模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

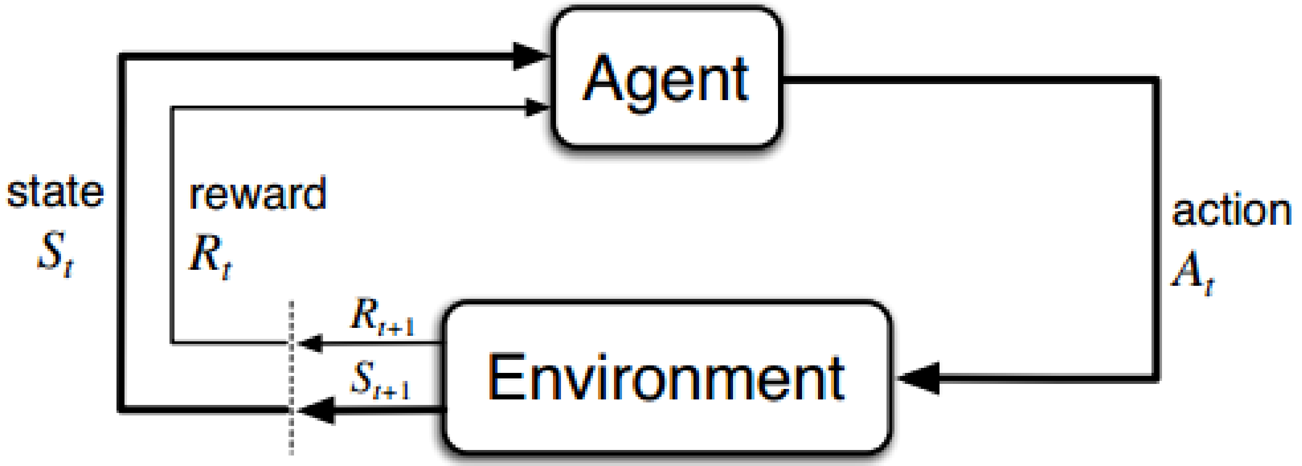
ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

Parrot: Data-Driven Behavioral Priors for Reinforcement Learning

Avi Singh*, Huihan Liu*, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, Sergey Levine
University of California, Berkeley

ICLR 2021

Background



$$MDP: \mathcal{M} = (S, A, \rho, P, r, \gamma)$$

sample inefficiency: agent needs millions of interactions with the environment to be well trained

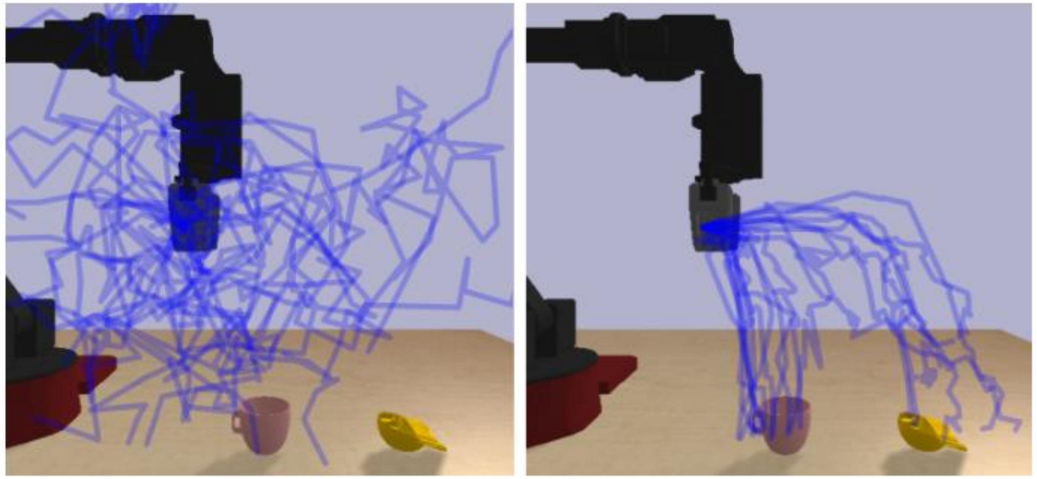
Motivation

In other machine learning fields, such as NLP or CV, pre-training on large, previously collected datasets to bootstrap learning for new tasks has emerged as a powerful paradigm to reduce data requirements when learning a new task.

try similarly useful pre-training for RL agents

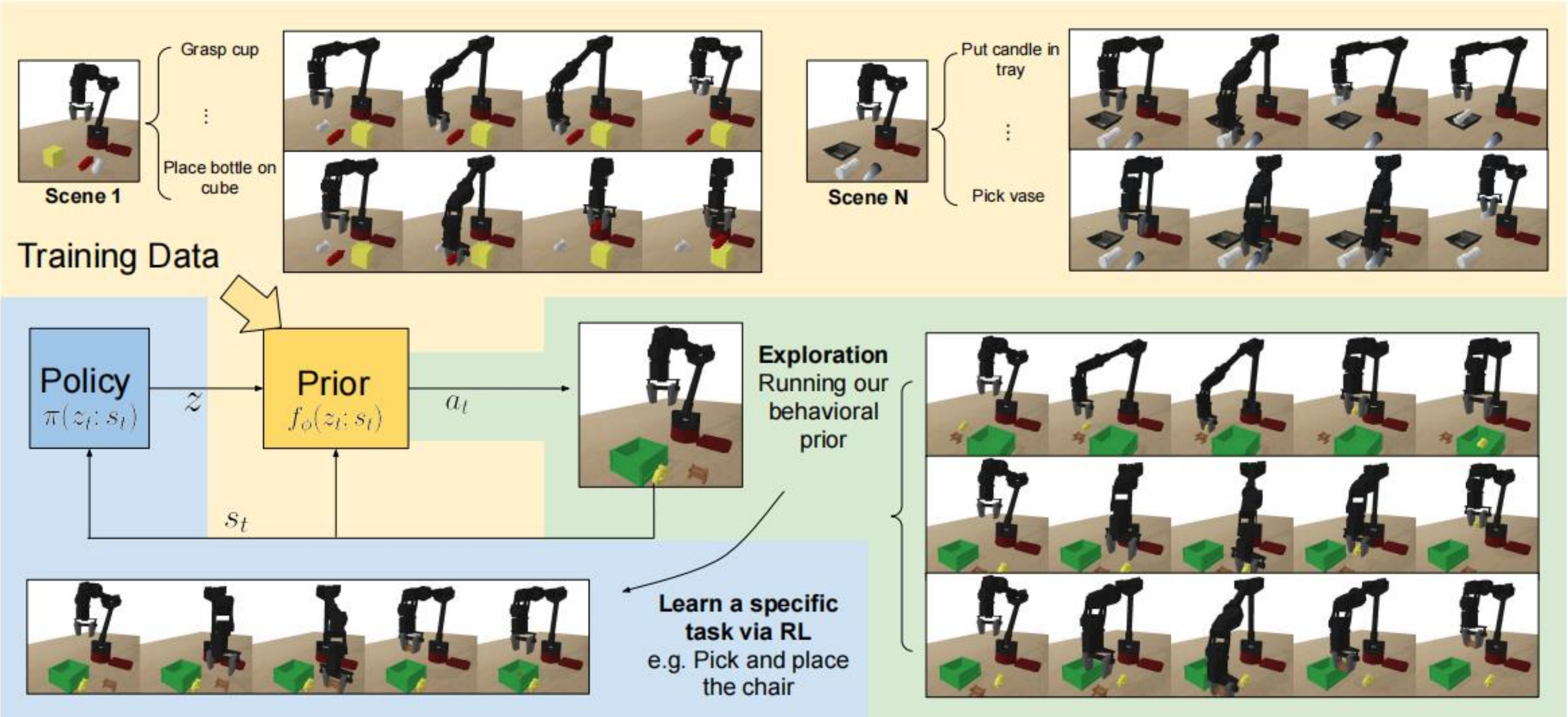
Benefit:

- reduce interactions and make a faster trainning process
- applicable to similar tasks



Without Behavioral Prior With Behavioral Prior

Method



The main idea: utilize manipulation data from a diverse range of prior tasks to train our behavioral prior, and then use it to bootstrap exploration for new tasks.

Convert $\mathcal{M} = (S, A, \rho, P, r, \gamma)$ to $\mathcal{M} = (S, Z, \rho, P, r, \gamma)$, then action a is obtained from s and z

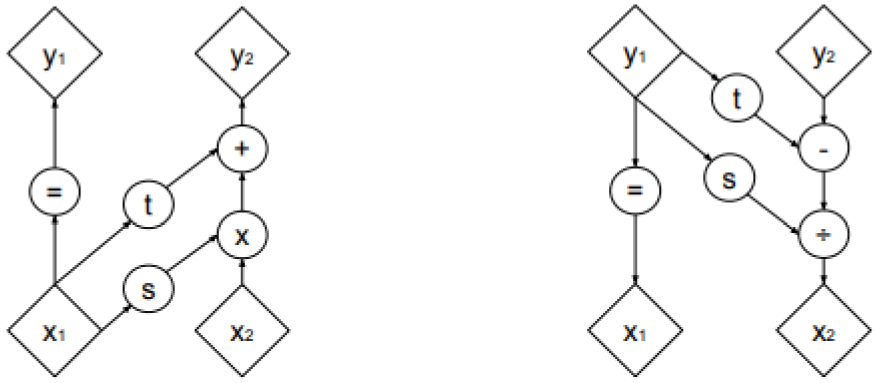
Method

Real NVP

$$\begin{cases} y_{1:d} & = x_{1:d} \\ y_{d+1:D} & = x_{d+1:D} \odot \exp\left(s(x_{1:d})\right) + t(x_{1:d}) \end{cases}$$

$$\Leftrightarrow \begin{cases} x_{1:d} & = y_{1:d} \\ x_{d+1:D} & = \left(y_{d+1:D} - t(y_{1:d})\right) \odot \exp\left(-s(y_{1:d})\right) \end{cases}$$

$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left[s\left(x_{1:d}\right)\right]\right) \end{bmatrix}$$



(a) Forward propagation (b) Inverse propagation

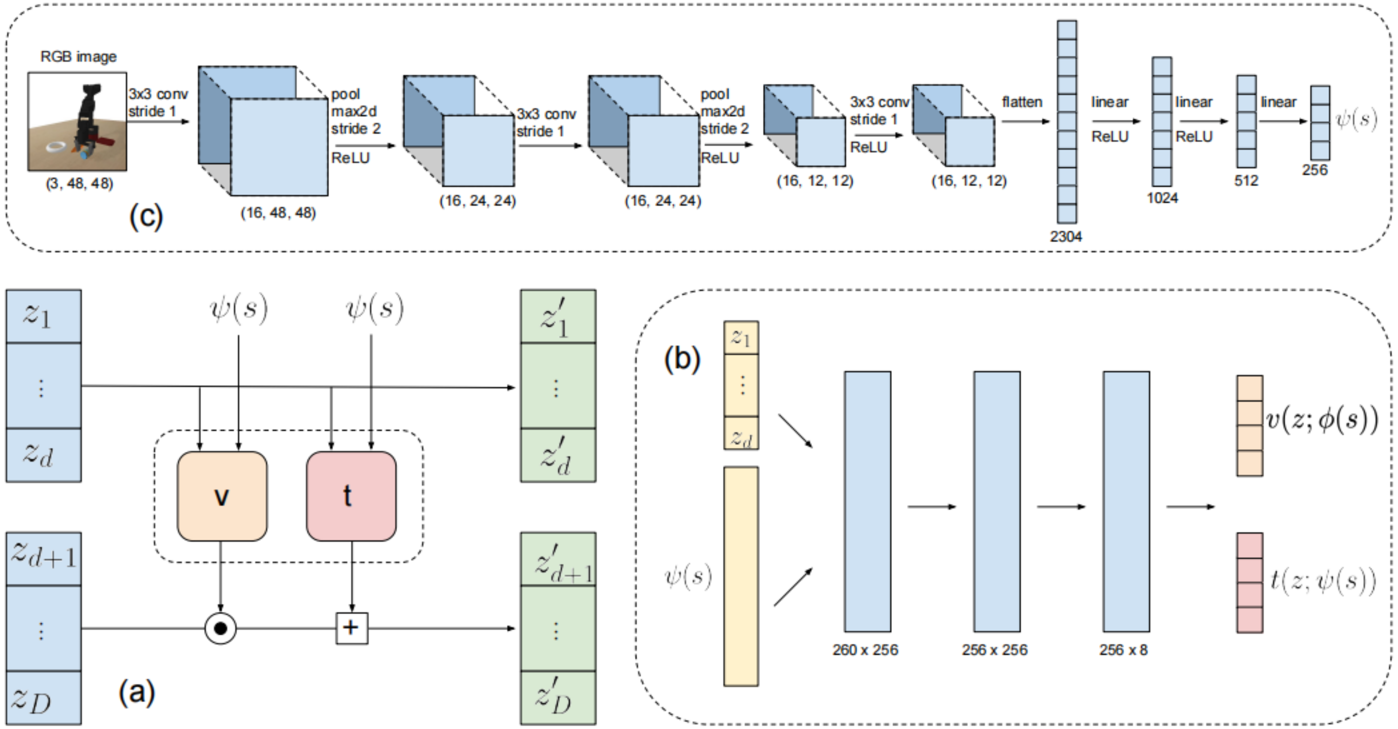
enable an invertible mapping,which makes the RL agent still retains full control over the action space of the original MDP

$$a = f_\phi(z; s)$$

$$p_{\text{prior}}(a|s) = p_z\left(f_\phi^{-1}(a;s)\right)\left|\det\left(\partial f_\phi^{-1}(a;s)/\partial a\right)\right|$$

maximize the likelihood term to learn behavioral prior

Method

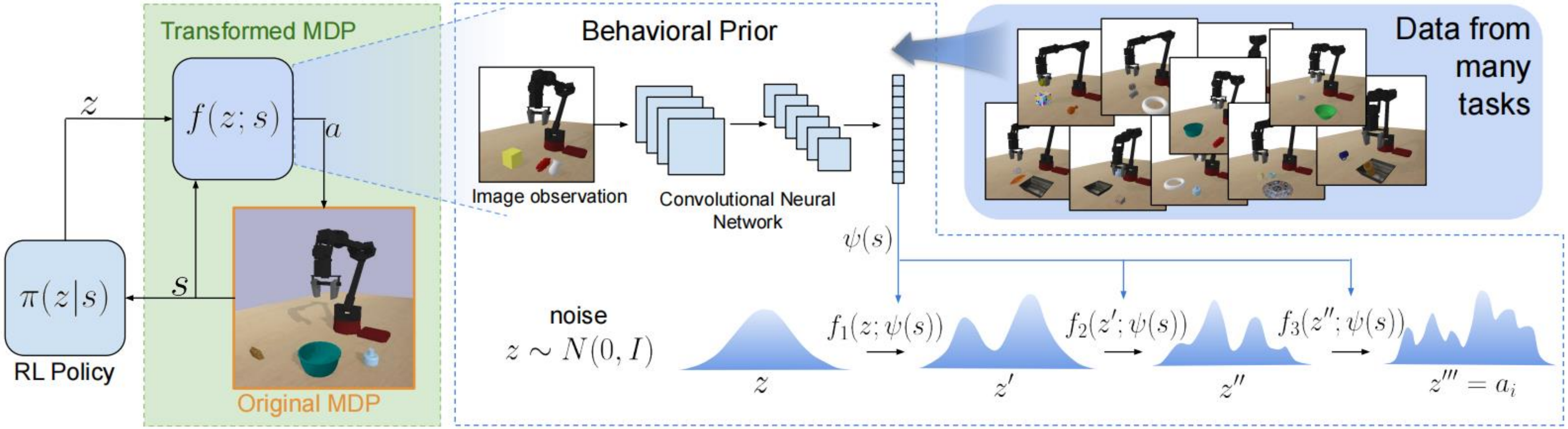


$$z'_{1:d} = z_{1:d}$$

$$z'_{d+1:D} = z_{d+1:D} \odot \exp(v(z_{1:d};\phi(s))) + t(z_{1:d};\phi(s))$$

The v , t and ψ are functions implemented using neural networks

Method



For a new task,initialize the RL policy to the base distribution used for training the prior, so that at the beginning of training, $\pi_\theta(z|s) = p_z(z)$

Update $\pi_\theta(z|s)$ with (s, z, s', r)

Method

模式识别与神经计算研究组
PAttern Recognition and NEural Computing

Algorithm 1 RL with Behavioral Priors

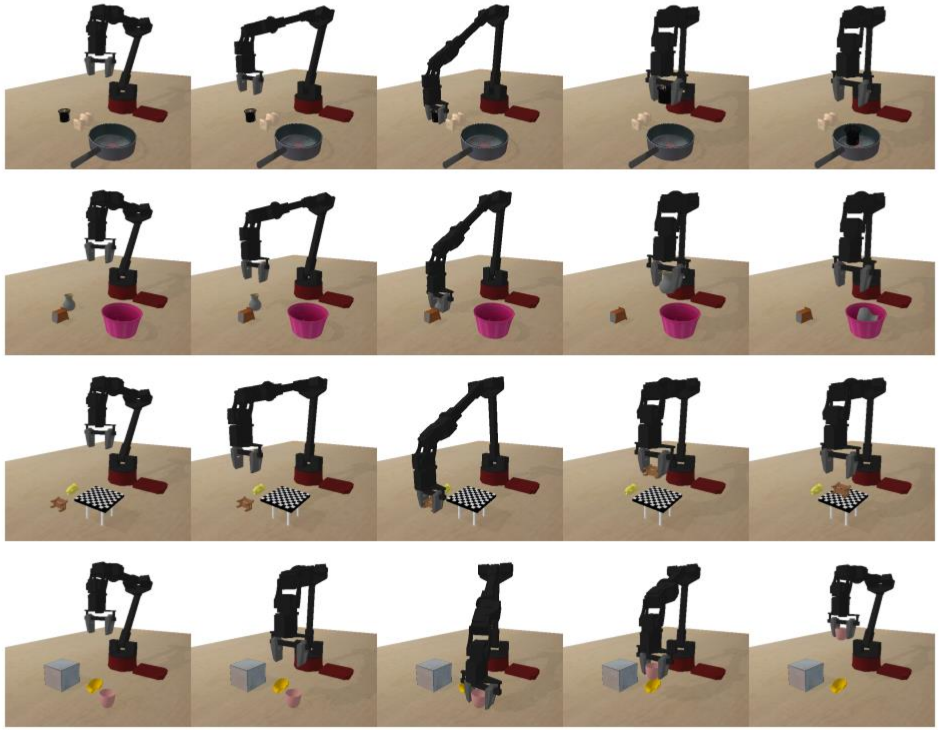
1: **Input**: Dataset $\mathcal{D}$ of state-action pairs (s, a) from previous tasks, new task $M^\star$
2: Learn f_ϕ by maximizing the likelihood term in Equation 2
3: **for** step k in $\{1, \ldots, \mathrm{N}\}$ **do**
4: $\quad s \leftarrow$ current observation
5: $\quad$ Sample $z \sim \pi_\theta(z|s)$
6: $\quad a \leftarrow f_\phi(z; s)$
7: $\quad s', r \leftarrow$ Execute a in $M^\star$
8: $\quad$ Update $\pi_\theta(z|s)$ with (s, z, s', r)
9: **end for**
10: **Return**: Policy $\pi_\theta(z|s)$ for task $M^\star$.

$$p_{\text{prior}}(a|s) = p_z\left(f_\phi^{-1}(a; s)\right) \left|\det\left(\partial f_\phi^{-1}(a;s)/\partial a\right)\right|$$

Experiment

Method	difference
Combining RL with demonstrations	Requires collecting demonstrations for the specific task
Generative modeling and RL	Non-invertible, lack of the ability of retaining full control over the action space
Hierarchical learning	focuses on modeling the temporal structure
Meta-learning/meta-IL	Need to interact with the prior tasks, with access to rewards and additional samples

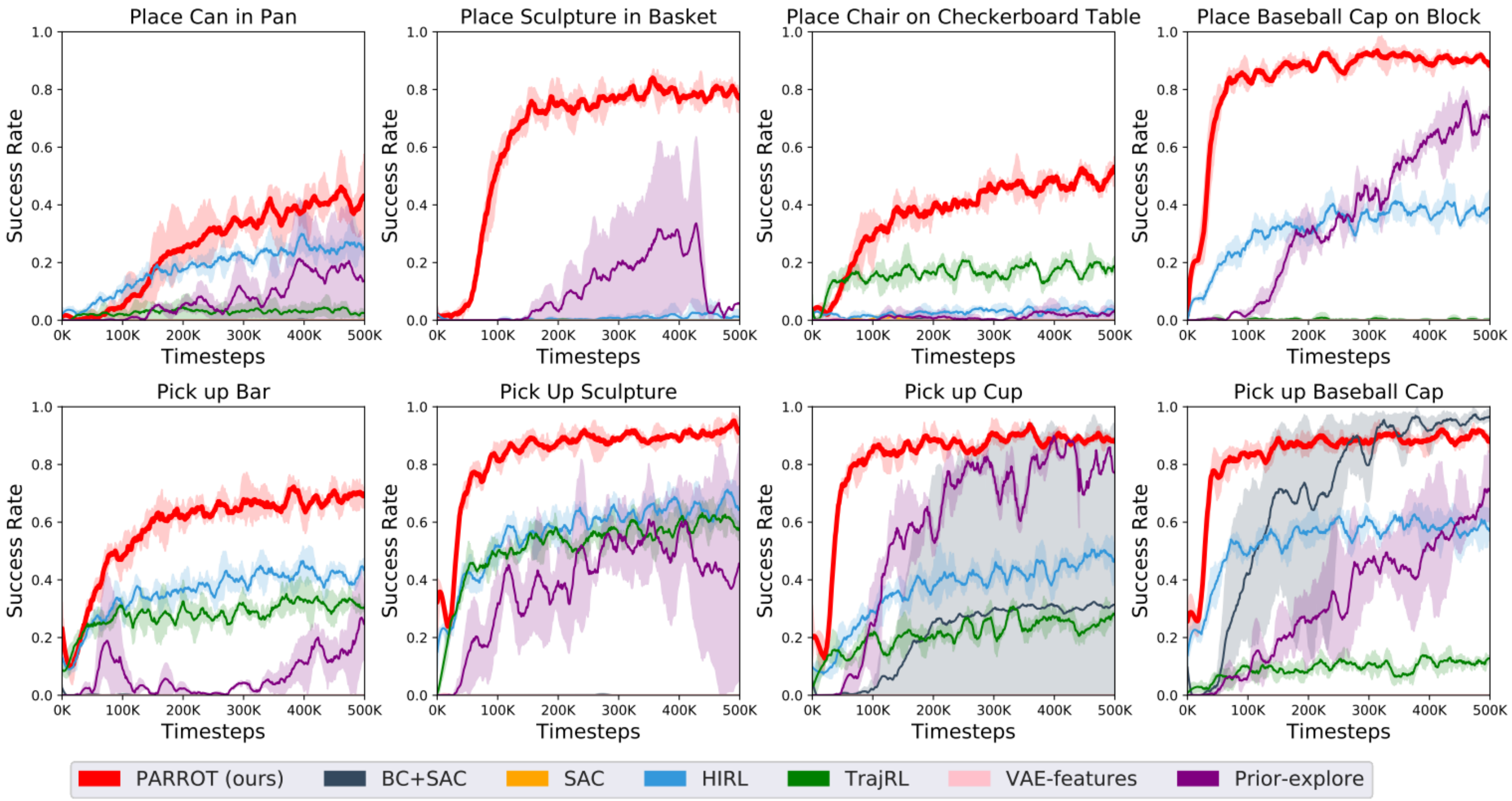
Experiment



Problem setting:

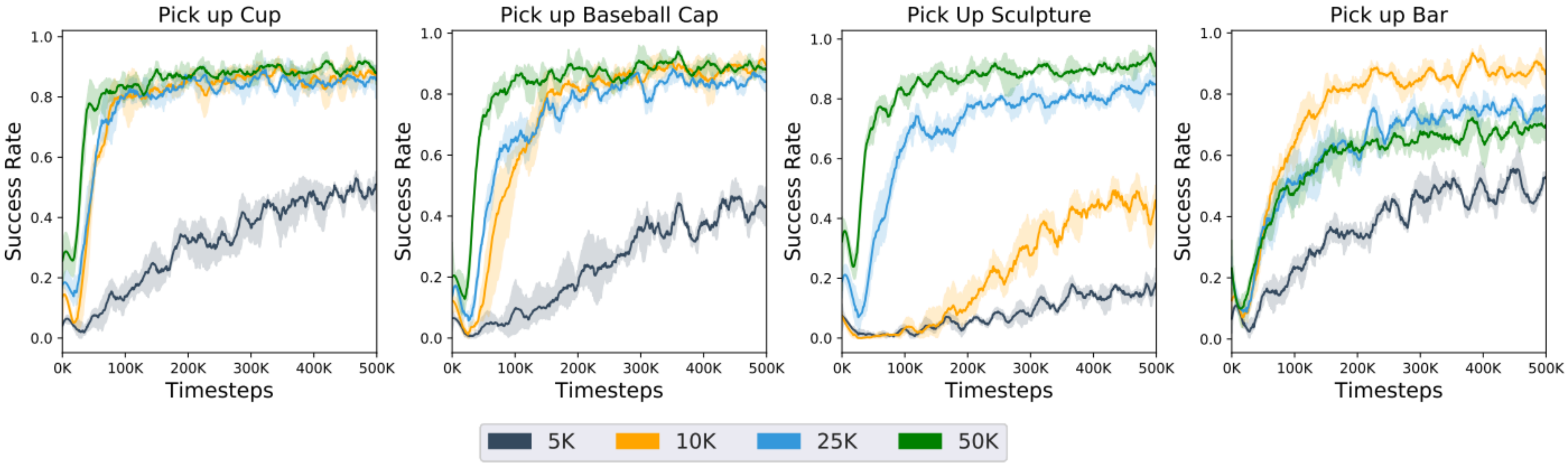
- Initial positions of all objects are randomized, and must be inferred from visual observations
- Not all objects in the scene are relevant to the current task
- A reward of +1 is provided when the objective for the task is achieved, and the reward is zero otherwise.

Experiment



Prior-explore: While collecting data, an action is executed from the prior with probability ϵ , else an action is executed from the learned policy

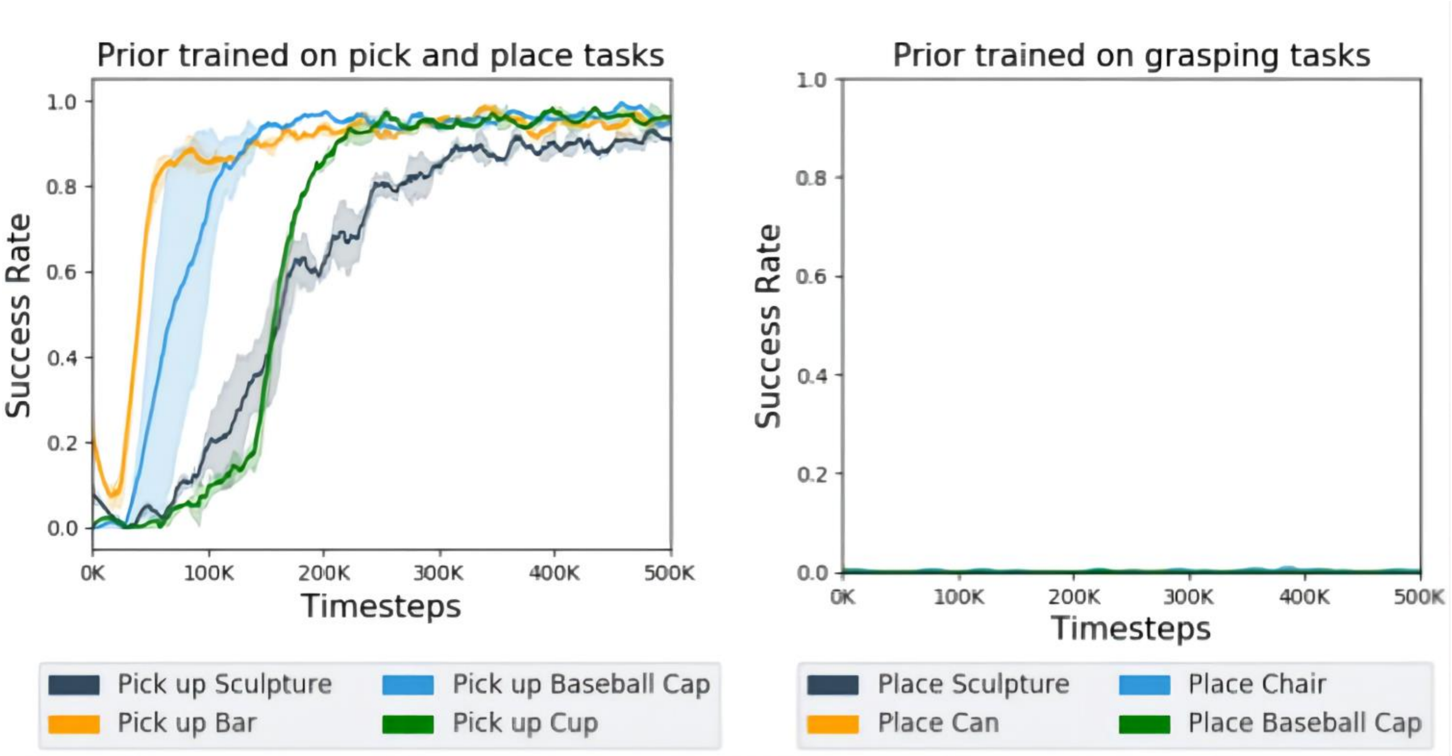
Experiment



- The size of the dataset positively correlates with performance, but about 10K trajectories are sufficient for obtaining good performance, and collecting additional data yields diminishing returns

- Note that initializing with even a smaller dataset size (like 5K trajectories) yields much better performance than learning from scratch

Experiment



The authors suspect this is due to the fact that pick and place tasks involve a completely new action (that of opening the gripper), which is never observed by the prior if it is trained only on grasping

Thanks