OpenAI

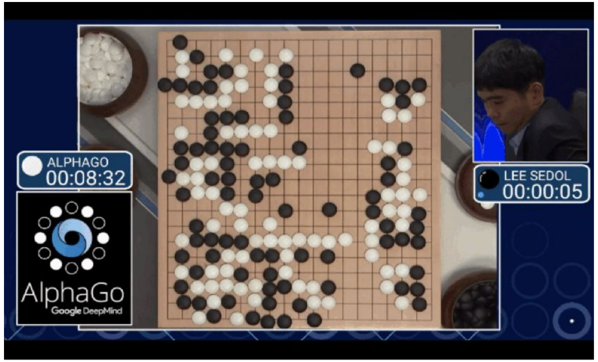
Asymmetric self-play for automatic goal discovery in robotic manipulation

lilian@openai, Apr 9 2021

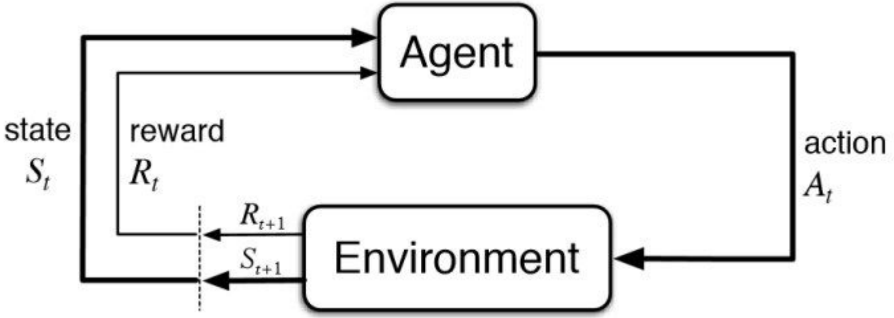
OpenAI

OpenAI

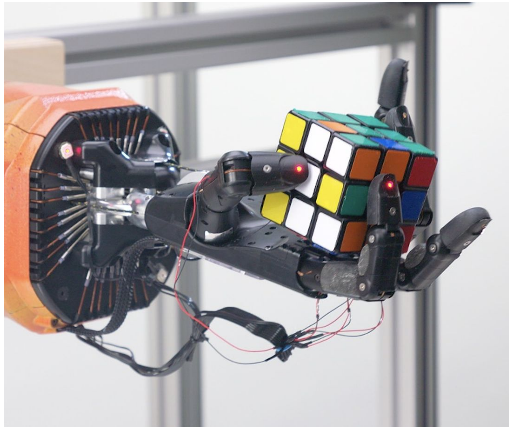
Reinforcement Learning Basics



Reinforcement Learning is **powerful**, but training needs **a lot of data**.

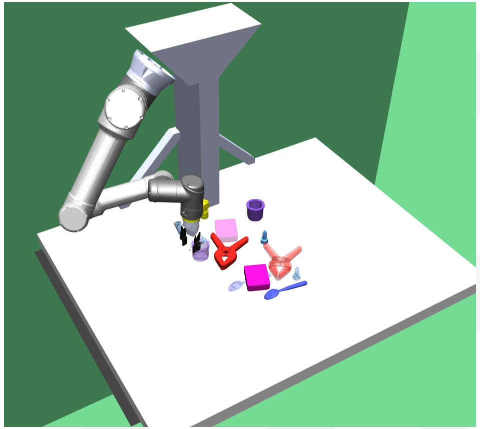


Robotic Manipulation Tasks



Solving rubik's cube with robot hand

The same RL control policy trained only in simulation can work in the real physical robot.

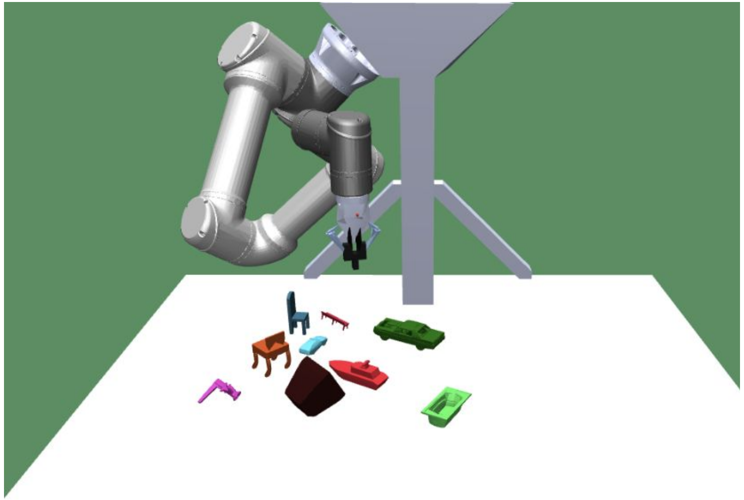


Object rearrangement on the tabletop

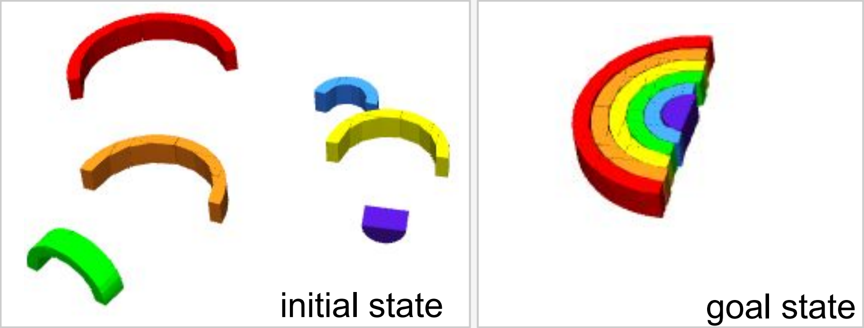
A single goal-conditioned policy can solve many manipulation tasks involving unseen arrangement and unseen objects.

Motivation

- Training a **single** goal-conditioned policy
- Solving **any** robotic manipulation task in an environment



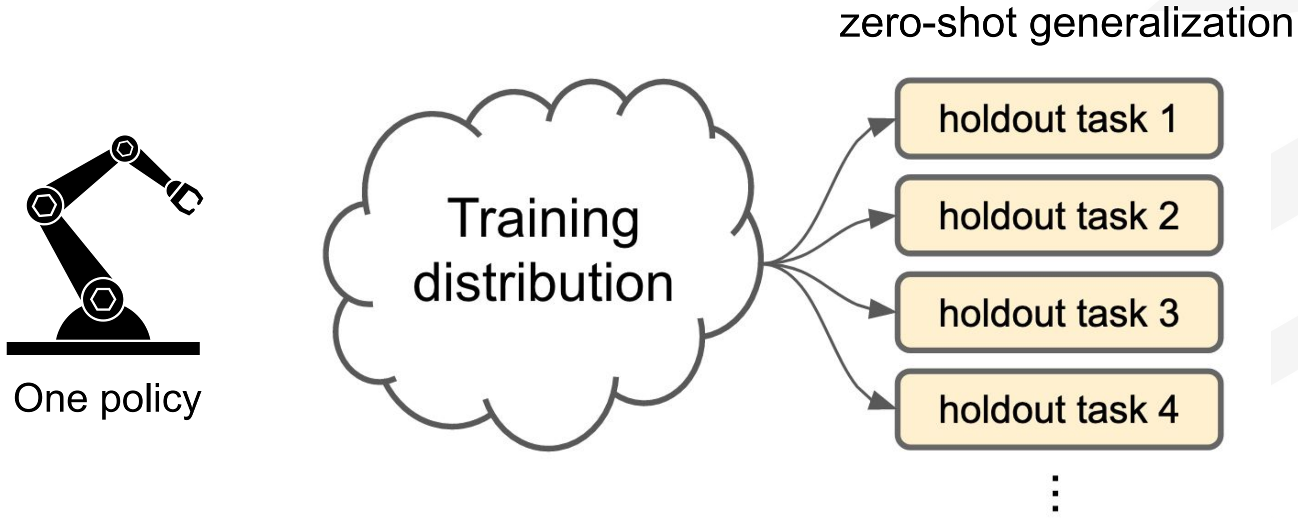
Robotic manipulation environment:
one UR robot + gripper + table surface



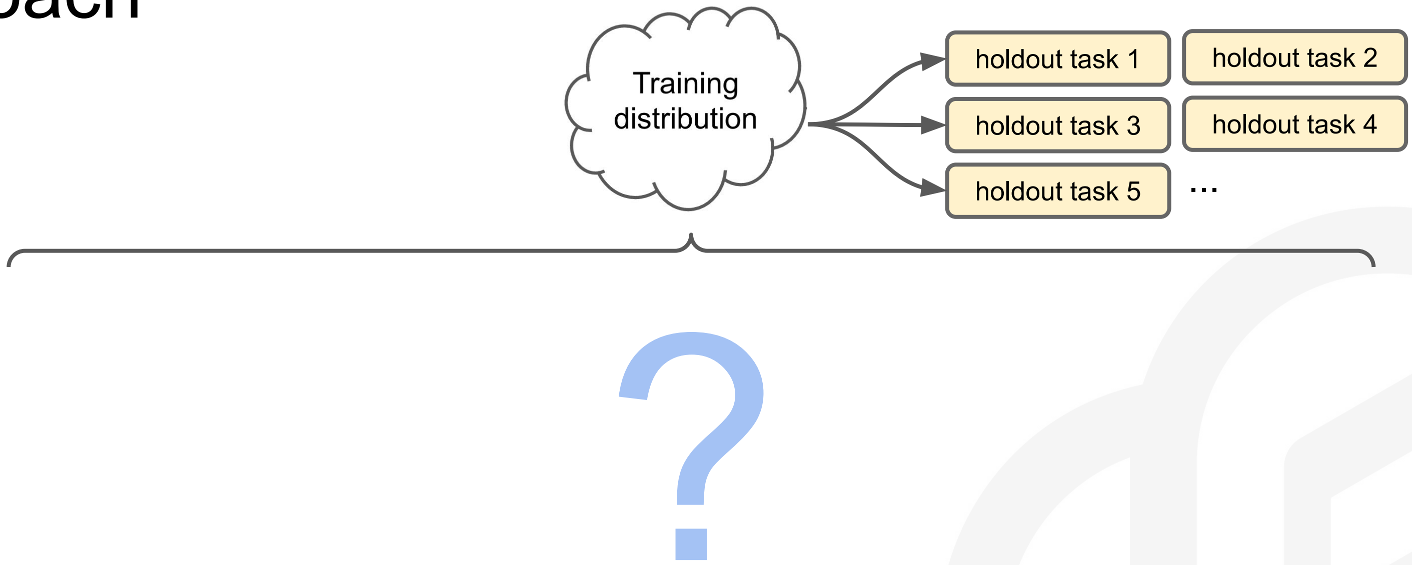
Task: Initial state → Goal state

Approach

- Goal: One policy for all tasks
  - Training on a large training distribution (initial + goal states)
  - Testing on unseen holdout tasks



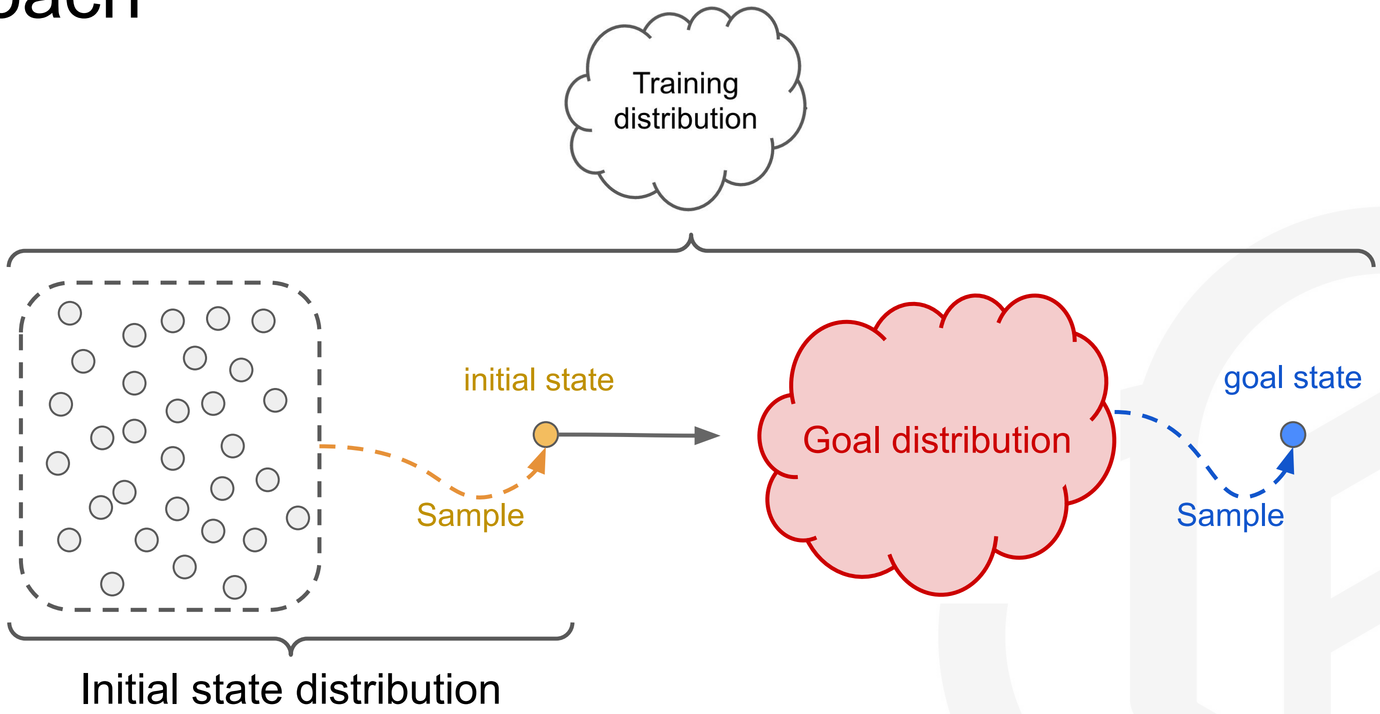
Approach



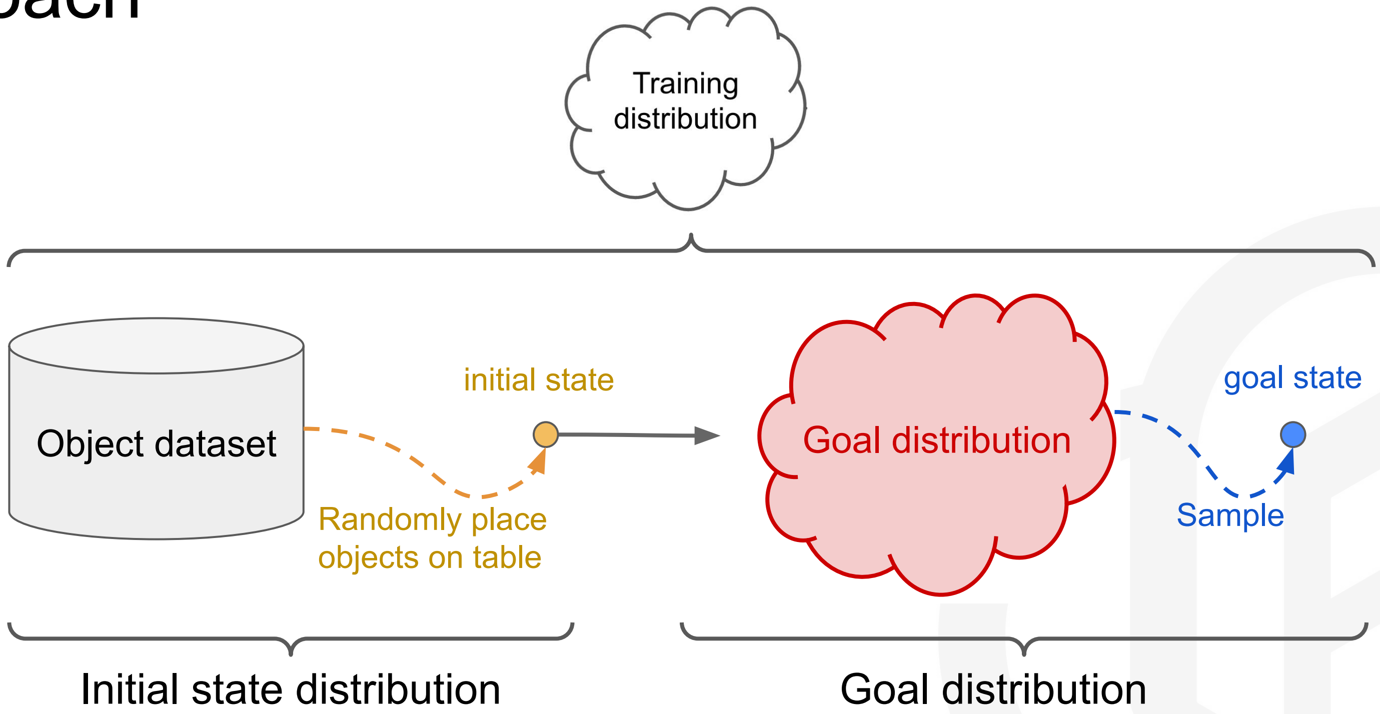
Training distribution
holdout task 1
holdout task 2
holdout task 3
holdout task 4
holdout task 5
...
?

OpenAI

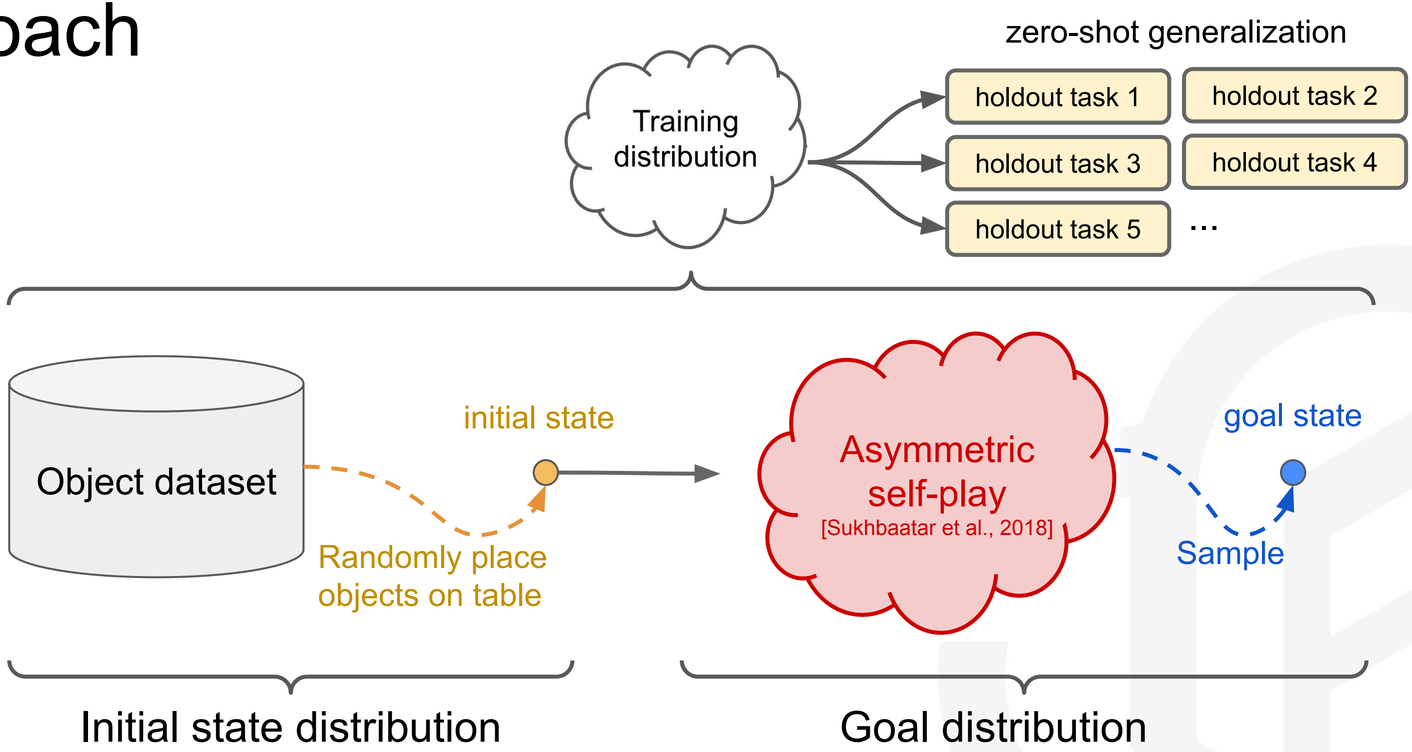
Approach



Approach

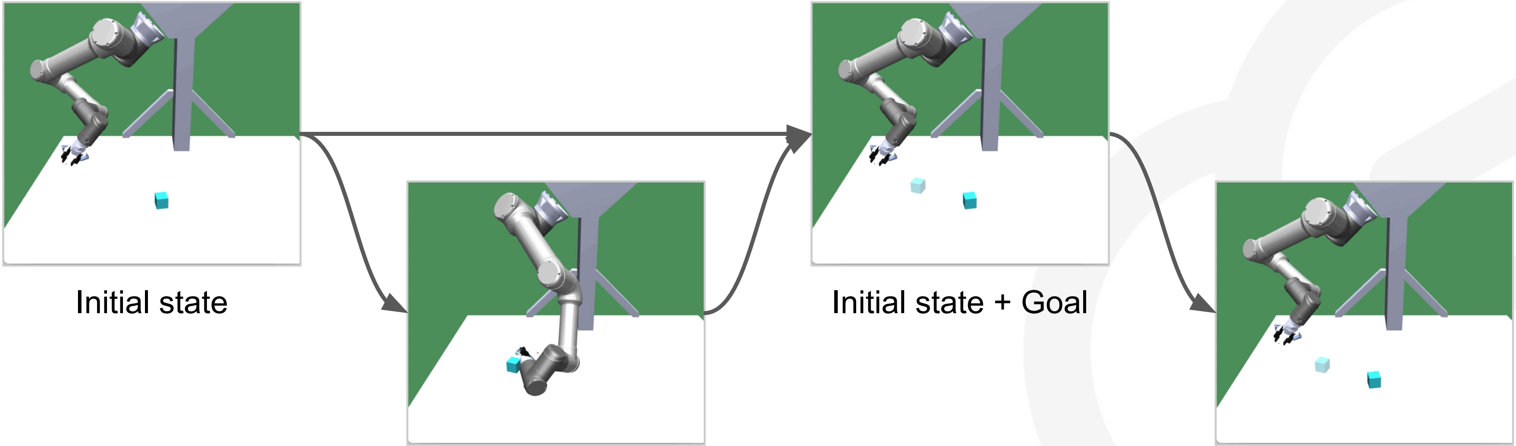

Training
distribution
Object dataset
initial state
Randomly place
objects on table
Goal distribution
goal state
Sample
Initial state distribution
Goal distribution

Approach


zero-shot generalization
Training distribution
holdout task 1
holdout task 2
holdout task 3
holdout task 4
holdout task 5
...
Object dataset
initial state
Randomly place objects on table
Asymmetric self-play
[Sukhbaatar et al., 2018]
goal state
Sample
Initial state distribution
Goal distribution

Asymmetric Self-play for Robotics Manipulation

- Learning to generate goals + Learning to solve them:
    - Train two policies (Alice, Bob) for the same robotic hardware



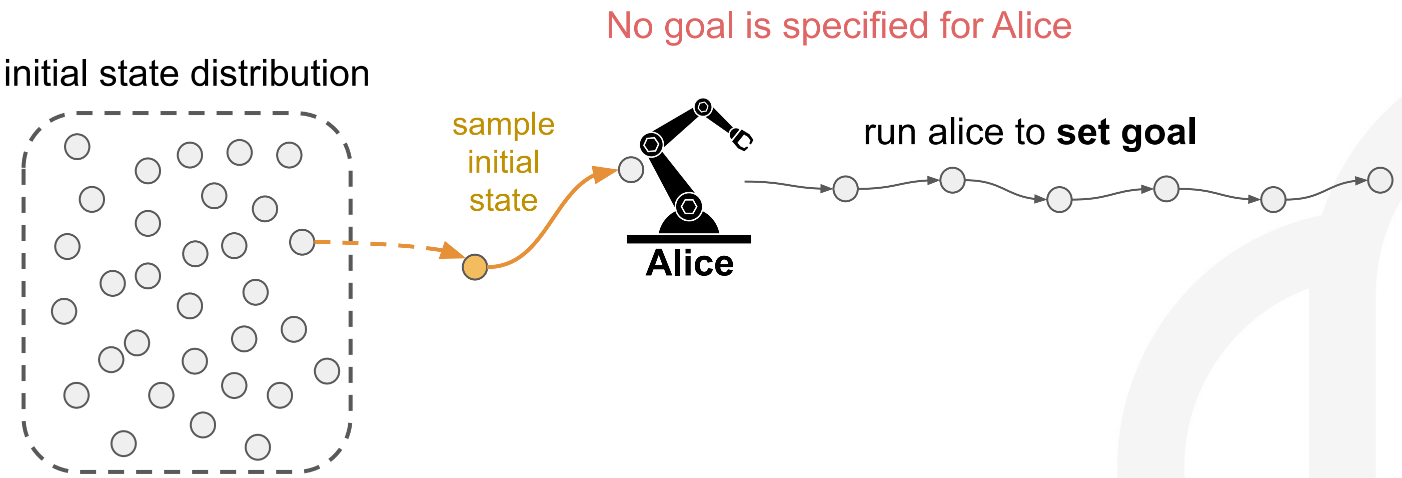
Initial state

Alice: Goal generation

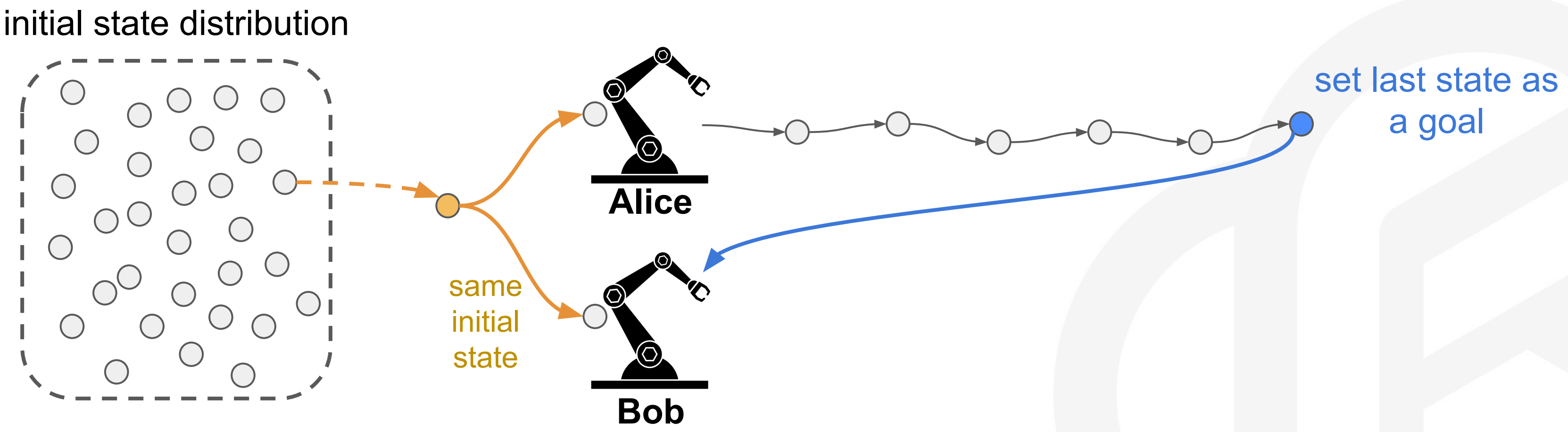
Initial state + Goal

Bob: Goal solving

Asymmetric Self-play for Robotics Manipulation

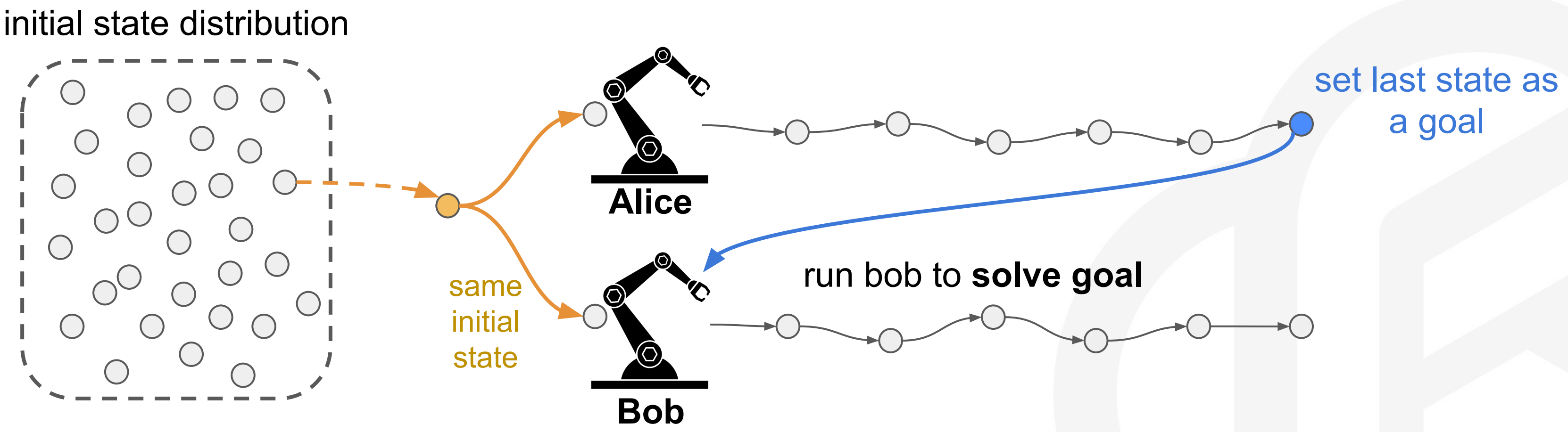


Asymmetric Self-play for Robotics Manipulation



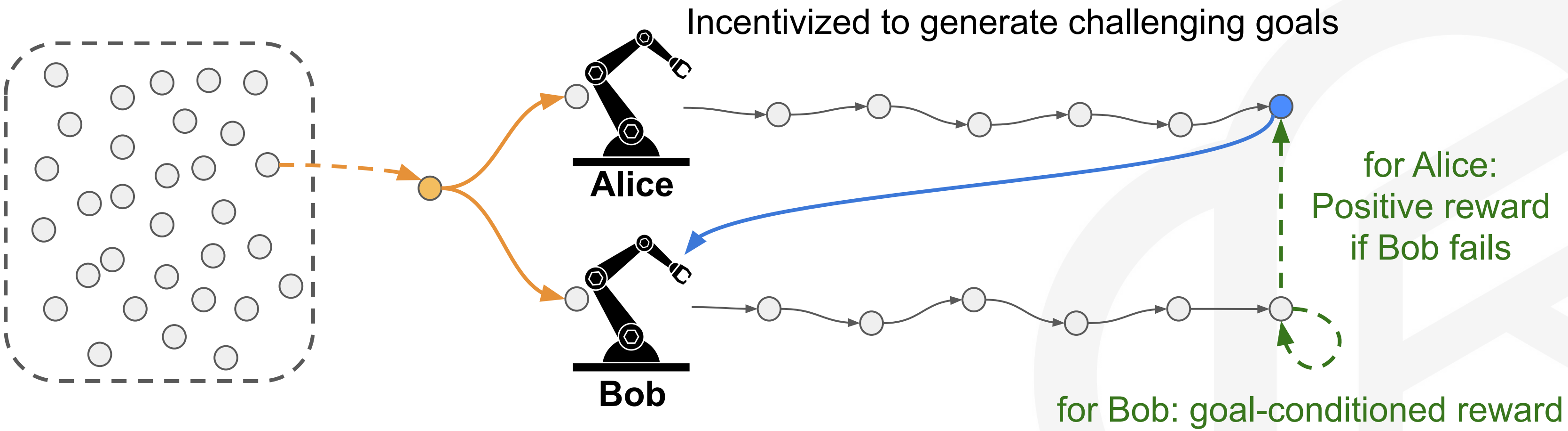
Bob is a goal-conditioned policy

Asymmetric Self-play for Robotics Manipulation

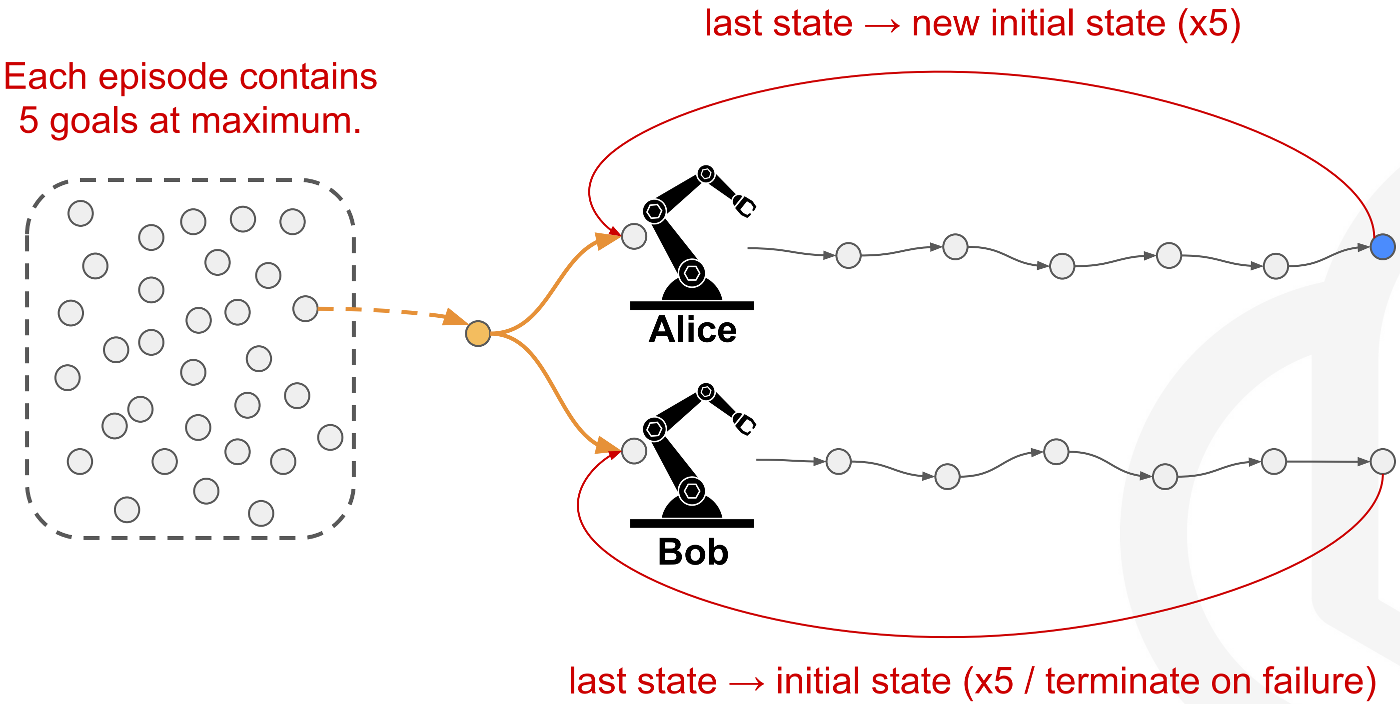


Bob is a goal-conditioned policy

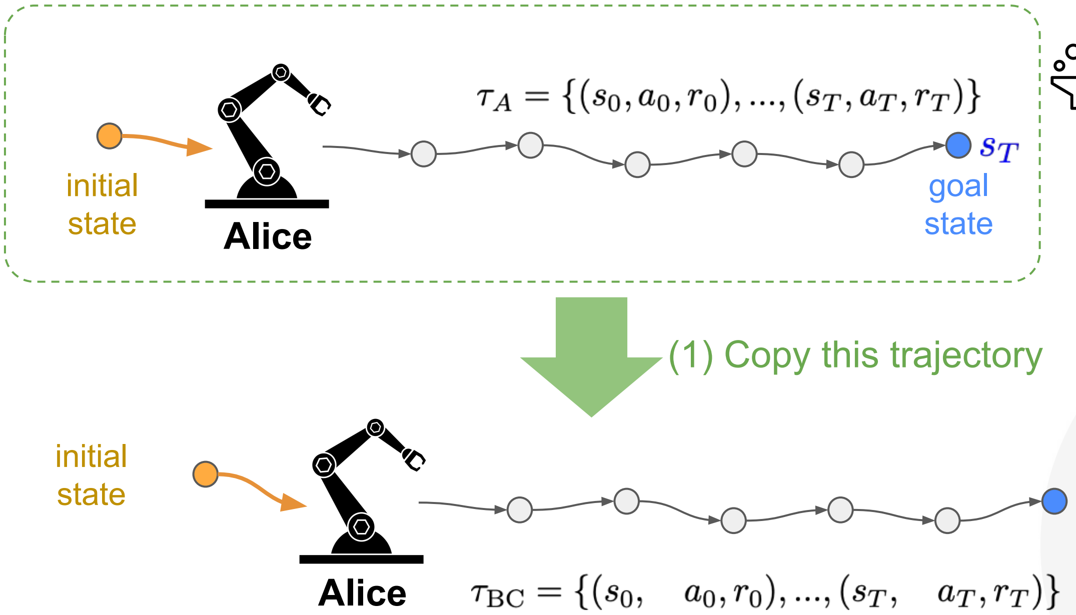
Asymmetric Self-play for Robotics Manipulation



Asymmetric Self-play for Robotics Manipulation



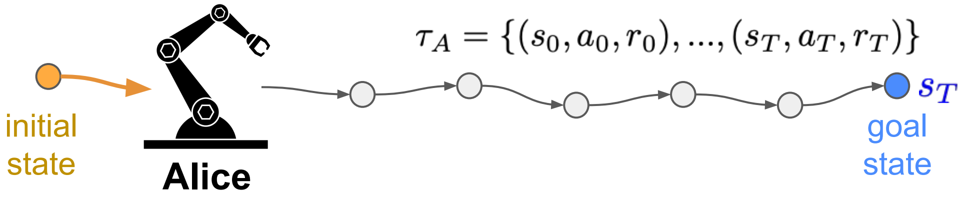
Alice Behavioral Cloning (ABC)



Demonstration trajectory filtering

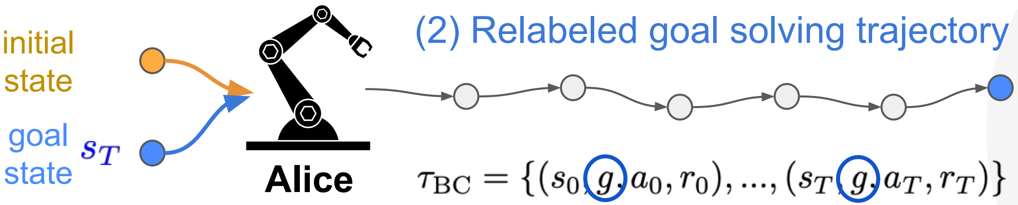
Only if Bob fails this goal

Alice Behavioral Cloning (ABC)



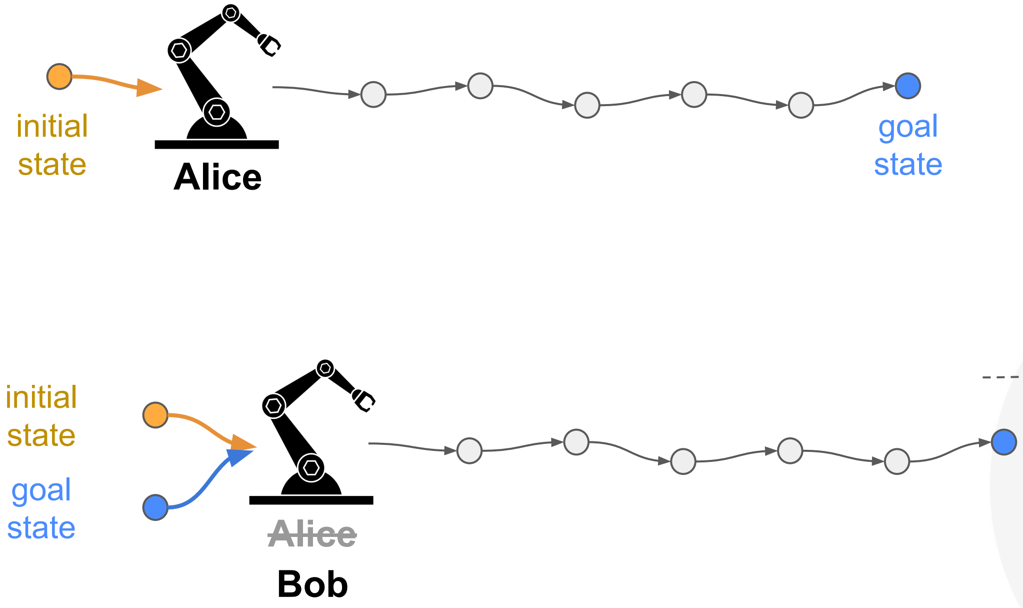
$$\tau_A = \{(s_0, a_0, r_0), \ldots, (s_T, a_T, r_T)\}$$

(2) Relabeled goal solving trajectory



$$\tau_{\mathrm{BC}} = \{(s_0, \underset{s_T}{g}, a_0, r_0), \ldots, (s_T, \underset{s_T}{g}, a_T, r_T)\}$$

Alice Behavioral Cloning (ABC)



For Bob:

$$\mathcal{L} = \mathcal{L}_{\mathrm{RL}} + \beta \underbrace{\mathcal{L}_{\mathrm{abc}}}_{\text{Behavioral cloning loss}}$$

(3) Use demonstration

Stabilizing Alice Behavioral Cloning (ABC)

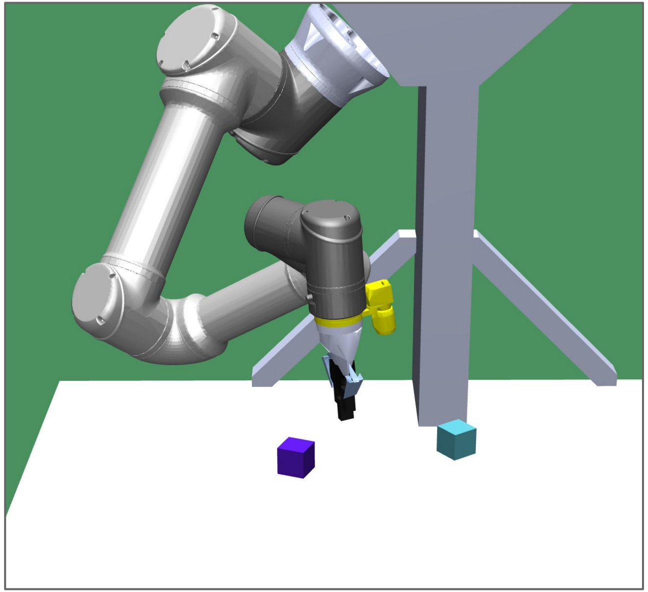
- Demonstration filtering:
  - Collect demonstration only for failed goal
- PPO (Schulman et al., 2017)-style clipping:
  - Prevent drastic policy change

$$\mathcal{L}_{\text{bc}} = -\mathbb{E}_{(s_t, g_t, a_t) \in \mathcal{D}_{\text{BC}}} \log \pi_B(a_t | s_t, g_t; \theta)$$

Naive BC loss

$$\mathcal{L}_{\text{abc}} = -\mathbb{E}_{(s_t, g_t, a_t) \in \mathcal{D}_{\text{BC}}} \left[\text{clip}\left(\frac{\pi_B(a_t | s_t, g_t; \theta)}{\pi_B(a_t | s_t, g_t; \theta_{\text{old}})}, 1-\epsilon, 1+\epsilon \right) \right]$$

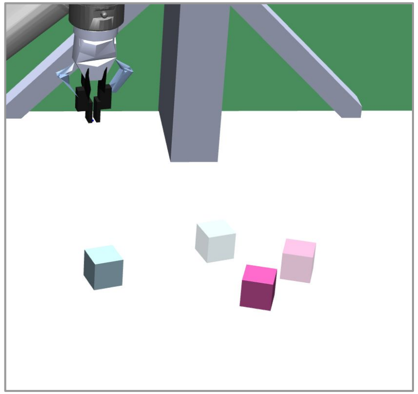
Block Environment



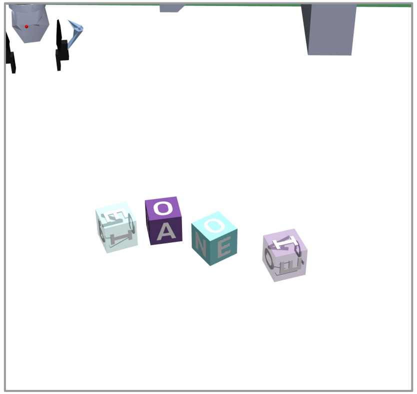
- [1, 2] blocks
- State policy
  - Current position & rotation of blocks
  - Target position & rotation of blocks

Evaluation: Skills to learn in the block environment

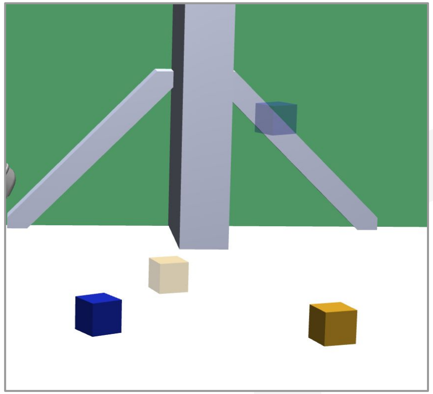
Pushing



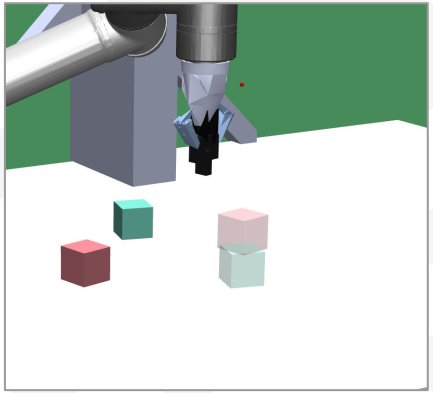
Flipping



Picking up



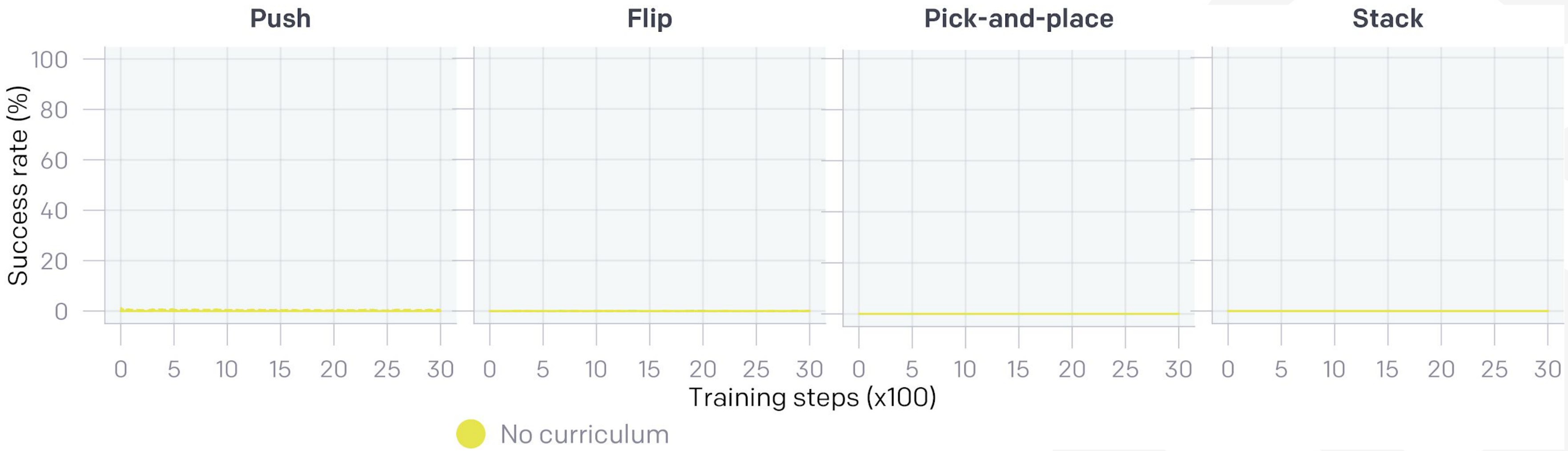
Stacking



Transparent blocks mark the goal state.

Generalize to unseen goals without manual curricula

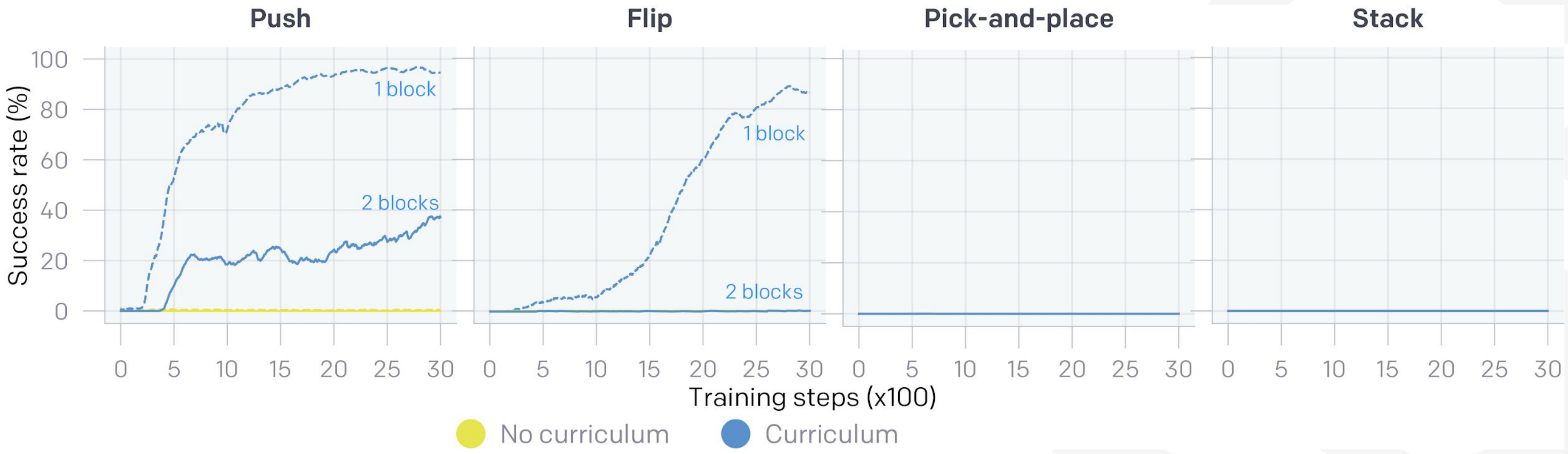
- PPO (Schulman et al., 2017) baseline without curriculum fails to learn



Generalize to unseen goals without manual curricula

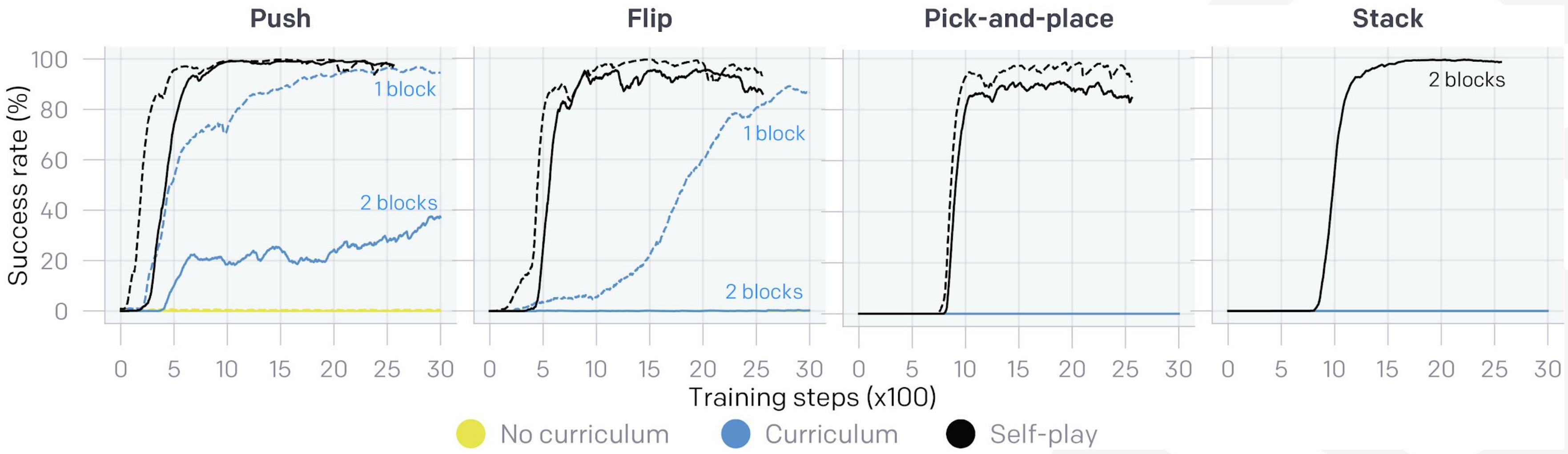
- PPO (Schulman et al., 2017) baseline completely fails to learn
- Domain knowledge-based manual curriculum is insufficient

goal distance ratio
goal rotation weight
probability of pick-and-place
probability of stacking



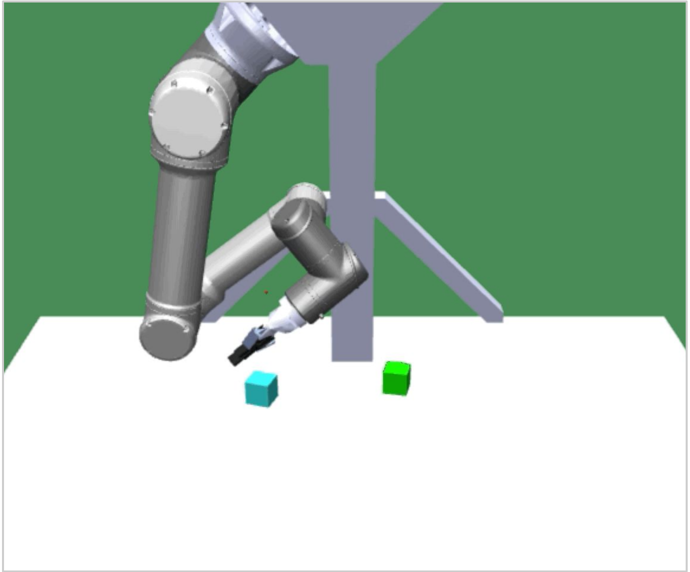
Generalize to unseen goals without manual curricula

- PPO (Schulman et al., 2017) baseline completely fails to learn
- Domain knowledge-based manual curriculum is insufficient
- Asymmetric self-play zero-shot generalizes to all tasks

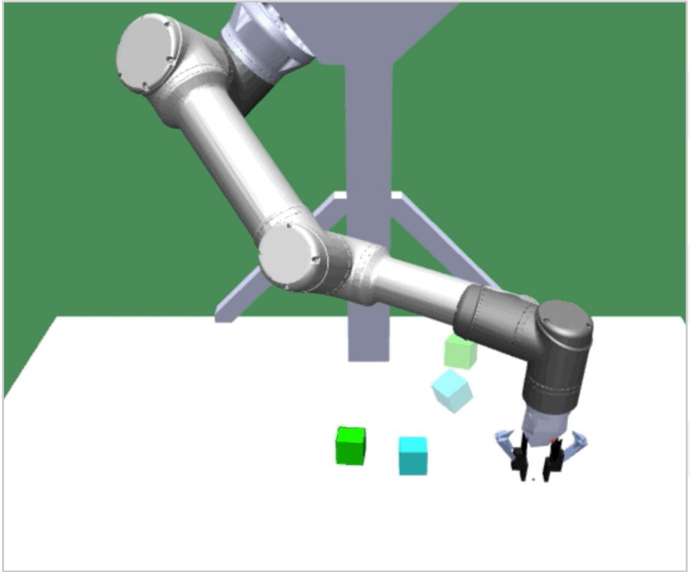


OpenAI

Discovery of Novel Goals / Solutions

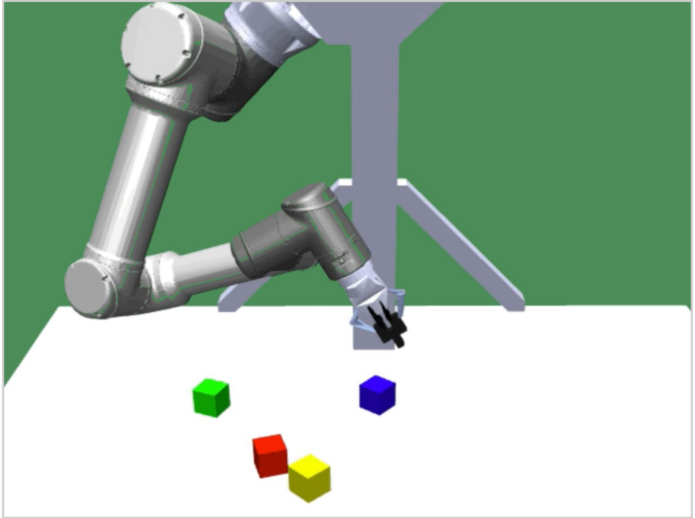


Novel Goals

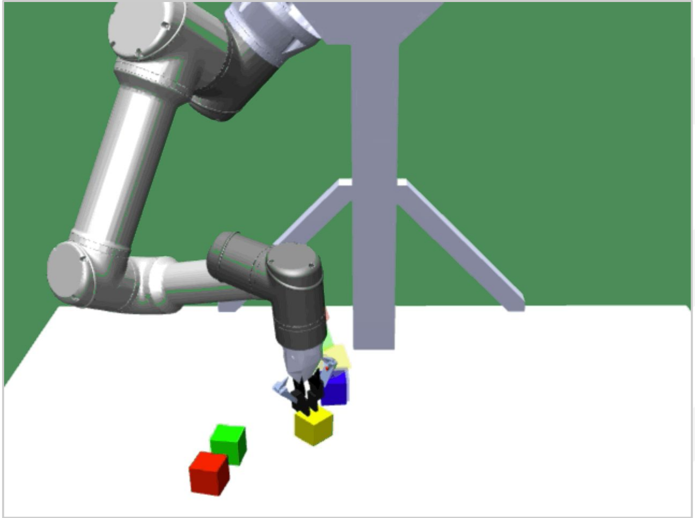


Novel Solutions

Discovery of Novel Goals / Solutions



Novel Goals

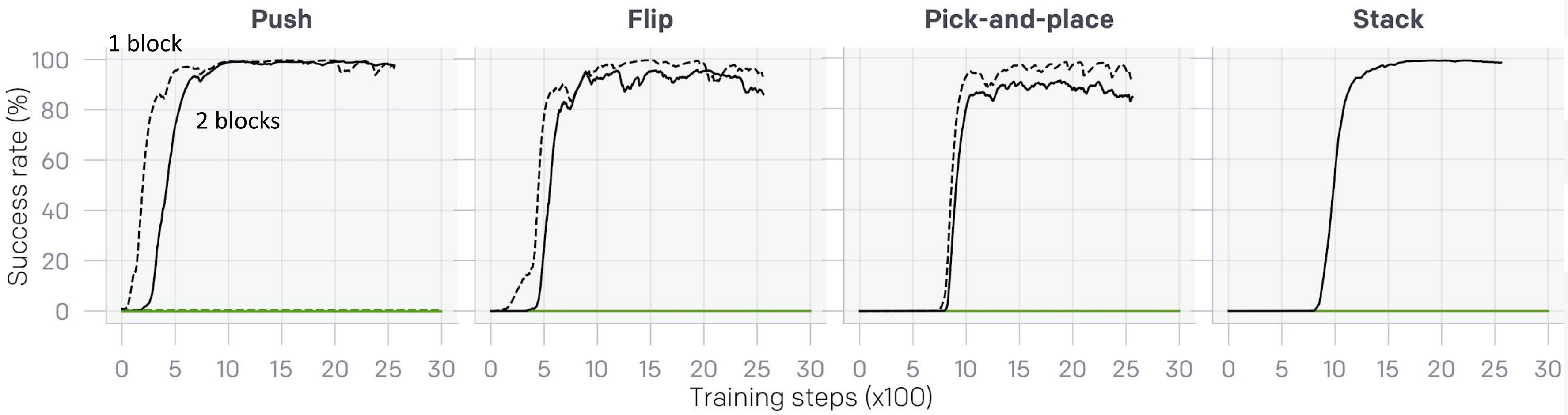


Novel Solutions

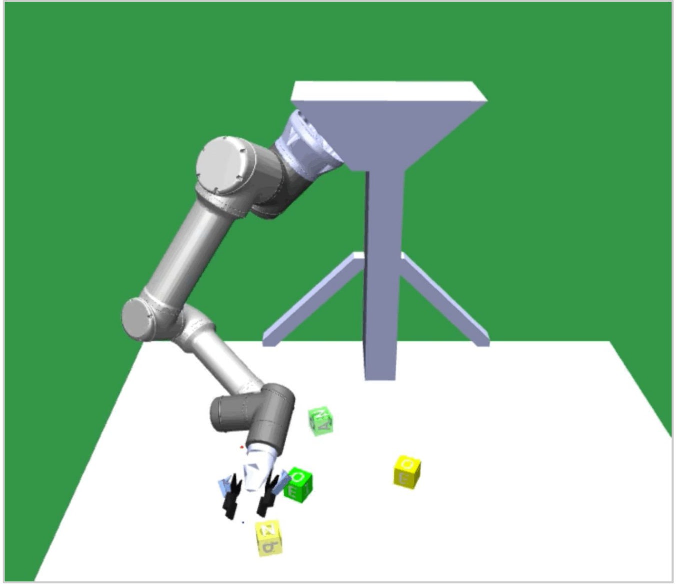
OpenAI

Ablation: ABC is critical

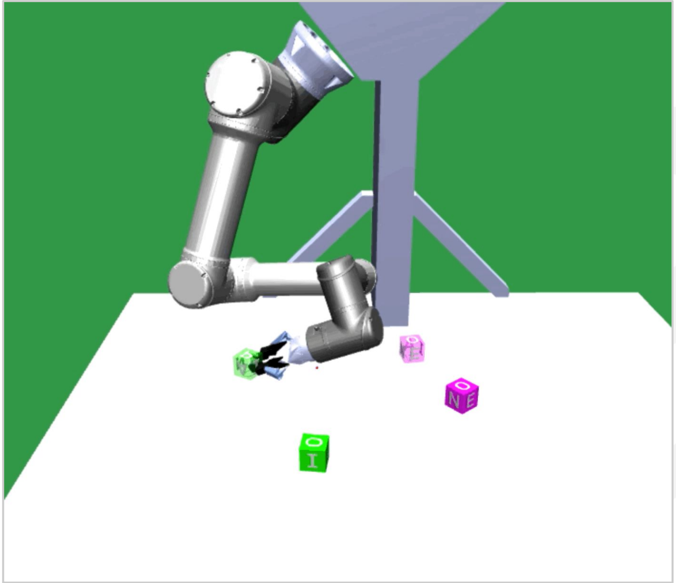
- ABC — Full setup with ABC
- No ABC — No behavioral cloning loss in Bob's training



Ablation: ABC is critical



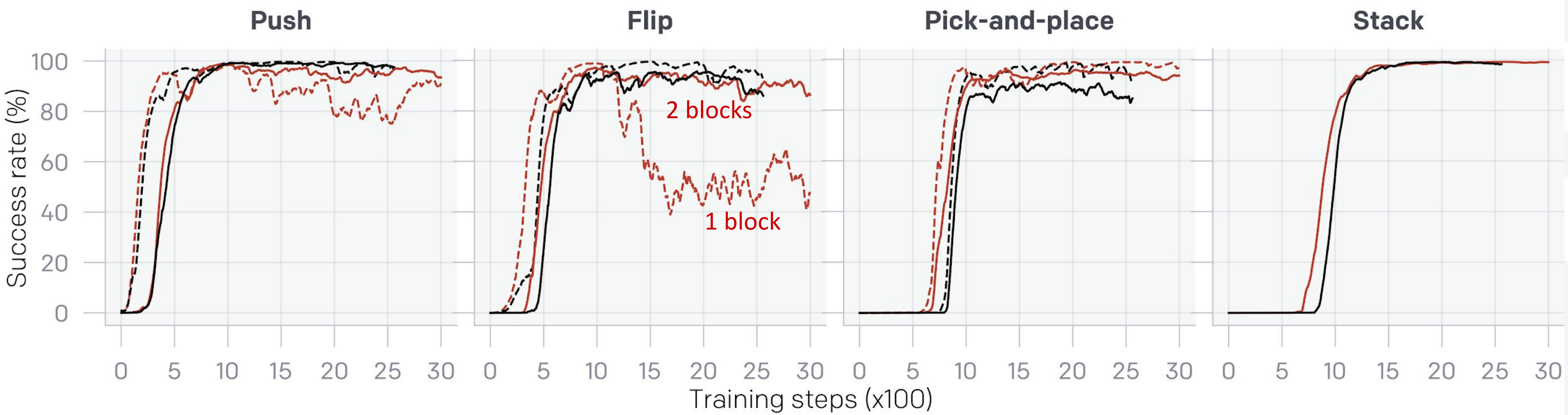
No ABC



With ABC

Ablation: ABC is critical

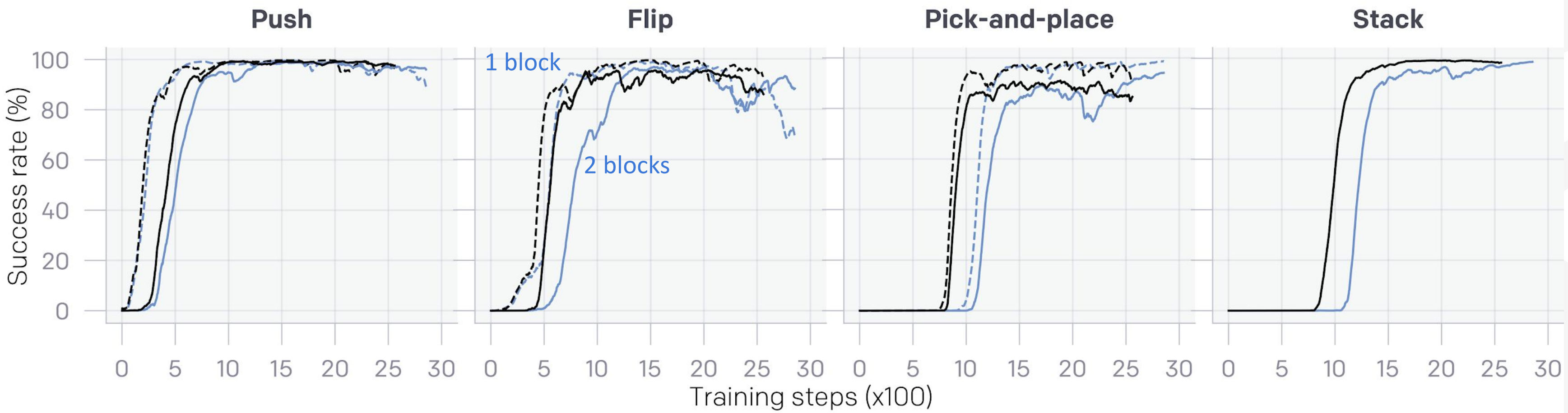
ABC	Full setup with ABC
No demonstration filter	Include all trajectories from Alice no matter Bob fails on this goal or not.



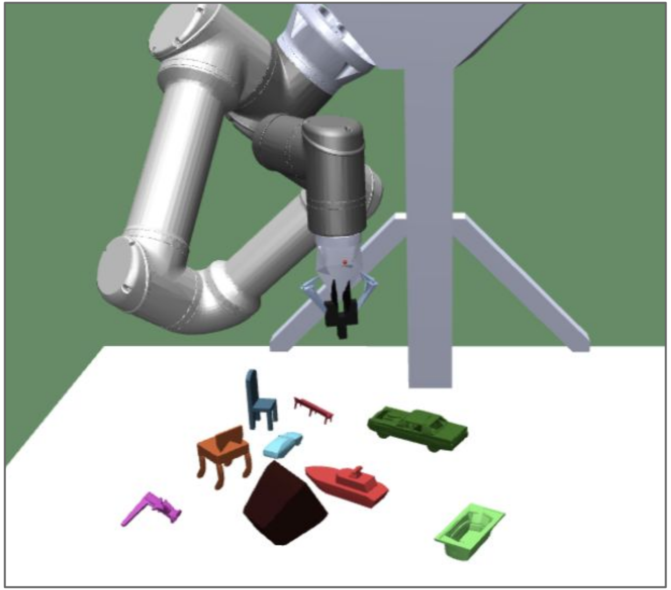
OpenAI

Ablation: ABC is critical

- ABC — Full setup with ABC
- No BC loss clipping — No PPO-style loss clipping in ABC loss

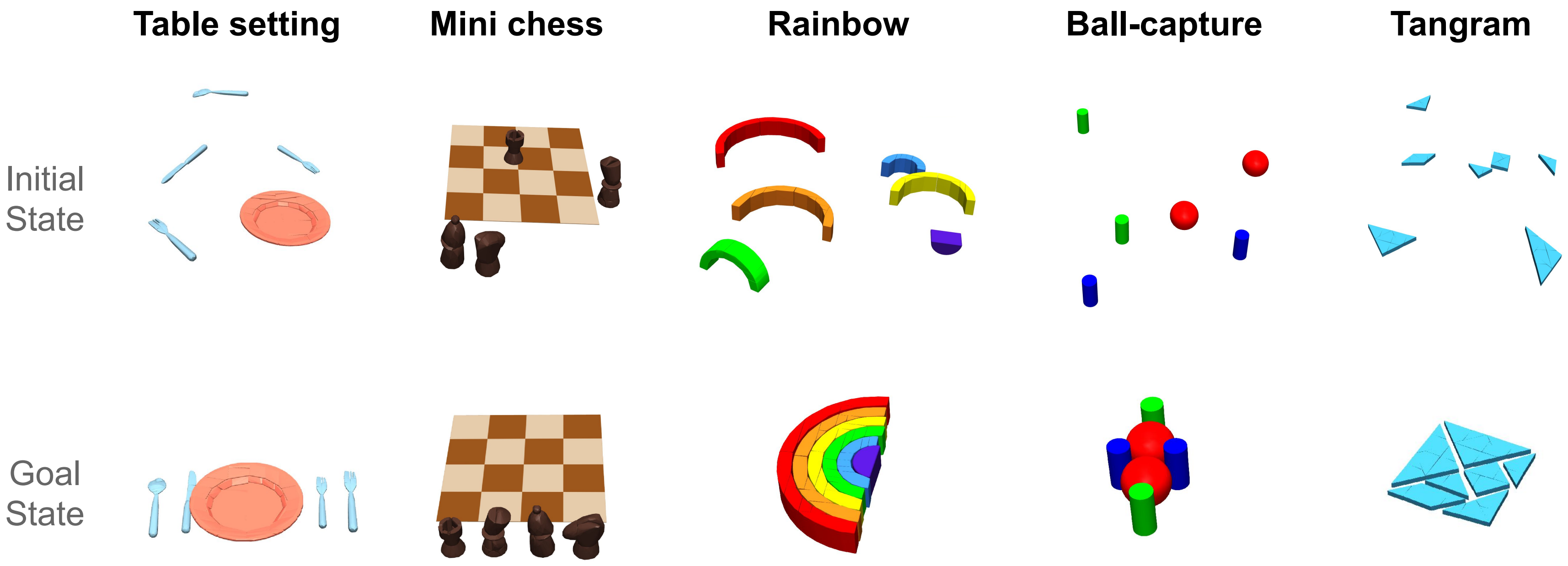


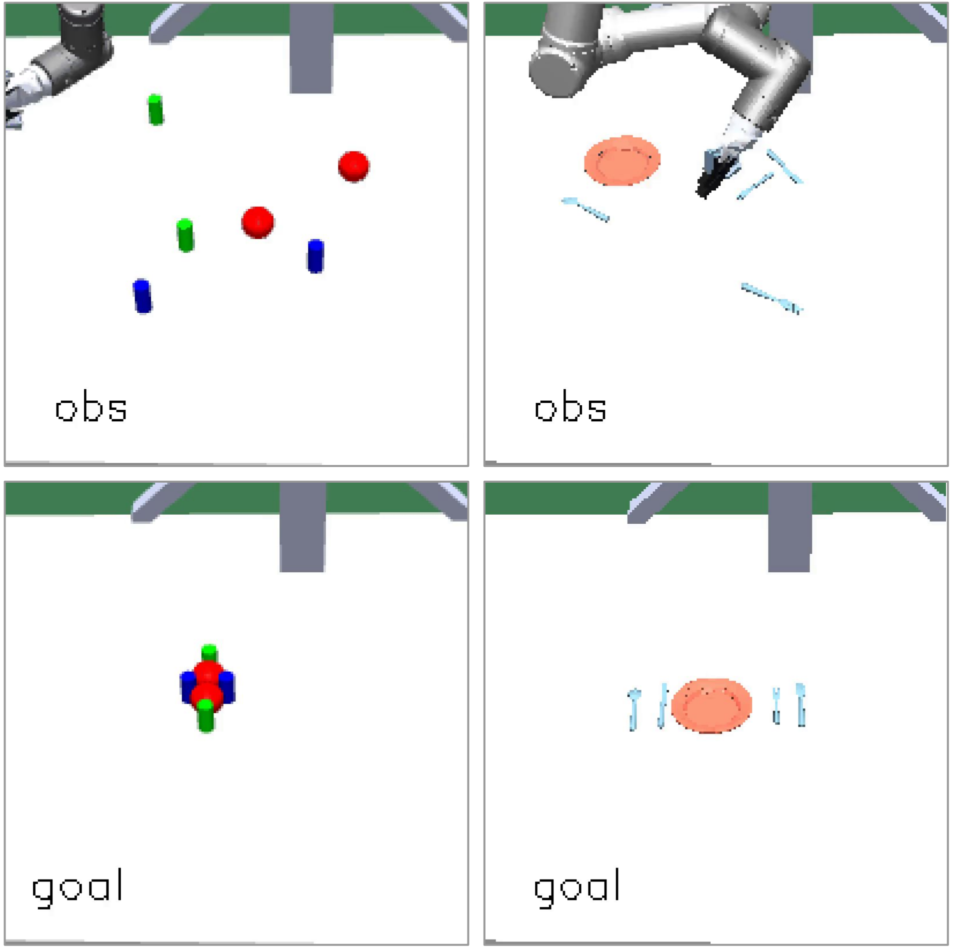
ShapeNet Environment



- [1, 10] objects from ShapeNet (Chang et al., 2015)
- Observation space: State + Vision
  - Current position & rotation of blocks
  - Target position & rotation of blocks
  - Images from front and wrist cameras
  - Target front-camera image

Holdout tasks with unseen objects



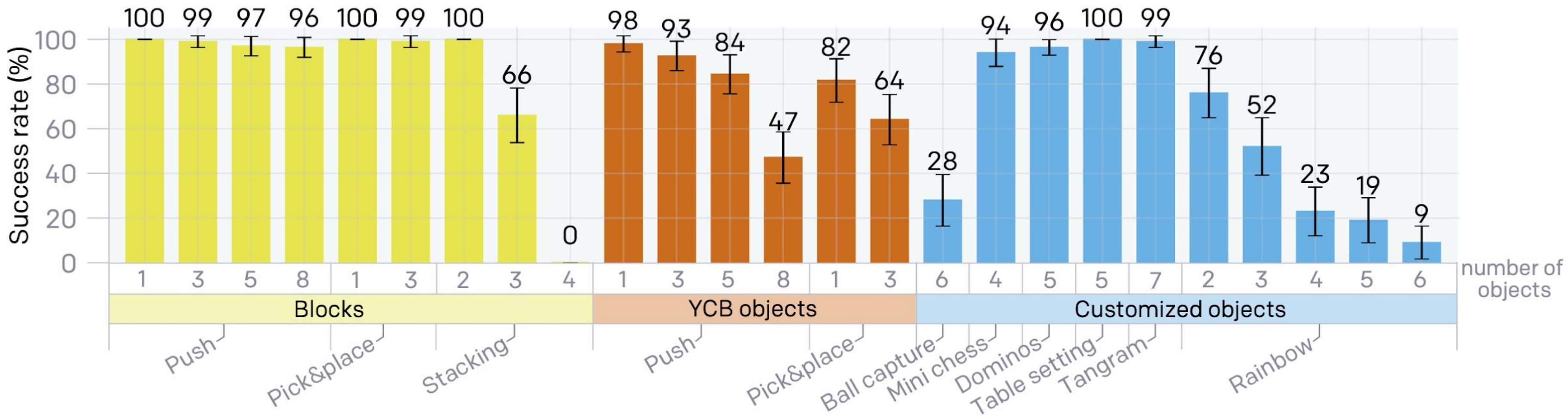


Zero-shot Generalization

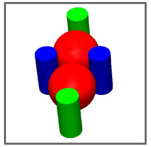
Check out more videos at

https://robotics-self-play.github.io/

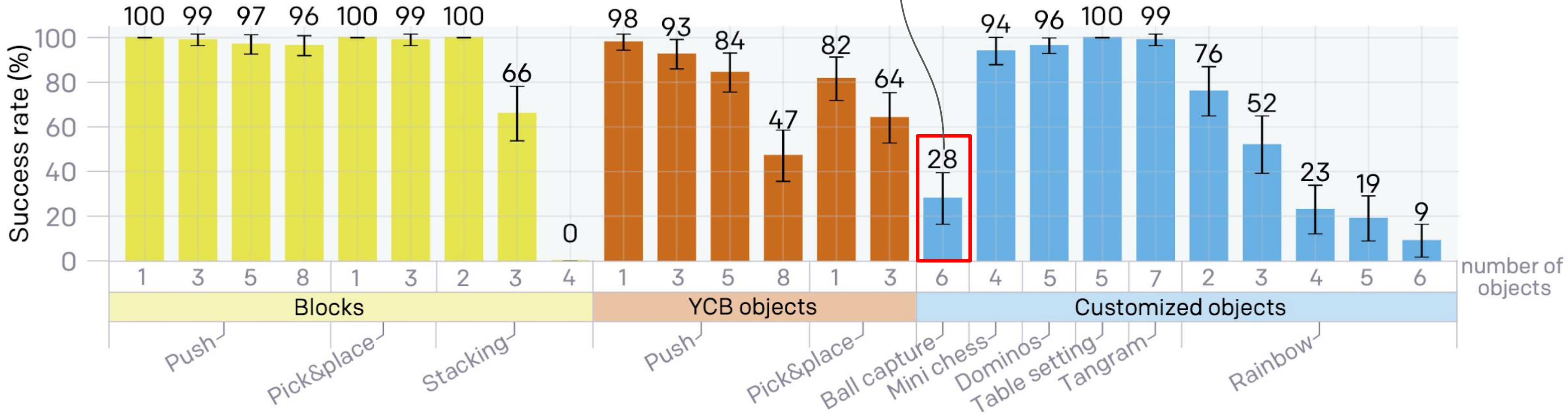
Zero-shot generalization



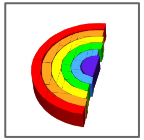
Zero-shot generalization



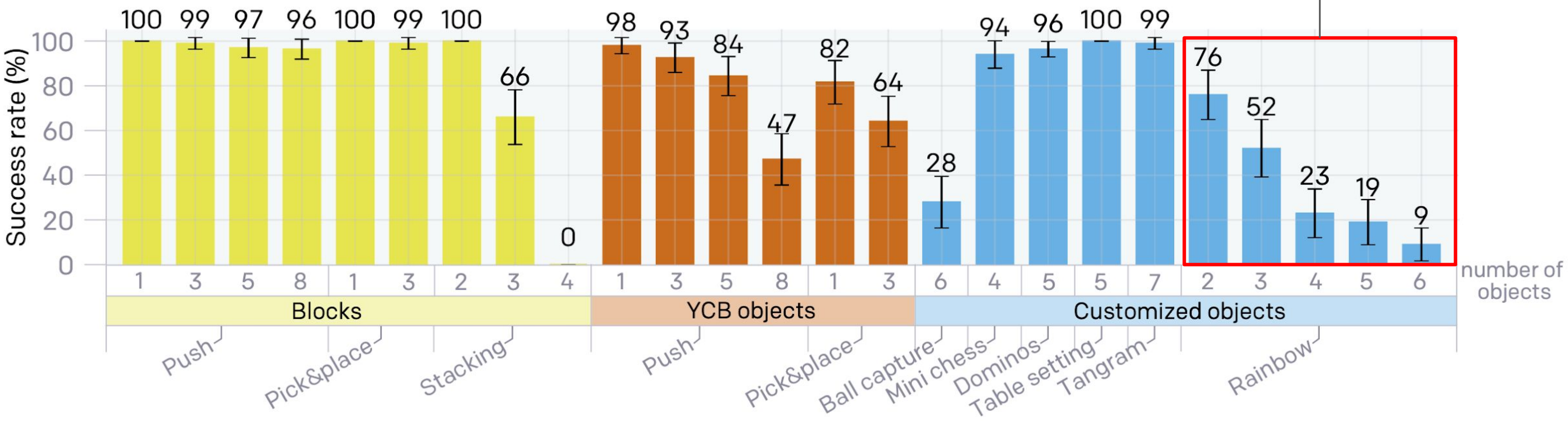
Delicate handling of rolling objects and lifting skills



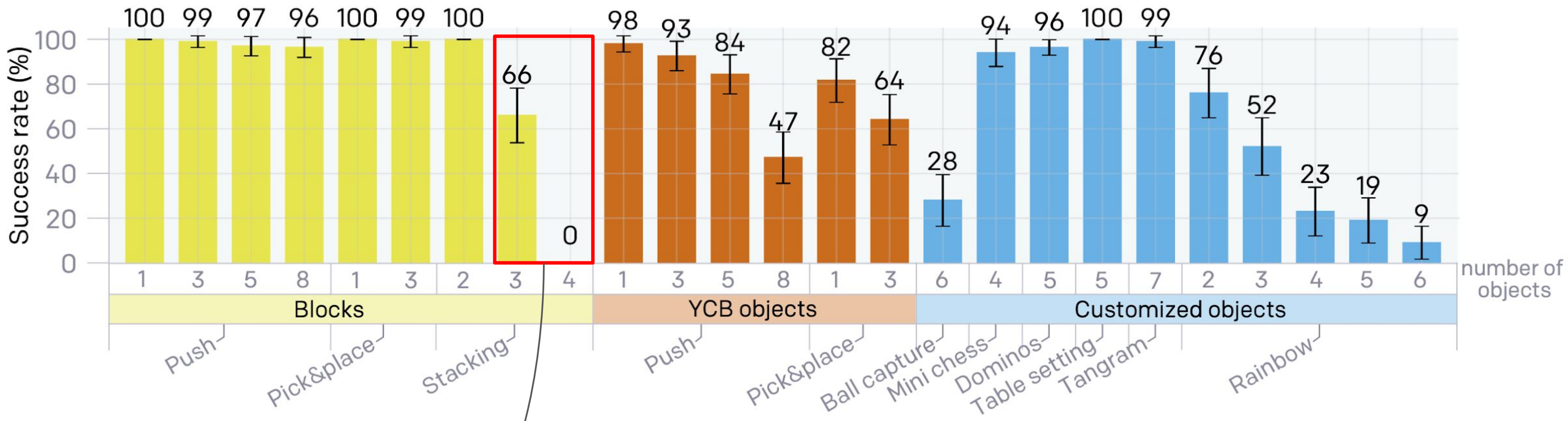
Zero-shot generalization

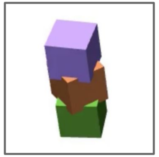


Understand concave shapes



Zero-shot generalization





Understand what's the correct order of placing objects.

Conclusion

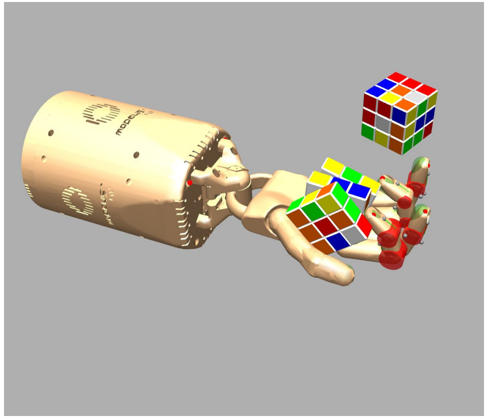
Asymmetric self-play can:

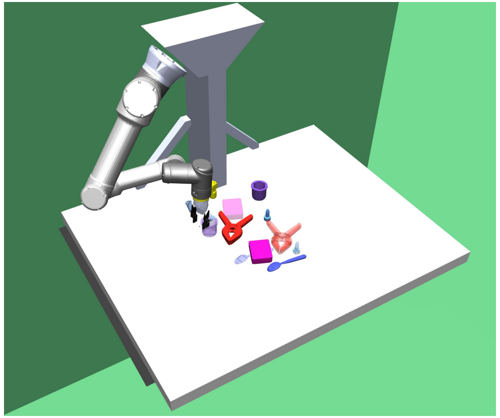
1. Train a policy that can **zero-shot generalize** to many unseen robotic manipulation tasks.
2. **Alleviate** the importance of **manual curriculum**.
3. Alice Behavior Cloning (ABC) is crucial.

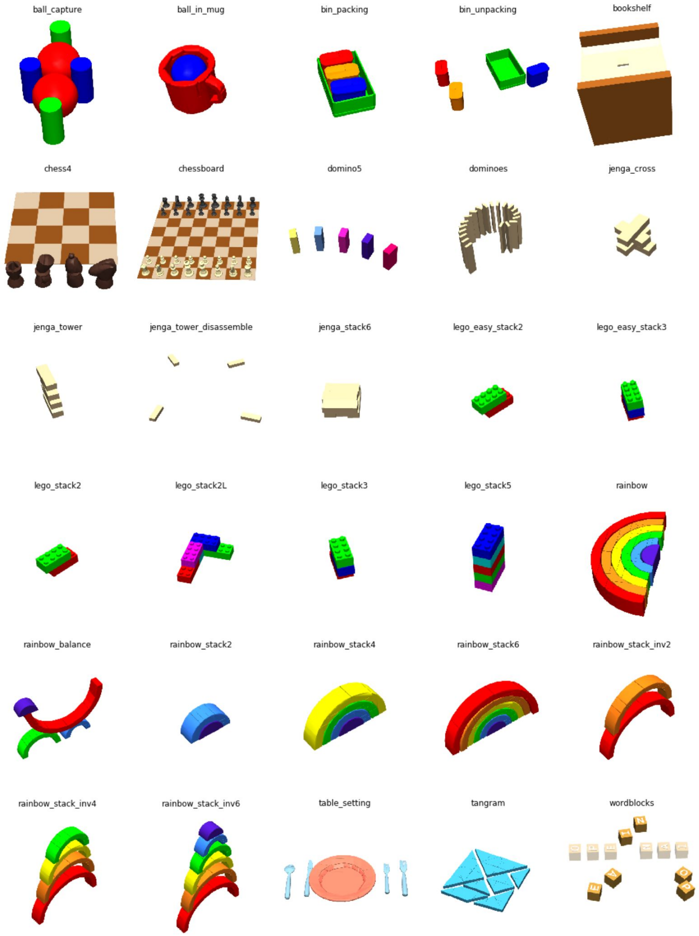
Announce: `robogym`

https://github.com/openai/robogym

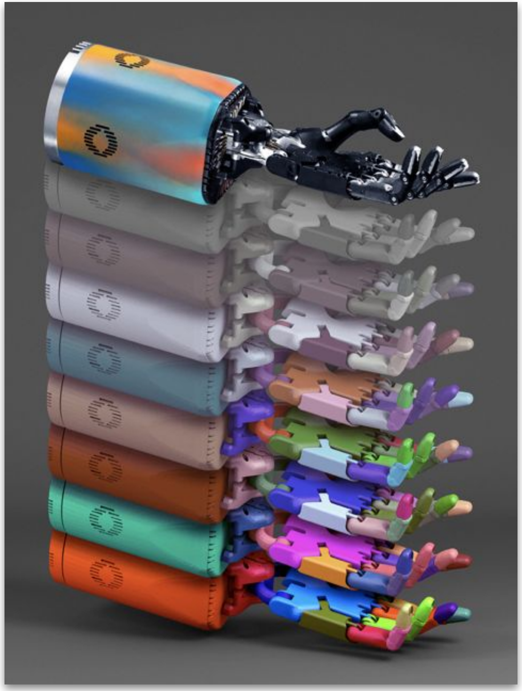
A simulation framework that uses OpenAI gym and MuJoCo simulator, including two environments: (1) in-hand manipulation with Rubik's cube; (2) table-top rearrange with one robot arm + gripper..







OpenAI



openai.com/blog/learning-dexterity

openai.com/blog/solving-rubiks-cube

arXiv:2101.04882v1 [cs.LG] 13 Jan 2021

ASYMMETRIC SELF-PLAY FOR AUTOMATIC GOAL DISCOVERY IN ROBOTIC MANIPULATION

OpenAI
Matthias Plappert*, Raul Sampedro*, Tao Xu*, Ilge Akkaya*, Vineet Kosaraju*, Peter Welinder*, Ruben D'Sa*, Arthur Petron*, Henrique Ponde de Oliveira Pinto*, Alex Paino*, Hyeonwoo Noh*, Lilian Weng*, Qiming Yuan*, Casey Chu*, Wojciech Zaremba*

ABSTRACT

We train a single, goal-conditioned policy that can solve many robotic manipulation tasks, including tasks with previously unseen goals and objects. We rely on asymmetric self-play for goal discovery, where two agents, Alice and Bob, play a game. Alice is asked to propose challenging goals and Bob aims to solve them. We show that this method can discover highly diverse and complex goals without any human priors. Bob can be trained with only sparse rewards, because the interaction between Alice and Bob results in a natural curriculum and Bob can learn from Alice's trajectory when relabeled as a goal-conditioned demonstration. Finally, our method scales, resulting in a single policy that can generalize to many unseen tasks such as setting a table, stacking blocks, and solving simple puzzles. Videos of a learned policy is available at https://robotics-self-play.github.io.

1 INTRODUCTION

We are motivated to train a *single* goal-conditioned policy (Kaelbling, 1993) that can solve *any* robotic manipulation task that a human may request in a given environment. In this work, we make progress towards this goal by solving a robotic manipulation problem in a table-top setting where the robot's task is to change the initial configuration of a variable number of objects on a table to match a given goal configuration. This problem is simple in its formulation but likely to challenge a wide variety of cognitive abilities of a robot as objects become diverse and goals become complex.

Motivated by the recent success of deep reinforcement learning for robotics (Levine et al., 2016; Gu et al., 2017; Hwangbo et al., 2019; OpenAI et al., 2019a), we tackle this problem using deep reinforcement learning on a very large training distribution. An open question in this approach is how we can build a training distribution rich enough to achieve generalization to many unseen manipulation tasks. This involves defining both an environment's initial state distribution and a goal distribution. The initial state distribution determines how we sample a set of objects and their configuration at the beginning of an episode, and the goal distribution defines how we sample target states given an initial state. In this work, we focus on a scalable way to define a rich goal distribution.

*Authors are listed at random and a detailed contribution section is at the end. Please cite as **OpenAI et al.**.

(a) Table-top setting with a robot arm (b) Example initial state for training (c) Example holdout tasks

Figure 1: (a) We train a policy that controls a robot arm operating in a table-top setting. (b) Randomly placed ShapeNet (Chang et al., 2015) objects constitute an initial state distribution for training. (c) We use multiple manually designed holdout tasks to evaluate the learned policy.

1

arxiv.org/abs/2101.04882

Thank you!

@lilianweng

lilianweng.github.io/lil-log