PARROT: DATA-DRIVEN BEHAVIORAL PRIORS FOR REINFORCEMENT LEARNING

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

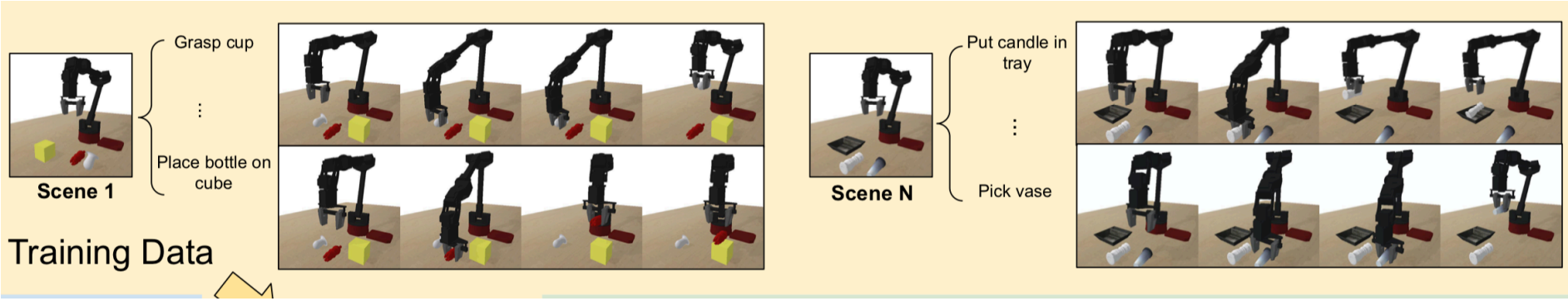
Problem Setting

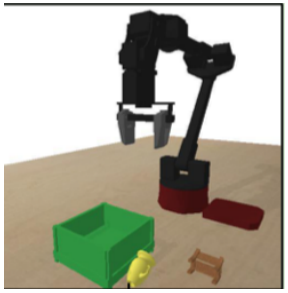
We have:

- (near optimal) trajectories from many tasks.
- a new task

We want to:

- help RL agent learn faster in the new task.





Possible solution of this problem

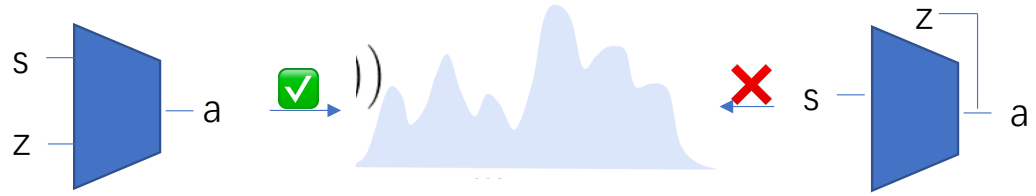
- Meta-RL
  - Agent can only interact with the new task, while meta-RL needs to interact with a set of training tasks.

- Meta-IL (meta-imitation learning)
  - meta-IL needs expert demonstration in new tasks.

Solution----learn an action prior from the dataset

- Learn a generative model from the dataset
  - We have (s, a) pairs from other tasks. We can model the dataset by a generative model $p_{\text{prior}}(a|s)$, which could help explore in the new task.
  - A distribution can be written as a deterministic function with a noise as input:
    - $a = f_\phi(z; s) \sim p_{\text{prior}}(a|s)$, where z is a unit Gaussian random variable.

Possible formulation



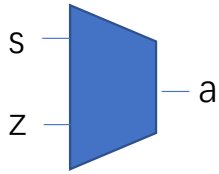
Like a conditional GAN

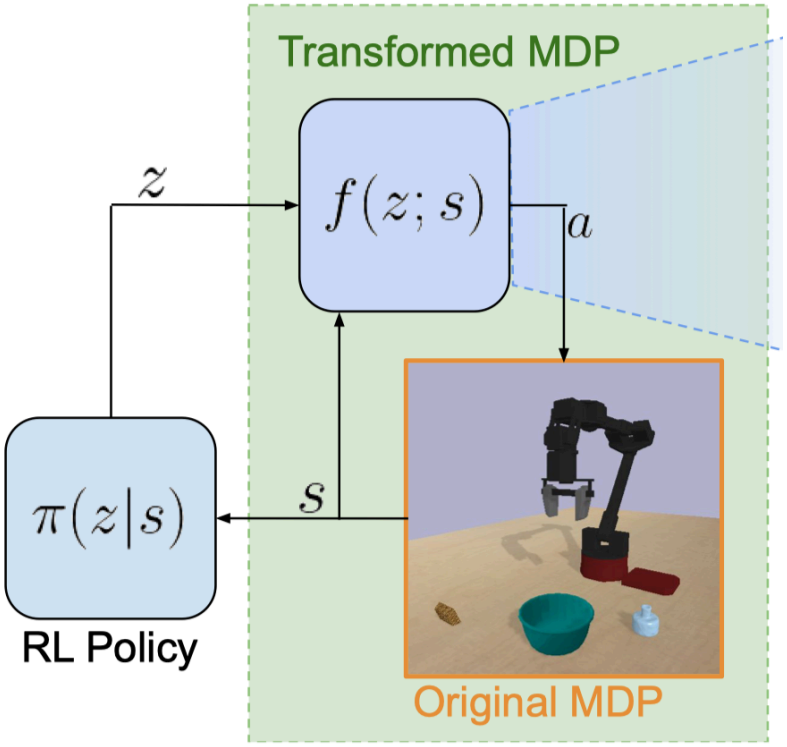
More impressive, especially in the multi-modal case

Gaussian policy we always use

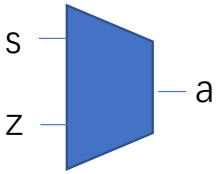
Easier to optimize; the probability p(a|s) is tractable

Solution----How to use the prior





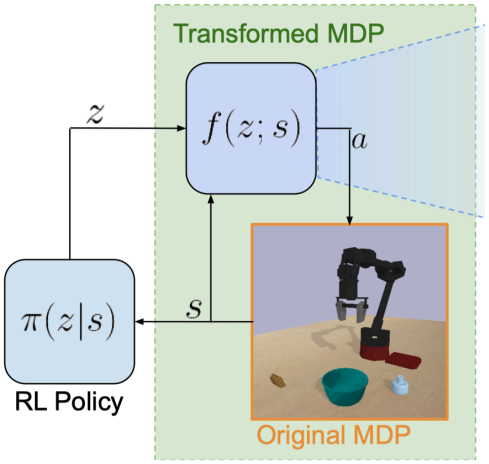
Solution----How to learn the prior



- Maximize likelihood estimation (MLE): max log_prob
  - Given a pair (s, a), how to calculate p(a|s)?
  - Recall:
    - if we know p(x), and y = f(x), what p(y) is?

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

    - f should be invertible, thus, f should be monotonous w.r.t. x
- Adversarial training: introducing a discriminator
  - the agent cannot retain full control over the action space
    - action space in such a way will be restricted to the dataset.
- We want the prior model to:
  - be capable of representing complex, multi-modal distributions ✅
  - state-conditioned ✅
  - provide a mapping for generating "useful" actions from noise samples when learning a new task ✅
  - allow easier learning in the reparameterized action space without hindering the RL agent's ability to attempt novel behaviors ❌



Solution: Using non-volume preserving (NVP) to construct invertible networks.

Non-volume Preserving (NVP)

x, y are D-dimensional vectors. s, t are functions from $R^d \mapsto R^{D-d}$

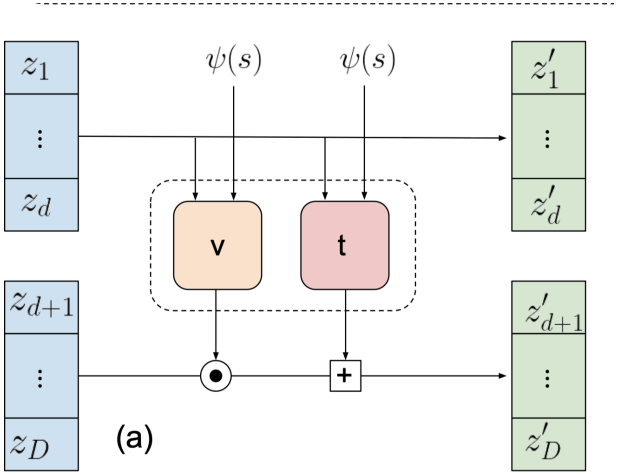
NVP

$$y_{1:d} = x_{1:d}$$

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

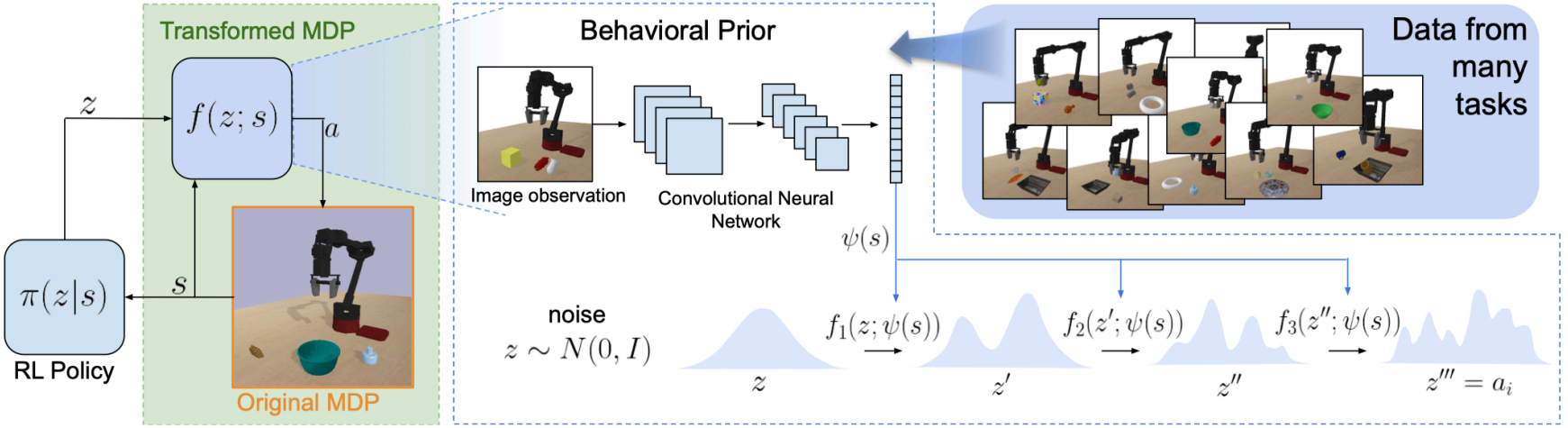
$$\begin{cases} y_{1:d} & = x_{1:d} \\ y_{d+1:D} & = x_{d+1:D} \odot \exp\left(s(x_{1:d})\right) + t(x_{1:d}) \end{cases}$$

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



Invertible Now !!

Method



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

Experiments----trajectories during exploration

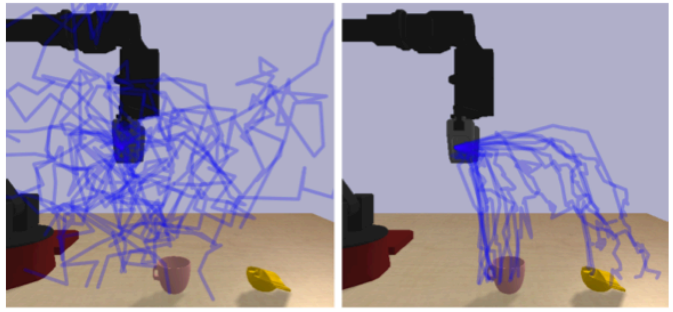


Figure 4: We plot trajectories from executing a random policy, with and without the behavioral prior. We see that the behavioral prior substantially increases the likelihood of executing an action that is likely to lead to a meaningful interaction with an object, while still exploring a diverse set of actions.

Experiment----Performance

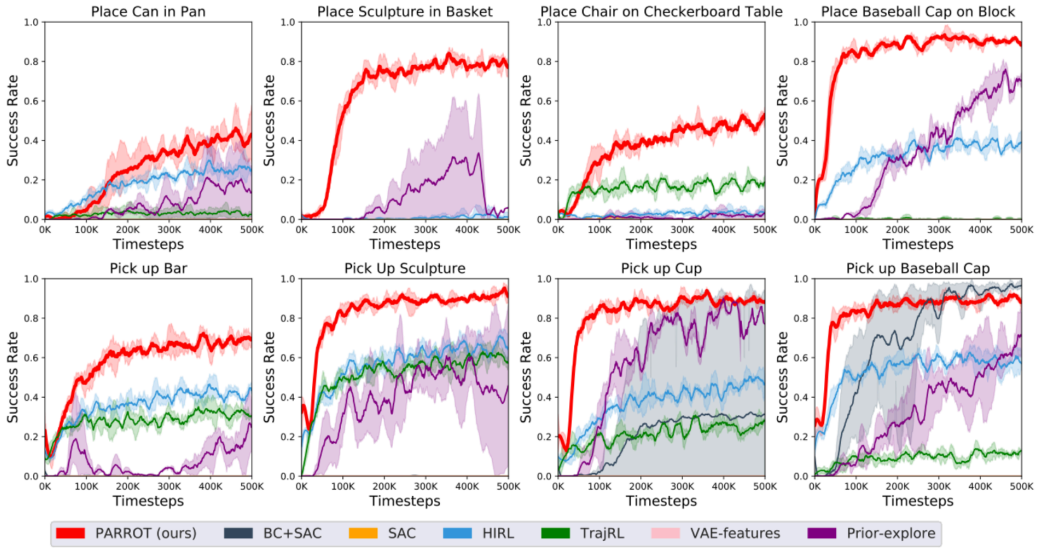


Figure 5: **Results.** The lines represent average performance across multiple random seeds, and the shaded areas represent the standard deviation. PARROT is able to learn much faster than prior methods on a majority of the tasks, and shows little variance across runs (all experiments were run with three random seeds, computational constraints of image-based RL make it difficult to run more seeds). Note that some methods that failed to make any progress on certain tasks (such as "Place Sculpture in Basket") overlap each other with a success rate of zero. SAC and VAE-features fail to make progress on any of the tasks.

Experiment----Sensitivity w.r.t. dataset size

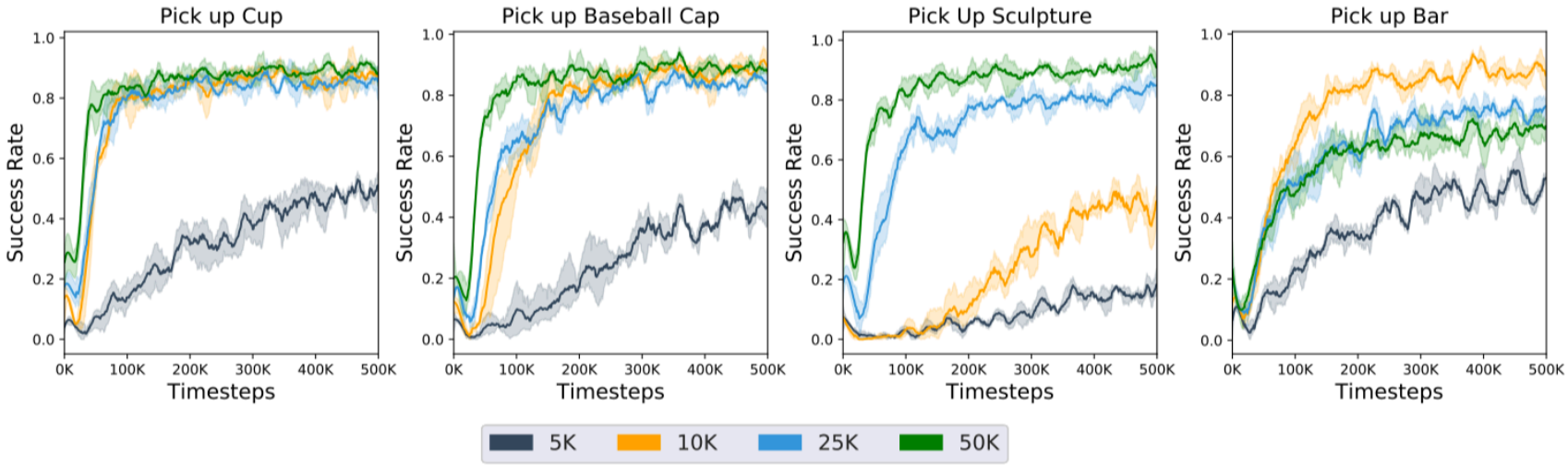


Figure 6: **Impact of dataset size on performance.** We observe that training on 10K, 25K or 50K trajectories yields similar performance.

Experiment---- Impact of train/test mismatch on performance

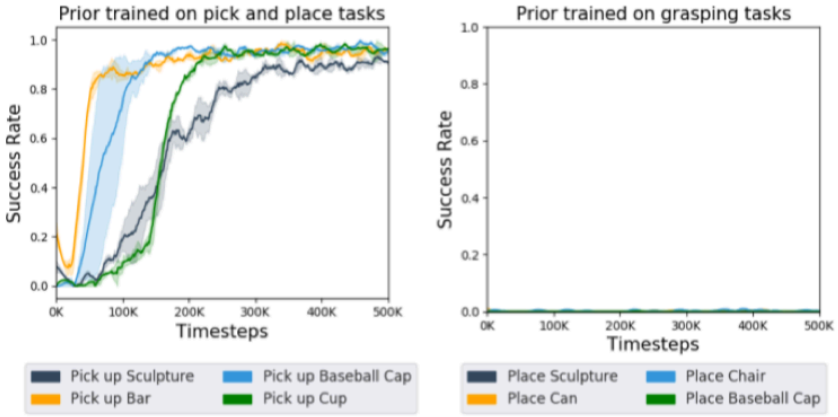


Figure 7: **Impact of train/test mismatch on performance.** Each plot shows results for four tasks. Note that for the pick and place tasks, the performance is close to zero, and the curves mostly overlap each other on the x-axis.