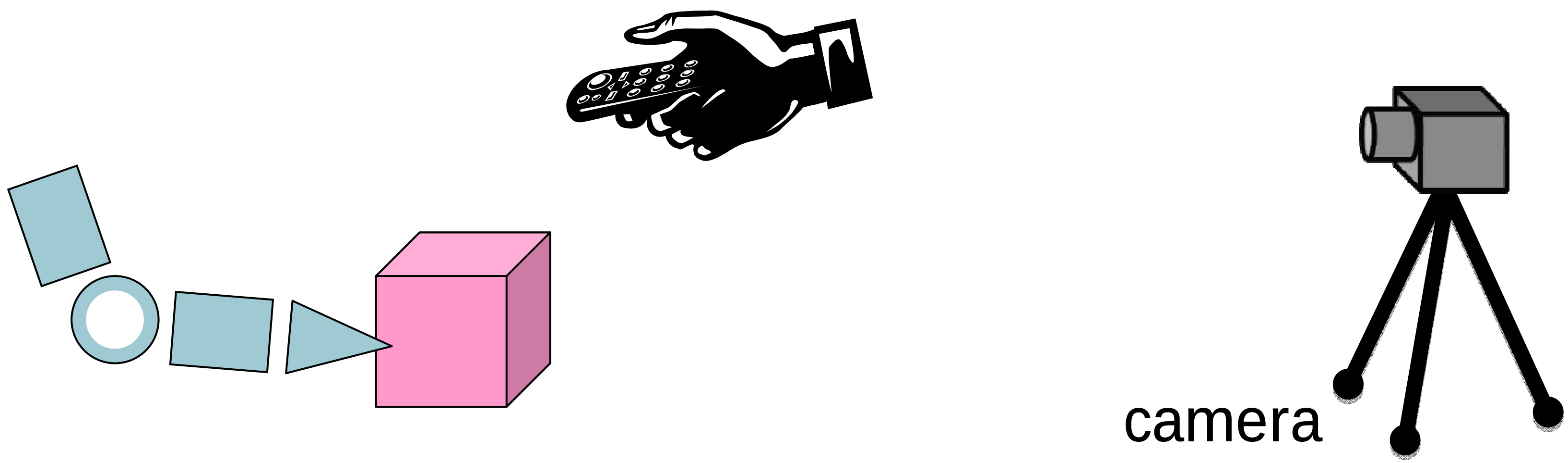
- Manipulation by grasping (pick-and-place)



- Graspless manipulation (pushing)



Overview of view-based teaching/playback with RL (1/3)

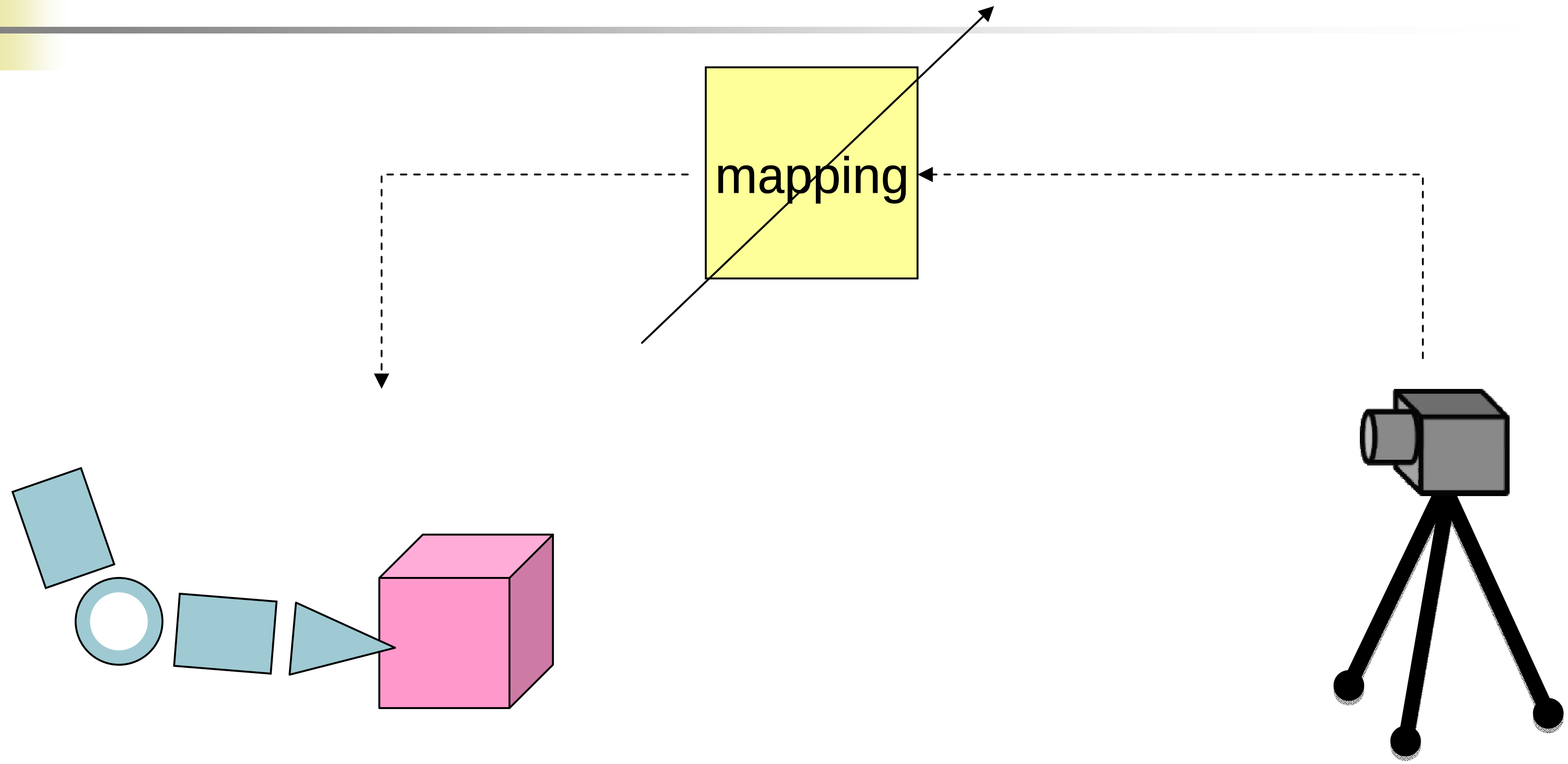


1. Human demonstration: Operator commands a robot to perform a manipulation task

Overview of view-based teaching/playback with RL (2/3)

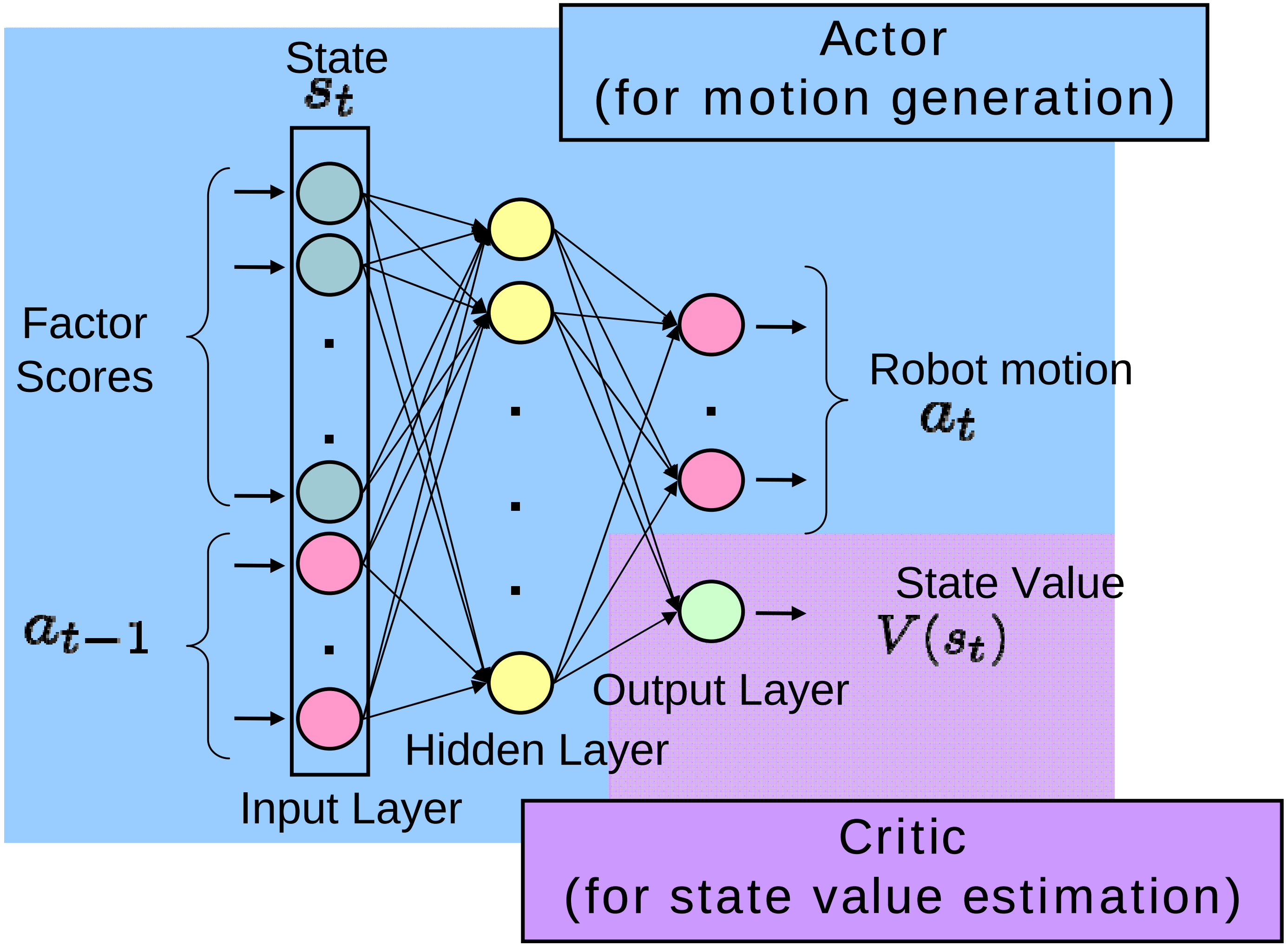
2. Obtain an **initial** mapping from image to motion

Overview of view-based teaching/playback with RL (3/3)

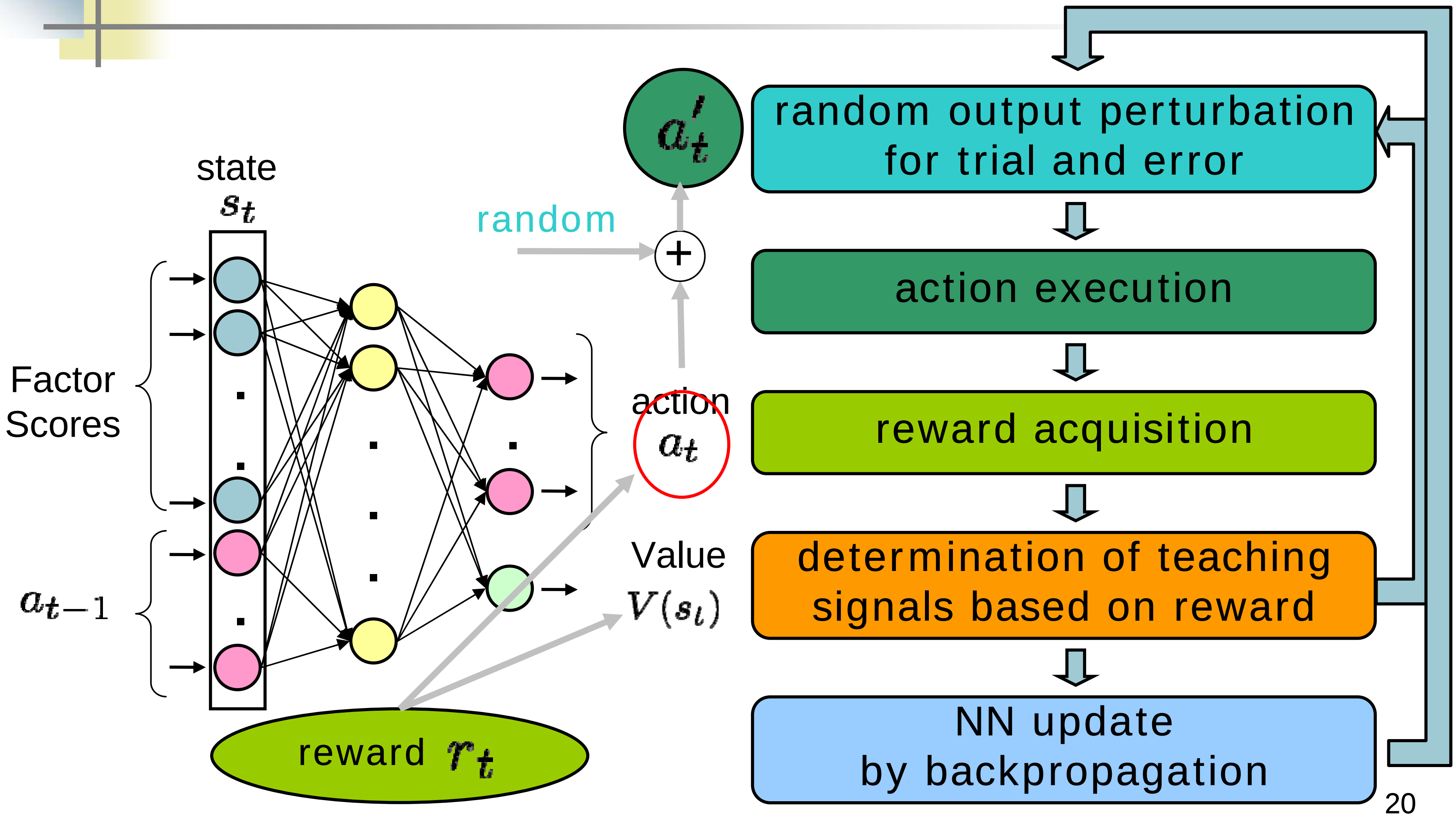


3. Repeated playback and reinforcement learning: Robot motion generation according to the mapping and its iterative update

Neural Network for actor-critic-based Reinforcement Learning



Actor-critic-based Reinforcement Learning



Details of reinforcement learning (based on [Shibata 03])

1. Random perturbation to actor output for trial and error

random perturbation for each episode

$$\boldsymbol{a}'_i(\boldsymbol{s}_t) = \boldsymbol{a}_i(\boldsymbol{s}_t) + \boldsymbol{R}_t + \boldsymbol{R}_e$$

random perturbation for each action

2. Calculation of TD error based on reward

$$\mathrm{TD}_{\mathrm{error}} = r_{t+1} + \gamma V(\boldsymbol{s}_{t+1}) - V(\boldsymbol{s}_t)$$

3. Setting teaching signals according to TD error

$$T_c(\boldsymbol{s}_t) = V(\boldsymbol{s}_t) + \beta(\mathrm{TD}_{\mathrm{error}}) \quad \text{(for critic)}$$

$$\boldsymbol{T}_a(\boldsymbol{s}_t) = \boldsymbol{a}(\boldsymbol{s}_t) + \rho(\mathrm{TD}_{\mathrm{error}})(\boldsymbol{a}'(\boldsymbol{s}_t) - \boldsymbol{a}(\boldsymbol{s}_t)) \quad \text{(for actor)}$$

Reward function for RL

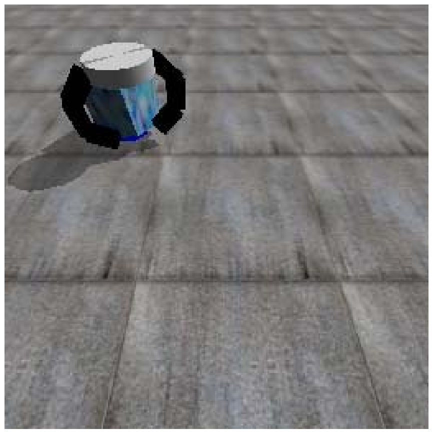
- Reward is necessary for reinforcement learning
  - Typical reward for manipulation: (negative of) distance between current object position and goal position

Not available because of view-based approach

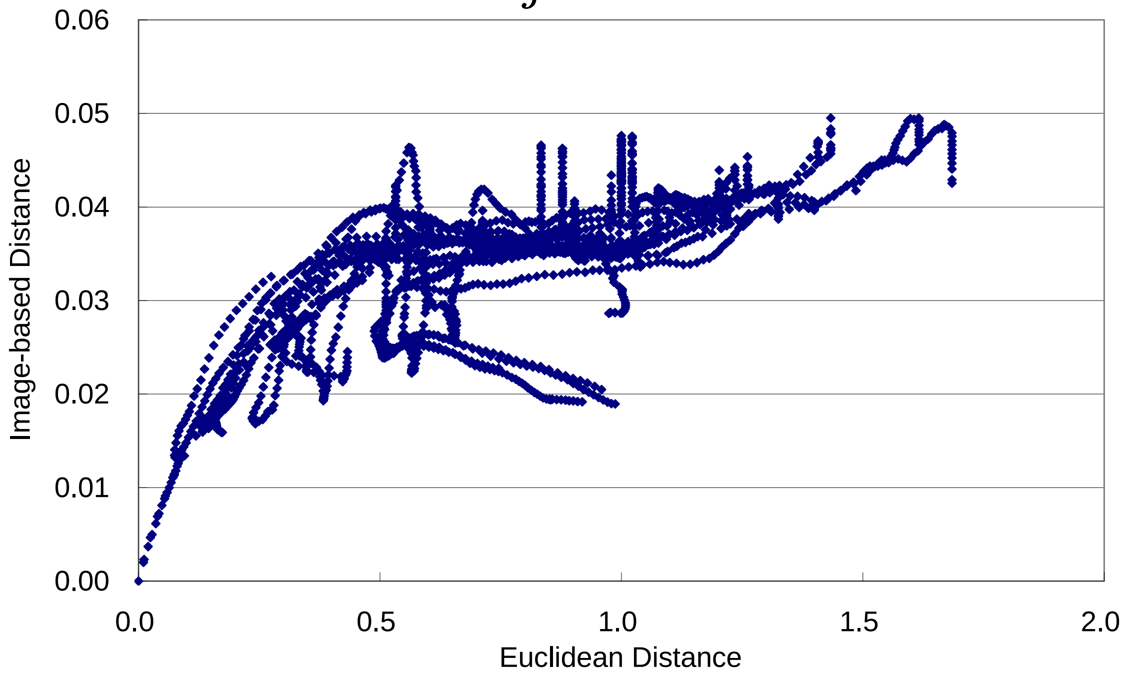
View-based reward

- Distance between current image and goal image

$$D_I(\boldsymbol{I}, \boldsymbol{I}_G) = \sum_{j=1}^{N_{\text{pixel}}} \frac{|I_j - I_{Gj}|}{N_{\text{pixel}}}$$



goal image



Target task 1: Pushing

- Push the object to the goal position

Hand

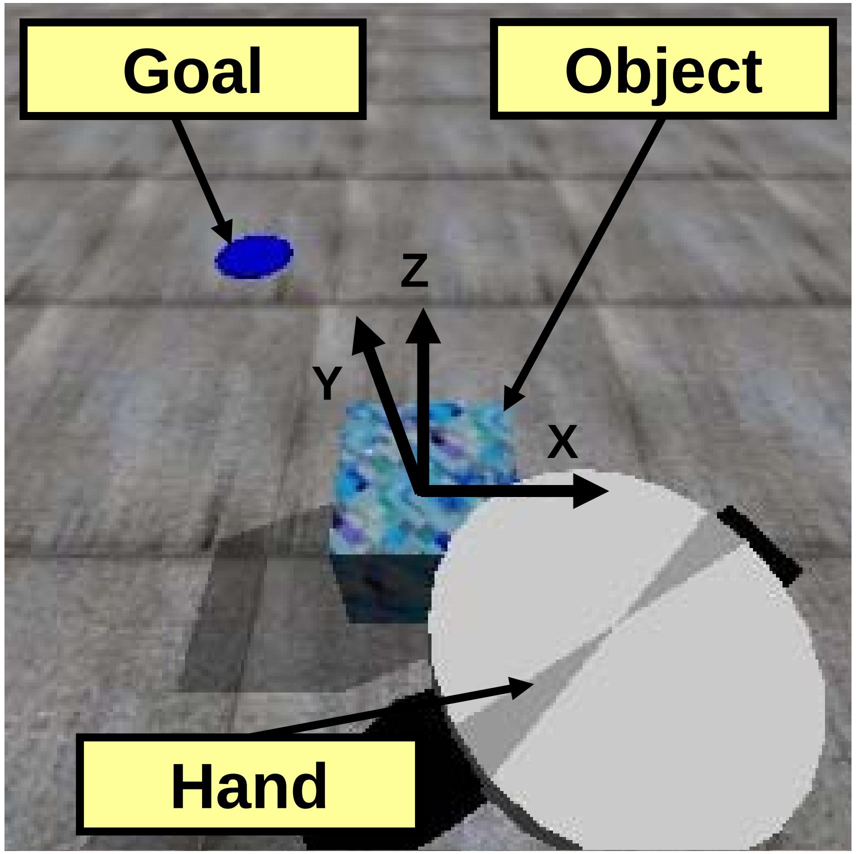
- Reduced to 3 DOF

horizontal translation + rotation

(x, y, θ)

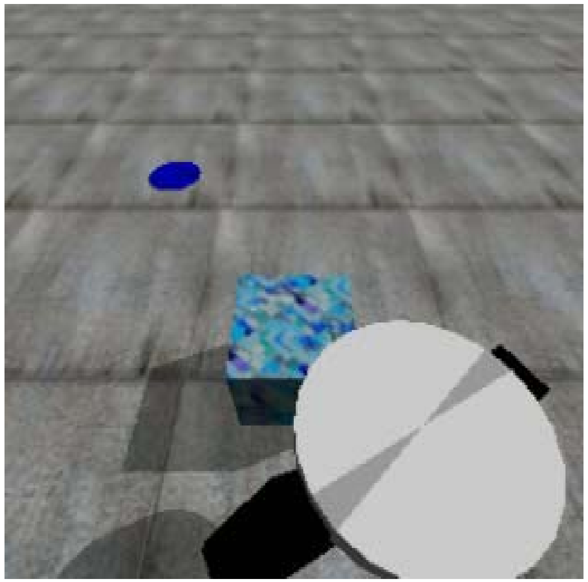
Object

- cube

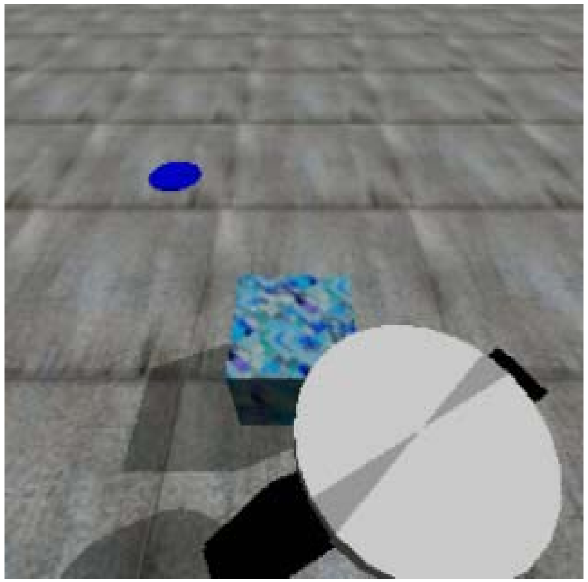


Playback **before** Reinforcement Learning

- Successful playback from the initial position of the demonstration



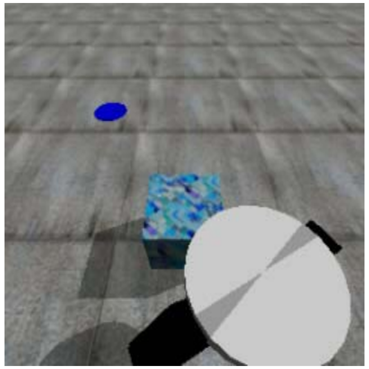
Demonstration



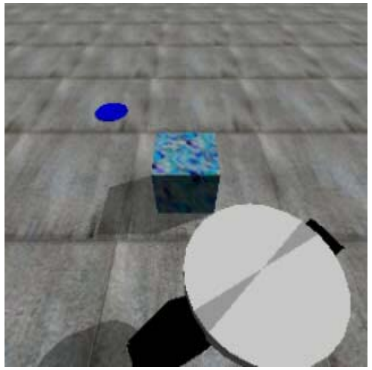
Playback

Reinforcement Learning

- Repeated manipulation from an initial position from which playback is not successful before reinforcement learning

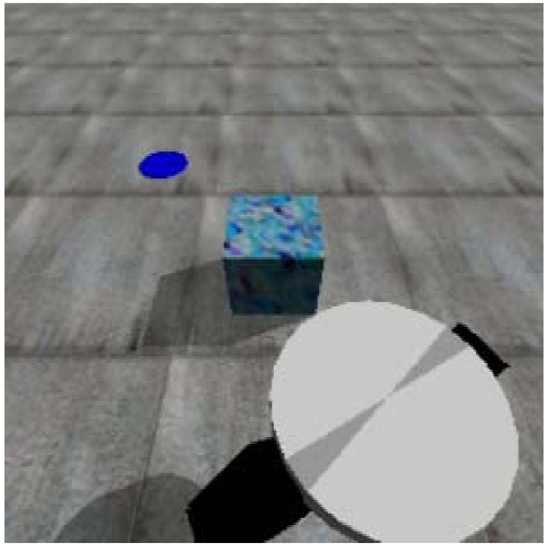


initial position
of human demonstration

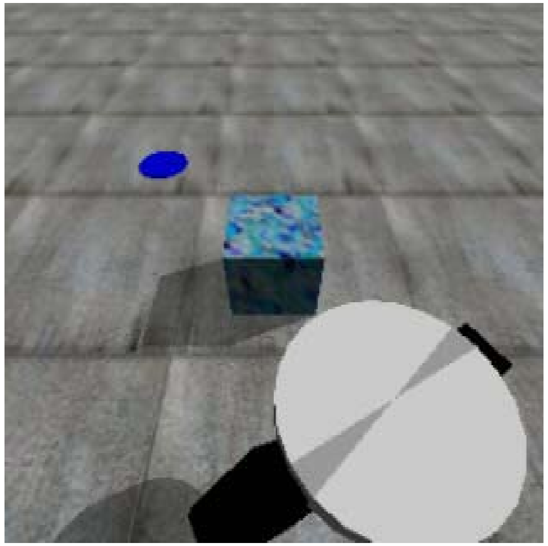


shifted initial position

Learning result



before RL

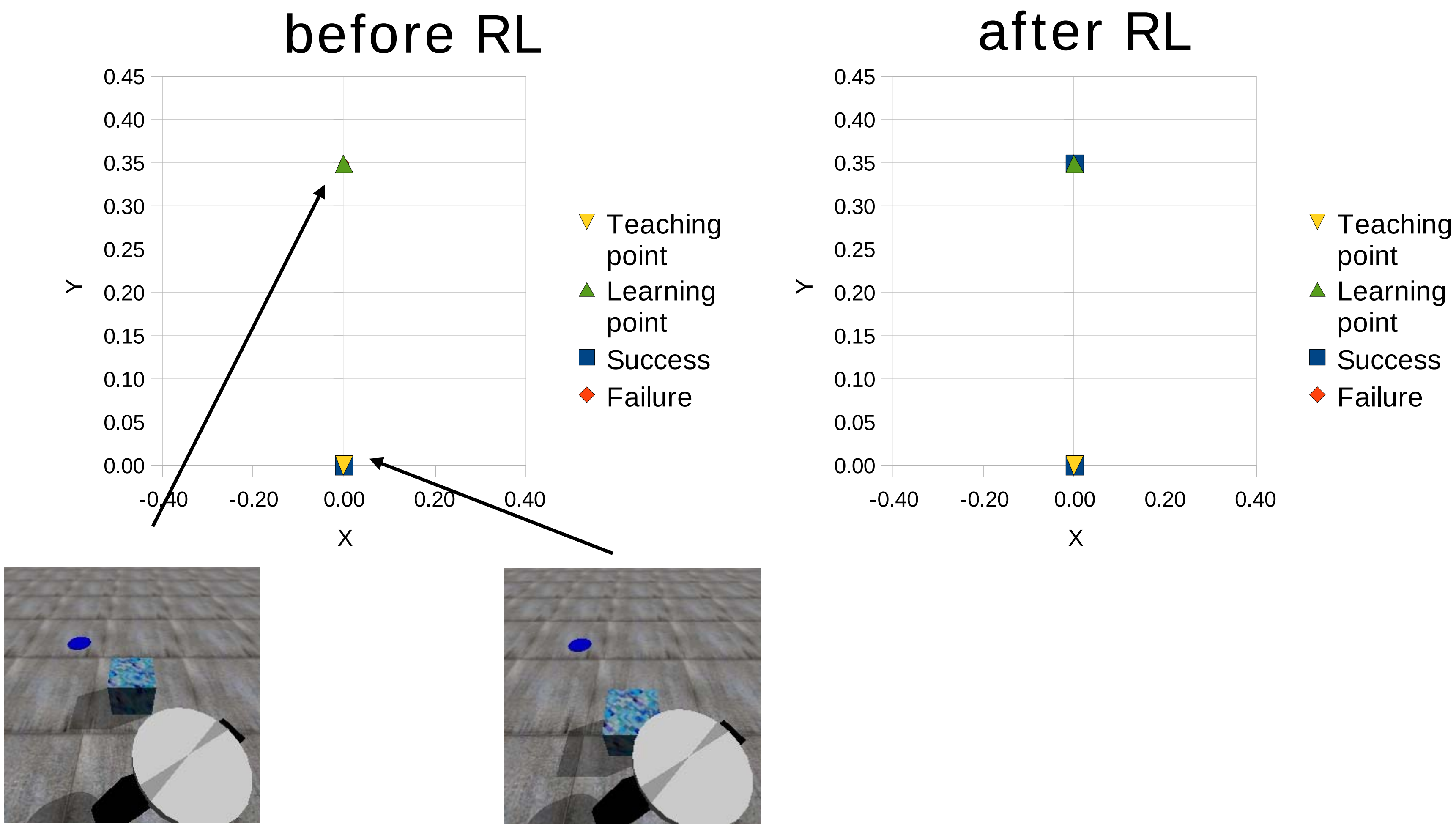


after RL

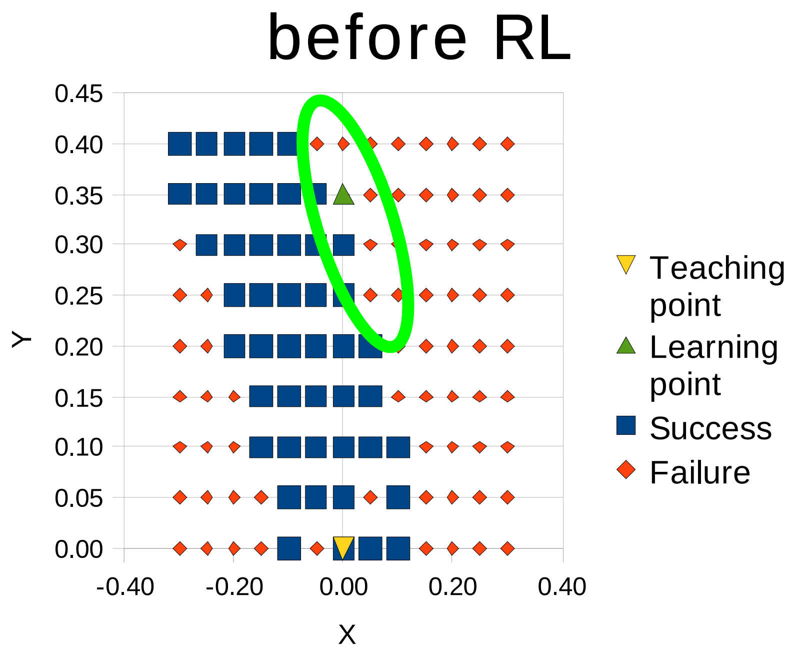
Computation time: 6554 [s] (CPU: core i7 870, 1000 episodes)

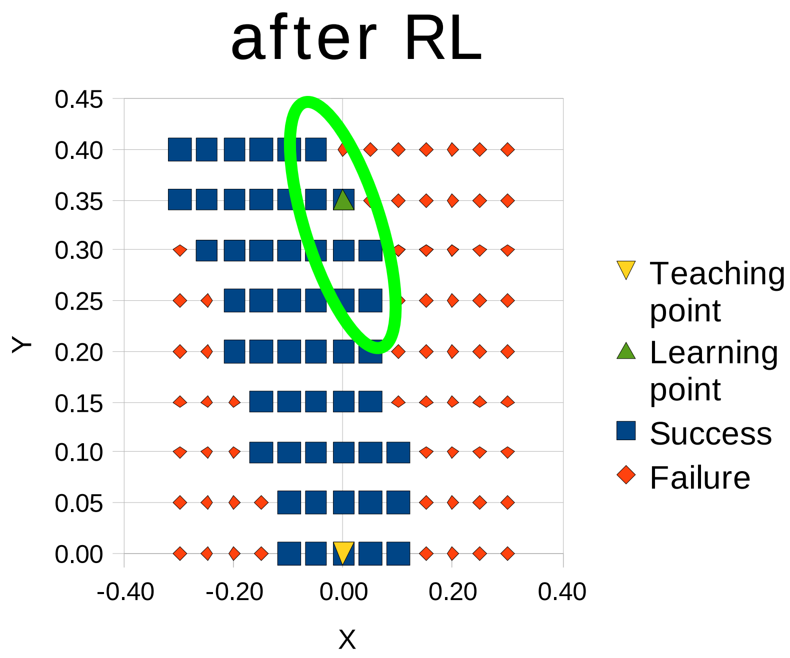
Successful manipulation from the shifted initial position from which it was not possible before RL

Range of initial positions for successful pushing



Range of initial positions for successful pushing





Wider success region after RL

Target task 2: Pick-and-place

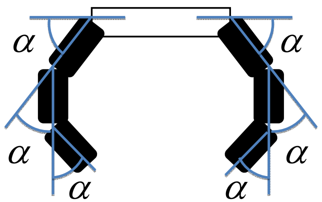
- Pick the object up and place it at the goal

Hand

- Reduced to 3 DOF

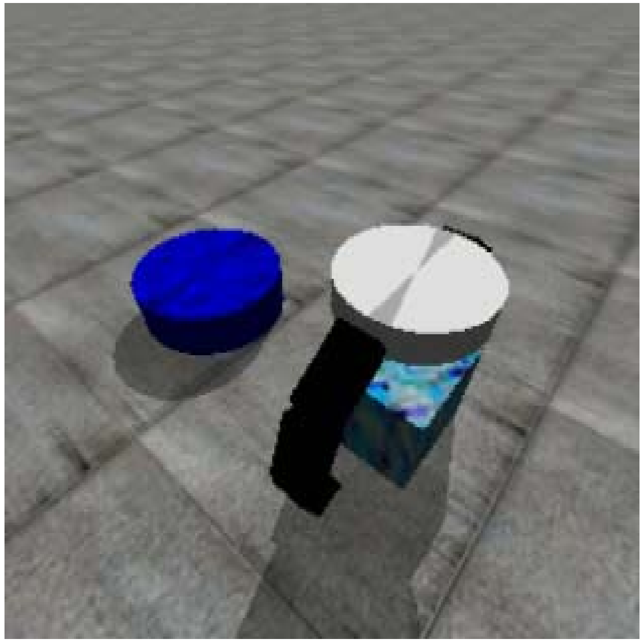
Translation in sagittal plane
+ finger bending

$$(y, z, \alpha)$$

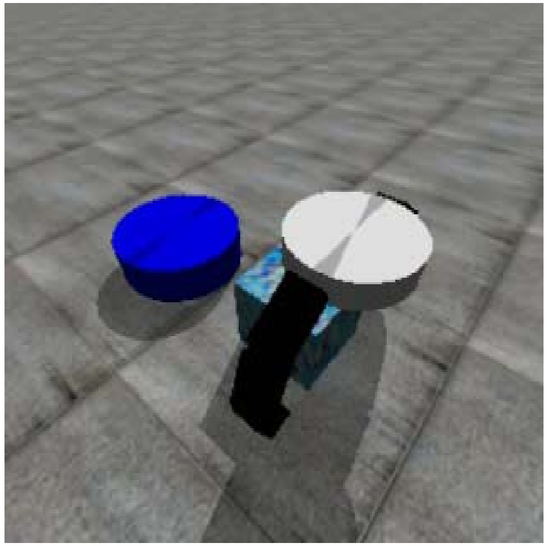


Object

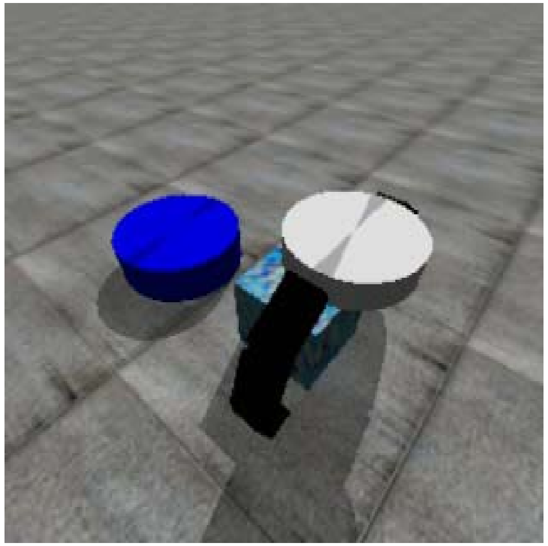
- cube



Learning result



before RL



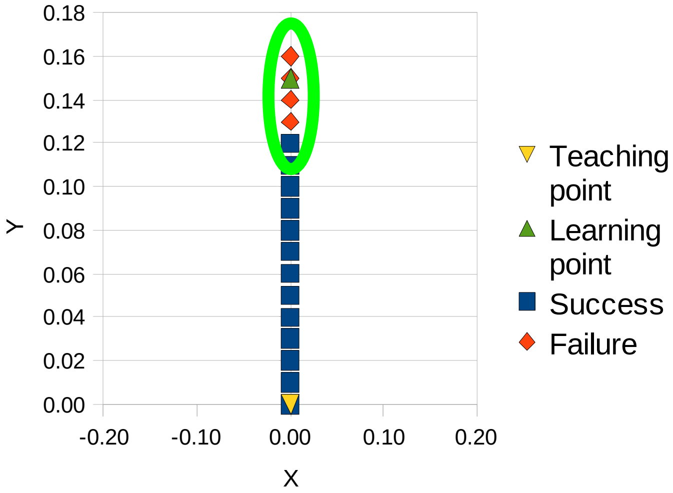
after RL

Computation time: 6244 [s] (CPU: core i7 870, 1000 episodes)

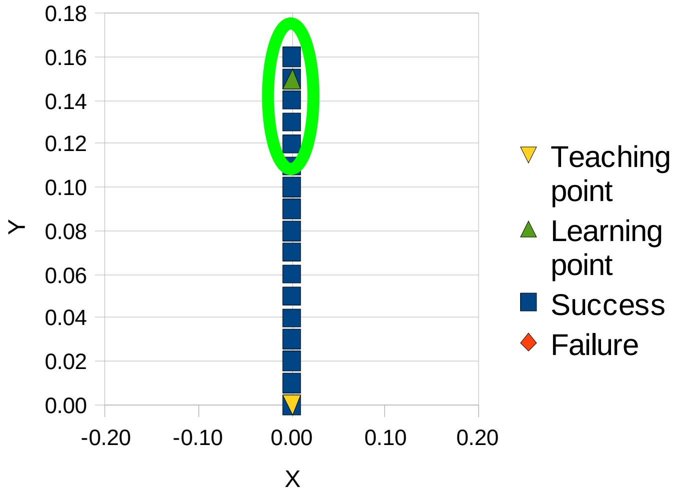
Successful manipulation from an initial position from which it was not possible before RL

Range of initial positions for successful pick-and-place

before RL



after RL



Wider success region after RL

Conclusion

- Reinforcement learning was integrated with our view-based teaching/playback
- Autonomous adaptation to wider task conditions was achieved on a virtual environment

Future Work

- Computation reduction
(current: ~7000 [s])
- Improvement of learning success rate
(current: ~30%)
- Application to various tasks that require higher DOF
- Application to actual robot systems