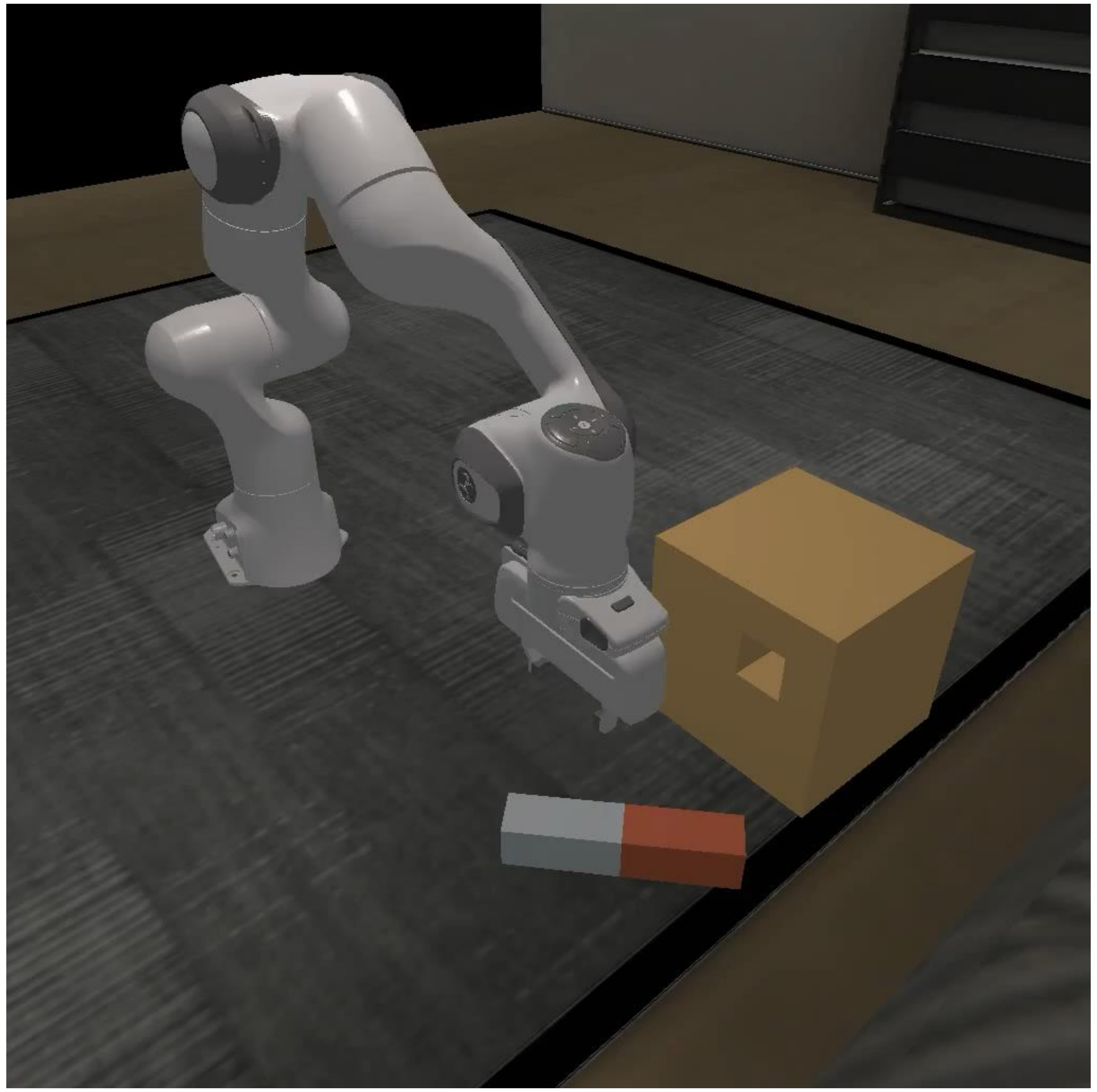
Policy Decorator: Model-Agnostic Online Refinement for Large Policy Model

Xiu Yuan*, Tongzhou Mu*, Stone Tao, Yunhao Fang, Mengke Zhang, Hao Su

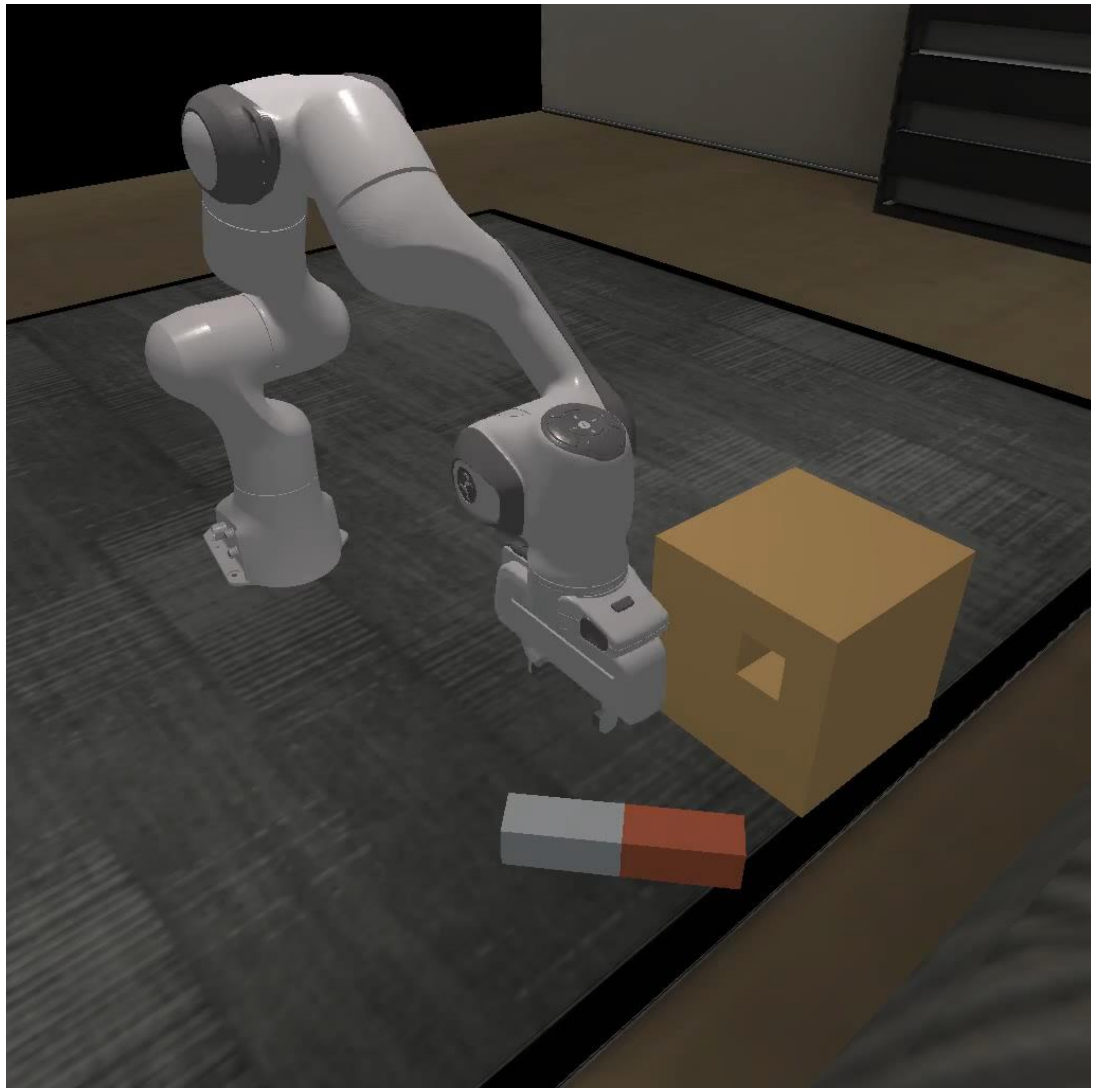
UC San Diego

This video has sound

Diffusion Policy

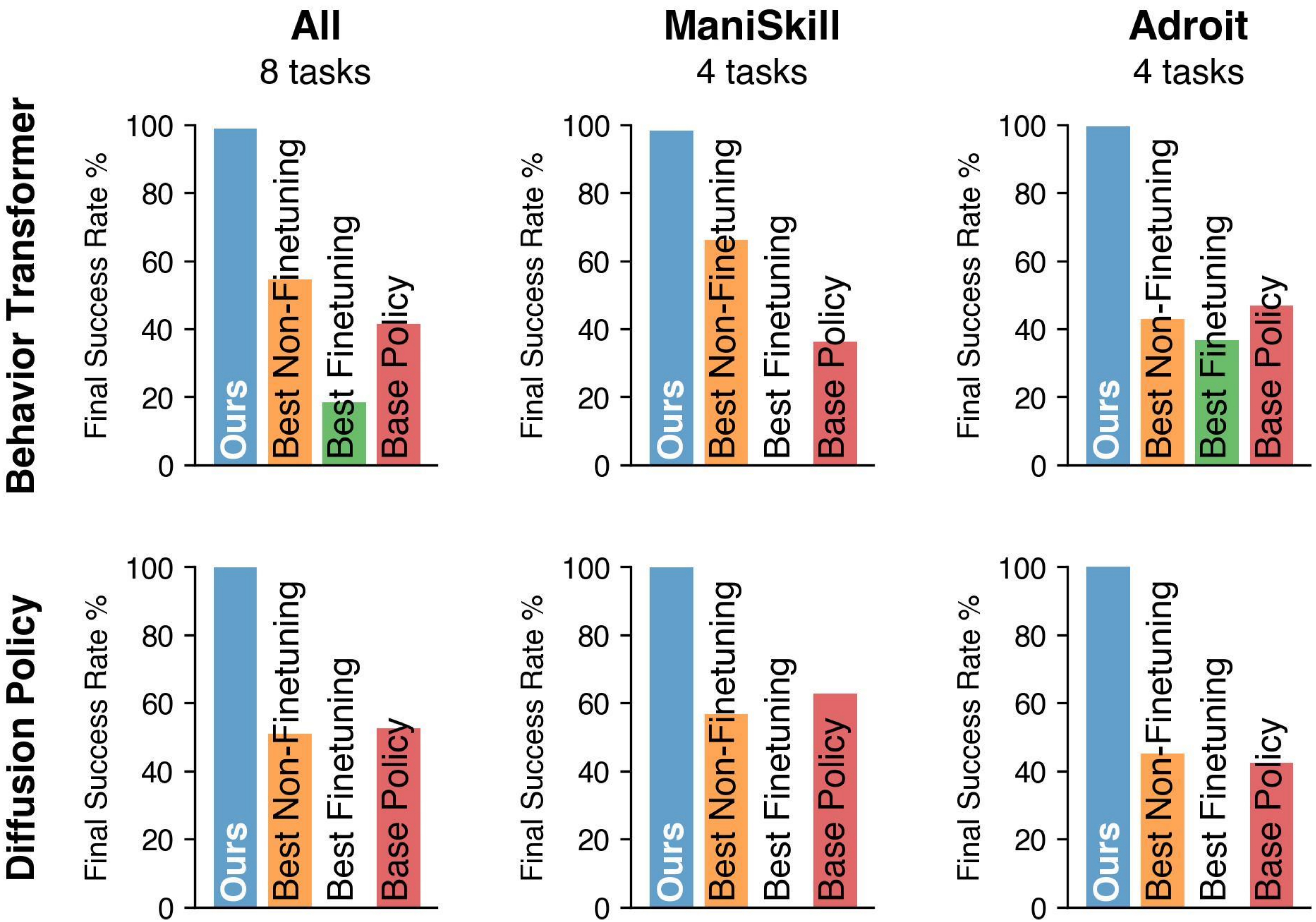


Ours



🛠 Fix It by a “Decorator”

Improve Various SOTA Policy Models



How do we achieve it?

How to Improve a Policy?

Collect 10x more demo? 🤔

Too Costly! 😔

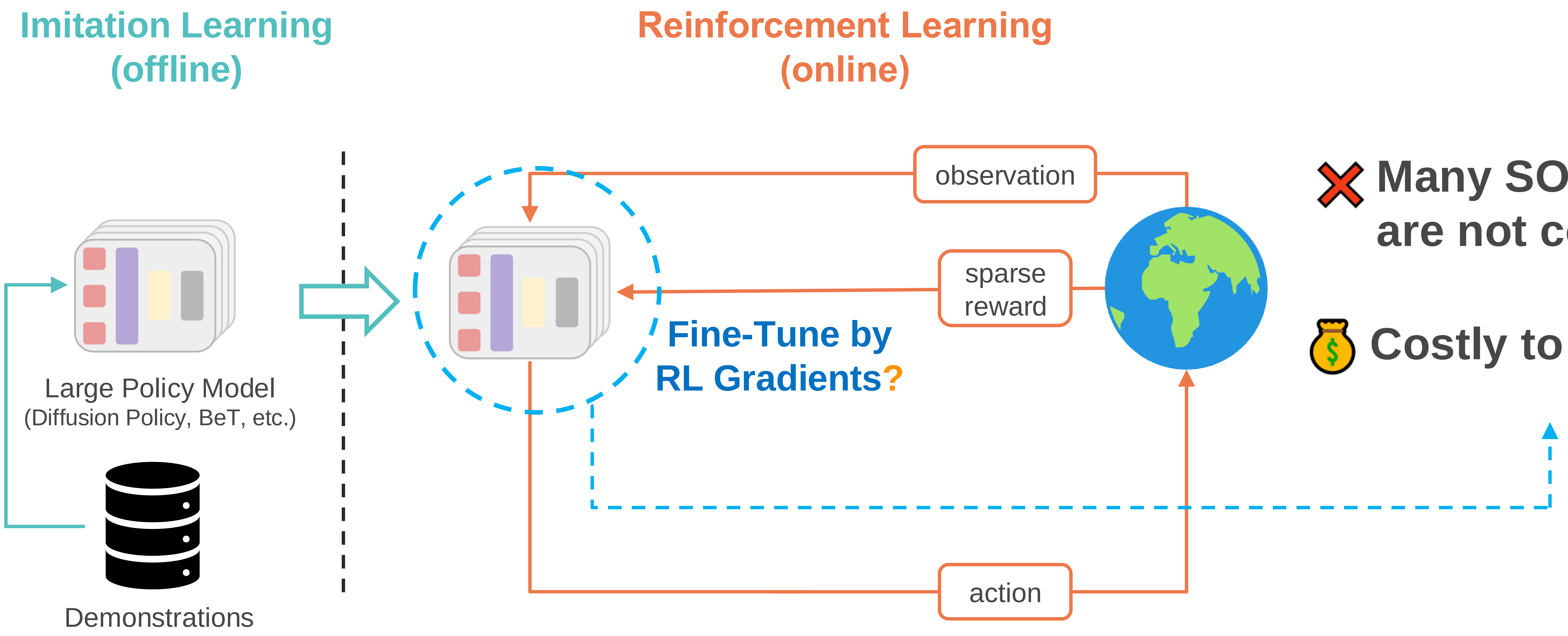
Video from iGibson2

Online RL w/ Sparse Reward

Less Human Effort,
Better Data Coverage 😄

Video from ManiSkill

Let’s Fine-Tune It?

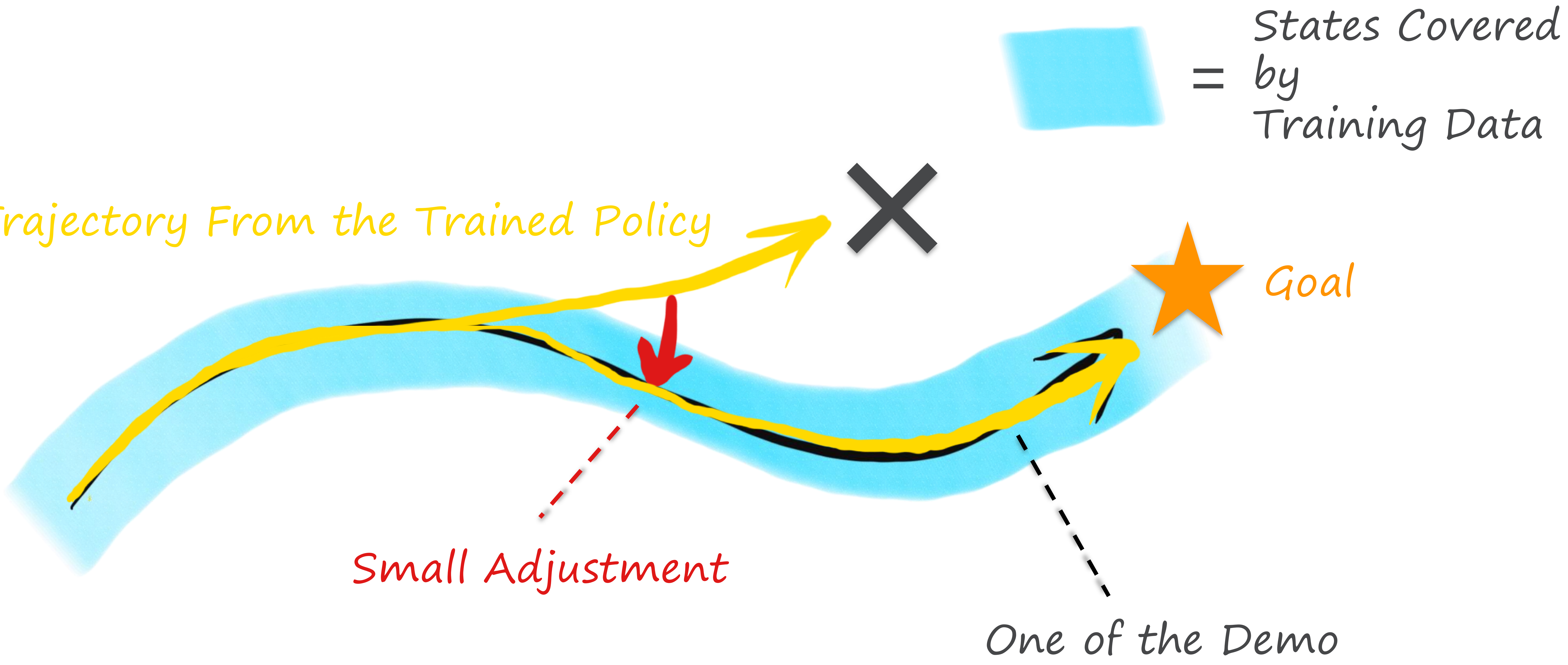


❌ Many SOTA policy models are not compatible with RL

💰 Costly to train large models

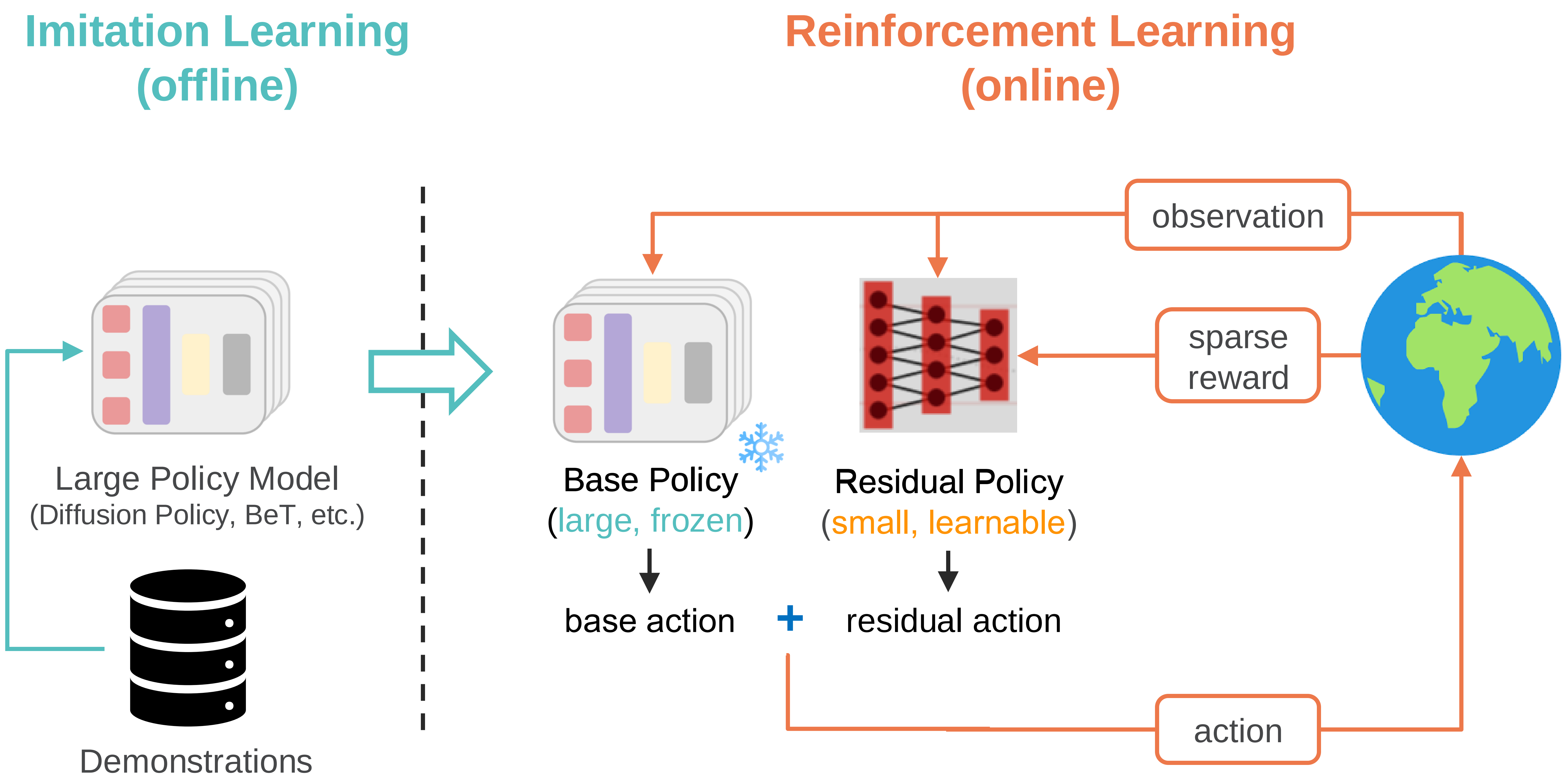
Any alternative solutions? 🤔

How Offline Imitation Learning Policy Fails


States Covered by Training Data
Trajectory From the Trained Policy
Goal
Small Adjustment
One of the Demo

Can we directly learn this "adjustment"? 🤔

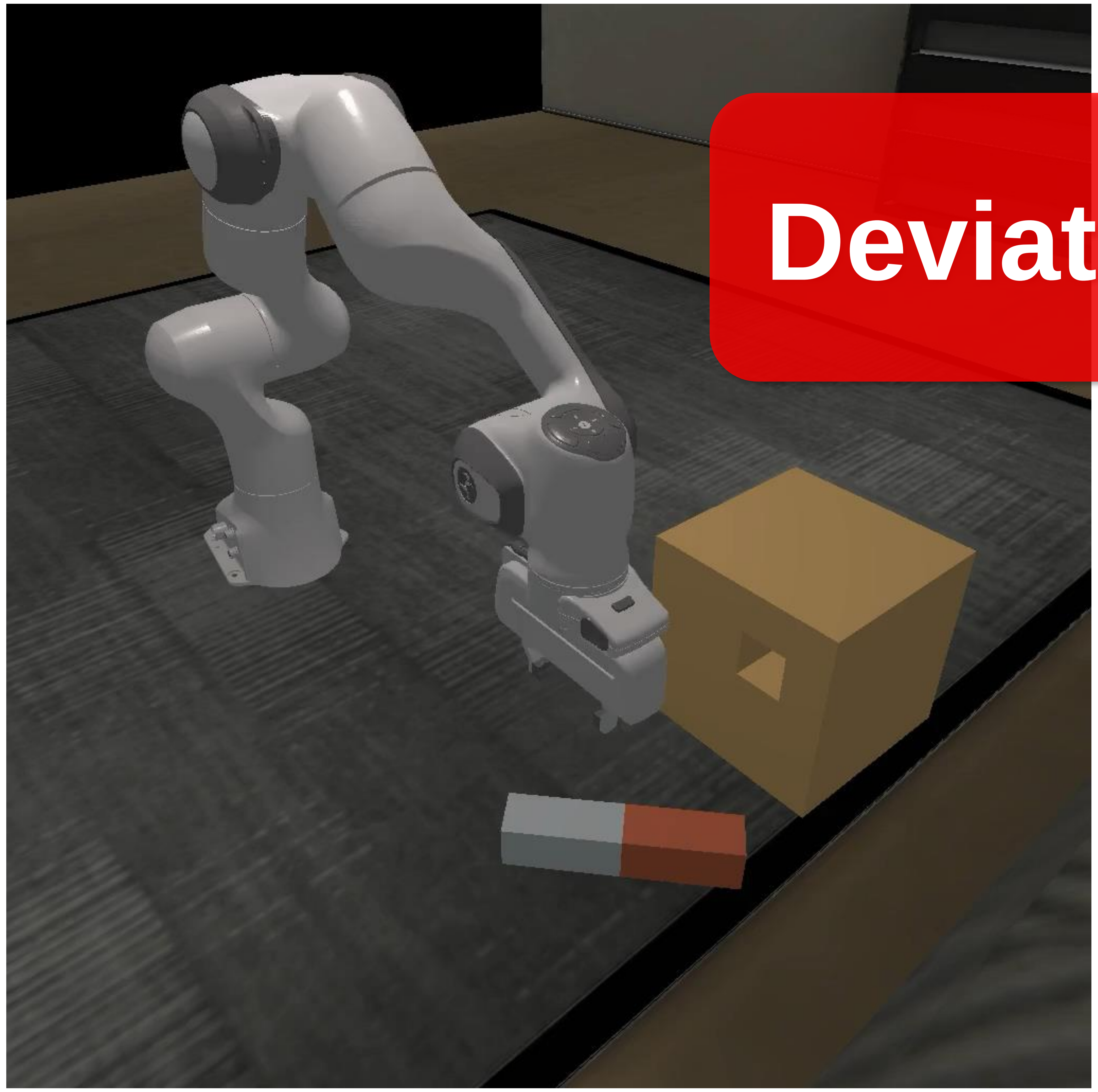
Residual Policy with Online RL



Are we done? 🤔

Virtually No Successes w/ Residual

Base Policy



Base + Random Residual Actions

Deviate Too Much! 😔

Controlled Exploration

Strategy 1: Bounded Residual Action

+

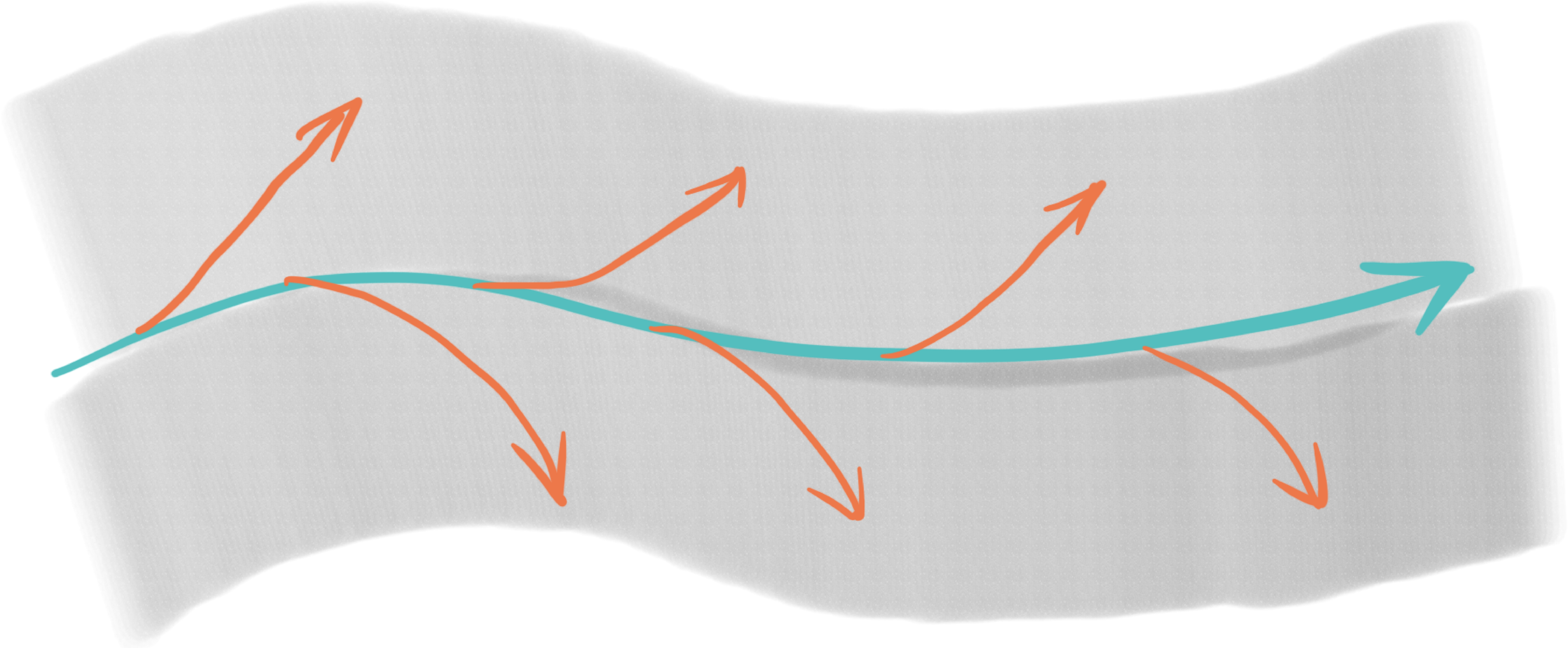
Strategy 2: Progressive Exploration Schedule

Strategy 1: Bounded Residual Action

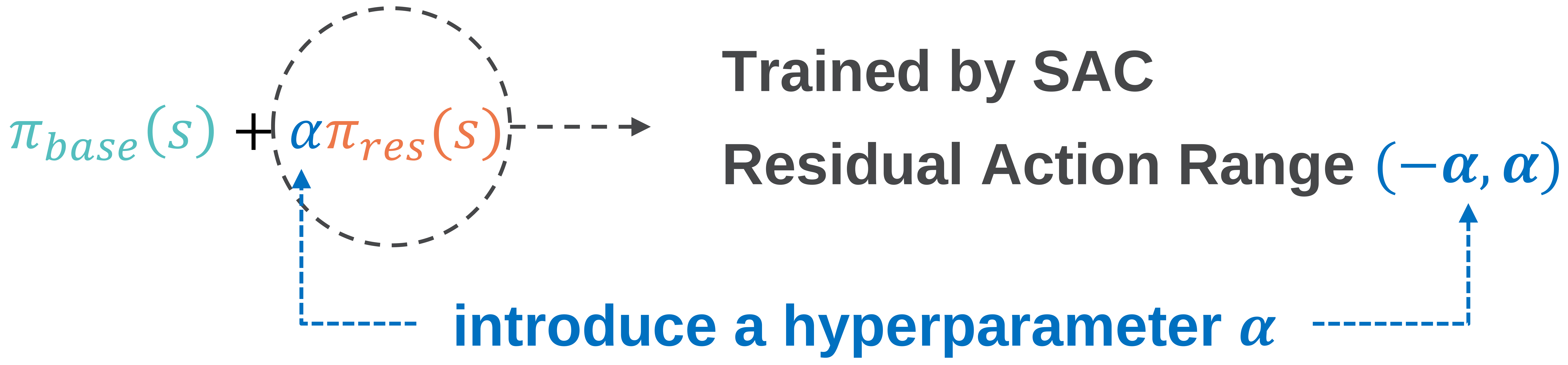
$\pi_{base}(s) + \pi_{res}(s)$

Trained by SAC

Residual Action Range $(\mathbf{-1, 1})$



Strategy 1: Bounded Residual Action

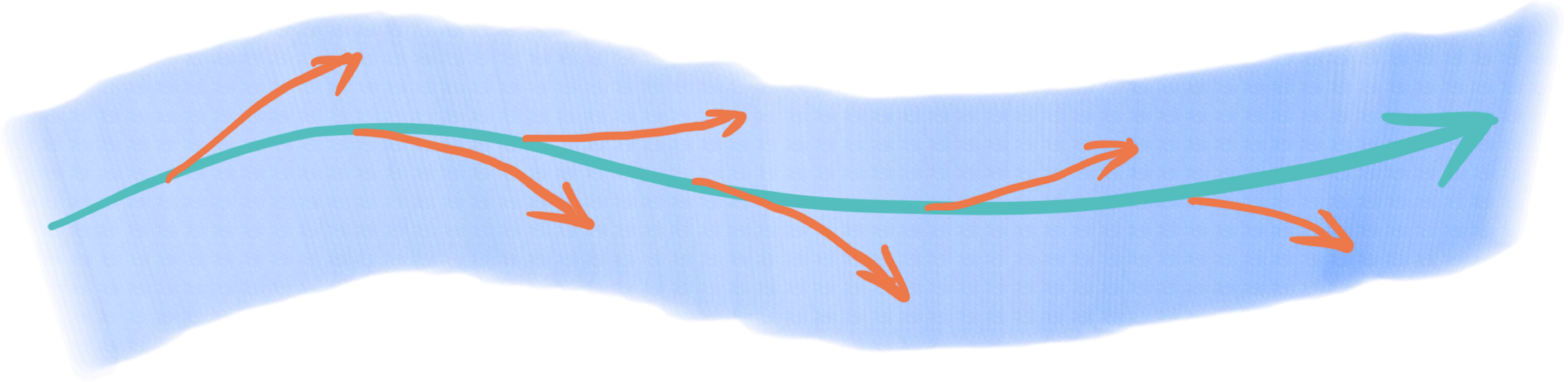


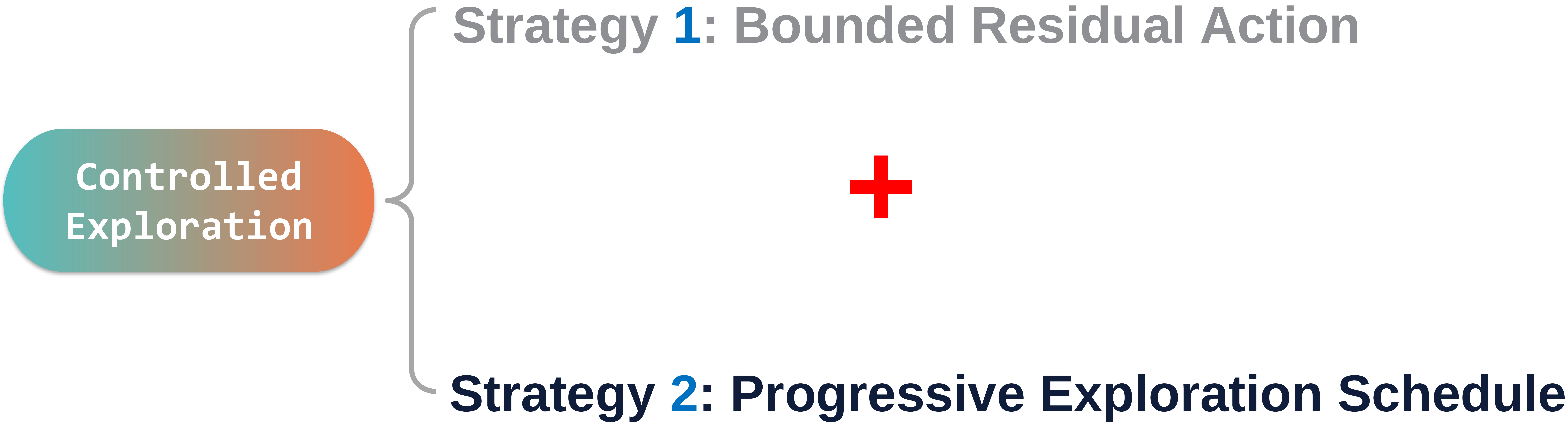
$\pi_{base}(s) + \alpha\pi_{res}(s)$

Trained by SAC

Residual Action Range $(-\alpha, \alpha)$

introduce a hyperparameter α





Controlled Exploration

Strategy 1: Bounded Residual Action

+

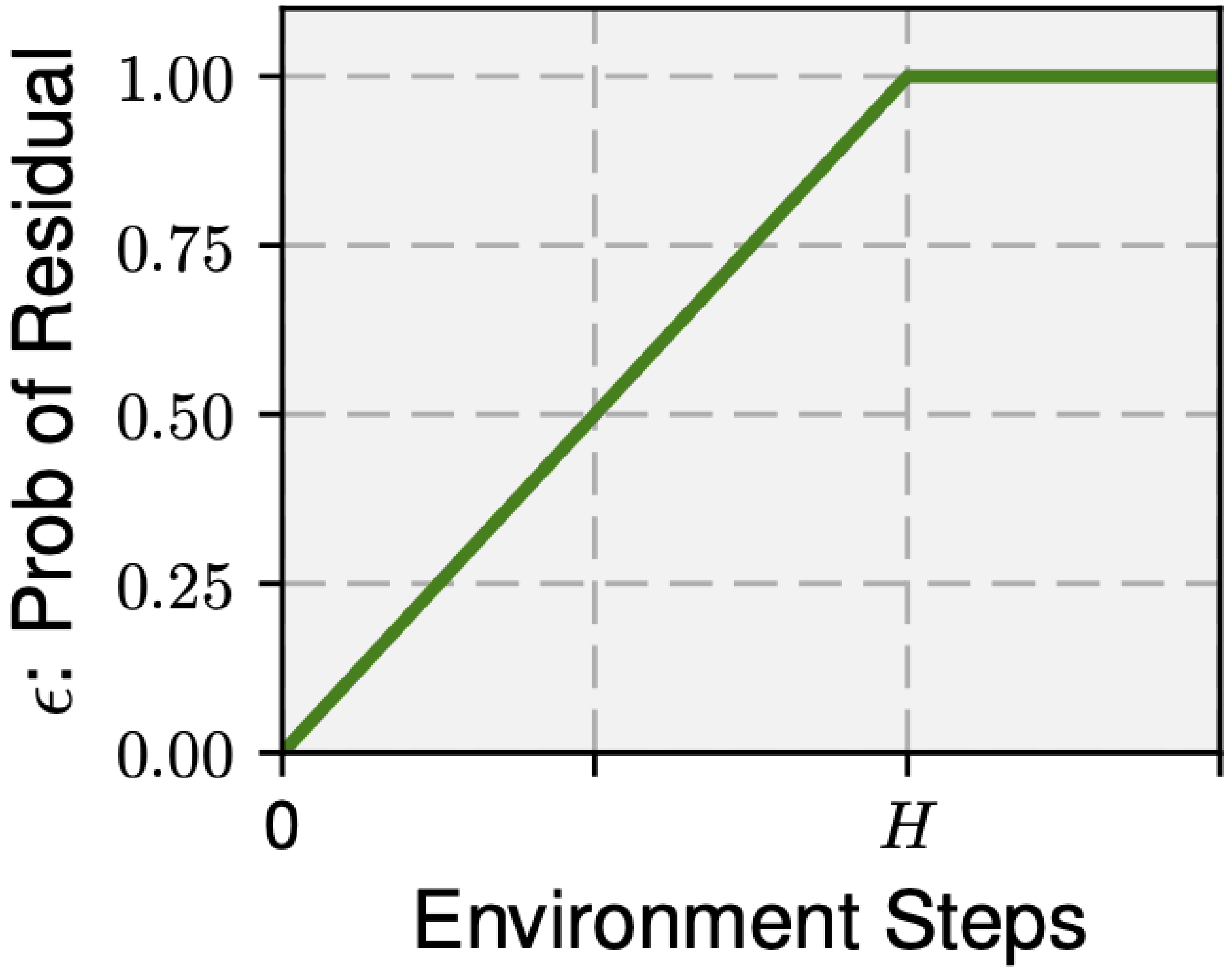
Strategy 2: Progressive Exploration Schedule

Strategy 2: Progressive Exploration Schedule

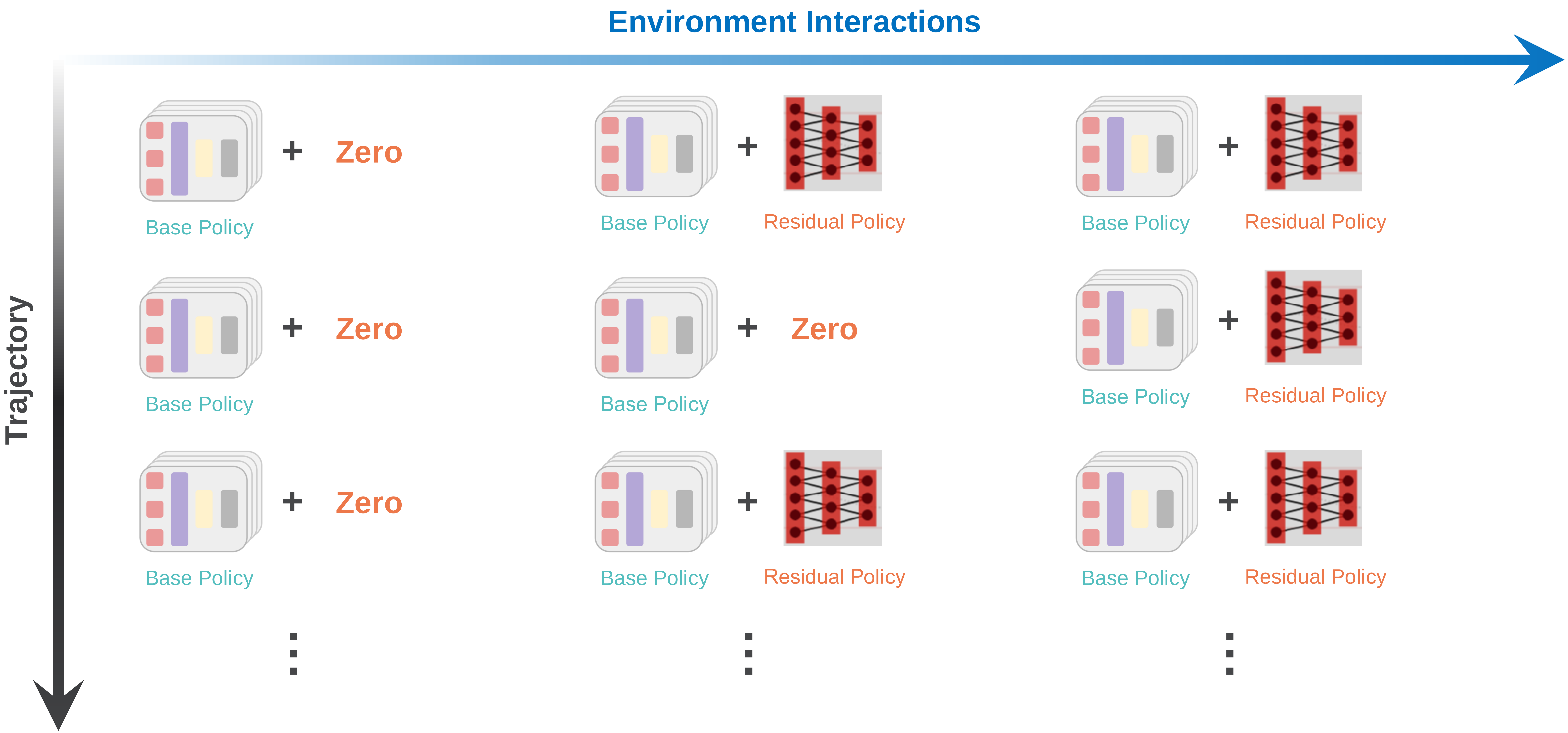
Enable Residual Actions w/ Probability ϵ

$$\pi(s) = \begin{cases} \pi_{base}(s) + \pi_{res}(s) & \text{Uniform}(0,1) < \epsilon \\ \pi_{base}(s) & \text{otherwise} \end{cases}$$

Progressively Increase ϵ



Strategy 2: Progressive Exploration Schedule



Environment Interactions

Trajectory

Base Policy + Zero | Base Policy + Residual Policy | Base Policy + Residual Policy

Base Policy + Zero | Base Policy + Zero | Base Policy + Residual Policy

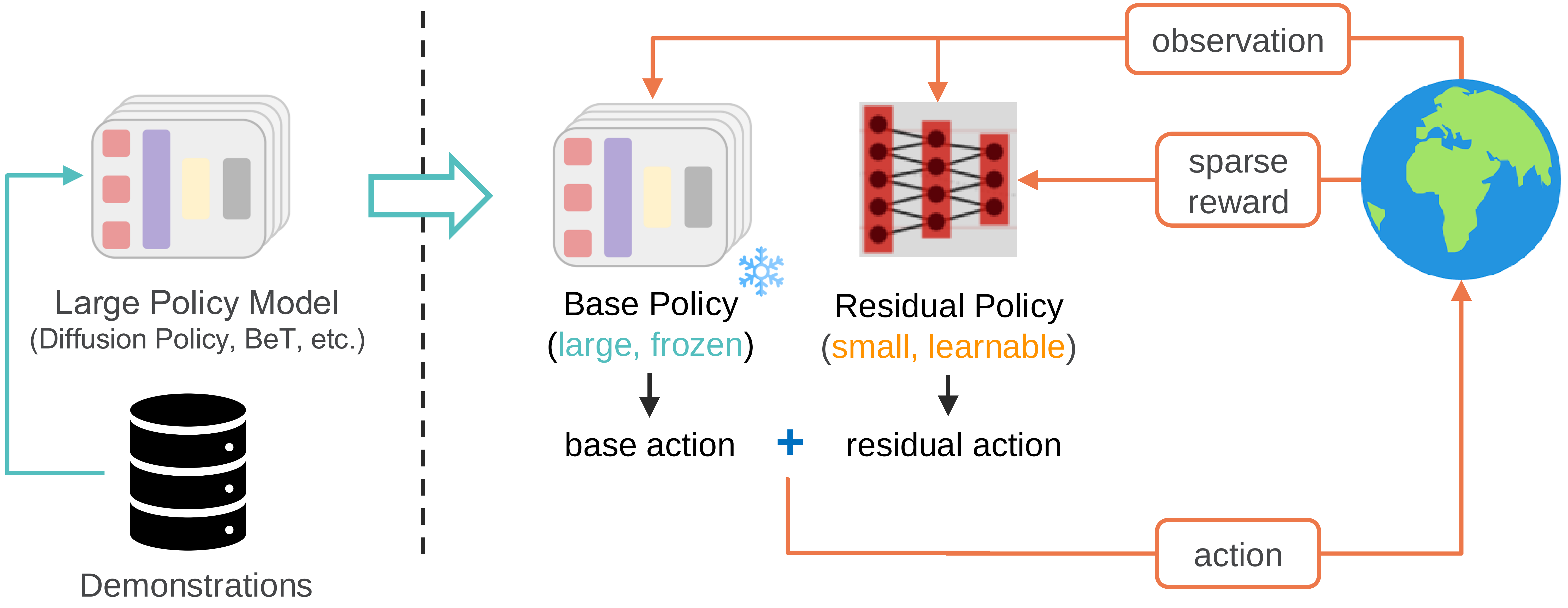
Base Policy + Zero | Base Policy + Residual Policy | Base Policy + Residual Policy

⋮ ⋮ ⋮

Residual Policy with Online RL

Imitation Learning (offline)

Reinforcement Learning (online)

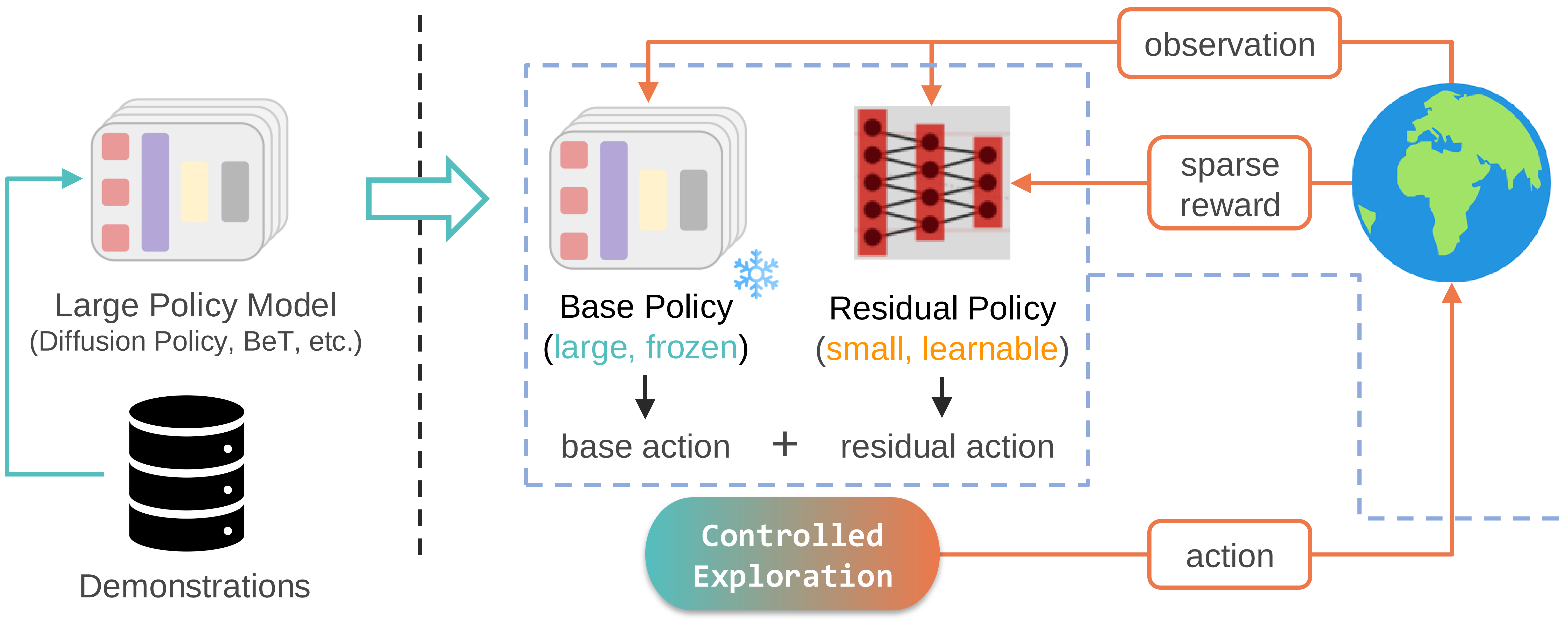


Res Policy Decorator RL

Imitation Learning (offline)

Reinforcement Learning (online)

Evaluation



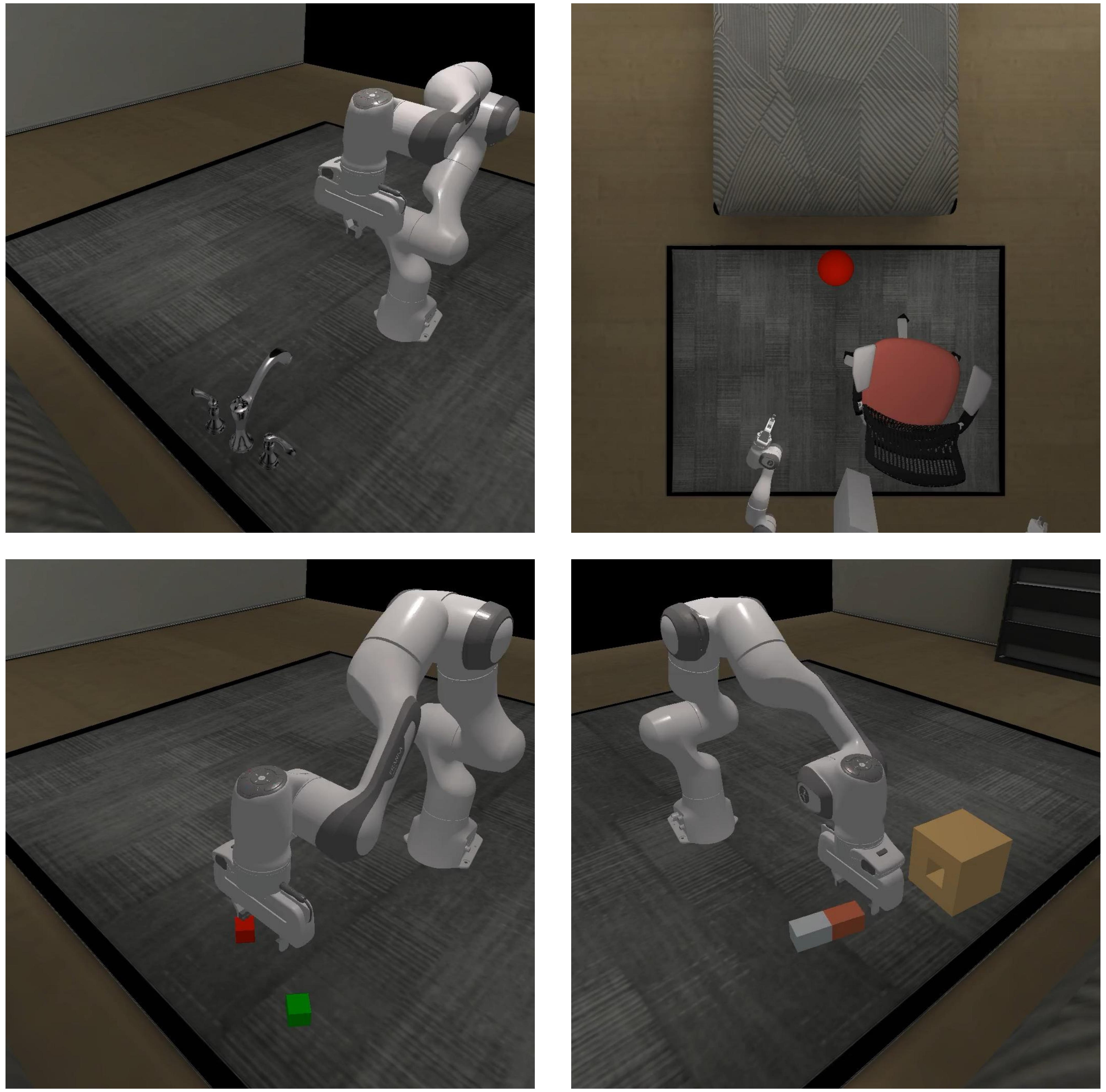
```
def policy_decorator(base_pi):
    def policy(obs):
        return base_pi(obs) +
residual_pi(obs)
    return policy

@policy_decorator
def base_policy(obs):
    ...
```

Model-Agnostic!

Evaluation on Diverse Tasks

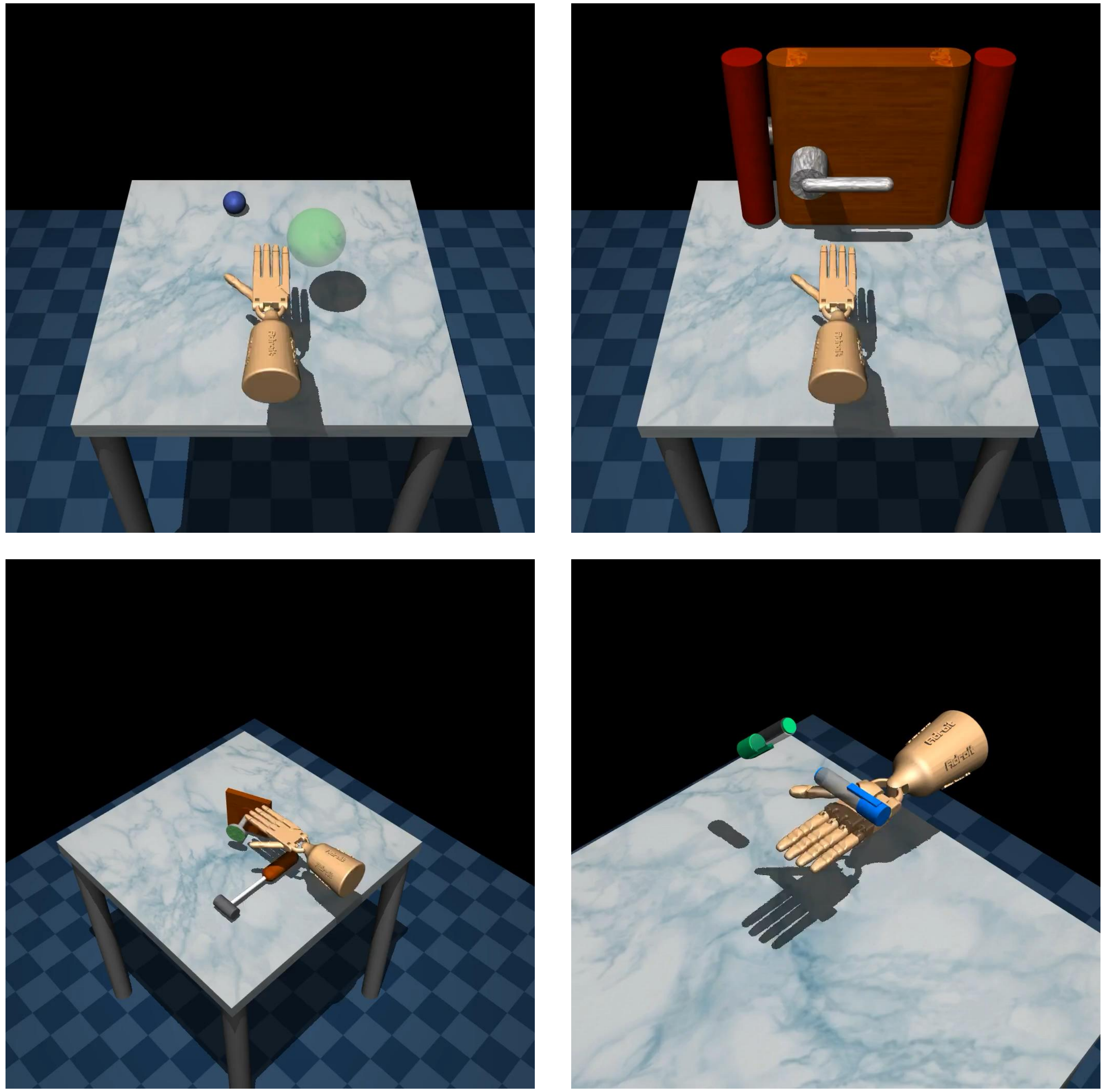
ManiSkill

Table-Top/Mobile, w/ Object Variations



Adroit

Dexterous Manipulation



Different Types of Strong Baselines

Fine-Tuning Methods

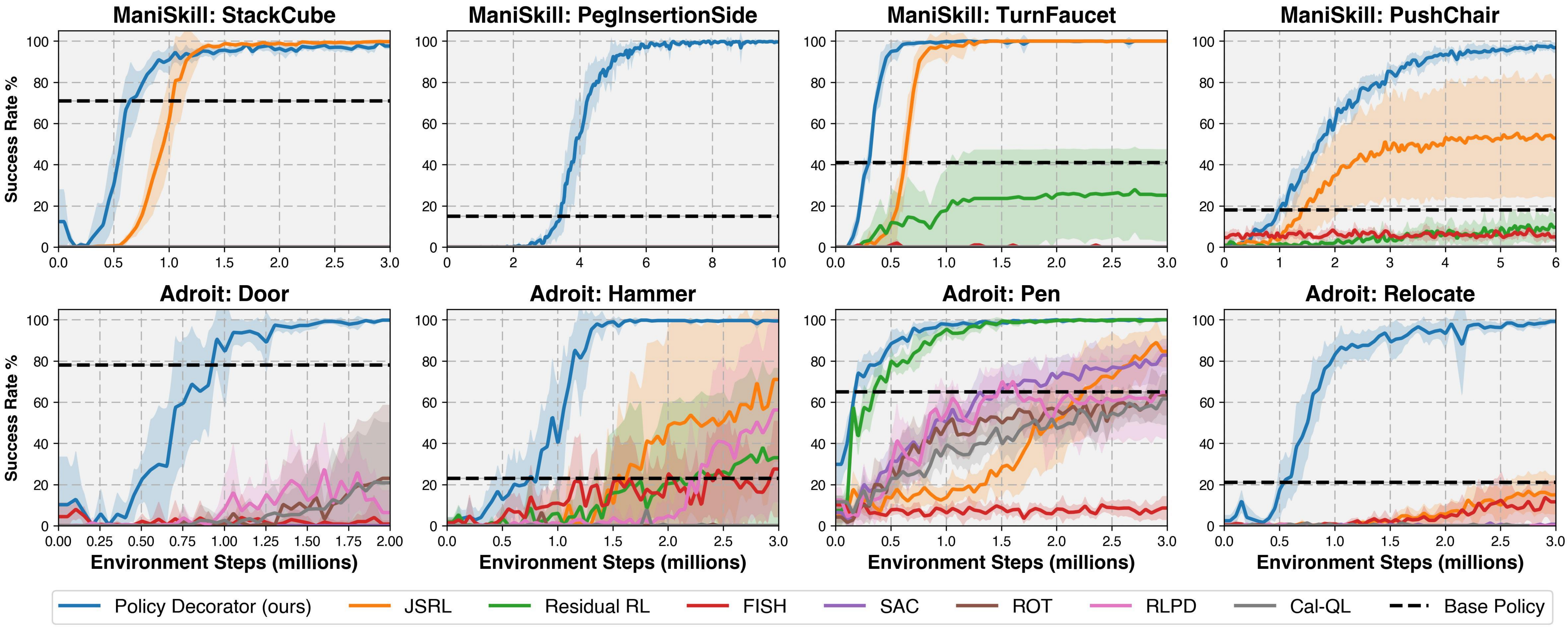
w/ A Lot of Special Designs + LoRA

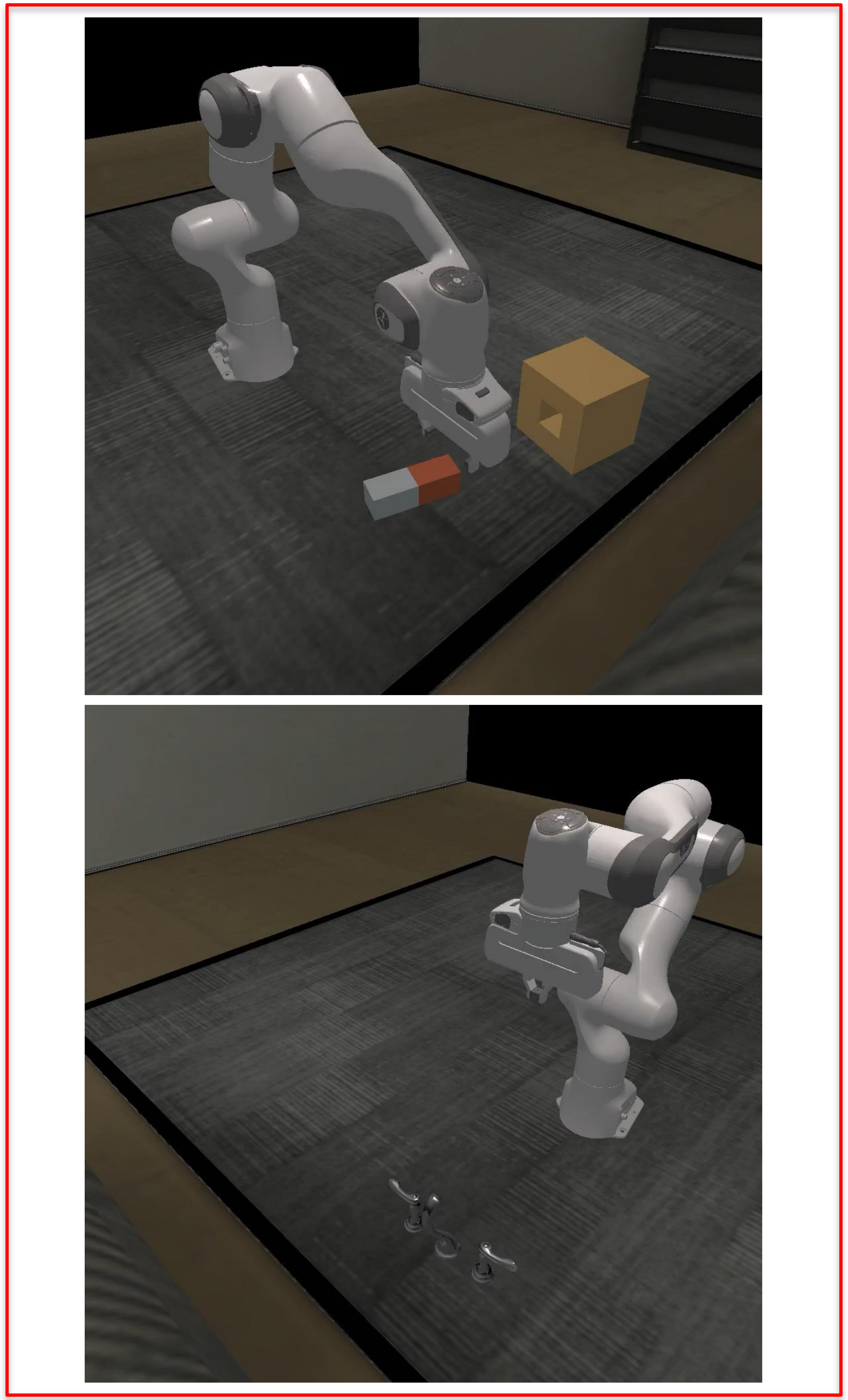
- **Basic RL**
  - SAC for Behavior Transformer
  - DIPO for Diffusion Policy
- **Boosting Basic RL with Demos**
  - Demo for Reward Learning: ROT
  - Demo as Off-Policy Experience: RLPD
  - Offline Value Pre-Training: Cal-QL

Non-Fine-Tuning Methods

- **Learning Residual Policy**
  - Vanilla Version: Residual RL
  - More Advanced Version: FISH
- **Utilize Base Policy to Build Curriculum**
  - JSRL

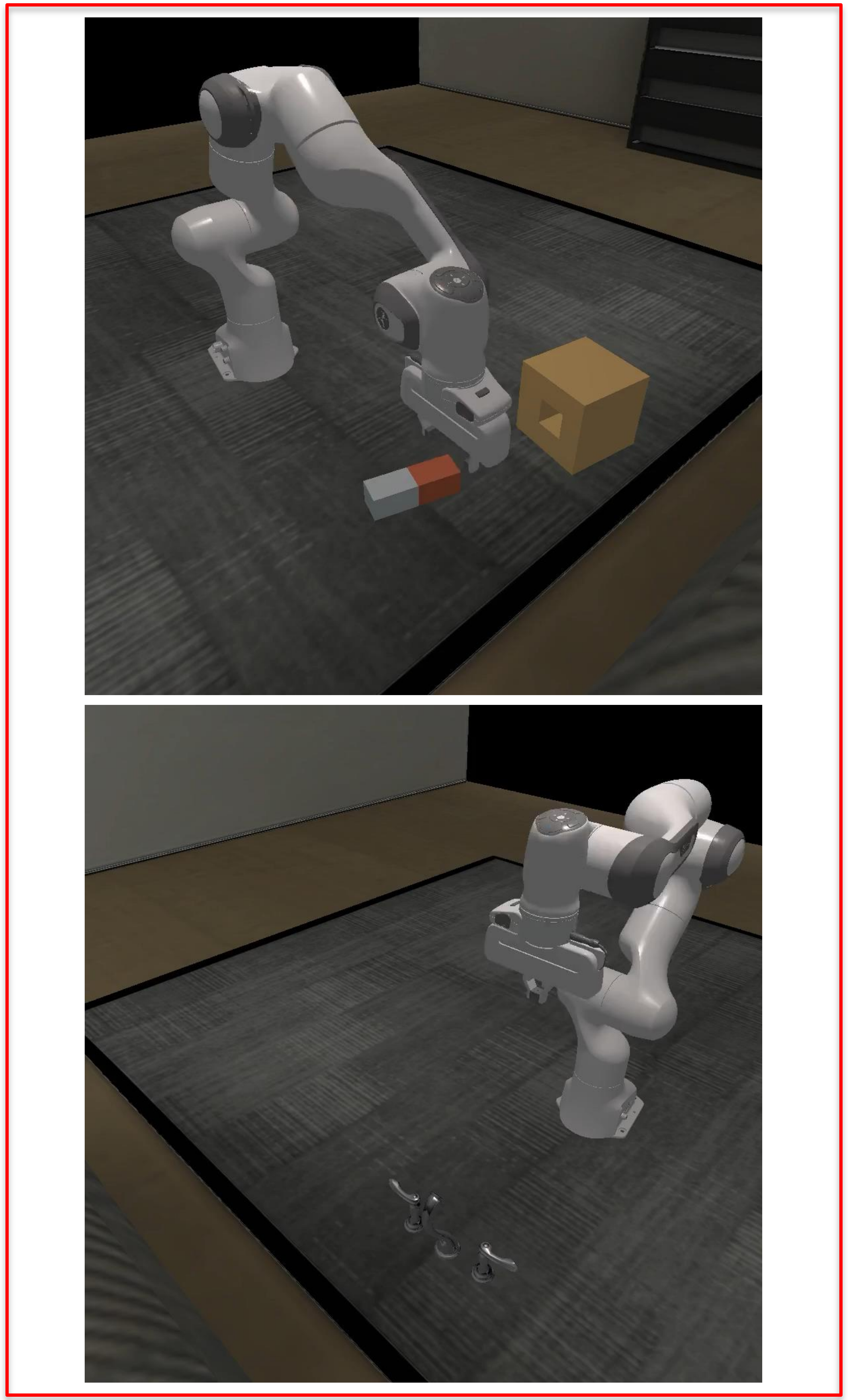
Effectively Improves Base Policies





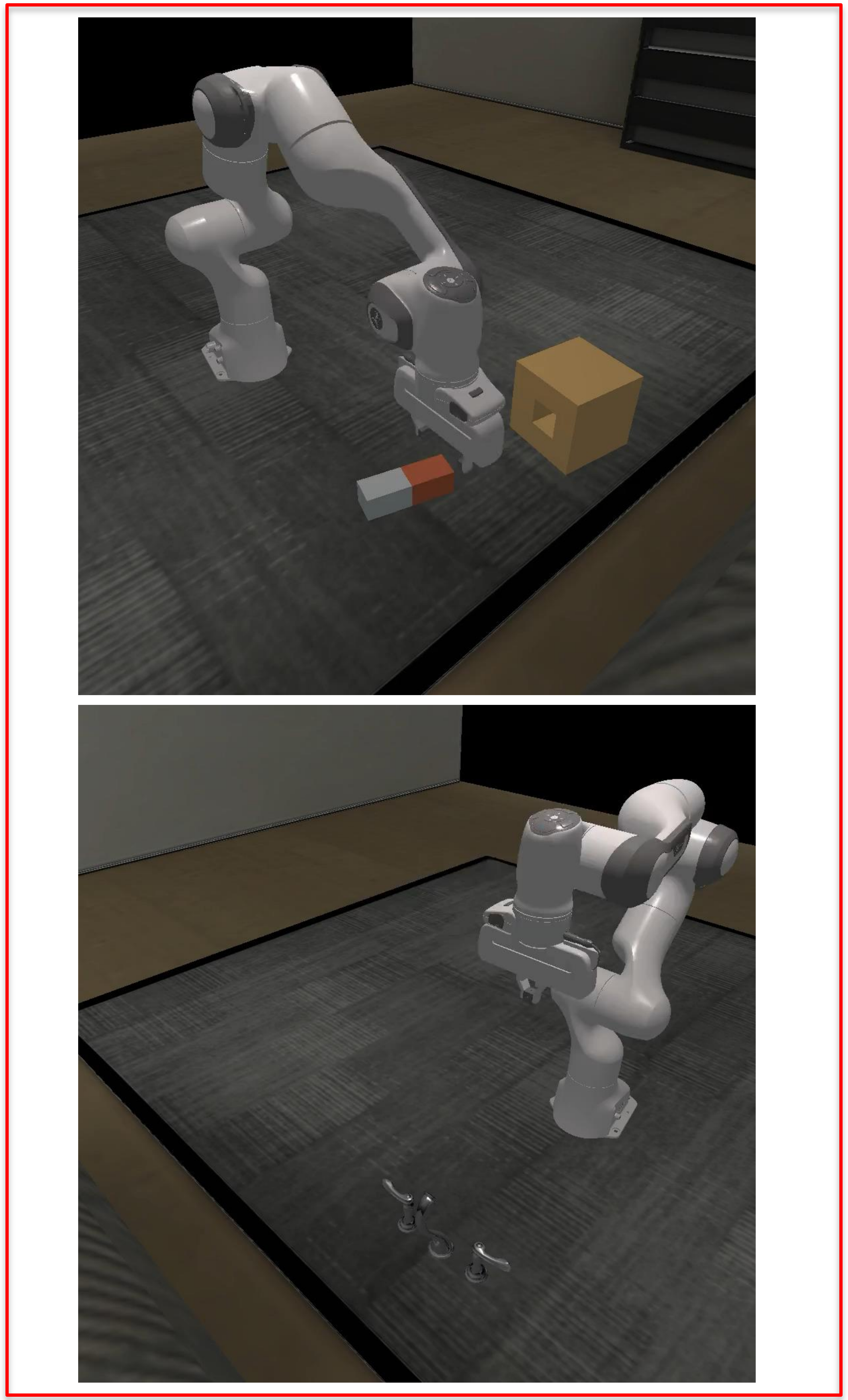
Base Policy

(w/o Online Learning)



Ours

(Base Policy + Online Residual)



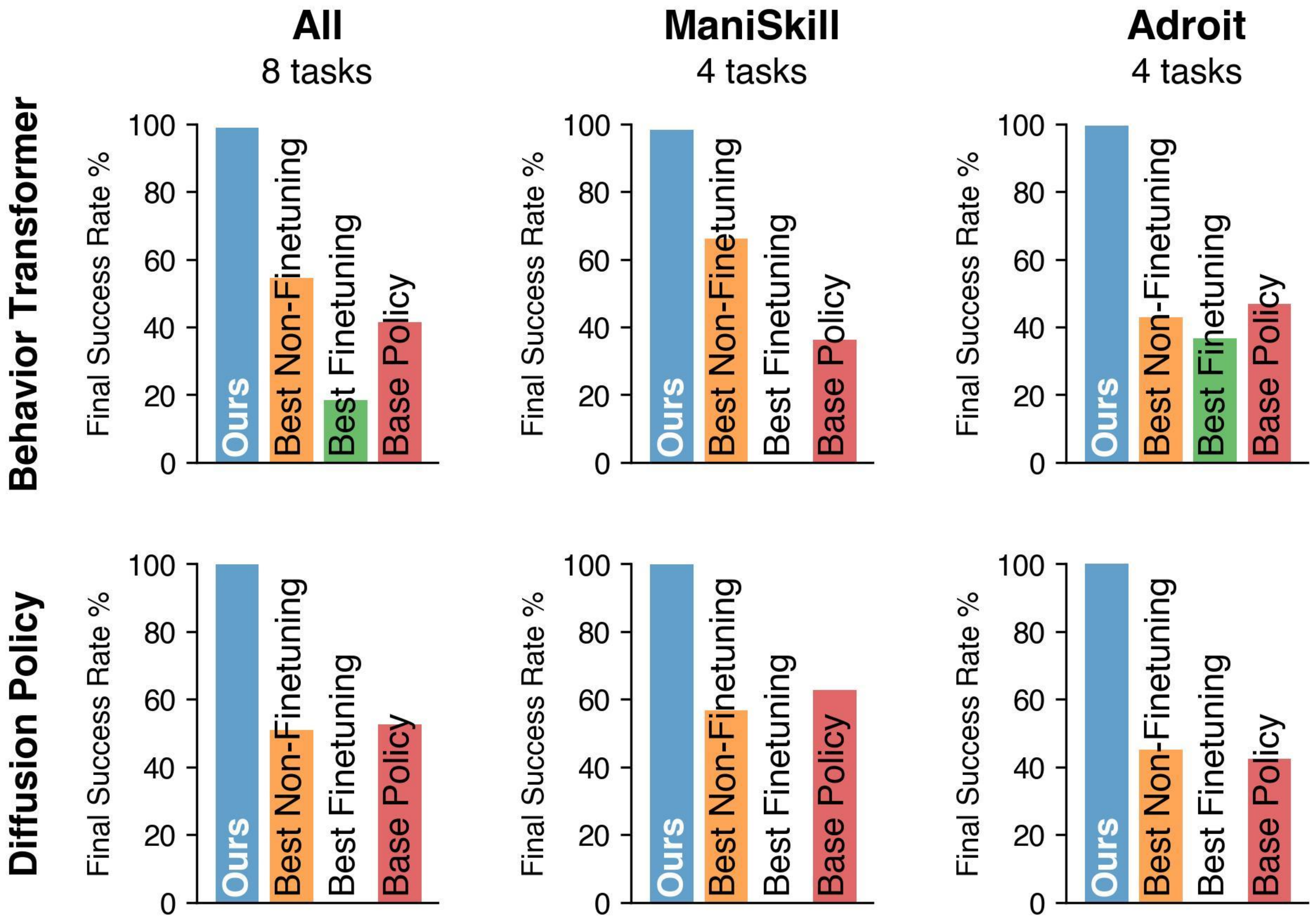
Online RL Policy

(w/o Base Policy)

Contributions

- `Policy Decorator`: A **model-agnostic** framework for refining large policy models through online learning.

- Effectively improve **2 SOTA large policy models** on **8 challenging robotic tasks**.



Thank you! 🥰