ReAgent:
Point Cloud Registration using Imitation and Reinforcement Learning

Dominik bauer, Timothy Patten and Markus Vincze
TU Wien
Vienna, Austria

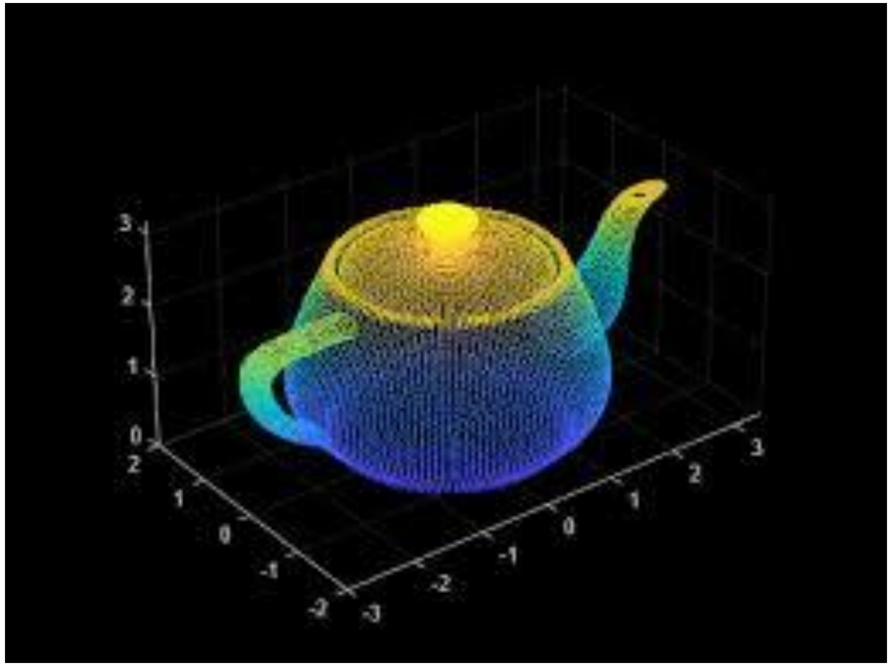
Presenter : 김진용

Content

- Background
    - Point Cloud Registration
    - RL, IL
- Define Problem
- Point Cloud Registration Agent
    - Reagent Architecture
    - Imitation an Expert Policy
    - Improving through Reinforcement
- Experiment

Background

What is Point Cloud?



Point Cloud

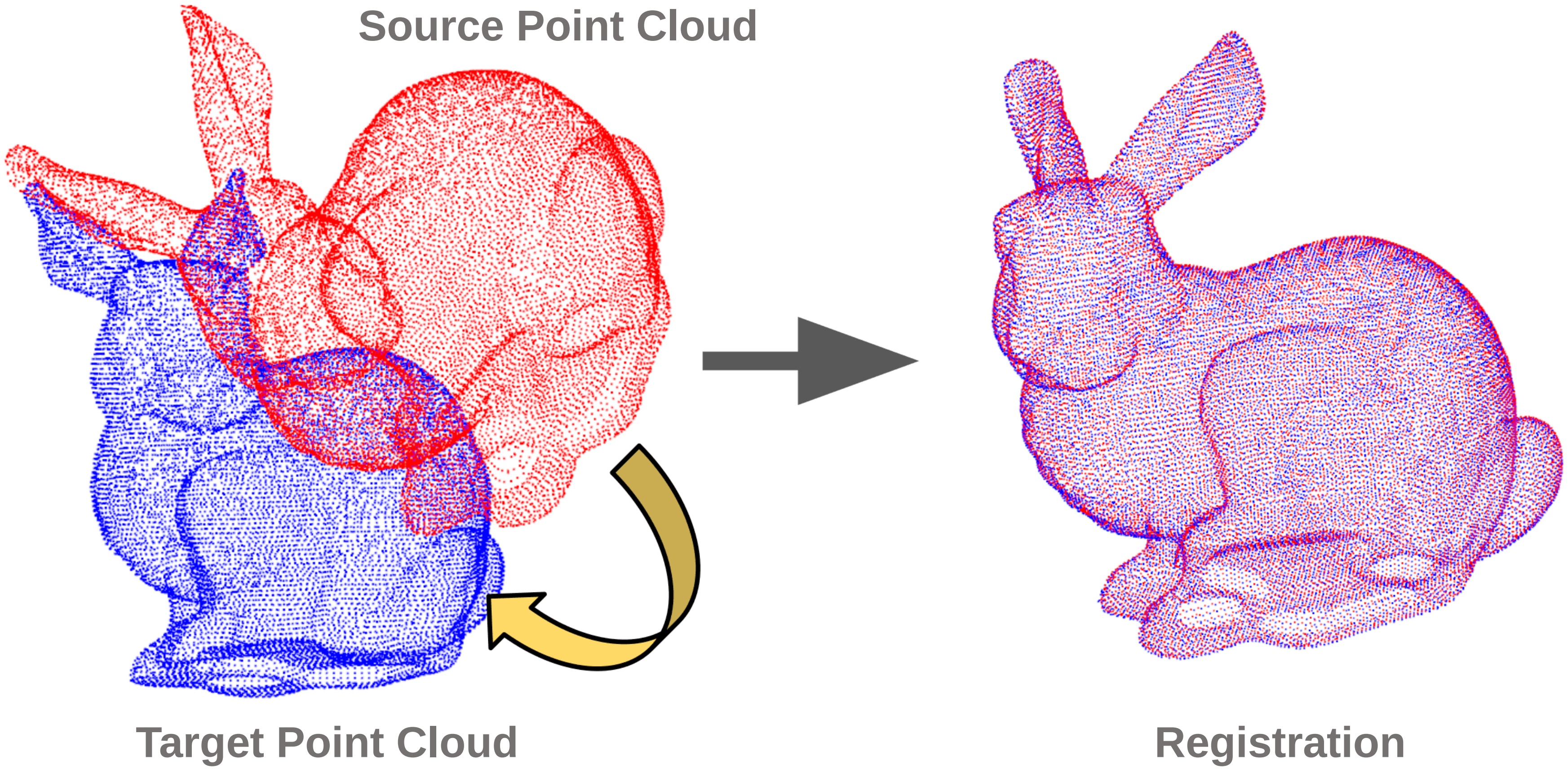
(X, Y, Z) n x 3 set

If number of point cloud is n,
it has below structure.

[[X1, Y1, Z1],
[X2, Y3, Z2],
…
[Xn, Yn, Zn]]

Background

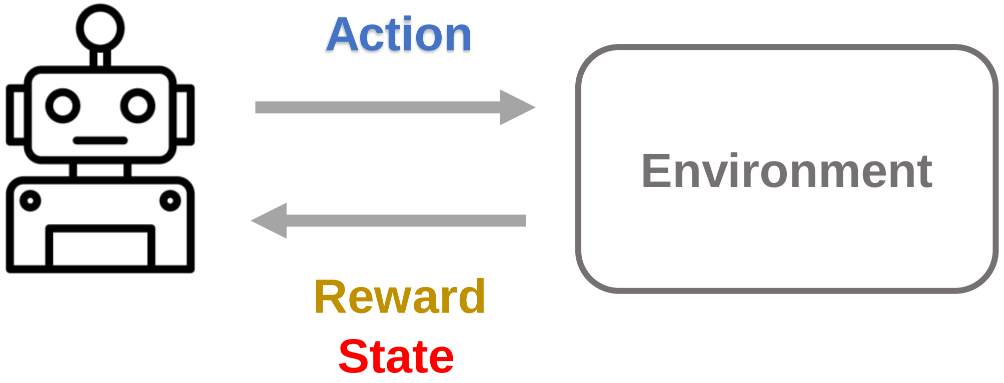
What is Point Cloud Registration?



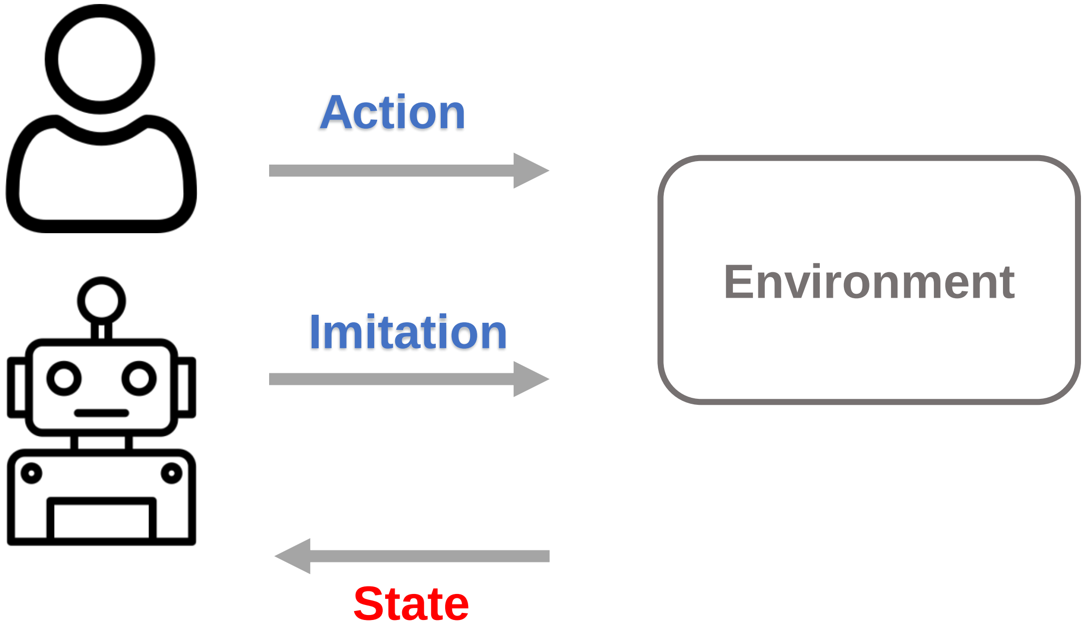
Background

What is Reinforcement Learning and Imitation Learning?

Reinforcement Learning



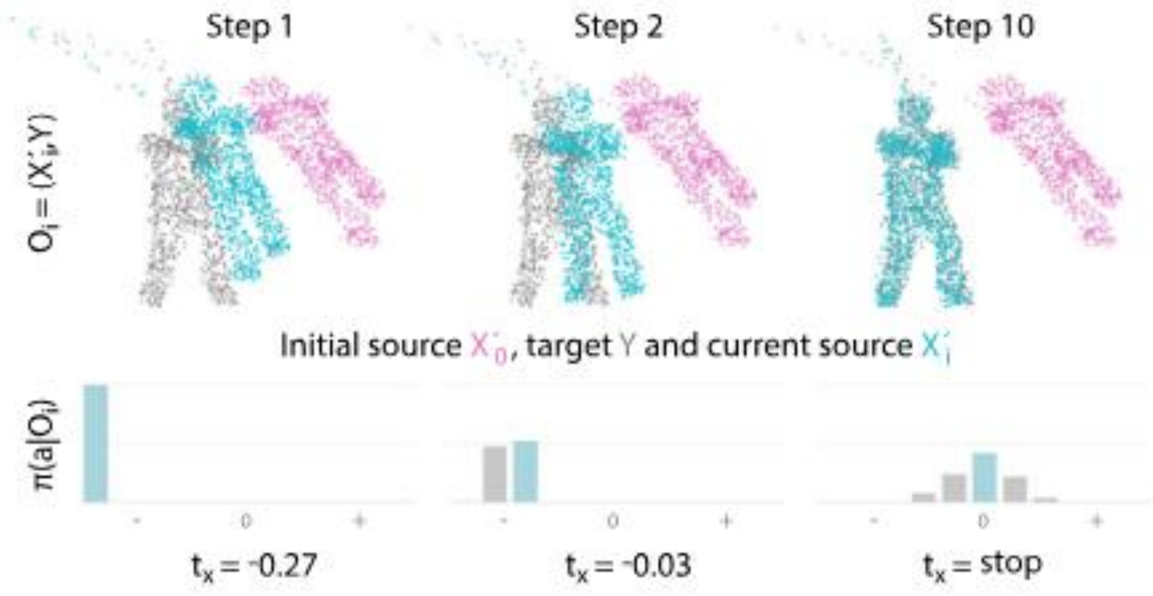
Imitation Learning



Define Problem

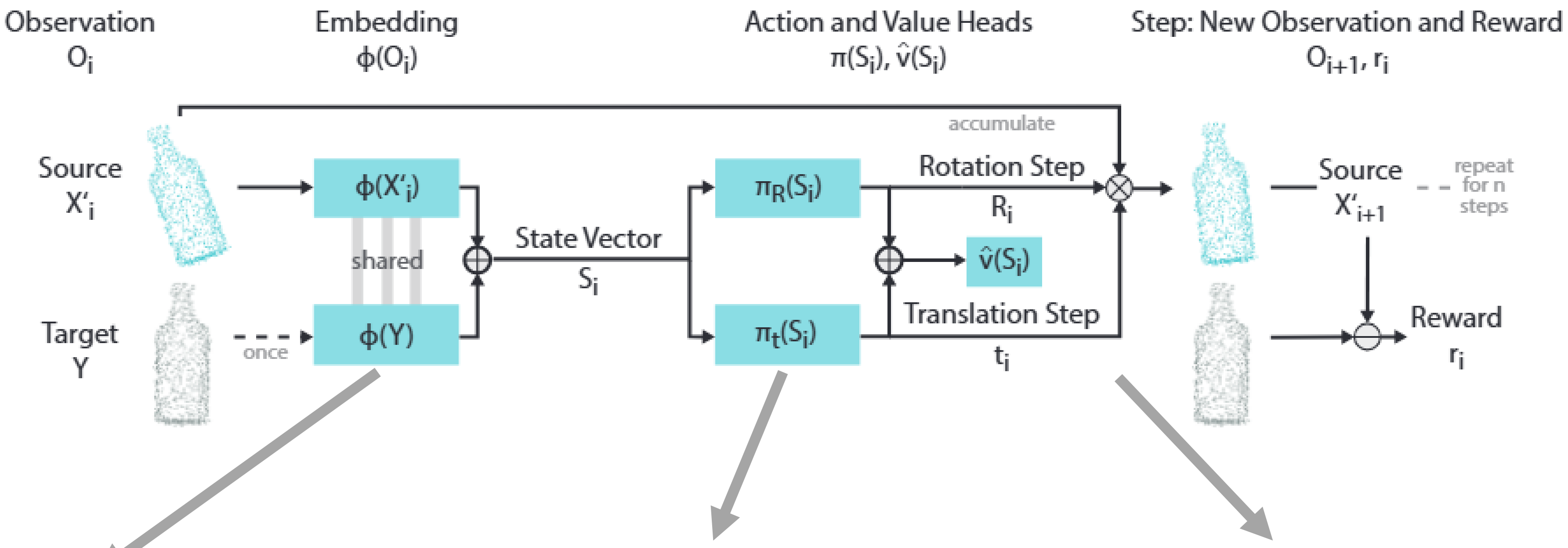
- Classical Registration Methods generalize well to novel domains but fail when **given a noisy observation or a bad initialization**.
- Learning-based methods, in contrast, are more robust but **lack in generalization capacity**.

RL + IL → Iterative Point Cloud Registration



Point Cloud Registration Agent

Reagent Architecture



State Vector
The number of point cloud is 1024.
We have to extract feature to reduce computational cost.
We may use MLP to get **state vector** S_i.

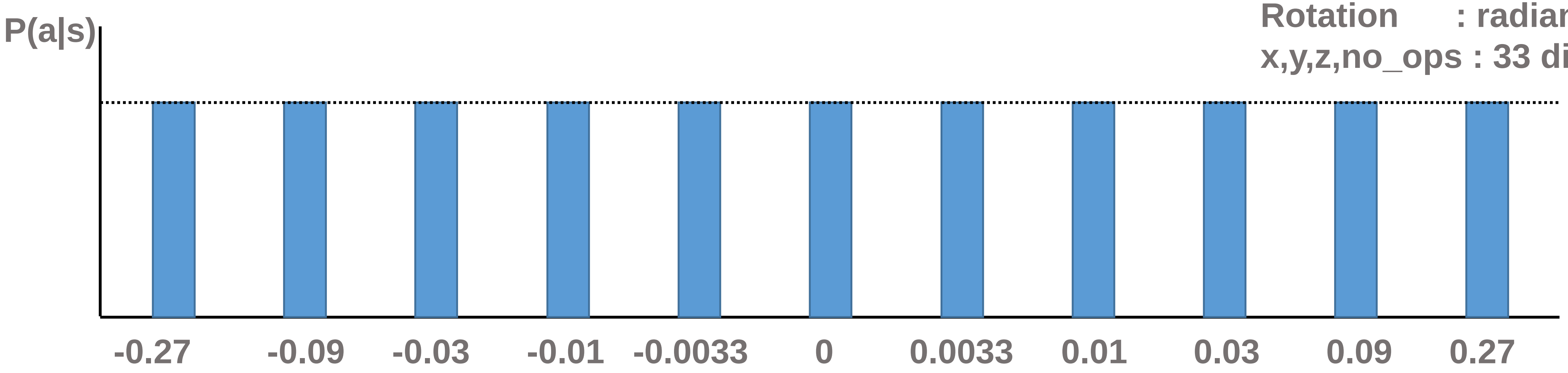
Action vector
From state vector, actor-critic predicts **rotation and translation action vector**.

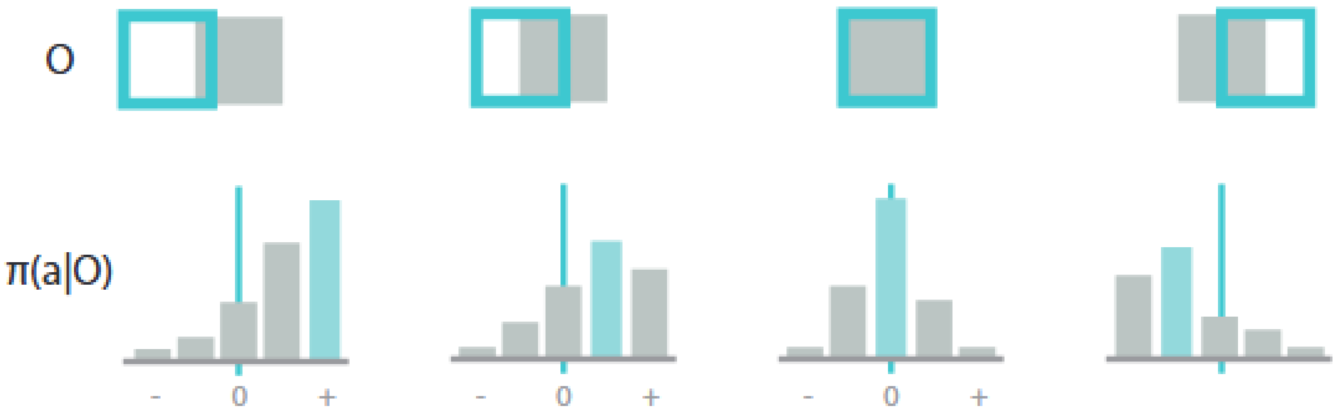
Step
We **transform previous observation to new observation** by using transformation matrix. And environment offers reward comparing chamfer distance between two observations.

Point Cloud Registration Agent

Discrete action space

Translation : cartesian
Rotation : radian
x,y,z,no_ops : 33 dim





Large step to right

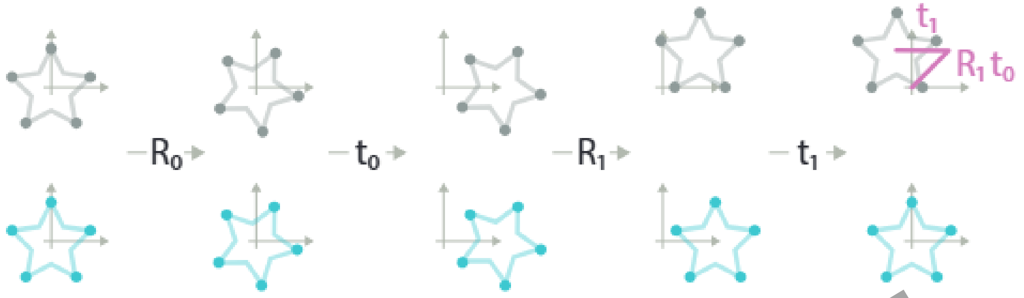
Small step to right

No operation

Small step to left

Point Cloud Registration Agent

Disentangled transformation



Global Transformation

$$\begin{bmatrix} R_1 & t_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_0 & t_0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1 R_0 & R_1 t_0 + t_1 \\ 0 & 1 \end{bmatrix}$$

$$R_1(R_0 X + t_0) + t_1 = R_1 R_0 X + R_1 t_0 + t_1 .$$

The center point of rotation matrix is global origin coordinate. It entangles rotation and translation matrices. It is hard to know global translation whether it is lead by rotation matrix or translation matrix.

Disentangled Transformation

$$X_i = (\prod^{i} R_i) X + \sum^{i} t_i .$$

$$R_i = \hat{R}_i R_{i-1}, \quad t_i = \hat{t}_i + t_{i-1},$$

$$X_i' = R_i (X' - \mu_{X'}) + \mu_{X'} + t_i .$$

We may use rotation matrix by object center point. It improves interpretability how model acts by using object's local rotation.
It means that translation and rotation matrices are separated.

Imitating an Expert Policy

Expert policy

$$Target = T_{gt} \times Source$$

State1 $$Pred_1 = T_{pred_1} \times Source$$

$$Target = \boxed{T_{gt}T^{-1}_{pred_1}} \times Pred_1$$

State2 $$Pred_2 = T_{pred_2} \times Pred_1$$

$$Target = \boxed{T_{gt}T^{-1}_{pred_1}T^{-1}_{pred_2}} \times Pred_2$$

State3 $$Pred_3 = T_{pred_3} \times Pred_2$$

$$Target = \boxed{T_{gt}T^{-1}_{pred_1}T^{-1}_{pred_2}T^{-1}_{pred_3}} \times Pred_3$$

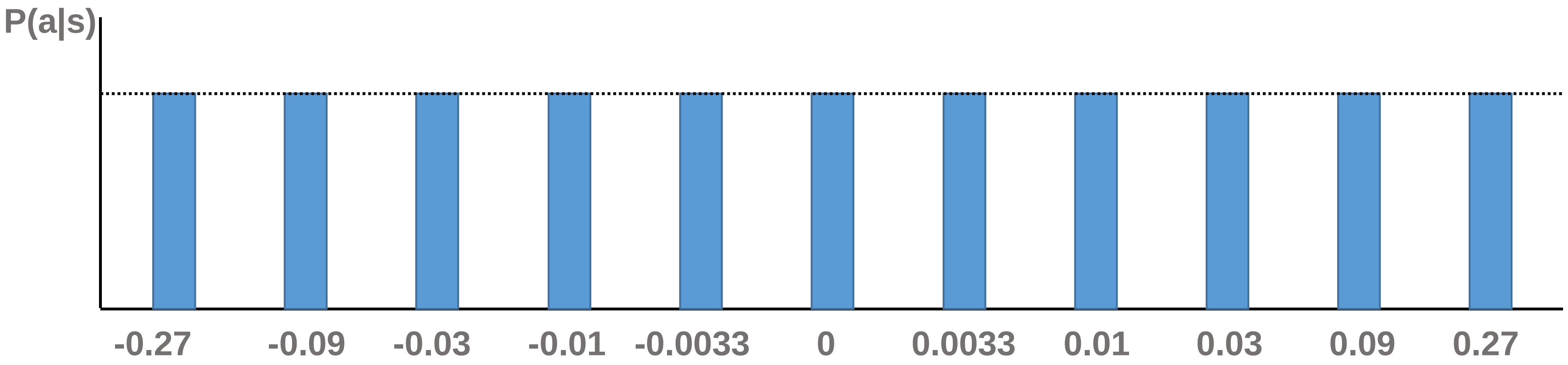
$$\cdots$$

$$\cdots$$

State10 $$Pred_{10} = T_{pred_{10}} \times Pred_9$$

Imitating an Expert Policy

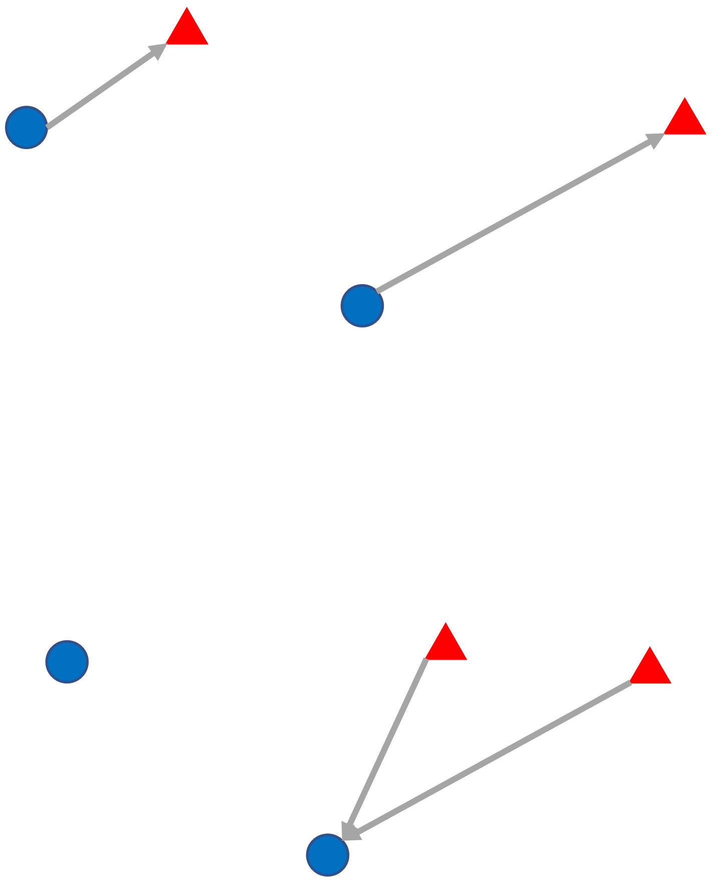
Data gathering



- If expert data of translation x is 0.3, since maximum size of expert policy is 0.27, 0.27 is chosen as gt.
- If expert data of translation x is 0.0001, 0 is chosen as gt.
- If expert data of translation x is -0.09, -0.09 is chosen as gt.

Improving through Reinforcement

Reward function



$$r = \begin{cases} -\varepsilon^-, & CD(X'_i, X) > CD(X'_{i-1}, X) \\ -\varepsilon^0, & CD(X'_i, X) = CD(X'_{i-1}, X) \\ \varepsilon^+, & CD(X'_i, X) < CD(X'_{i-1}, X). \end{cases}$$

$$d_{\mathrm{ch}}(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \|a - b\|^2 + \frac{1}{|B|} \sum_{b \in B} \min_{a \in A} \|b - a\|^2$$

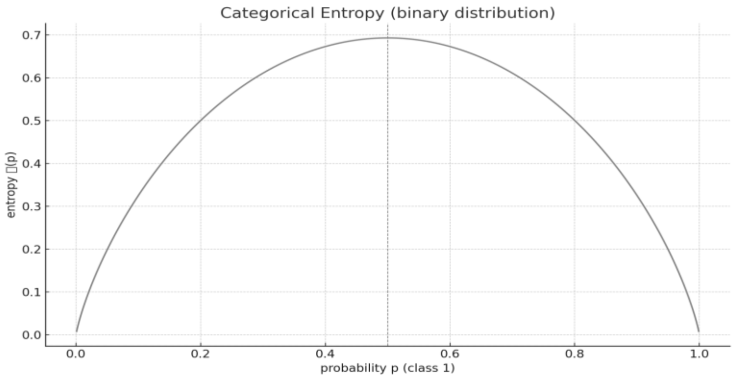
Improving through Reinforcement

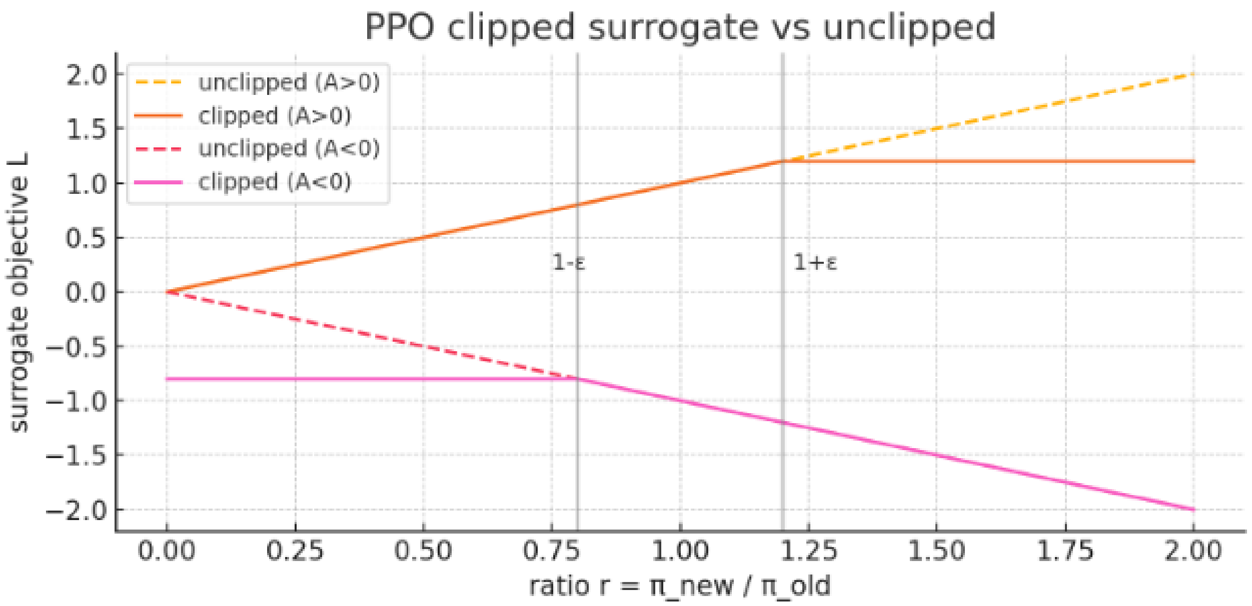
PPO $$\underbrace{-L^{\text{clip}}}_{\text{policy}} + \; c_v \underbrace{\tfrac{1}{2}(V_\theta(s) - G_t)^2}_{\text{value}} - \; c_e \underbrace{\mathcal{H}[\pi_\theta]}_{\text{entropy}}$$

Policy Gradient : If the action is good behavior, log probability of action is encouraged.

Value loss : High advantage means that value function doesn't predict value well. It is encouraged to reduce advantage to predict value precisely.

Entropy loss : Low probability is encouraged to be increased and High probability is encouraged to be reduced. Exploration is maximize.





PPO Clip

- **If action is good and ratio is over 1+E, surrogate objective is limited not to be increased for training stability.**
- **If action is good and ratio is under 1-E, loss is encouraged to be increased.**

Improving through Reinforcement

Algorithm 1 Combined Imitation and Reinforcement Learning using a Replay Buffer

for all observations O in $\mathcal{O}$ **do**
  % Gather replay buffer
  for N trajectories **do**
    for n refinement steps **do**
      agent predicts policy $\pi(O)$ and value $\hat{v}$
      action a is sampled from policy $\pi(O)$
      take action a, receive reward r and next O'
      add sample to buffer b, step observation $O = O'$
    end for
  end for
  % Process replay buffer
  compute return R, shuffle buffer b
  for all samples in buffer b **do**
    agent predicts new policy $\pi'(O)$ and value $\hat{v}'$
    % Imitate expert
    expert predicts action a^*
    compute cross-entropy loss l_{IL} of $\pi'(O)$ and a^*
    % Reinforce
    compute PPO loss l_{RL} of $\pi'(O)$ and $\pi(O)$
    % Update agent
    $l = l_{IL} + l_{RL} \cdot \alpha$
    backpropagate combined loss l
  end for
  clear buffer b
end for

Experiment

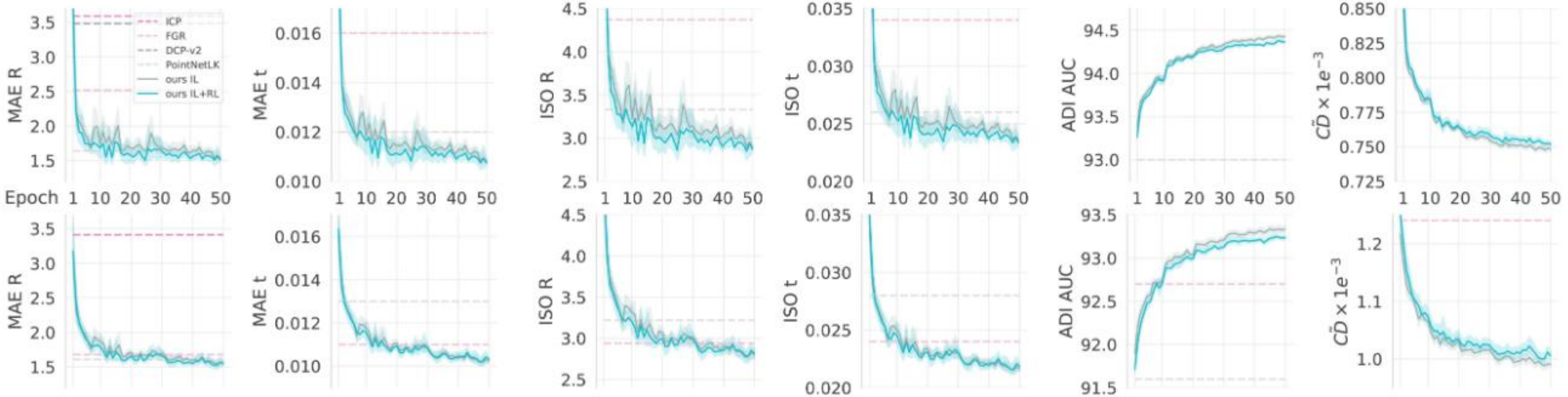


Fig. 5: Convergence of ReAgent with 10 random seeds on held-out models (top) and categories (bottom) of ModelNet40. The lines show the mean and the shaded areas indicate the 95%-confidence intervals. Best viewed digitally.

Experiment

	Segmented Objects						T (↓)
	MAE (↓)		ISO (↓)		ADI (↑)	$\tilde{CD}$ (↓)	
	R	t	R	t	AUC	$\times 1e^{-3}$	[ms]
ICP	5.34	0.036	10.47	0.076	88.1	2.99	**19**
FGR$^+$	**0.11**	**0.001**	**0.19**	**0.001**	**99.7**	**0.16**	131
DCP-v2	7.42	0.050	14.93	0.102	72.4	4.93	54
PointNetLK	0.90	0.010	1.74	0.020	92.5	1.09	45
ours IL	0.77	0.006	1.33	0.012	95.7	0.30	21
ours IL+RL	0.93	0.007	1.66	0.014	95.4	0.34	

TABLE IV: Results on ScanObjectNN with the object segmented from the observation. Learning-based methods use the model trained on ModelNet40. Note that ↓ indicates that smaller values are better. Runtimes are for a single registration and 2048 points per cloud. $^+$ indicates that FGR additionally uses normals.

Thank you