Multi-Task and Goal-Conditioned Reinforcement Learning

CS 224R

Reminders

5/17: Homework 3 is due; Homework 4 is out

The Plan

Recap

Multi-task imitation and policy gradients

Multi-task Q-learning

Goal-conditioned RL

Key learning goals:

- Familiarity with multi-task learning challenges
- Hindsight relabeling in goal-conditioned RL

The Plan

Recap

Multi-task imitation and policy gradients

Multi-task Q-learning

Goal-conditioned RL

Recap: CS224R so far

Fundamentals:

- Imitation
- On-policy, off-policy and offline RL
- Model-free and model-based RL
- Reward functions

Biggest challenge so far?

<u>Sample complexity</u>

Next two weeks:

- Amortize the data complexity across many tasks/scenarios
- Go beyond single task

The Plan

Recap

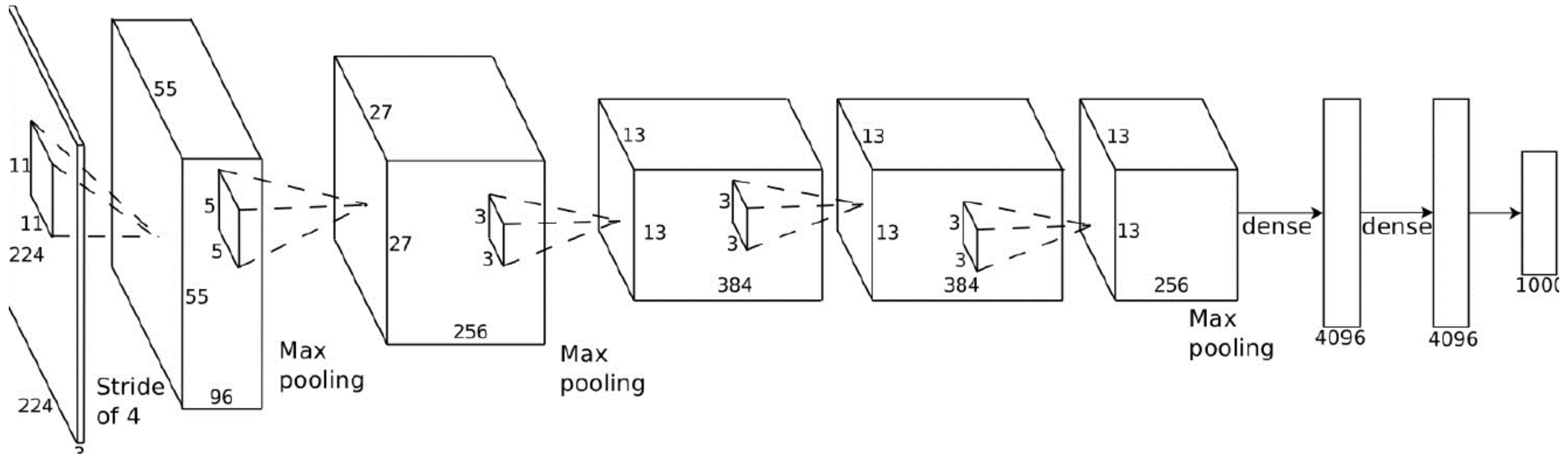
Multi-task imitation and policy gradients

Multi-task Q-learning

Goal-conditioned RL

Multi-task imitation learning

$\mathbf{o}_t$

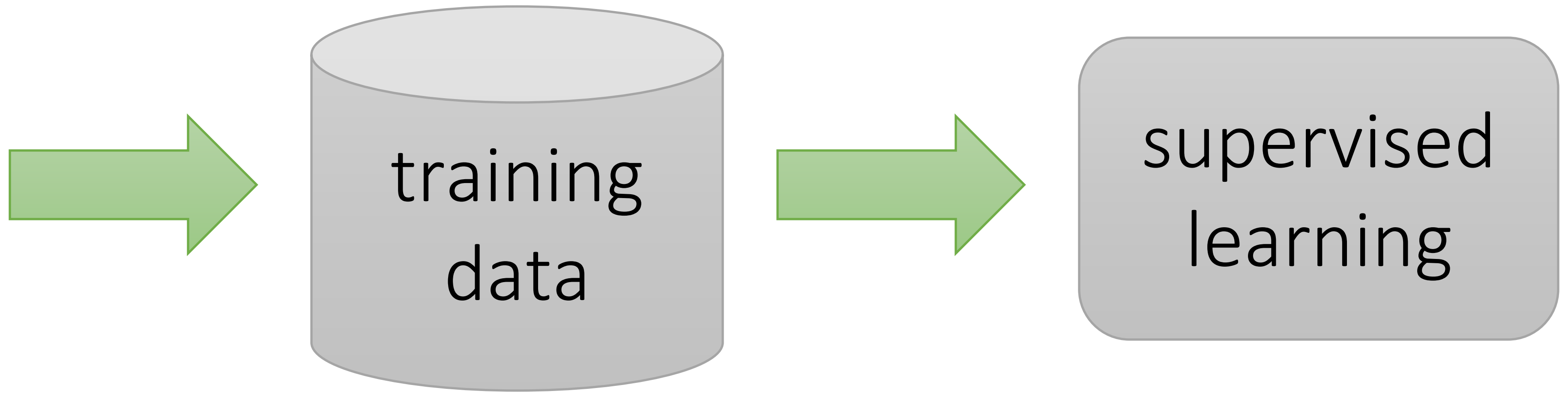


$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

$\mathbf{a}_t$

$\mathbf{o}_t$

$\mathbf{a}_t$



$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

Images: Bojarski et al. '16, NVIDIA

How to optimize multi-task IL?

$$\min_{\theta} -E_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}} \log \pi_\theta(\mathbf{a}|\mathbf{s})$$

Data: Given trajectories collected by an expert

"*demonstrations*" $\mathscr{D} := \{(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T)\}$

Training: Train policy to mimic expert: $\min_{\theta} - \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathscr{D}}[\log \pi_\theta(\mathbf{a}\,|\,\mathbf{s})]$

i.e. minimize cross-entropy loss or ℓ_2 loss between predicted & expert actions.

How to optimize multi-task IL?

$$\min_\theta \mathcal{L}(\theta, \mathcal{D}) \longrightarrow \min_\theta \sum_{i=1}^{T} \mathcal{L}(\theta, \mathcal{D}_i)$$

Same as supervised learning!

Same architectures, stratified sampling, etc.

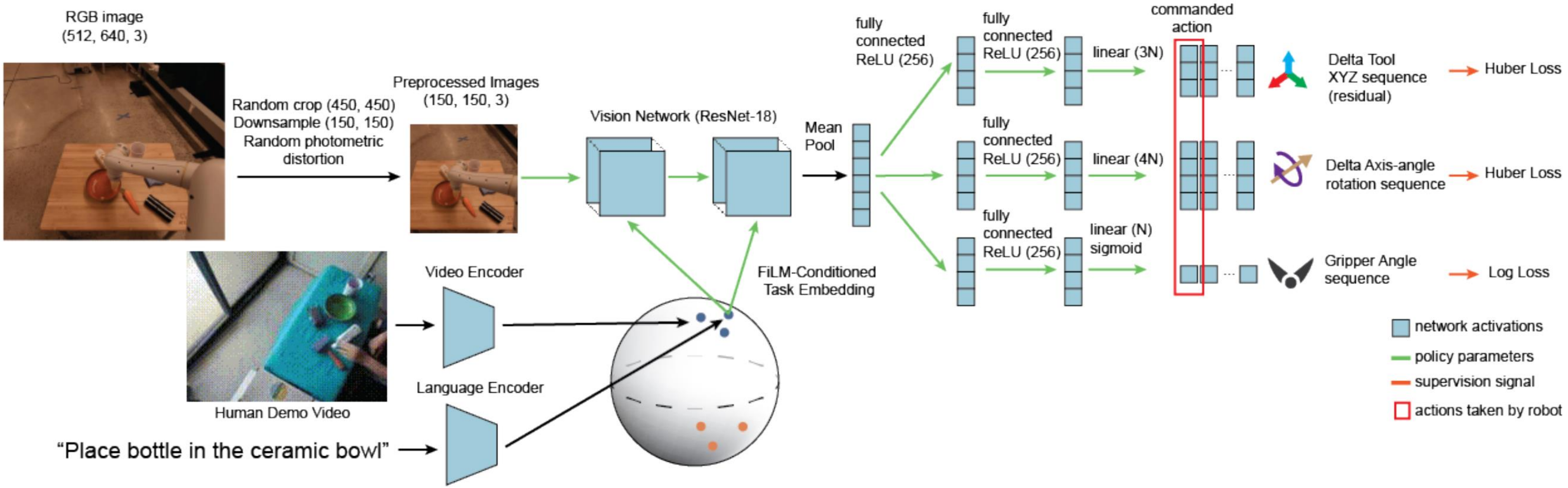
Data: Given trajectories collected by an expert

"*demonstrations*" $\mathscr{D} := \{(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T)\}$

Training: Train policy to mimic expert: $\min_\theta -\mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathscr{D}}[\log \pi_\theta(\mathbf{a}\,|\,\mathbf{s})]$

i.e. minimize cross-entropy loss or ℓ_2 loss between predicted & expert actions.

How to specify a task?



$$\min \sum_{\text{task } i} \sum_{\substack{(s,a)\sim\mathcal{D}_e^i \\ w_h\sim\mathcal{D}_h^i \cup \mathcal{D}_e^i}} \underbrace{-\log \pi(a|s, z^i)}_{\text{behavior cloning}} + \underbrace{D_{\cos}(z_h^i, z_\ell^i)}_{\text{language regression}} \text{, where } \underbrace{z_h^i \sim q(\cdot|w_h)}_{\text{video encoder}}, \underbrace{z_\ell^i \sim q(\cdot|w_\ell^i)}_{\text{language encoder}}$$

Jang et al. BC-Z. CoRL 2021

How to specify a task?

Skill	Held-out tasks (no demos during training)	Lang-conditioned performance
pick-place	'place sponge in tray'	82% (9.2)
	'place grapes in red bowl'	75% (10.8)
	'place apple in paper cup'	33% (12.2)
pick-wipe	'wipe tray with sponge'	0% (0)
pick-place	'place banana in ceramic bowl'	75% (9.7)
	'place bottle in red bowl'	75% (9.7)
	'place grapes in ceramic bowl'	70% (10.3)
	'place bottle in table surface'	50% (11.2)
	'place white sponge in purple bowl'	45% (11.2)
	'place white sponge in tray'	40% (11.0)
	'place apple in ceramic bowl'	20% (8.9)
	'place bottle in purple bowl'	20% (8.9)
	'place banana in ceramic cup'	0% (0)
	'place banana on white sponge'	0% (0)
	'place metal cup in red bowl'	0% (0)
grasp	'pick up grapes'	65% (10.7)
	'pick up apple'	55% (11.2)
	'pick up towel'	42.8% (18.7)
	'pick up pepper'	35% (10.7)
	'pick up bottle'	30% (10.3)
	'pick up the red bowl'	0% (0)
pick-drag	'drag grapes across the table'	14% (13.2)
pick-wipe	'wipe table surface with banana'	10% (6.7)
	'wipe tray with white sponge'	0% (0)
	'wipe ceramic bowl with brush'	0% (0)
push	'push purple bowl across the table'	30% (10.3)
	'push tray across the table'	25% (9.7)
	'push red bowl across the table'	0% (0)
	Holdout Task Overall	32%

Shows non-zero success for 20/28 hold-out tasks

Average 32% success over all 28 tasks

"Push purple bowl across the table"

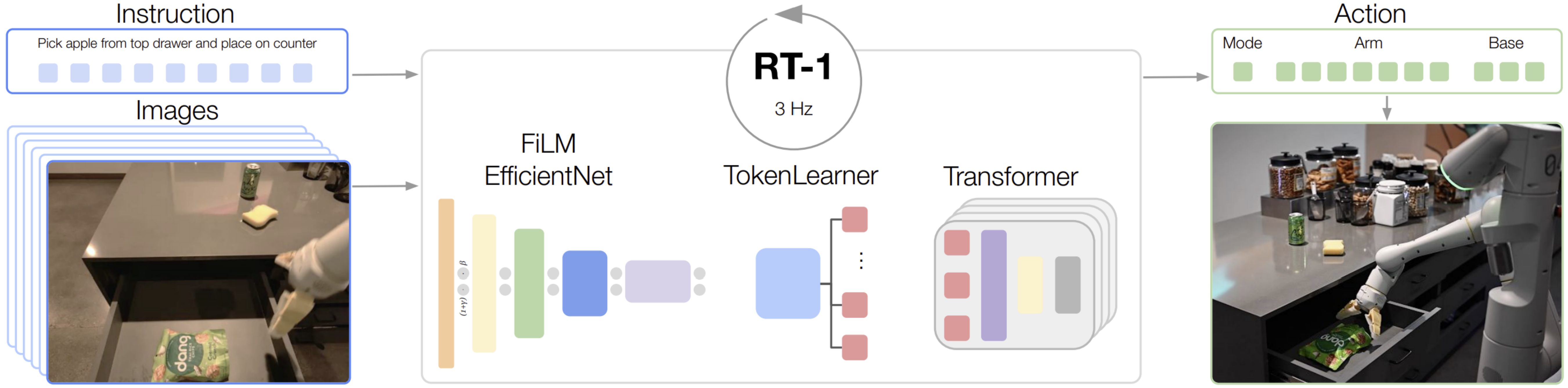
"Place bottle in tray"

Jang et al. BC-Z. CoRL 2021

Scaled-up version: Robotics Transformer (RT-1)

100%

RT-1 (ours)



(a) RT-1 takes images and natural language instructions and outputs discretized base and arm actions. Despite its size (35M parameters), it does this at 3 Hz, due to its efficient yet high-capacity architecture: a FiLM (Perez et al., 2018) conditioned EfficientNet (Tan & Le, 2019), a TokenLearner (Ryoo et al., 2021), and a Transformer (Vaswani et al., 2017).

Tasks

Brohan et al. RT-1, 2022

What is a reinforcement learning task?

Reinforcement learning

action space

dynamics

A task: $\mathcal{T}_i \triangleq \{\mathcal{S}_i, \mathcal{A}_i, p_i(\mathbf{s}_1), p_i(\mathbf{s}'|\mathbf{s},\mathbf{a}), r_i(\mathbf{s},\mathbf{a})\}$

state space

initial state distribution

reward

An alternative view:

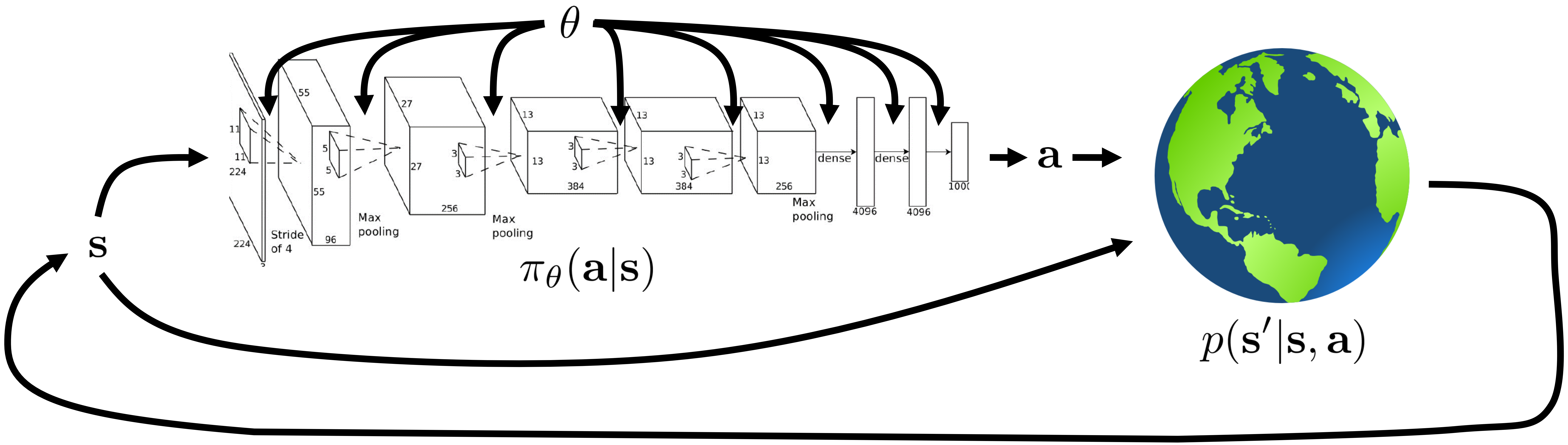
A task identifier is part of the state: $\mathbf{s} = (\bar{\mathbf{s}}, \mathbf{z}_i)$

original state

$$\mathcal{T}_i \triangleq \{\mathcal{S}_i, \mathcal{A}_i, p_i(\mathbf{s}_1), p(\mathbf{s}'|\mathbf{s},\mathbf{a}), r(\mathbf{s},\mathbf{a})\} \longrightarrow \{\mathcal{T}_i\} = \left\{\cup\mathcal{S}_i, \cup\mathcal{A}_i, \frac{1}{N}\sum_i p_i(\mathbf{s}_1), p(\mathbf{s}'|\mathbf{s},\mathbf{a}), r(\mathbf{s},\mathbf{a})\right\}$$

It can be cast as a standard Markov decision process!

The goal of multi-task reinforcement learning



Multi-task RL

The same as before, except:

a task identifier is part of the state: $\mathbf{s} = (\bar{\mathbf{s}}, \mathbf{z}_i)$

e.g. one-hot task ID

language description

desired goal state, $\mathbf{z}_i = \mathbf{s}_g$ ← "goal-conditioned RL"

What is the reward?

The same as before

Or, for goal-conditioned RL:

$$r(\mathbf{s}) = r(\bar{\mathbf{s}}, \mathbf{s}_g) = -d(\bar{\mathbf{s}}, \mathbf{s}_g)$$

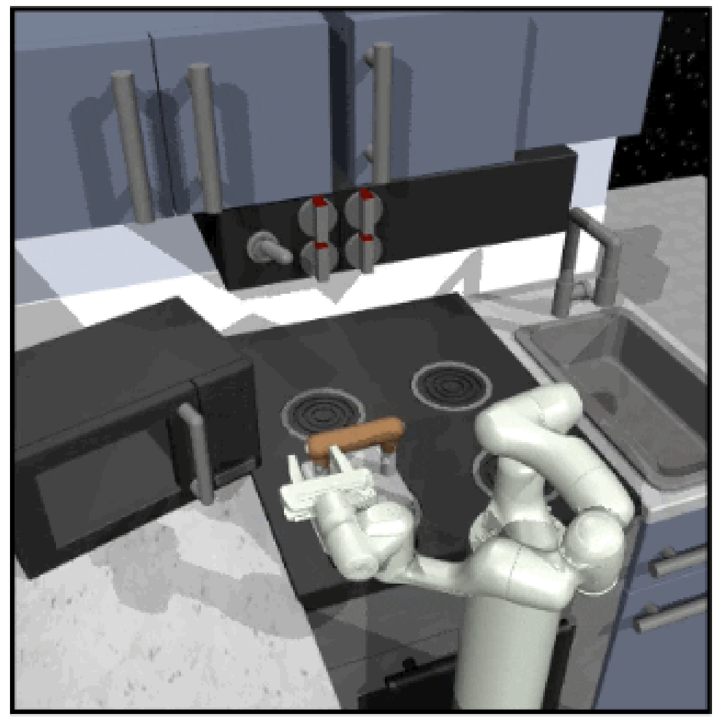
Distance function d examples:

- Euclidean ℓ_2
- sparse 0/1

Multi-task (RL) benefits

Cross-task generalization

Easier exploration



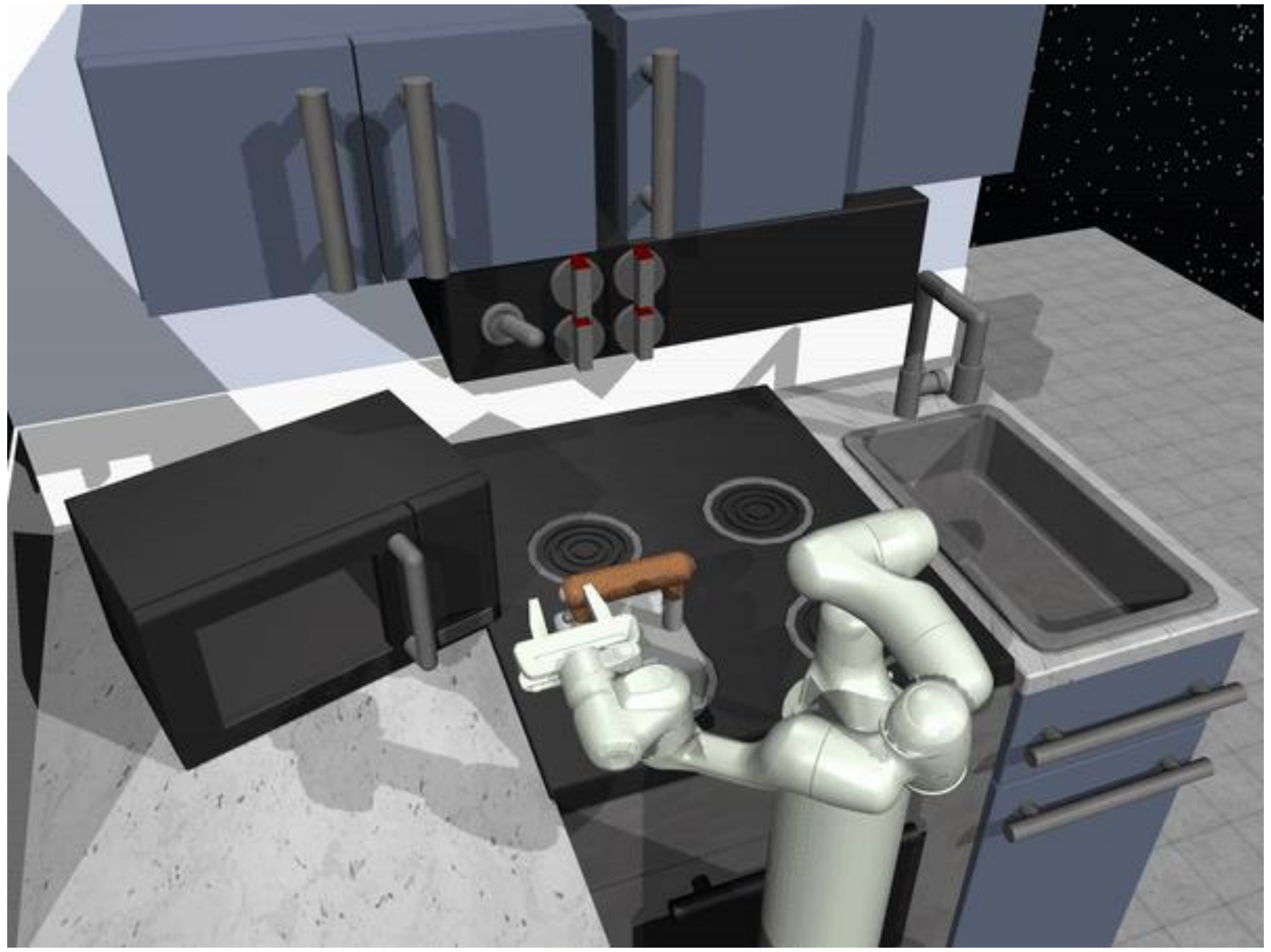
Pertsch et al. SPiRL

Multi-task (RL) benefits

Cross-task generalization

Easier exploration

Sequencing for long-horizon tasks



Gupta et al. Relay Policy Learning

Multi-task (RL) benefits

Cross-task generalization

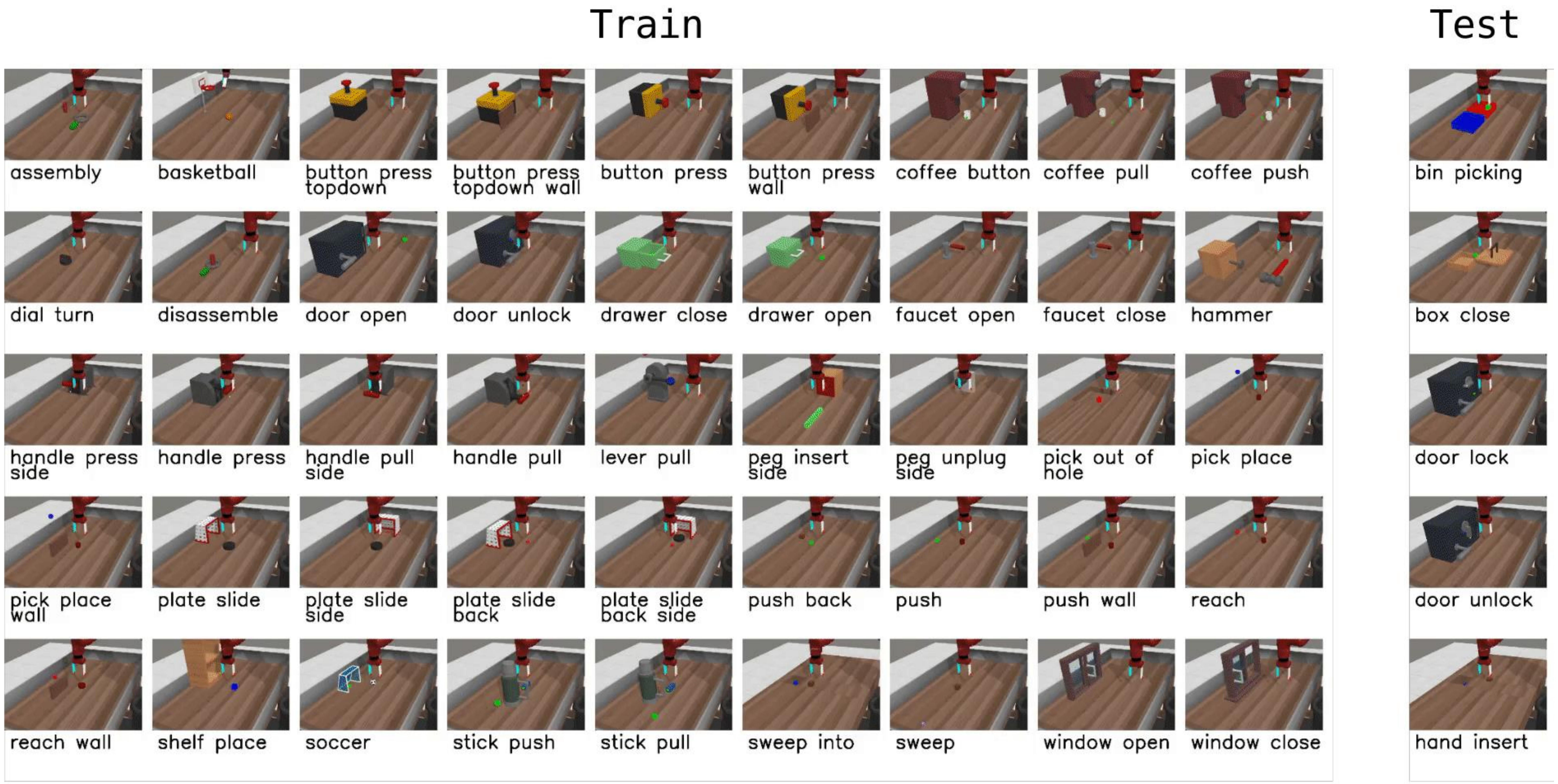
Easier exploration

Sequencing for long-horizon tasks

Reset-free learning

Per-task sample-efficiency gains

Multi-task RL benchmark: Meta-World



[Meta-World, Yu*, Quillen*, He*, et al., CoRL 2019]

Meta-world: why poor results?

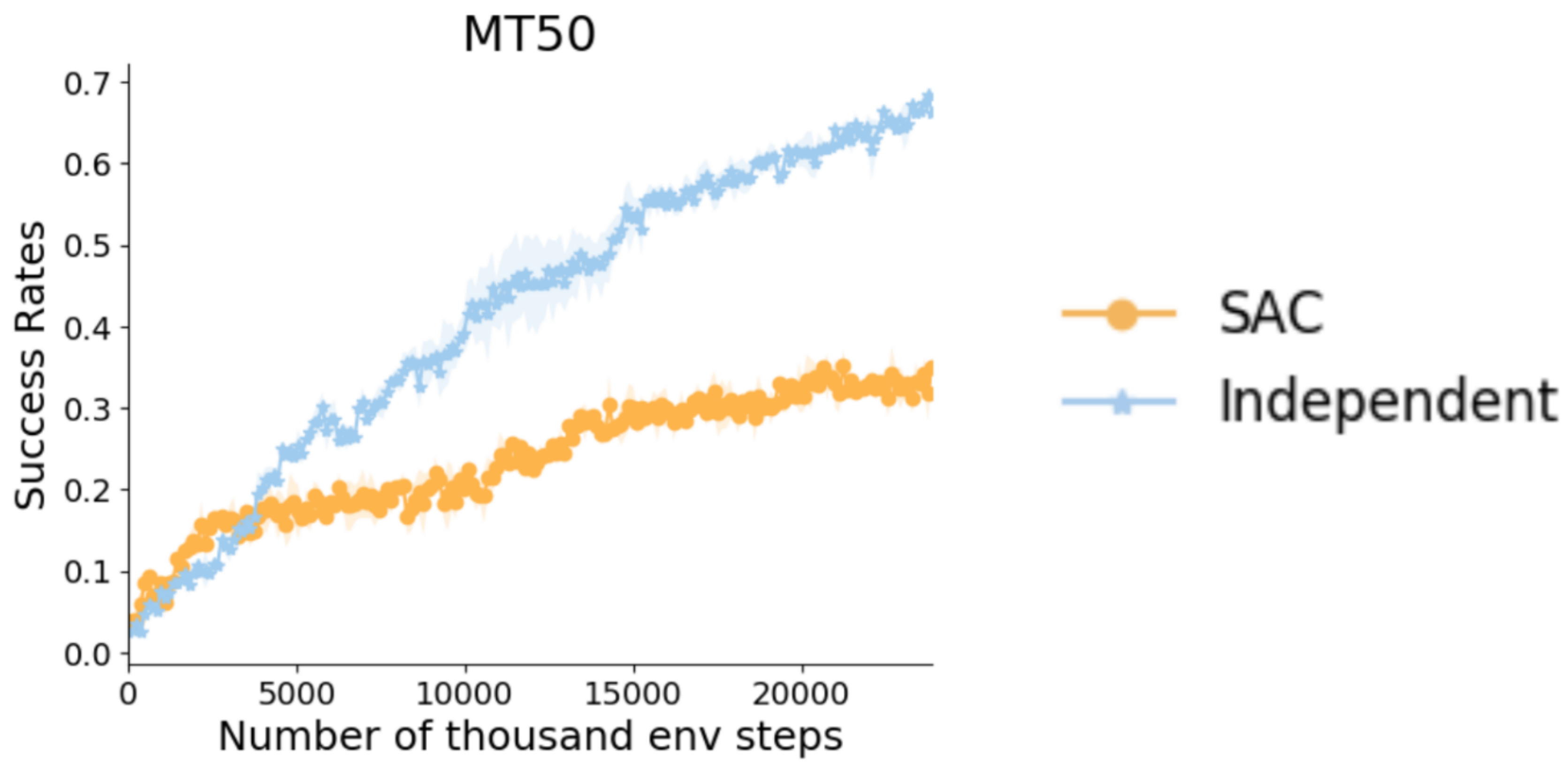
Methods	MT10	MT50
Multi-task PPO	25%	8.98%
Multi-task TRPO	29%	22.86%
Task embeddings	30%	15.31%
Multi-task SAC	39.5%	28.83%
Multi-task multi-head SAC	**88%**	**35.85%**

- Exploration?
- Data scarcity?
- Model capacity?

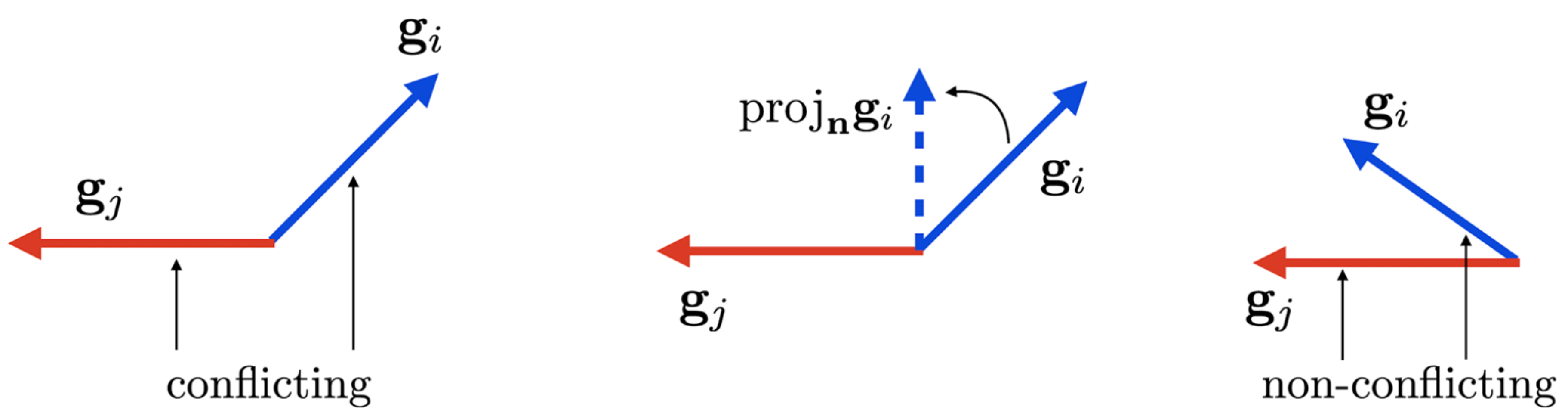
✓ All tasks are solvable individually
✓ Plenty of samples
✓ Plenty of capacity

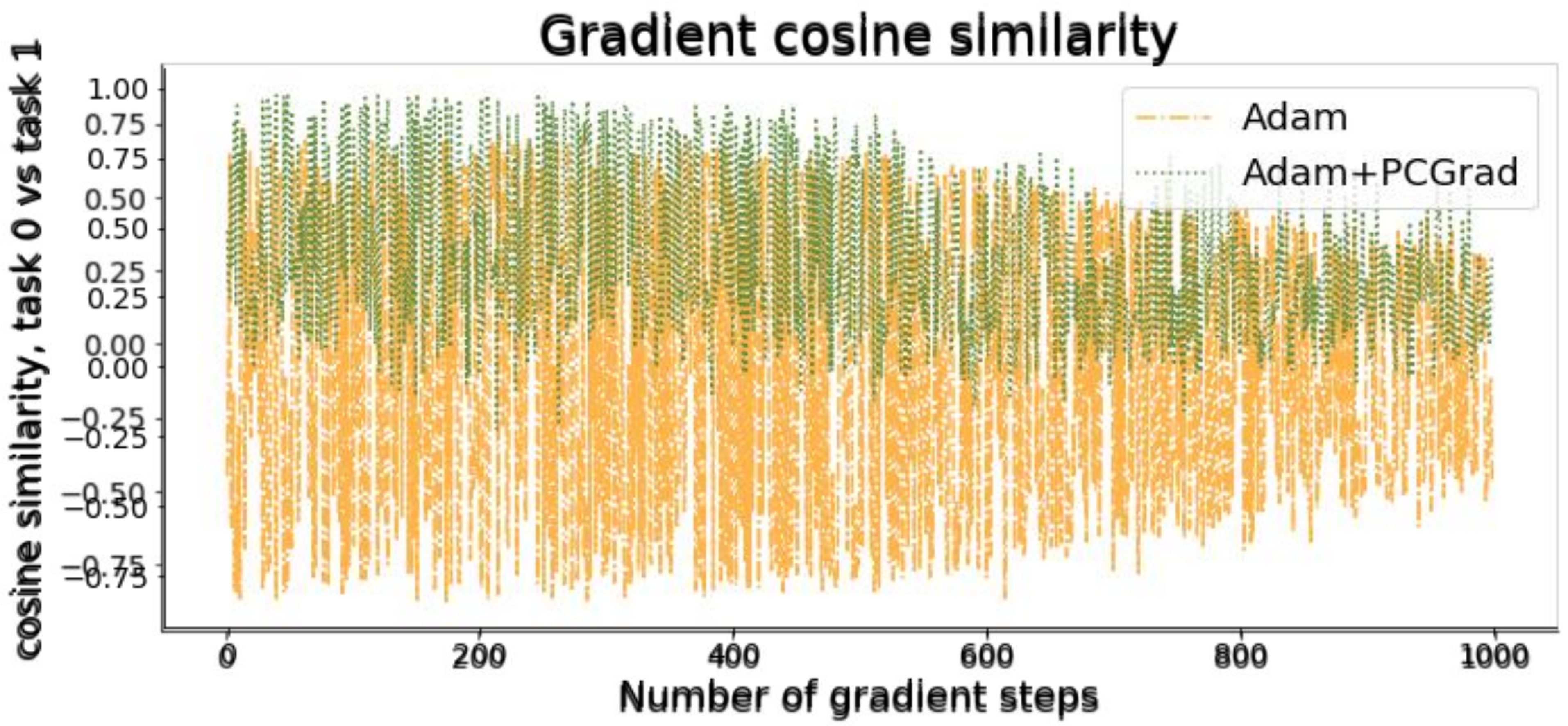
Optimization challenge?

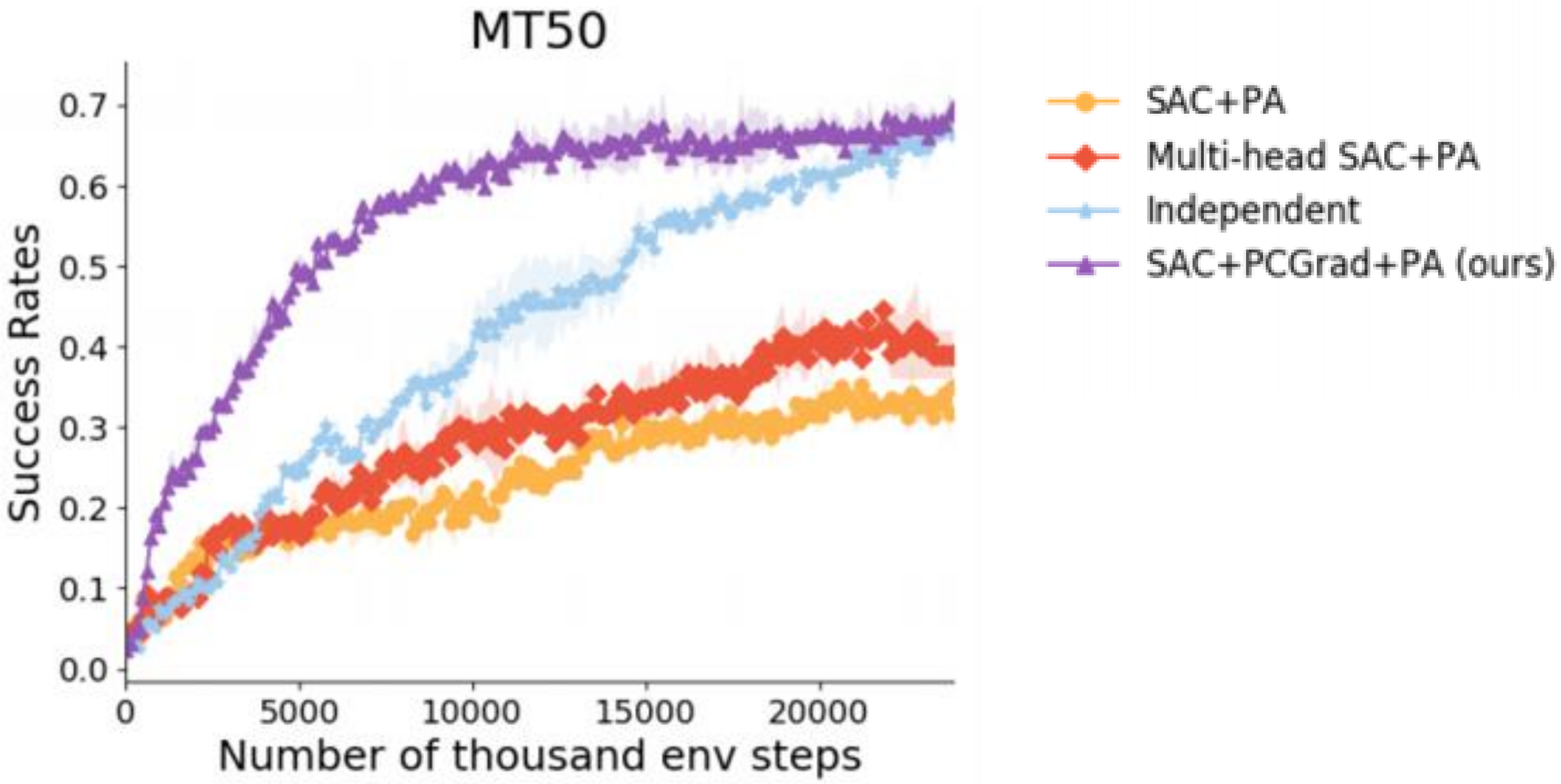
Multi-task (RL) difficulties



if two gradients conflict, project each onto the normal plane of the other, else, don’t do anything.







Yu et al. PCGrad. NeurIPS ‘20

Multi-task RL algorithms

Policy: $\pi_\theta(\mathbf{a}|\bar{\mathbf{s}})$ —> $\pi_\theta(\mathbf{a}|\bar{\mathbf{s}}, \mathbf{z}_i)$

Q-function: $Q_\phi(\bar{\mathbf{s}}, \mathbf{a})$ —> $Q_\phi(\bar{\mathbf{s}}, \mathbf{a}, \mathbf{z}_i)$

Analogous to multi-task supervised learning

If it's still a standard Markov decision process,

then, why not apply standard RL algorithms? You can! You can often do better.

What is different about reinforcement learning?

The data distribution is
controlled by the agent!

Should we share data in addition to sharing weights?

The Plan

Recap

Multi-task imitation and policy gradients

Multi-task Q-learning

Goal-conditioned RL

An example

Task 1: passing

Task 2: shooting goals

What if you accidentally perform a good pass when trying to shoot a goal?

Store experience as normal. *and* Relabel experience with task 1 ID & reward and store.

“hindsight relabeling” "hindsight experience replay” (HER)

Multi-task RL with relabeling

1. Collect data $\mathcal{D}_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{z}_i, r_{1:T})\}$ using some policy
2. Store data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_k$
3. Perform hindsight relabeling:

$k++$

   a. Relabel experience in $\mathcal{D}_k$ for task $\mathcal{T}_j$:
      $\mathcal{D}'_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{z}_j, r'_{1:T}\}$ where $r'_t = r_j(\mathbf{s}_t)$

   b. Store relabeled data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}'_k$

4. Update policy using replay buffer $\mathcal{D}$

<— Which task $\mathcal{T}_j$ to choose?
- randomly
- task(s) in which the trajectory gets high reward
- other

Eysenbach et al. Rewriting History with Inverse RL
Li et al. Generalized Hindsight for RL
Kalashnikov et al. MT-Opt
Yu et al. Conservative Data-Sharing

When can we apply relabeling?
- reward function form is known, evaluatable
- dynamics consistent across goals/tasks
- using an off-policy algorithm*

Kaelbling. Learning to Achieve Goals. IJCAI ‘93
Andrychowicz et al. Hindsight Experience Replay. NeurIPS ‘17

Another example:

Task 1: close a drawer

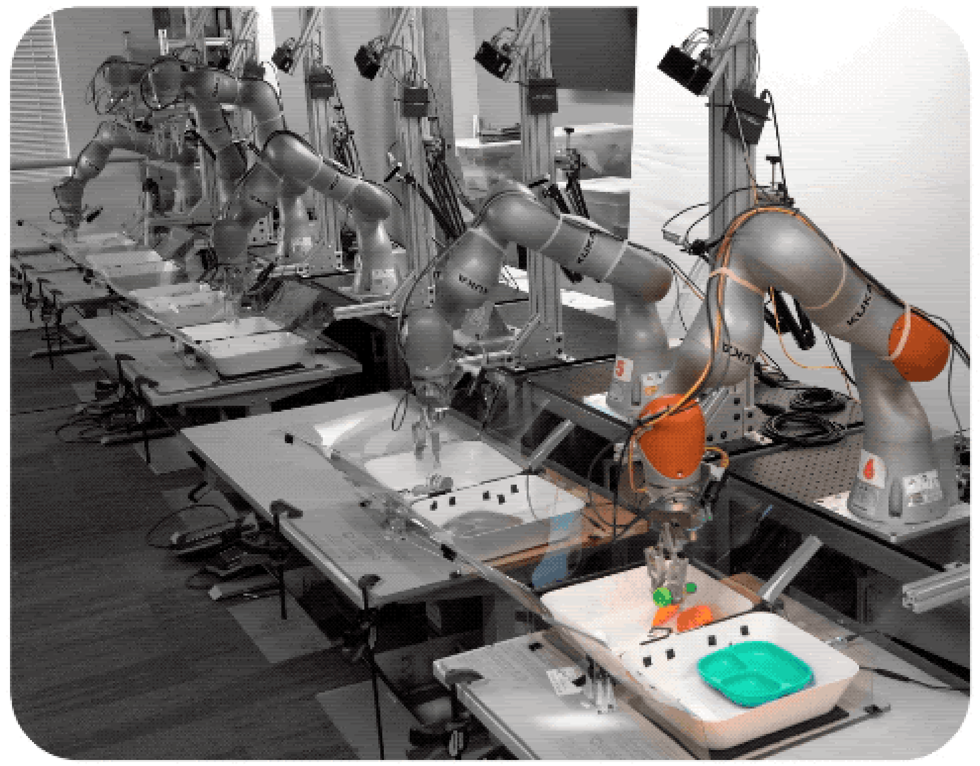
Task 2: open a drawer

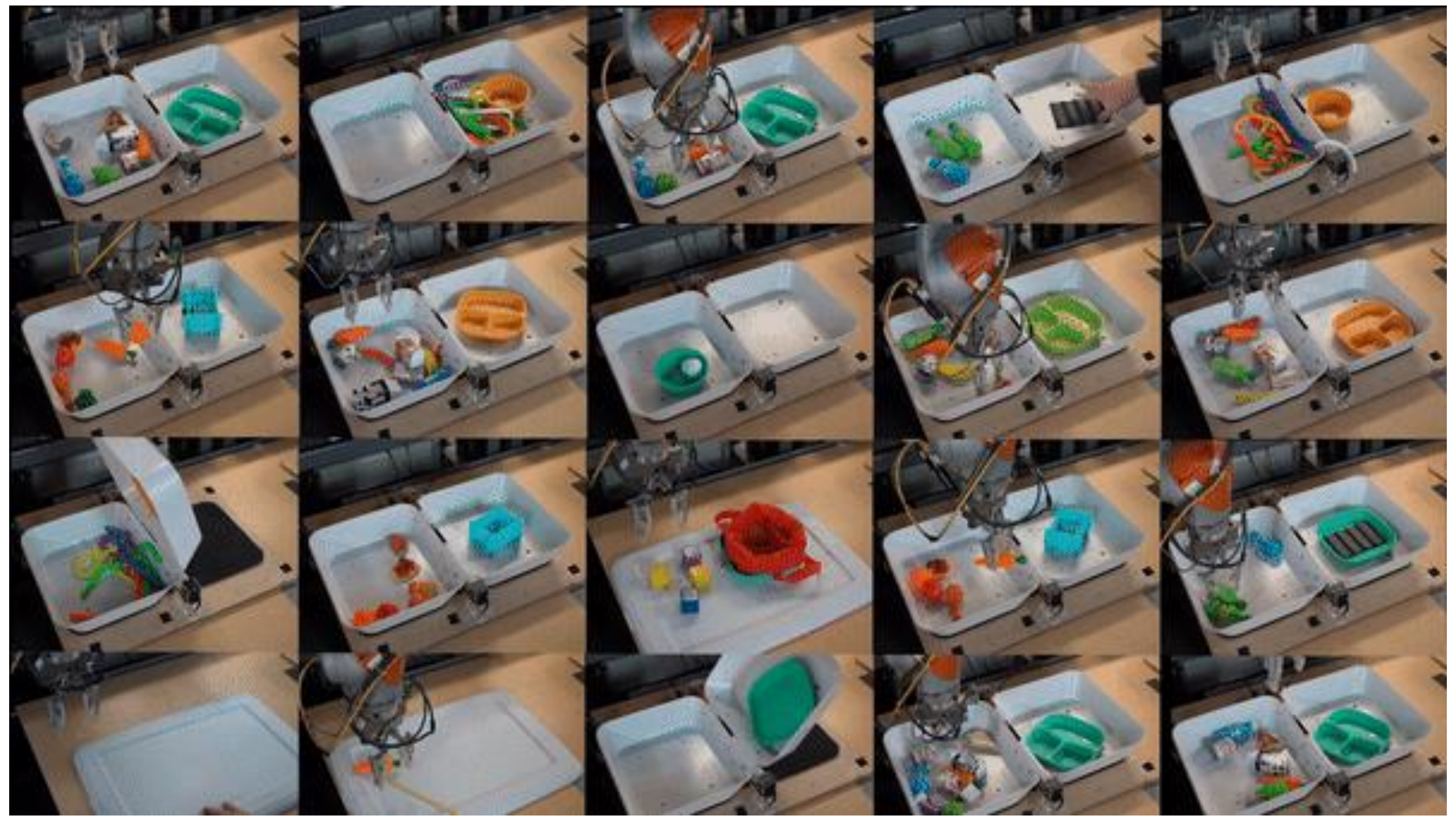
Can we use episodes from drawer opening task for drawer closing task?

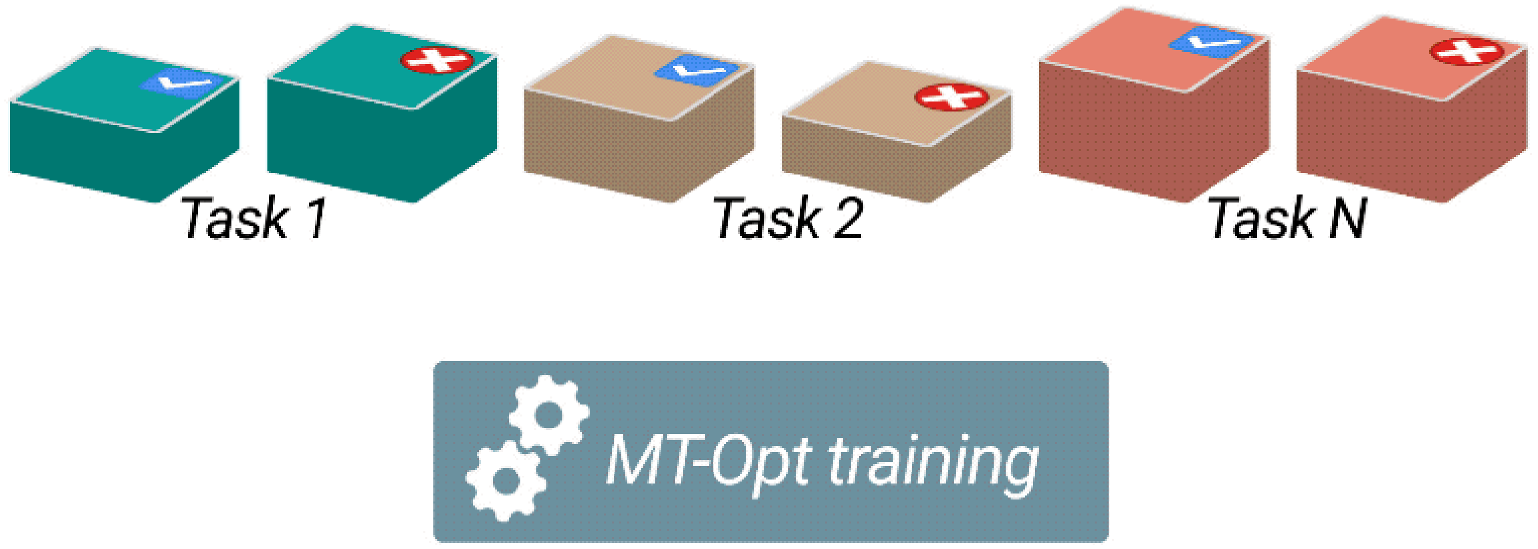
How does that answer change for Q-learning vs Policy Gradient?

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \right) \left(\sum_{t=1}^{T} r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right)$$

Example of multi-task Q-learning: MT-Opt



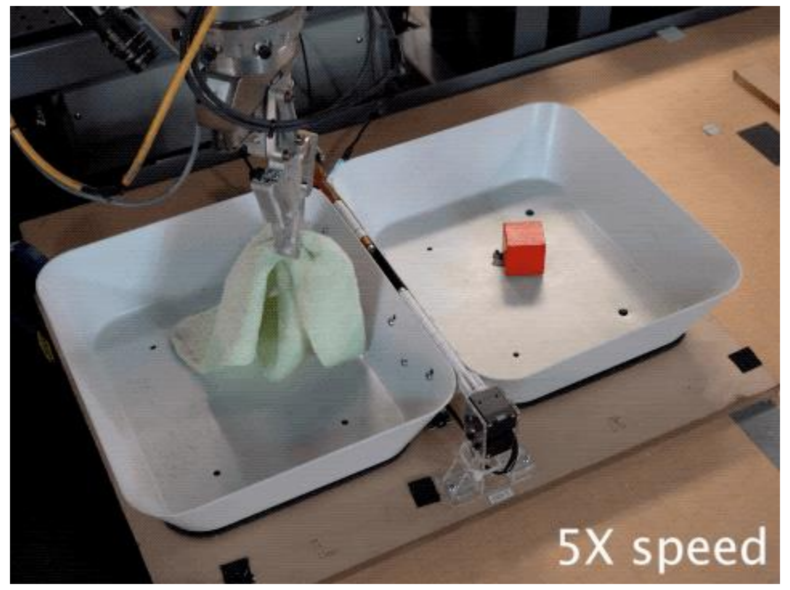




80% avg improvement over baselines across all the ablation tasks (4x improvement over single-task)

~4x avg improvement for tasks with little data

Fine-tunes to a new task (to 92% success) in 1 day



Kalashnikov et al. MT-Opt. CoRL '21

The Plan

Recap

Multi-task imitation and policy gradients

Multi-task Q-learning

Goal-conditioned RL

Goal-conditioned RL with hindsight relabeling

$k++$

1. Collect data $\mathcal{D}_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{s}_g, r_{1:T})\}$ using some policy
2. Store data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_k$
3. Perform hindsight relabeling:
   a. Relabel experience in $\mathcal{D}_k$ using last state as goal:
   $\mathcal{D}'_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{s}_T, r'_{1:T}\}$ where $r'_t = -d(\mathbf{s}_t, \mathbf{s}_T)$
   b. Store relabeled data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}'_k$
4. Update policy using replay buffer $\mathcal{D}$

Result: exploration challenges alleviated

Kaelbling. Learning to Achieve Goals. IJCAI '93
Andrychowicz et al. Hindsight Experience Replay. NeurIPS '17

Goal-conditioned RL with hindsight relabeling

$k++$

1. Collect data $\mathcal{D}_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{s}_g, r_{1:T})\}$ using some policy
2. Store data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_k$
3. Perform hindsight relabeling:
   a. Relabel experience in $\mathcal{D}_k$ using last state as goal:
      $\mathcal{D}'_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{s}_T, r'_{1:T}\}$ where $r'_t = -d(\mathbf{s}_t, \mathbf{s}_T)$
   b. Store relabeled data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}'_k$
4. Update policy using replay buffer $\mathcal{D}$

<— Other relabeling strategies?

use any state from the trajectory

Result: exploration challenges alleviated

Kaelbling. Learning to Achieve Goals. IJCAI '93
Andrychowicz et al. Hindsight Experience Replay. NeurIPS '17

Hindsight relabeling for goal-conditioned RL

Example: goal-conditioned RL, simulated robot manipulation

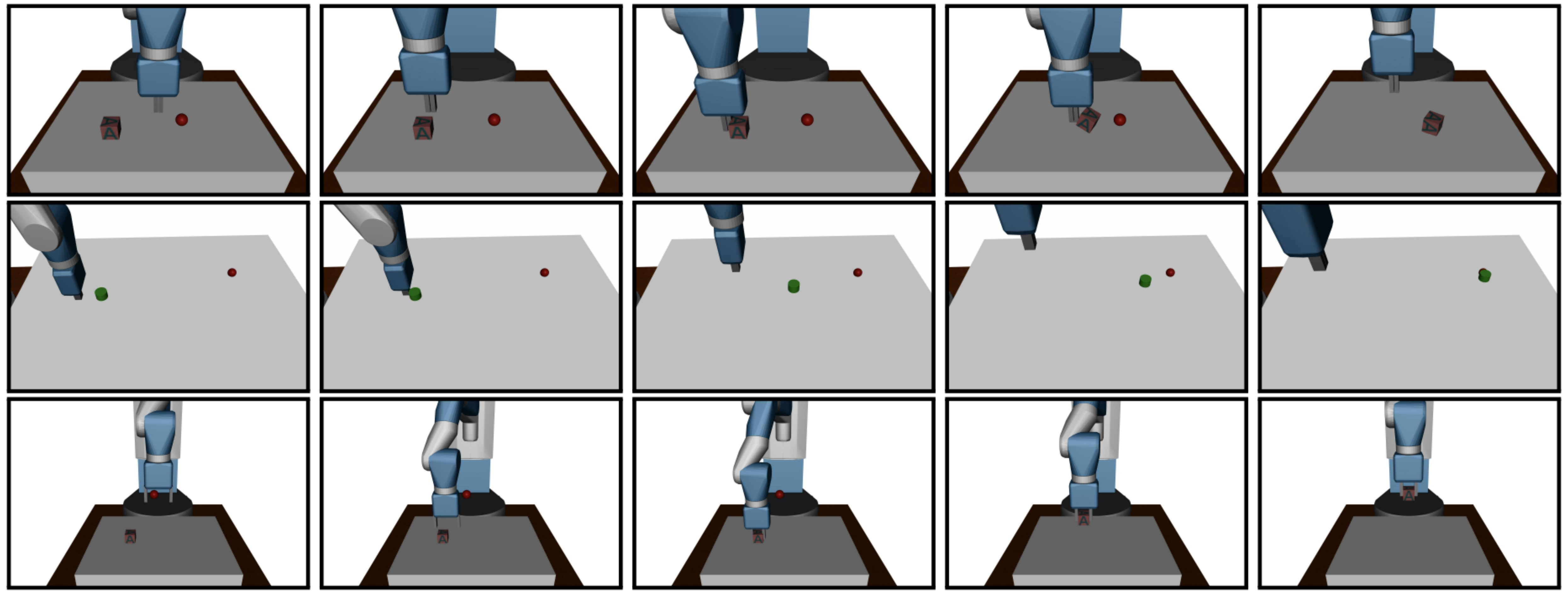
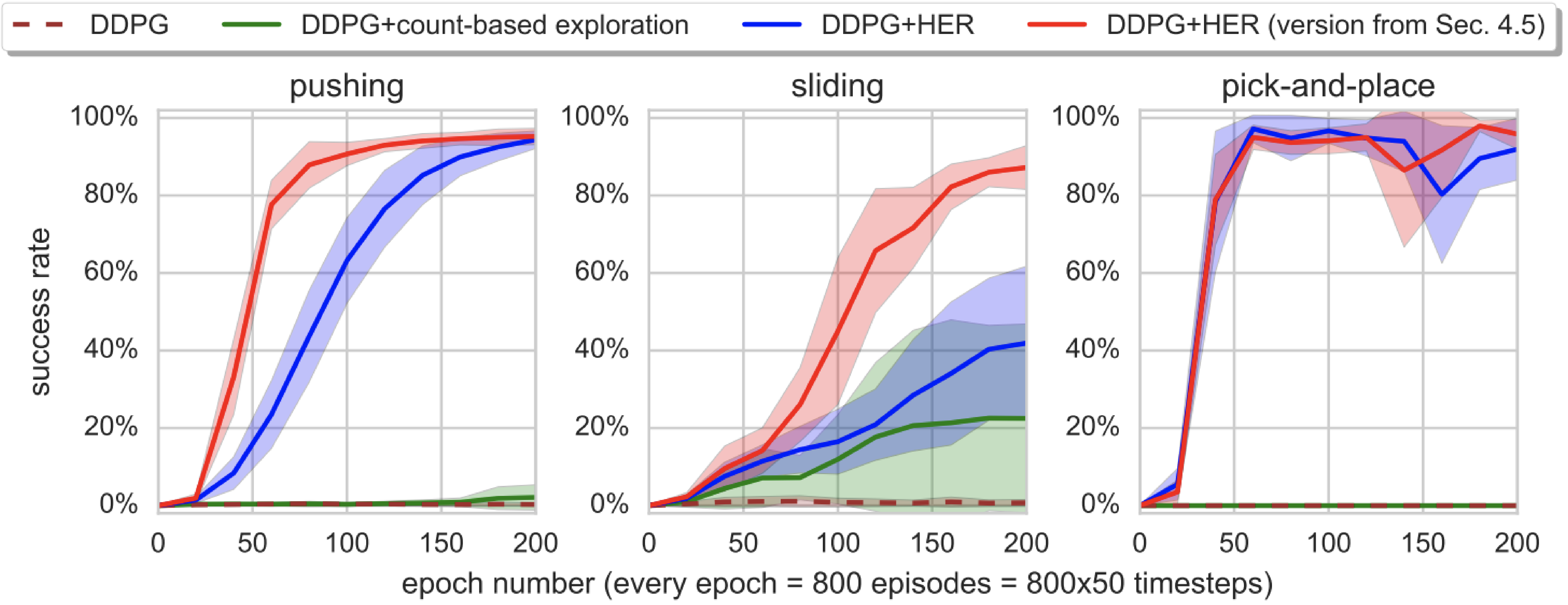


Figure 2: Different tasks: *pushing* (top row), *sliding* (middle row) and *pick-and-place* (bottom row). The red ball denotes the goal position.



Kaelbling. Learning to Achieve Goals. IJCAI ‘93
Andrychowicz et al. Hindsight Experience Replay. NeurIPS ‘17

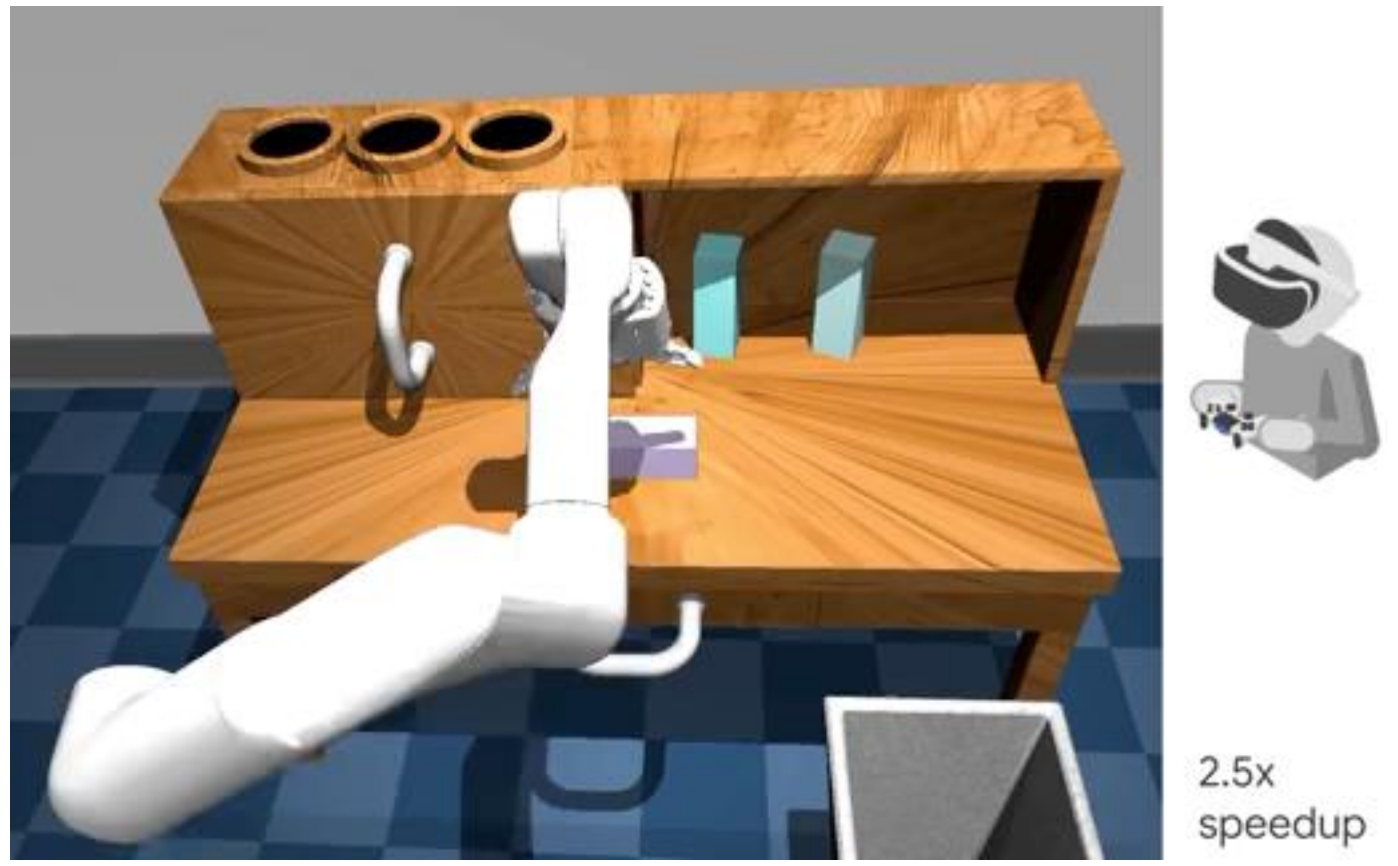
Can we use this insight for better learning?

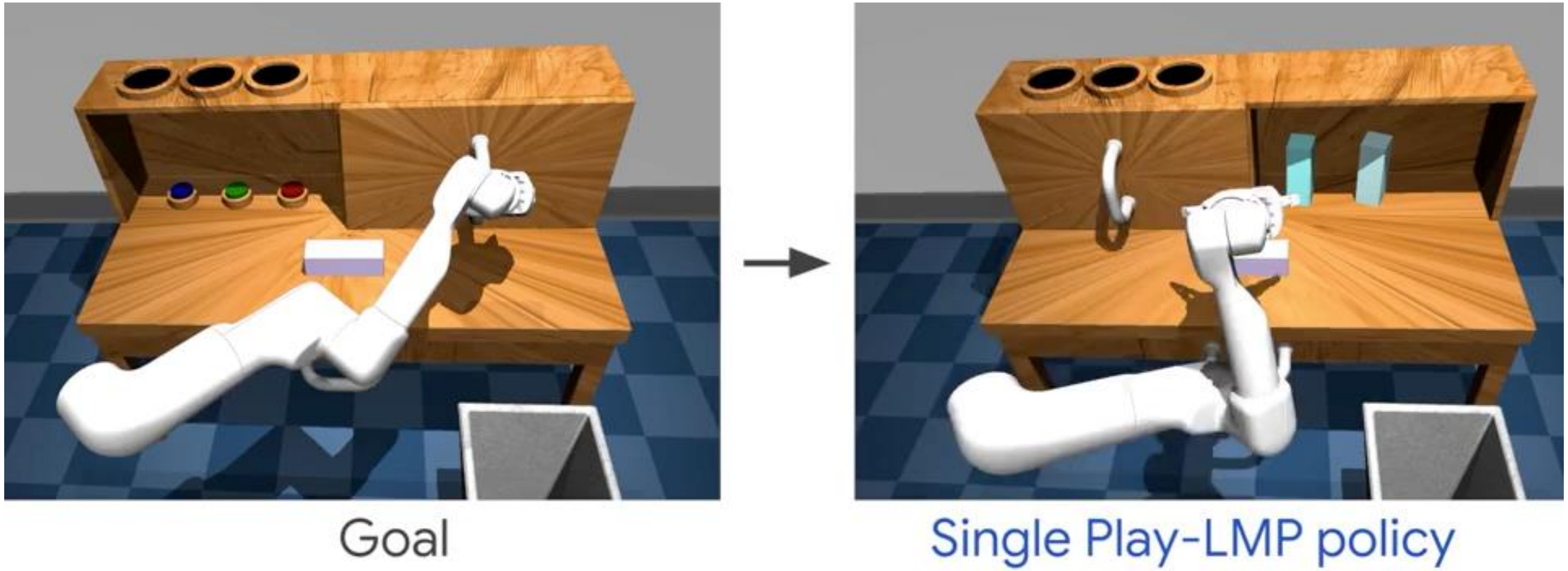
If the data is optimal, can we use supervised imitation learning?

1. Collect data $\mathcal{D}_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})\}$ using some policy
2. Perform hindsight relabeling:
   a. Relabel experience in $\mathcal{D}_k$ using last state as goal:
   $\mathcal{D}'_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{s}_T, r'_{1:T}\}$ where $r'_t = -d(\mathbf{s}_t, \mathbf{s}_T)$
   b. Store relabeled data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}'_k$
3. Update policy using **supervised imitation on** replay buffer $\mathcal{D}$

Eysenbach, Kumar, Gupta, RL is supervised learning on optimized data. BAIR blogpost, 2020
Ghosh, Gupta et al. Learning to Reach Goals via Iterated Supervised Learning. ICLR '21

Collect data from "human play”, perform goal-conditioned imitation.





Lynch, Khansari, Xiao, Kumar, Tompson, Levine, Sermanet. Learning Latent Plans from Play. ‘19

Recap

Key learning goals:

- Familiarity with multi-task learning challenges
- Hindsight relabeling in goal-conditioned RL

MTRL challenges:

- Optimization challenges
- Data sharing challenges

Goal-conditioned RL:

- An instance of MTRL
- Hindsight relabeling can help with exploration and learning

Next

Guest lecture by Jie Tan from Google

Can policies transfer between environments?

Can we use that for training agents in sim and transferring their behavior to real?