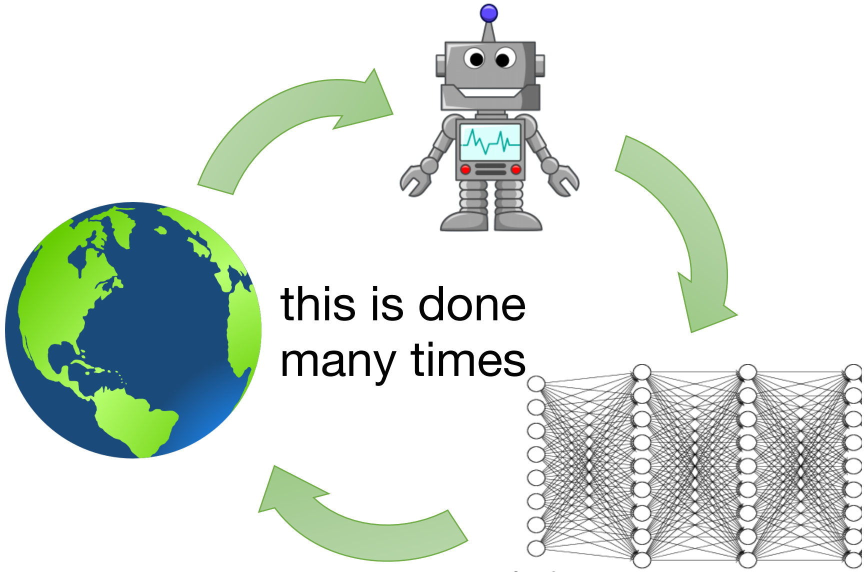
Offline RL for Real-Robot Pre-Training and Fine-Tuning

Aviral Kumar

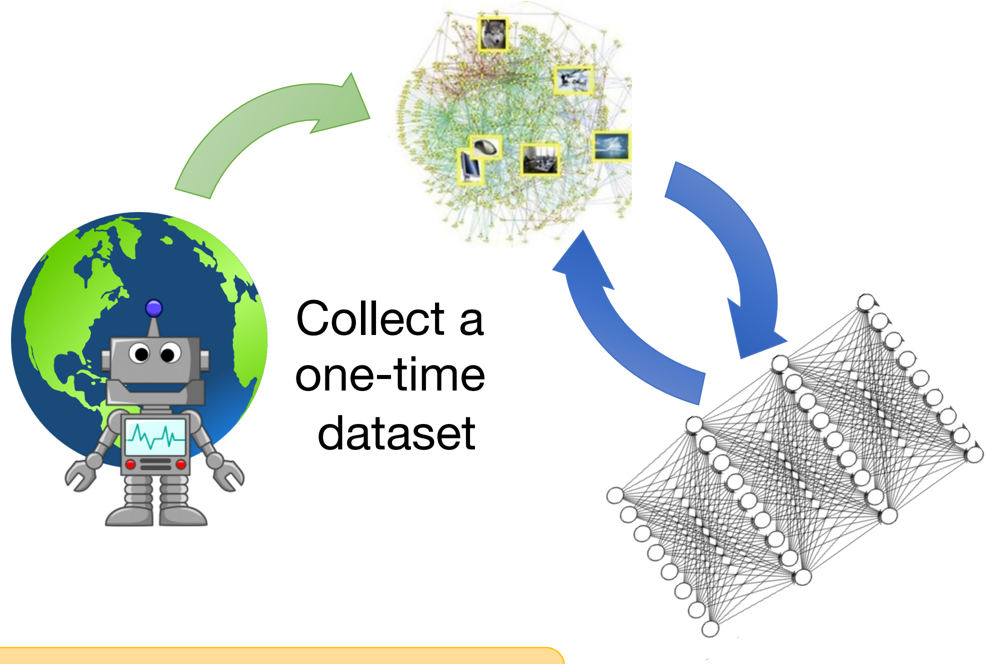
Berkeley
UNIVERSITY OF CALIFORNIA

Offline Reinforcement Learning

Standard Online RL



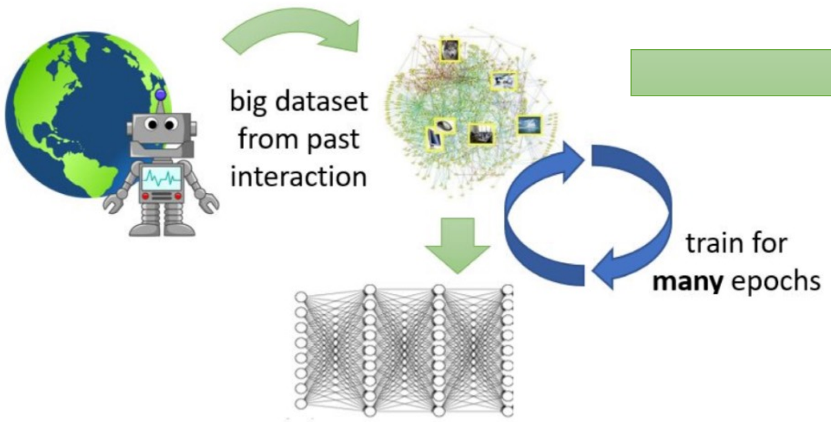
Offline RL



No unsafe or costly exploration

Potential to bring generalization benefits of supervised learning

What this Picture Actually Looks Like



a big general-purpose dataset

+

limited task data

Several key properties

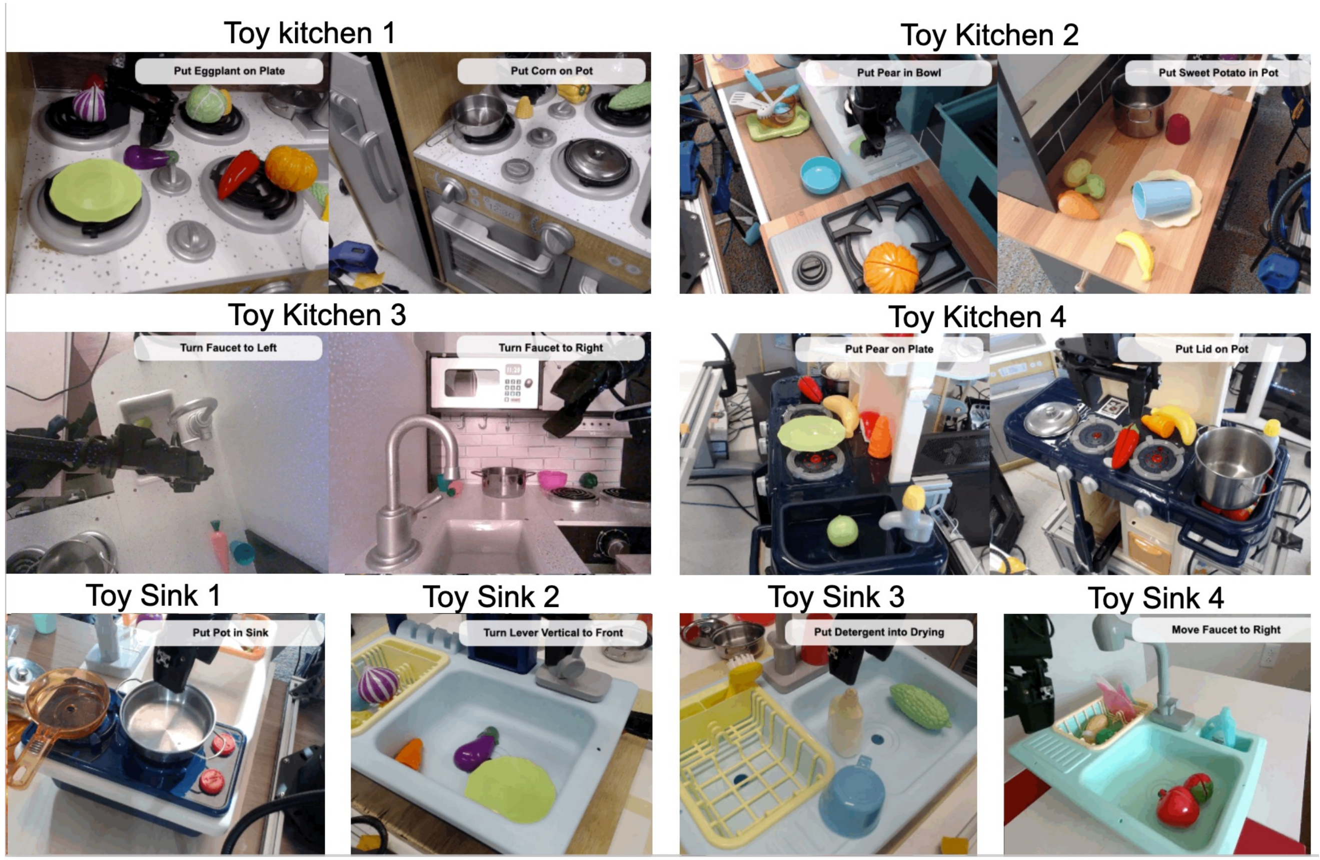
Multi-task data, no reward

Humans or “other agents”

Directly from raw visual observations

How can we apply offline RL in the presence of all of this?

Learning from Diverse Robot Datasets



Pre-training on broad data → **Fine-tuning on <u>limited,</u> task-specific data**

Imitation learning

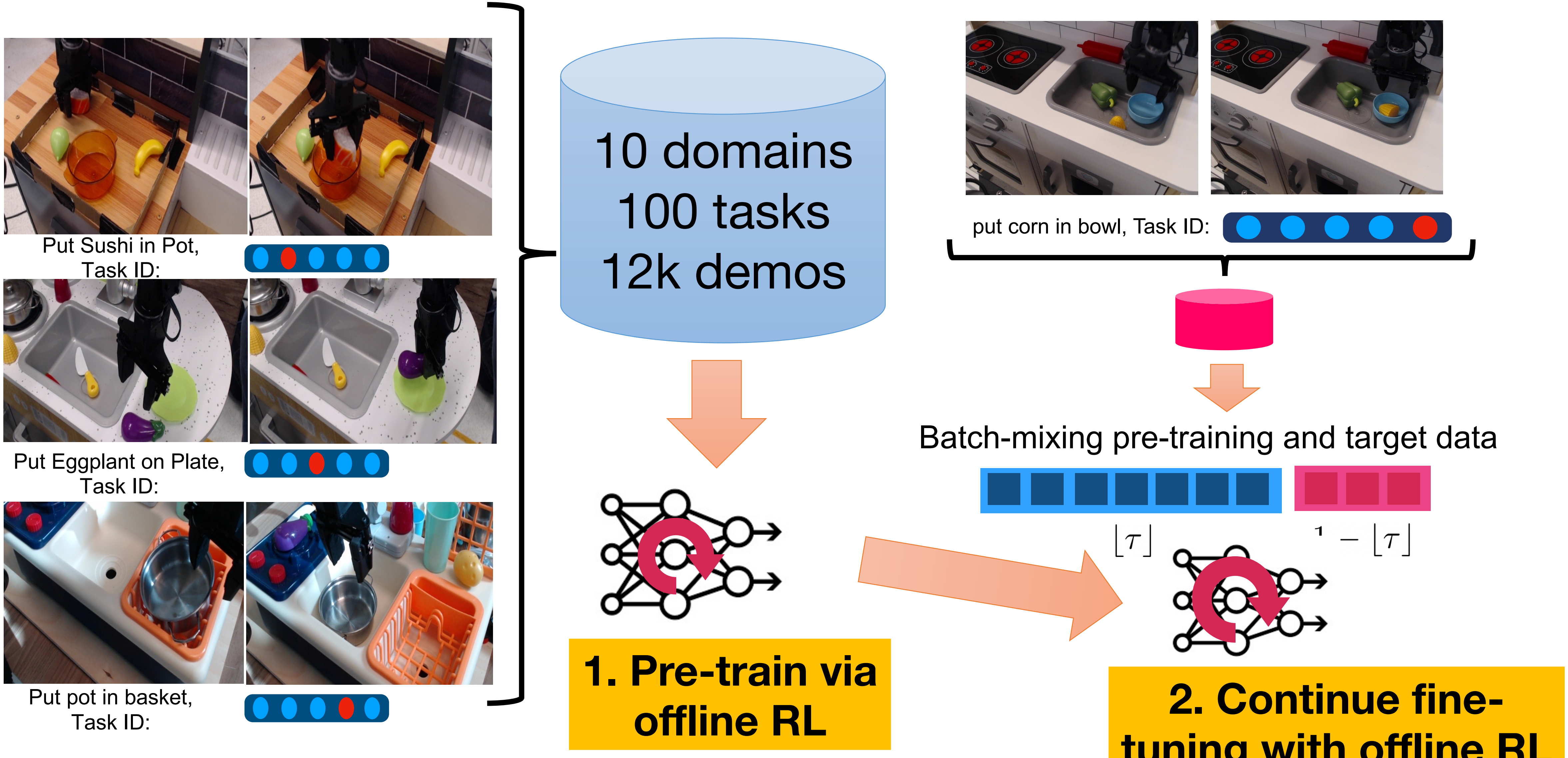
Ebert et al. 2021
Young et al. 2021
and many more

Pre-trained representation
↓
Imitation

Nair
Sh... al. 2022
... more....

Can we instead use offline RL for <u>both</u> pre-training & fine-tuning?

Pre-Training for Robots Using Offline RL



Ebert*, Yang* et al. **Bridge data: Boosting Generalization of Robotic Skills.** RSS 2022.

Ingredient 1: Conservative Q-Learning

The issue in offline RL is erroneous Q-values at **out-of-distribution (OOD) actions**

$$Q(\mathbf{s},\mathbf{a}) \leftarrow r(\mathbf{s},\mathbf{a}) + \underline{\gamma \mathbb{E}_{\mathbf{a}' \sim \pi_\theta(\mathbf{a}'|\mathbf{s}')}[Q(\mathbf{s}',\mathbf{a}')]}$$

Control this value somehow!

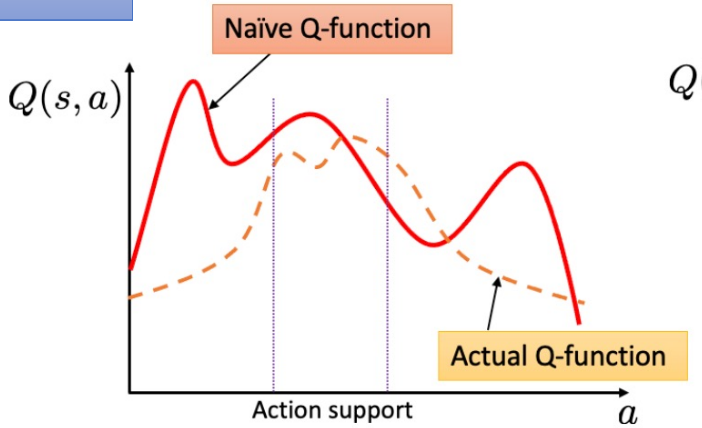
Maximize the data Q-values

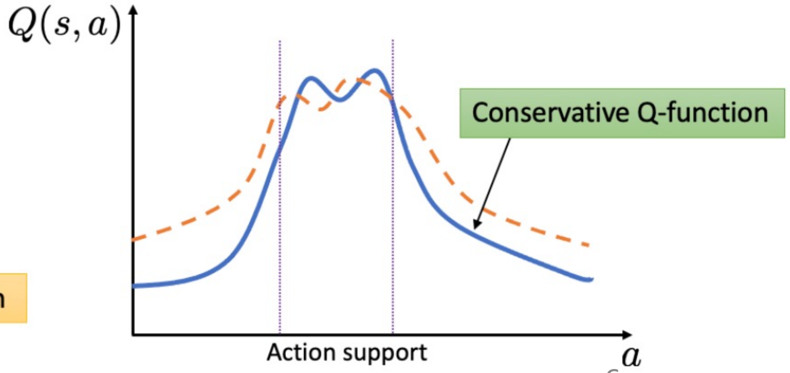
Standard TD error

$$Q_{\mathrm{CQL}}(\mathbf{s},\mathbf{a}) := \arg\min_{Q}\max_{\mu} \quad \left(\underline{\mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\mathbb{E}_{\mathbf{a}\sim\mu(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]} - \overline{\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}[Q(\mathbf{s},\mathbf{a})]}\right) + \frac{1}{2\alpha}\underline{\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[(Q(\mathbf{s},\mathbf{a}) - y(\mathbf{s},\mathbf{a}))^2\right]}$$

Minimize OOD Q-values

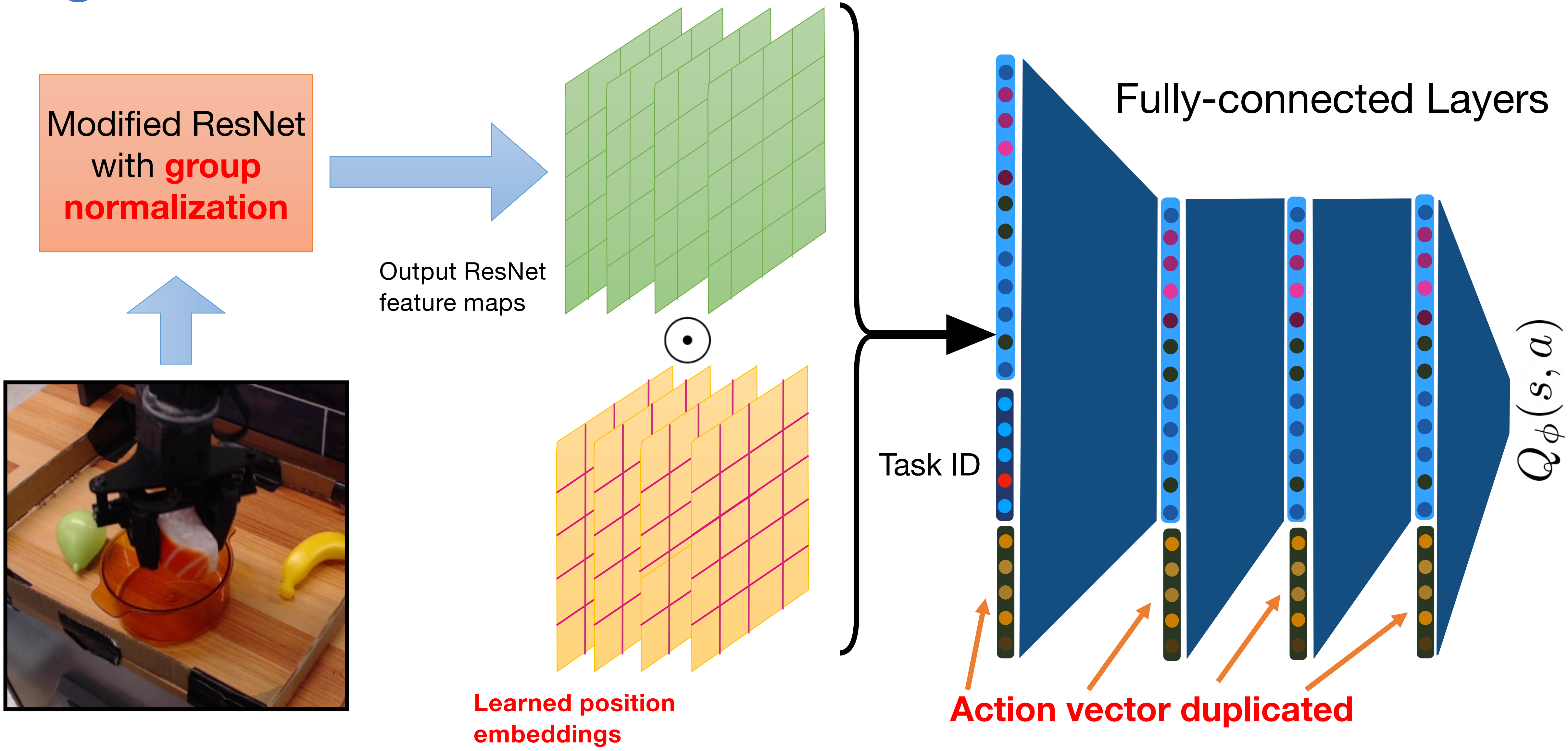
"Bake the pessimism"
into the Q-function





K., Zhou, Tucker, Levine. **Conservative Q-Learning for Offline Reinforcement Learning**. NeurIPS 2020

Ingredient 2: Architecture



Ingredient 3: Rewards & Checkpoint Selection

Rewards

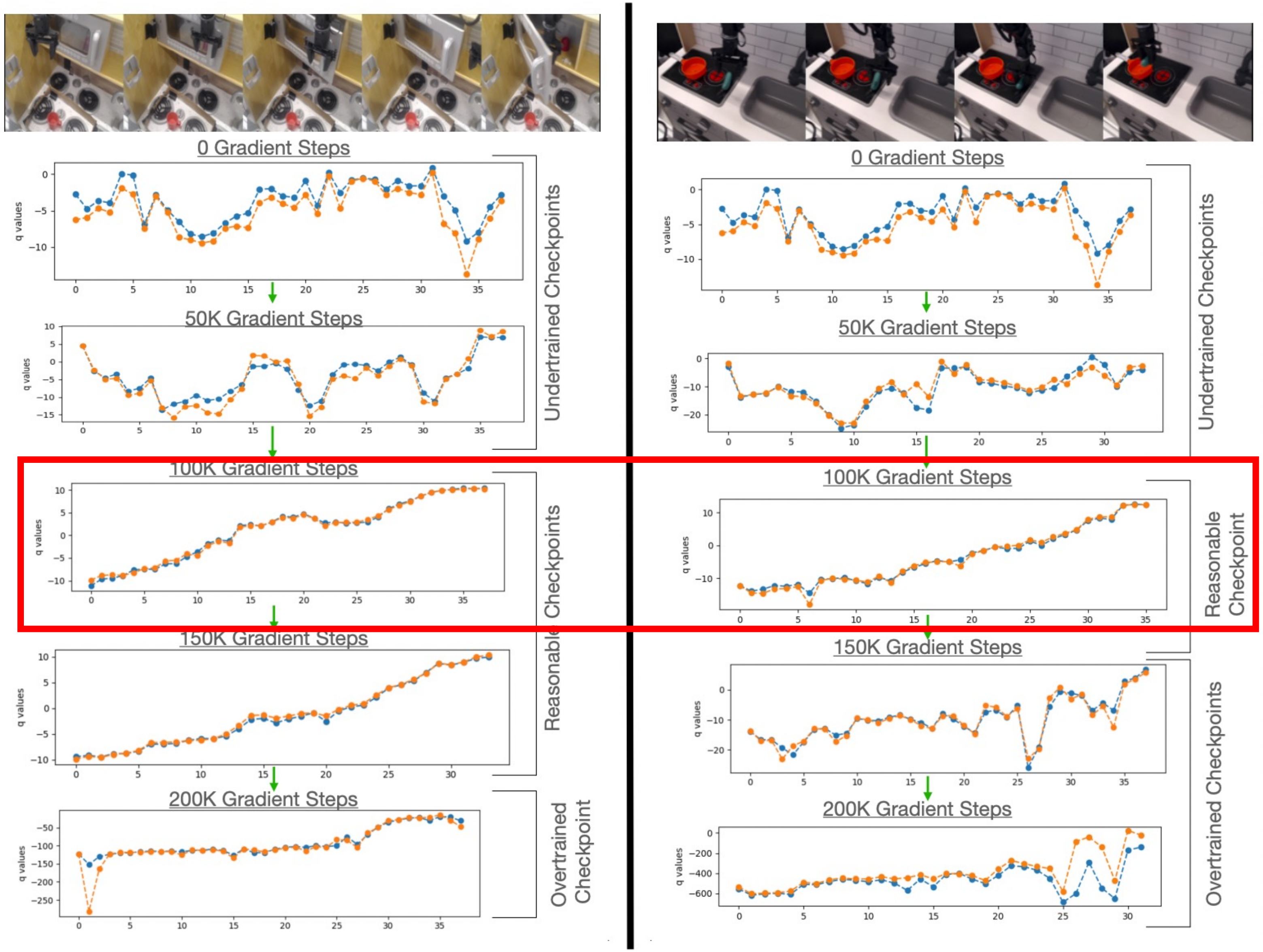
Sparse, binary rewards,
+1 at the end of the trajectory

Important: the binary values matter (-1, +10)

Checkpoint selection

Worst case: impossible!

But can use the knowledge that data is “expert”



Summary of Ingredients in PTR

Ingredient 1: An Offline RL Algorithm (CQL)

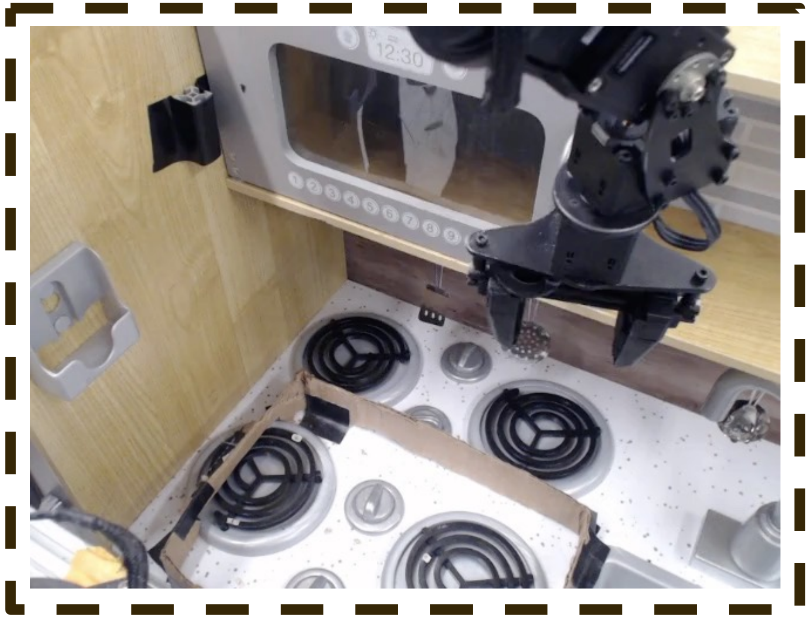
Ingredient 2: A high-capacity architecture
(ResNet + group normalization + action duplication + learned spatial embeddings)

Ingredient 3: Reward functions + checkpoint selection heuristic

Now some Empirical Results….

Task: Solving A Task in A New Domain

1. Pre-Train on Bridge Data, 12 doors 800 demonstrations



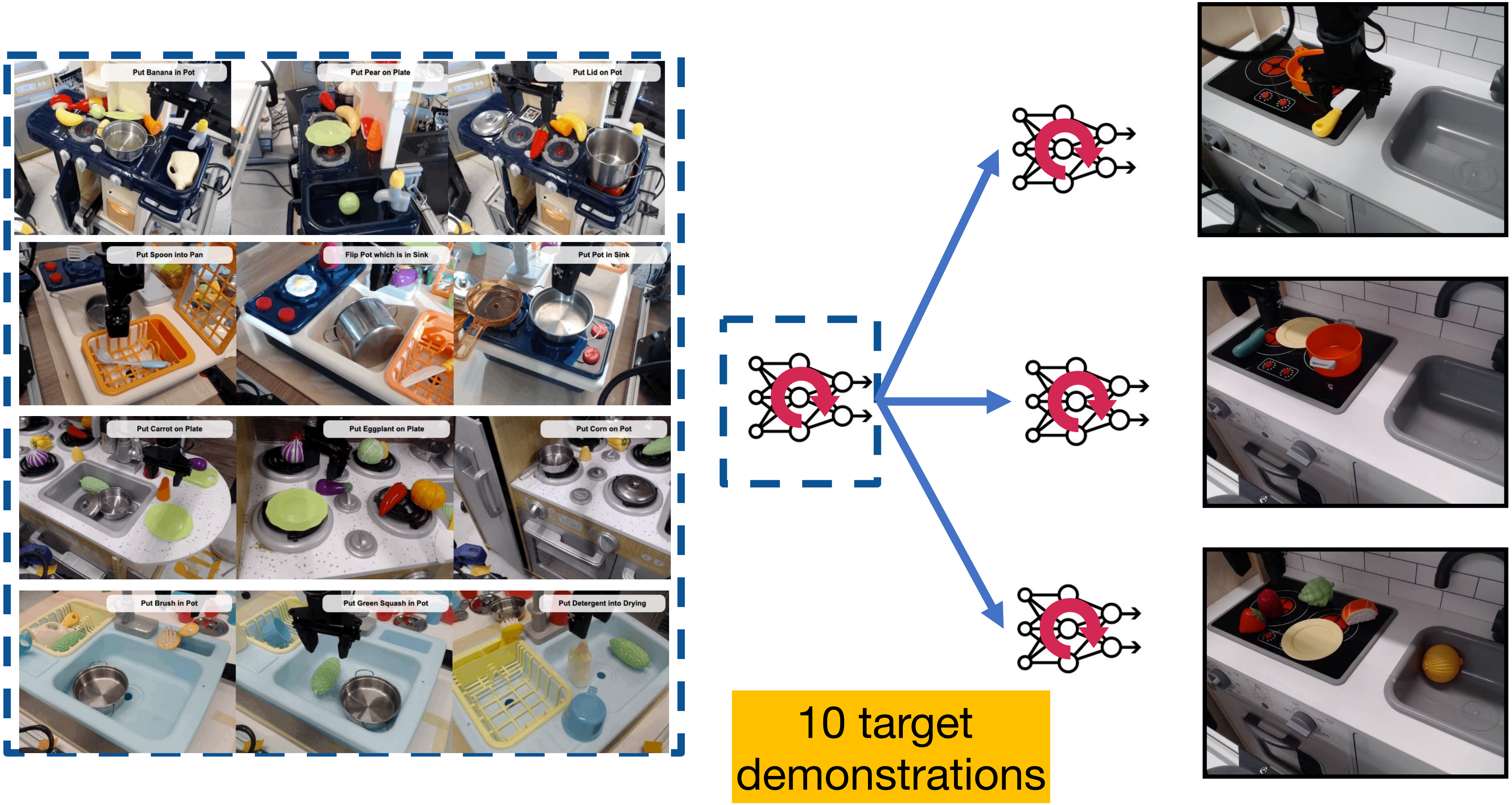
2. Fine-Tune on Target Domain Data:
1 door, 10 demonstrations

Results: Solving A Task in A New Domain

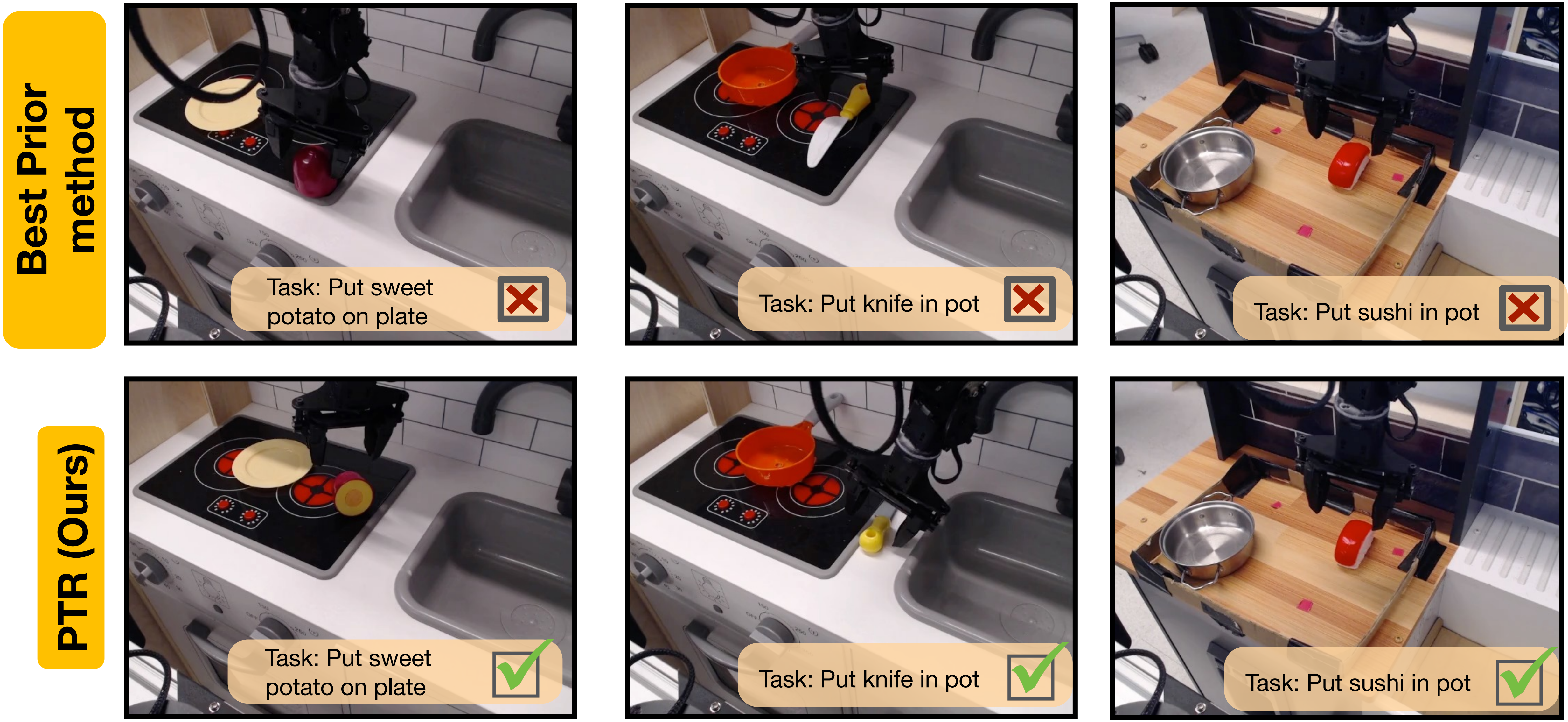
Method:
Imitation (Best prior method)

Method:
PTR (Ours)

Task: Solving New Tasks in New Domains



Results: Solving New Tasks in New Domains



Some Quantitative Results

Joint training vs pre-training??

		BC finetuning			Joint training		Target data only		Pre-train. rep. + BC finetune	
Task	**PTR (Ours)**	**BC (fine.)**	**Autoreg. BC**	**BeT**	**COG**	**BC**	**CQL**	**BC**	**R3M**	**MAE**
Take croissant from metal bowl	**7/10**	3/10	5/10	1/10	4/10	4/10	0/10	1/10	1/10	3/10
Put sweet potato on plate	**7/20**	1/20	1/20	0/20	0/20	0/20	0/20	0/20	0/20	1/20
Place knife in pot	**4/10**	2/10	2/10	0/10	1/10	3/10	3/10	0/10	0/10	0/10
Put cucumber in pot	**5/10**	0/10	1/10	0/10	2/10	1/10	0/10	0/10	0/10	0/10

Imitation (using transformers, auto-regressive)

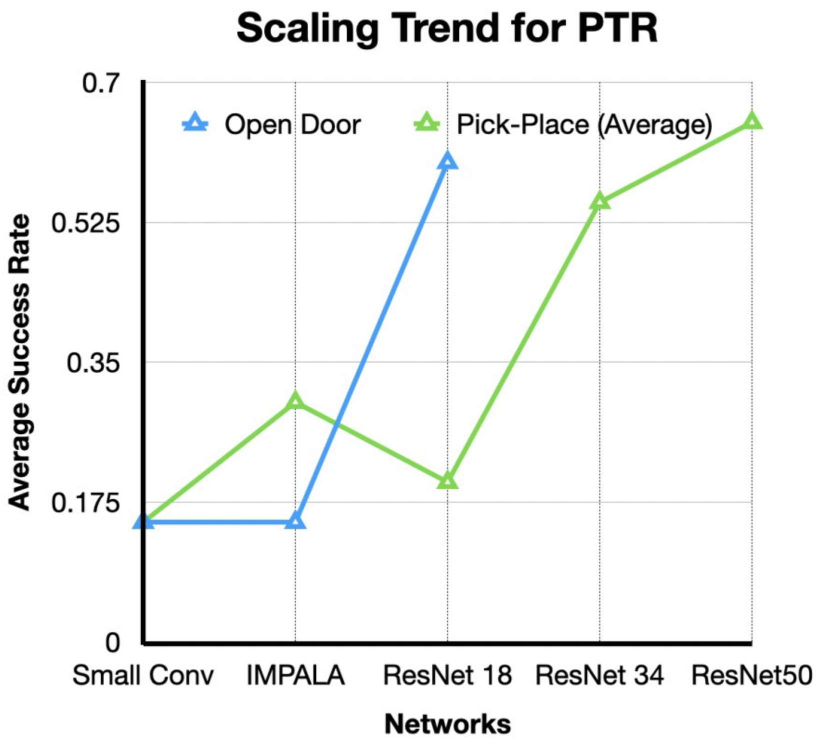
Better fine-tuning!

Self-supervised pre-training from videos / bridge data

Representation learning

Takeaway: Offline RL learns useful representations + better fine-tuning

Scaling Curve And Analysis

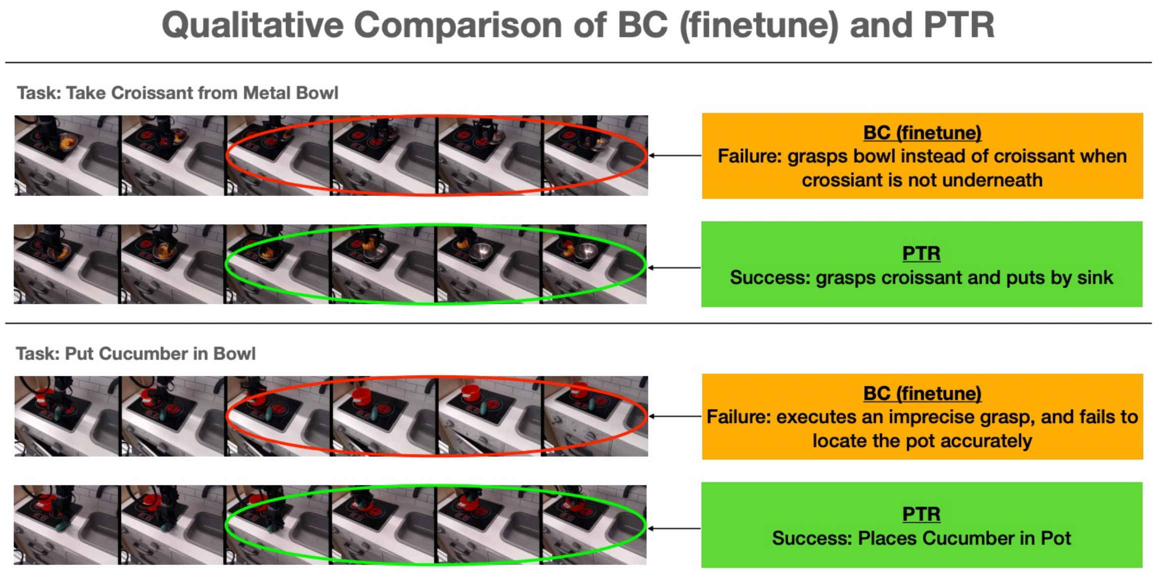


Better performance with larger networks!

Why would RL enable better performance…

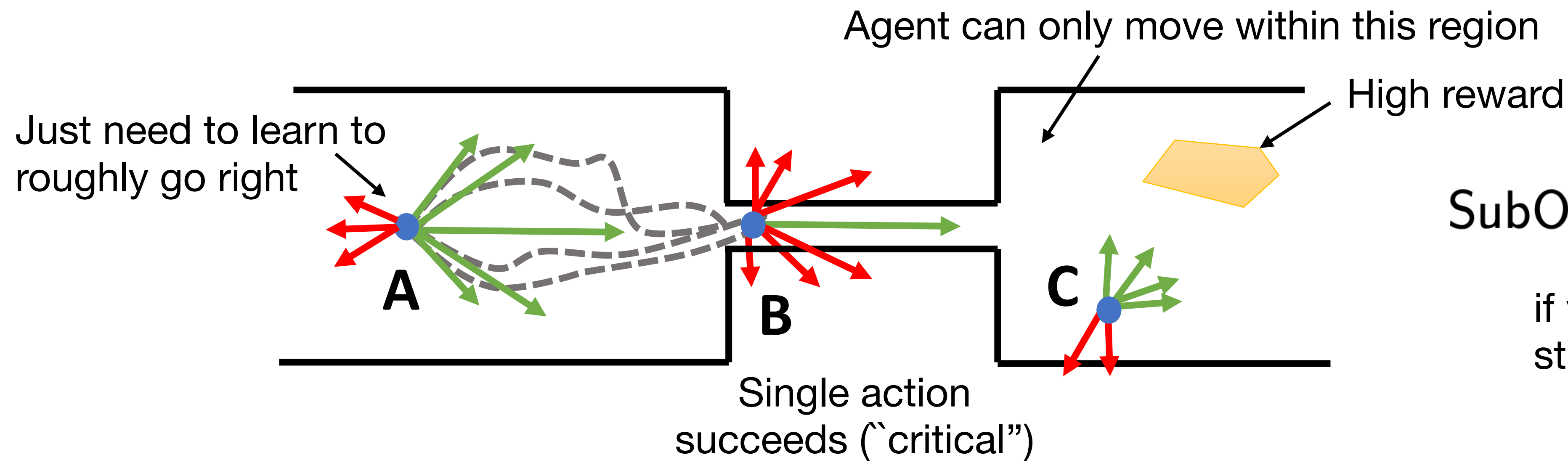
….when the data is collected via human teleoperation?

Preview: Value-functions can learn what's critical!



Analysis of Why PTR Outperforms Imitation

With near-expert data



$$\mathrm{SubOpt}(\pi_{\mathrm{RL}}) \leq \mathrm{SubOpt}(\pi_{\mathrm{BC}})$$

if volume of non-critical states in a trajectory is large

Test: Run weighted BC, where weights come from the learned Q-function!

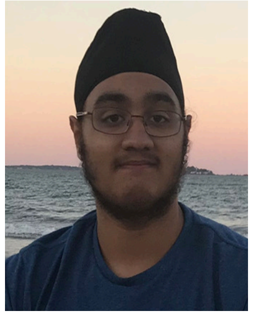
Task	BC (finetune)	PTR (Ours)	Advantage-weighted BC (finetune)
Put cucumber in pot	0/10	5/10	5/10
Take croissant from metal bowl	3/10	7/10	6/10

K., Hong, Singh, Levine. **When Should Offline RL Be Preferred Over BC?** ICLR 2022.

Takeaways and Future Directions

- Offline RL can be good for both **representation learning** and **control, even with human demonstration data**

- Future Directions:
  - Extend to use videos and multi-robot data on more dexterous tasks
  - **Goal specification:** language? goals? reward learning?
  - **Workflows**: "How should a practitioner tune this approach on their problem"




Thank You!



Paper: https://arxiv.org/abs/2210.05178
Code: https://github.com/Asap7772/PTR