Imitation Learning

CS 224R

Course reminders

- Start forming **final project groups** (survey due Mon April 17)
- **Homework 1** out today, due Weds April 19
- Fill out **AWS form** with account ID by this Friday April 7

News

- Thursday PyTorch tutorial (4:30 pm) moved to **Skilling Auditorium**
- Up to **2% extra credit** for providing TA-endorsed answers on Ed

The plan for today

Imitation Learning

1. Where does the data come from?
2. What can go wrong?
3. Learning from online interventions
4. Case study in fine robotic manipulation

Topic of homework 1!

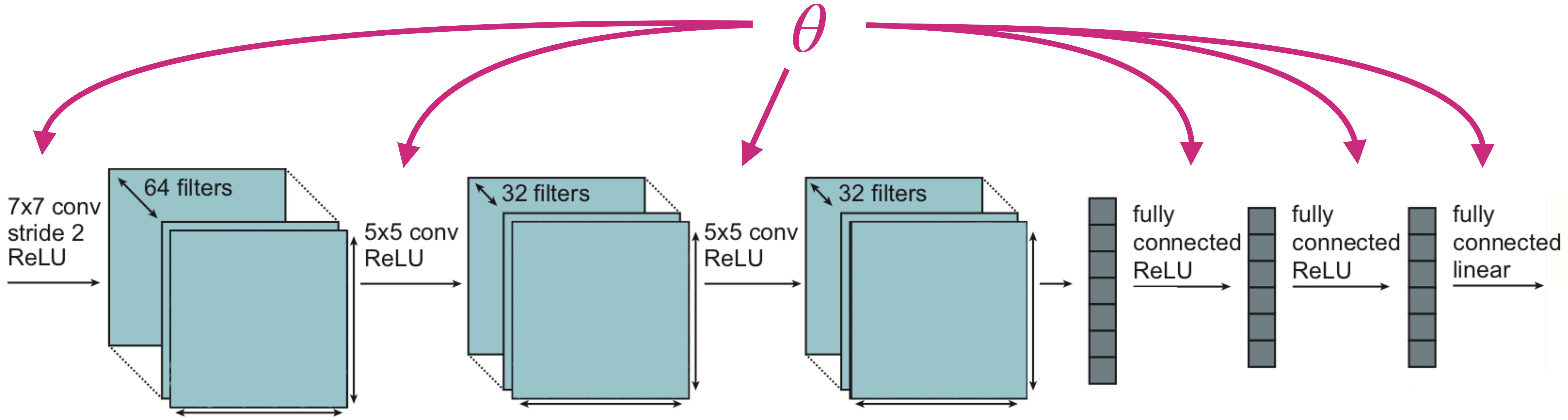
Key learning goals:

- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

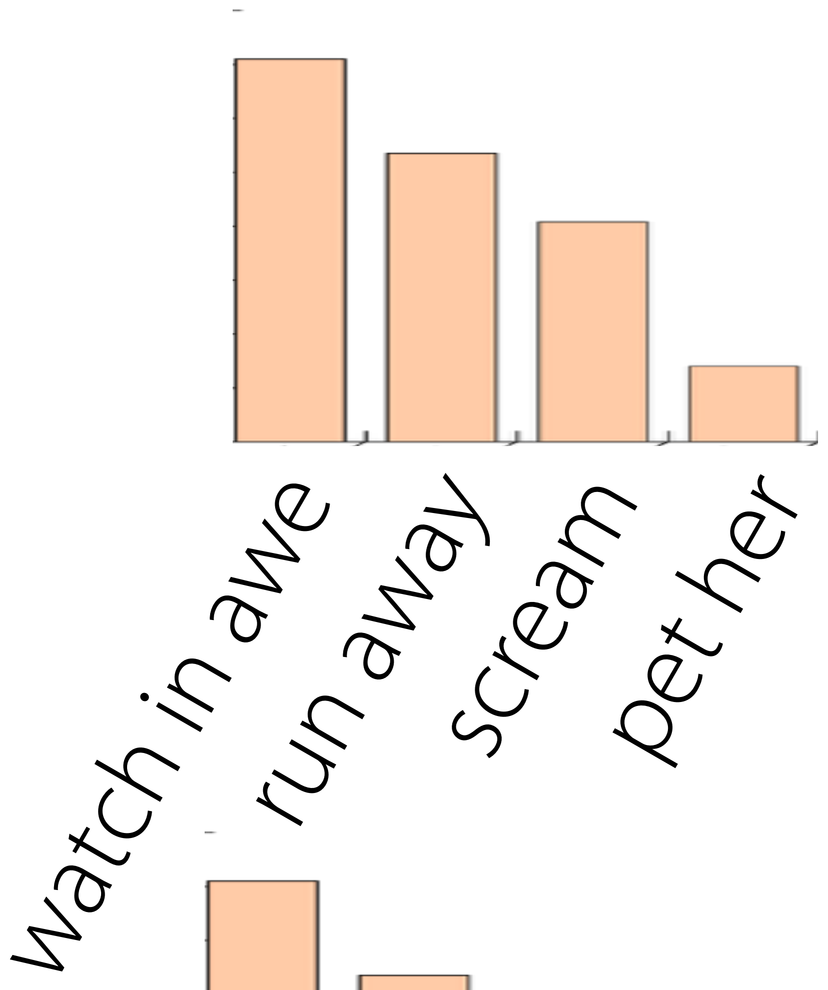
A formalization of behavior

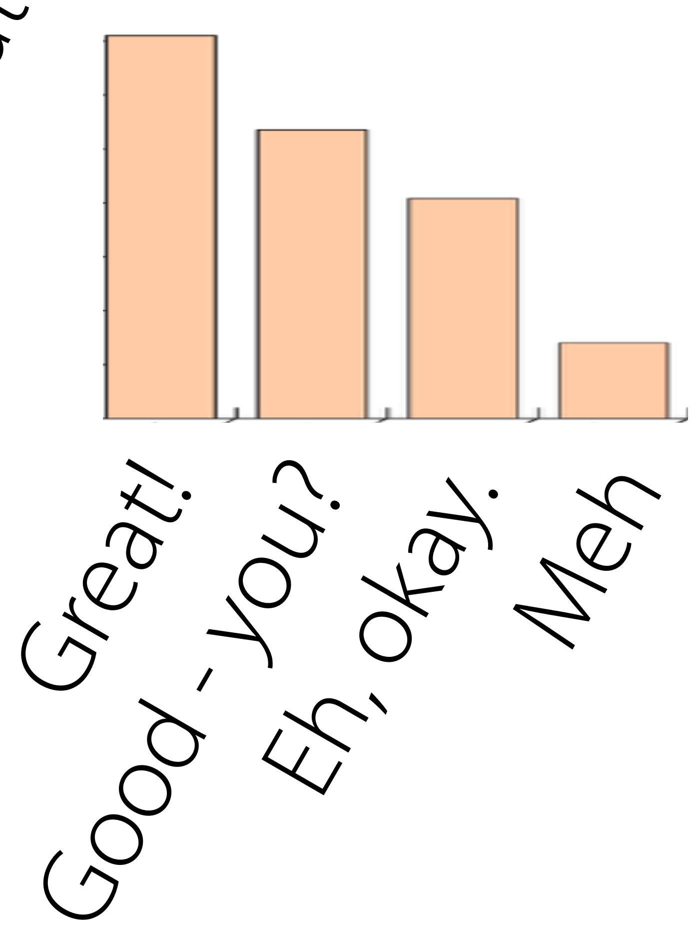
“Hi, how are you?”

$\mathbf{s}$



$\mathbf{a}$





$\pi_\theta(\mathbf{a} \,|\, \mathbf{s})$

Observe *state* $\mathbf{s}_t$

Take *action* $\mathbf{a}_t$ (e.g. by sampling from *policy* $\pi_\theta(\cdot \,|\, \mathbf{s}_t)$)

Observe next state $\mathbf{s}_{t+1}$ sampled from unknown world *dynamics* $p(\cdot \,|\, \mathbf{s}_t, \mathbf{a}_t)$

Result: a *trajectory* $\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T$. also called a policy *roll-out*

The basics of imitation learning

Key idea: Train policy using supervised learning

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: $\min_{\theta} - \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathscr{D}}[\log \pi_\theta(\mathbf{a} \mid \mathbf{s})]$

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

The plan for today

Imitation Learning

1. ***Where does the data come from?***
2. What can go wrong?
3. Learning from online interventions
4. Case study in fine robotic manipulation

Key learning goals:

- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

How to collect demonstrations?

In some domains: People already collect demonstrations that can be recorded

e.g. driving cars, writing text messages

What about robotics?

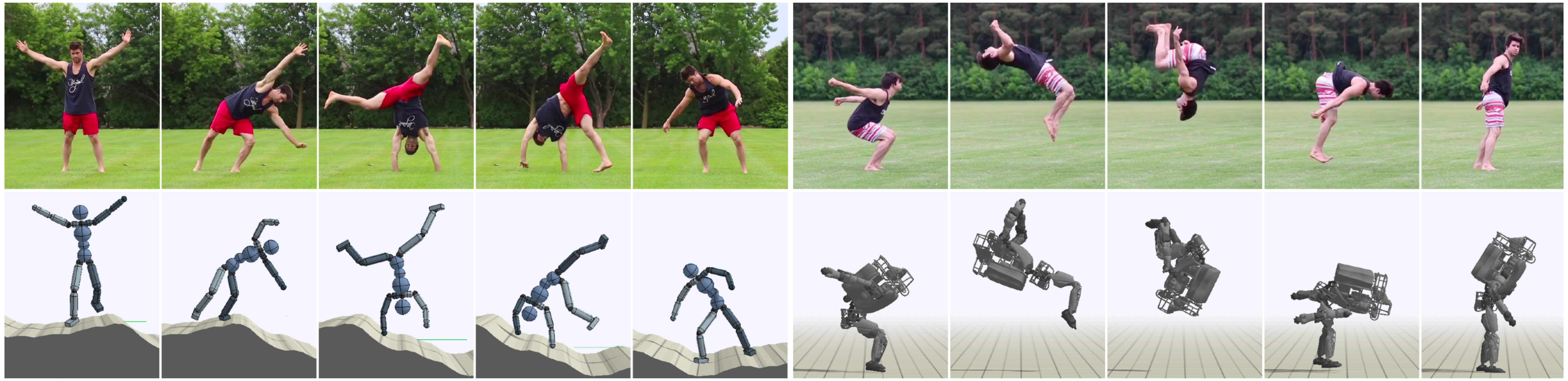
Kinesthetic teaching	Remote controllers	Puppeteering
		
+ easy interface	~ interface ease varies	+ easy interface
- human visible in scene		- requires double hardware

In other domains: It may not be viable to collect demos! (e.g. quadruped robot)

Can we directly use videos of people, animals?

Embodiment gap:
- difference in appearance
- difference in physical capabilities, degrees of freedom

Hard to directly imitate human & animal data, but can guide exploration.



Peng, Kanazawa, Malik, Abbeel, Levine. SFV: Reinforcement Learning of Physical Skills from Videos. SIGGRAPH Asia 2018.

The plan for today

Imitation Learning

1. Where does the data come from?
2. ***What can go wrong?***
3. Learning from online interventions
4. Case study in fine robotic manipulation

Key learning goals:

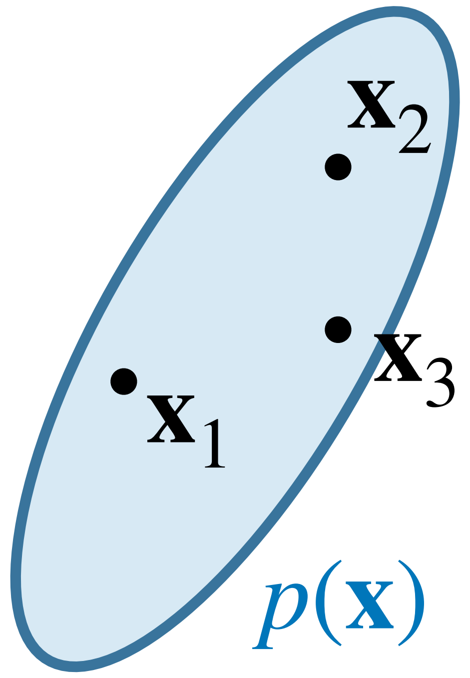
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

What can go wrong in imitation learning?

What can go wrong in imitation learning?

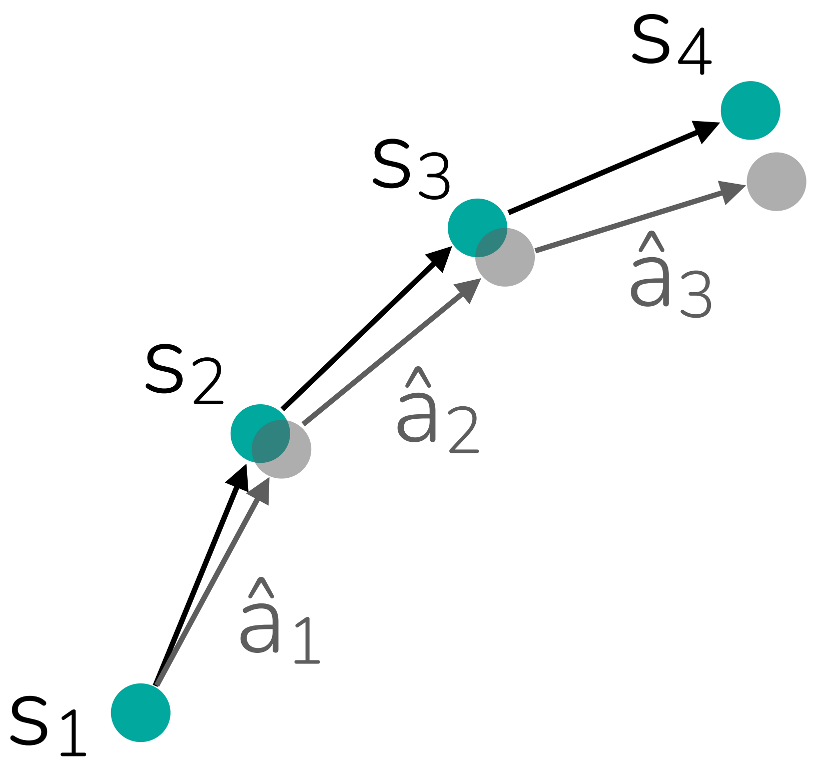
1. Compounding errors

Supervised learning



Inputs independent of predicted labels $\hat{\mathbf{y}}$

Supervised learning of behavior



Predicted actions affect next state.

Errors can lead to drift away from the data distribution!

Errors can then compound!

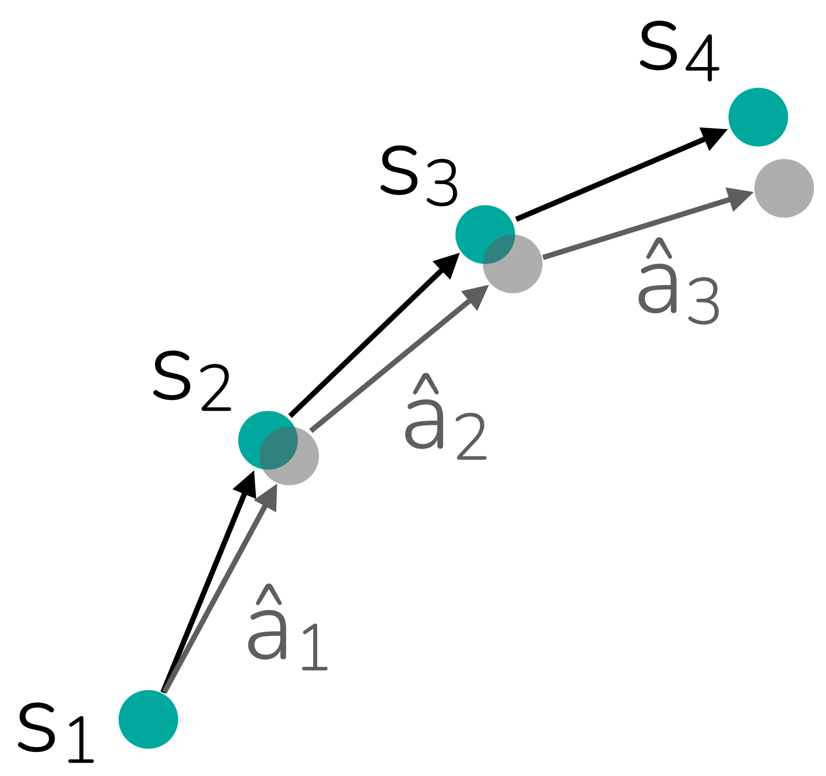
$$\underbrace{p_{expert}(\mathbf{s})}_{\text{states visited by expert}} \neq \underbrace{p_{\pi}(\mathbf{s})}_{\text{states visited by learned policy } \pi}$$

"covariate shift"

What can go wrong in imitation learning?

1. Compounding errors

Supervised learning of behavior



Predicted actions affect next state.

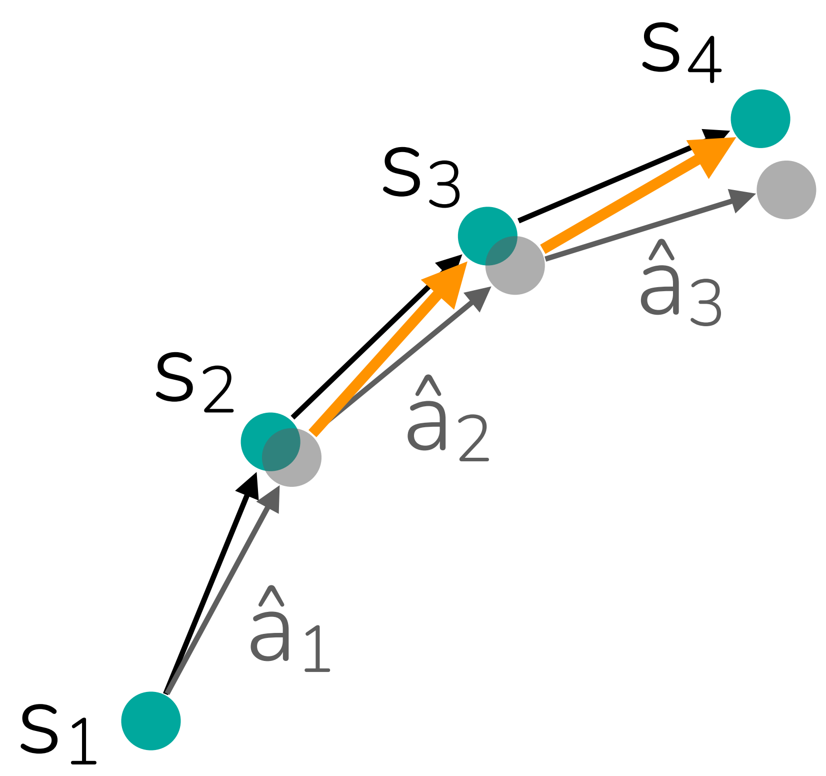
Errors can lead to drift away from the data distribution!

Errors can then compound!

$$p_{expert}(\mathbf{s}) \neq p_\pi(\mathbf{s})$$

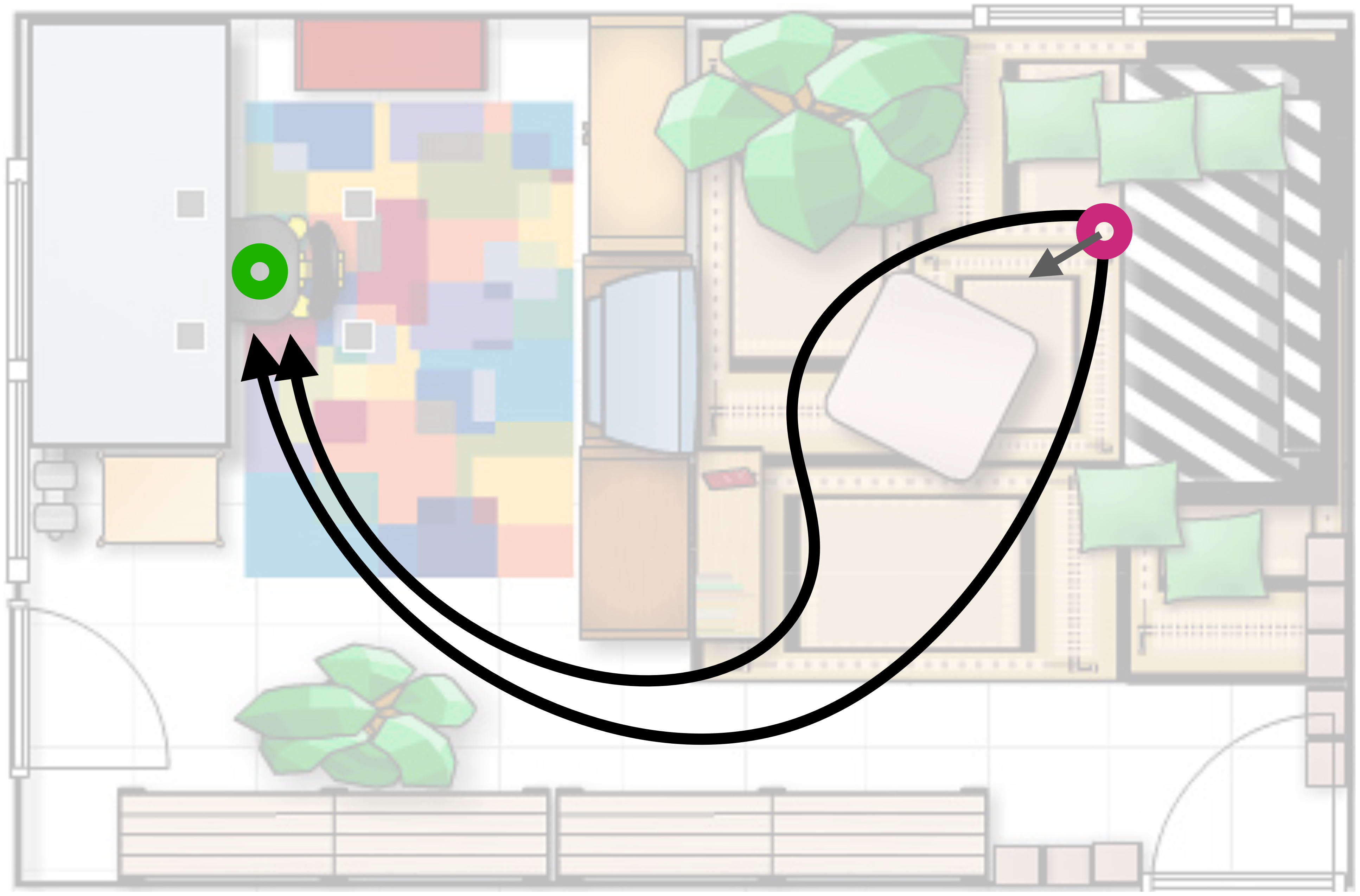
Solutions?

1. Collect A LOT of demo data & hope for the best.
2. Collect corrective behavior data



What can go wrong in imitation learning?

2. Multimodal demonstration data



The data takes two different actions **here**!

If we use ℓ_2 loss, what action will the agent take?

When does this happen in practice? All time time!

Esp. when data collected by multiple people.

Solution? Use expressive distribution class to fit $p(a \mid s)$.

- capture all modes of the data distribution
- e.g. Gaussian mixture, Categorical, VAEs, diffusion models

What can go wrong in imitation learning?

3. Mismatch in observability between expert & agent

Example demos scraped from conversations:

s: Hi, how are you?

a: Great, how was the basketball game last weekend?

…

s: Hey, how are you?

a: I'm good. Looking forward to getting lunch tomorrow!

…

Problem: Expert has more information than is observed by the agent.

Impossible to accurately imitate.

Solutions:

- Give as much contextual information to the agent as possible.
- Collect demos in a way that gives expert same information as agent.

What can go wrong in imitation learning?

1. Compounding errors
2. Multimodal demonstration data
3. Mismatch in observability between expert & agent

The plan for today

Imitation Learning

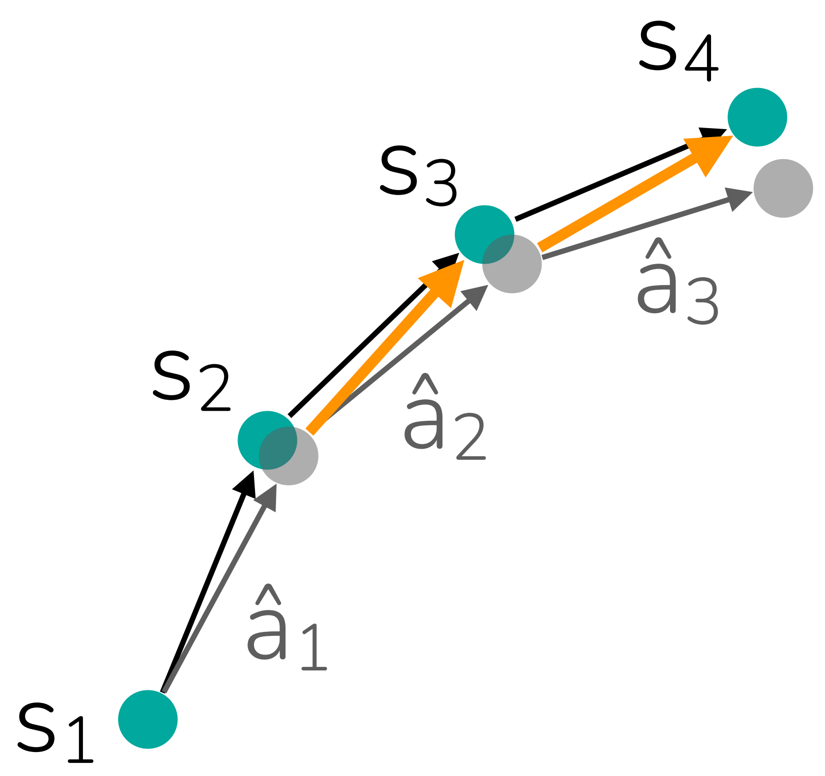
1. Where does the data come from?
2. What can go wrong?
3. ***Learning from online interventions***
4. Case study in fine robotic manipulation

Key learning goals:

- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

Addressing Compounding Errors with DAgger

Collect corrective behavior data

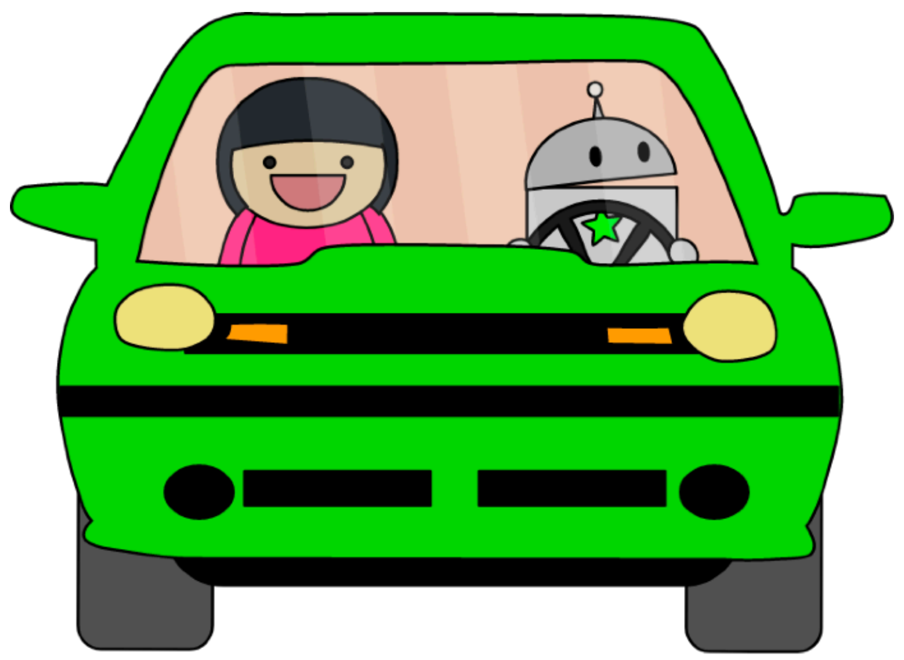


1. Roll-out learned policy π_θ: $\mathbf{s}'_1, \hat{\mathbf{a}}_1, \ldots, \mathbf{s}'_T$
2. Query expert action at visited states $\mathbf{a}^* \sim \pi_{expert}(\cdot \mid \mathbf{s}')$
3. Aggregate corrections with existing data $\mathscr{D} \leftarrow \mathscr{D} \cup \{(\mathbf{s}', \mathbf{a}^*)\}$
4. Update policy $\min_\theta \mathscr{L}(\pi_\theta, \mathscr{D})$

"dataset aggregation" (DAgger)

+ data-efficient way to learn from an expert

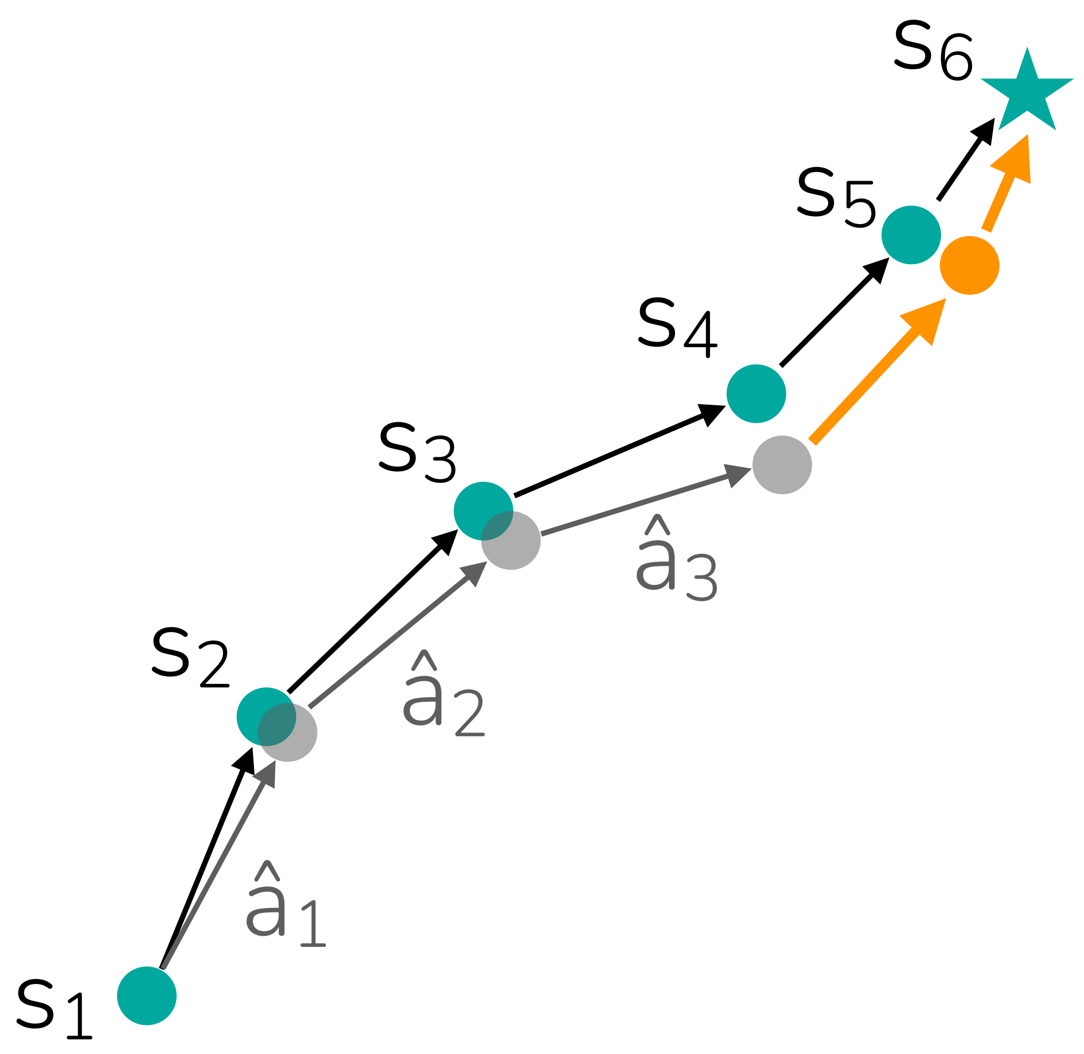
- can be challenging to query expert when agent has control



Is there another way to collect corrective data?

Addressing Compounding Errors with DAgger

Collect corrective behavior data while *taking full control*



1. Start to roll-out learned policy π_θ: $\mathbf{s}'_1, \hat{\mathbf{a}}_1, \ldots, \mathbf{s}'_t$
2. Expert intervenes at time t when policy makes mistake
3. Expert provides (partial) demonstration $\mathbf{s}'_t, \mathbf{a}^*_t, \ldots, \mathbf{s}'_T$
4. Aggregate new demos with existing data $\mathscr{D} \leftarrow \mathscr{D} \cup \{(\mathbf{s}'_i, \mathbf{a}^*_i)\}; i \geq t$
5. Update policy $\min_\theta \mathscr{L}(\pi_\theta, \mathscr{D})$

"human gated DAgger"

+ (much) easier interface for providing corrections

- can be hard to catch mistakes quickly in some application domains

The plan for today

Imitation Learning

1. Where does the data come from?
2. What can go wrong?
3. Learning from online interventions
4. ***Case study in fine robotic manipulation***

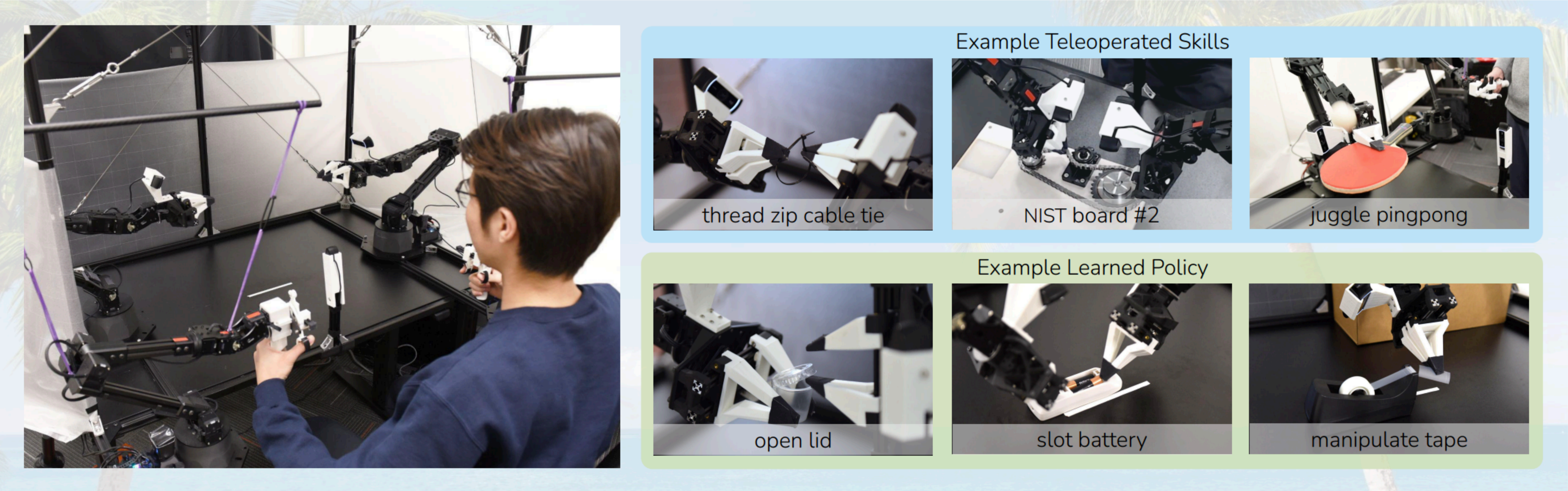
Key learning goals:

- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

Case study: Can robots learn fine-grained manipulation skills from demonstrations?

Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware

Tony Z. Zhao[1] Vikash Kumar[3] Sergey Levine[2] Chelsea Finn[1]
[1] Stanford University [2] UC Berkeley [3] Meta



Goal: Solve tasks where ***precision*** and ***closed-loop feedback*** are important, with objects that are ***difficult to simulate***

—> by learning from real-world data.

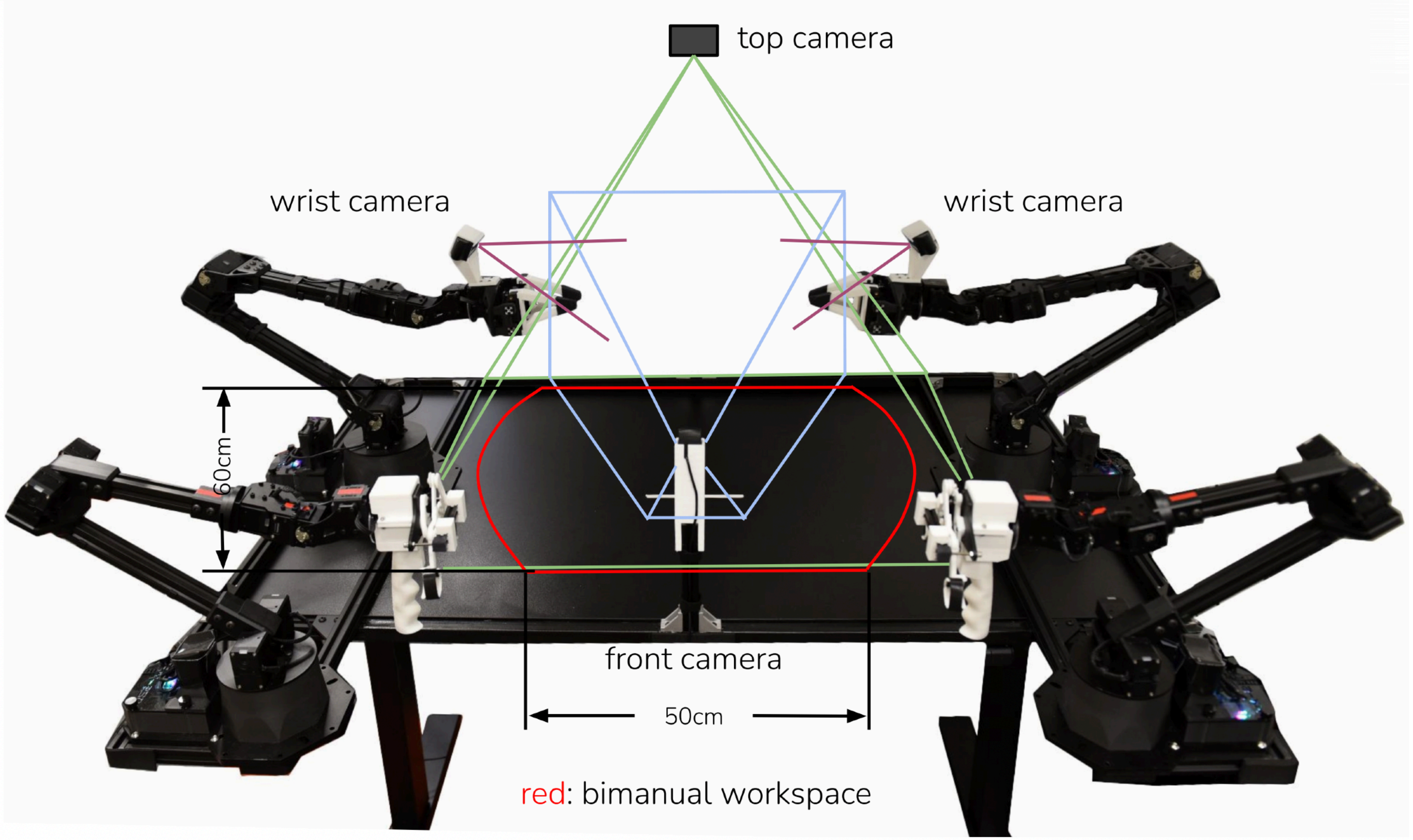
Hardware Set-Up

- *Total cost*: <$20k
- Off-the-shelf arms + open-sourced parts & code
- Fun to play with. :)

Hardware Set-Up

- *Total cost*: <$20k
- Off-the-shelf arms + open-sourced parts & code

- Off-the-shelf 6-DoF arms with 3D-printed fingers
- Map joint angles across robots during teleoperation
- 50 Hz control
- Record RGB images from 4 cameras
- No force feedback (beyond weight of "leader" robot)



Imitation Learning System

Train neural network policy to map from images to target joint positions.

Challenge 1: Supervised imitation learning struggles with **compounding errors**, particularly at 50 Hz.

Challenge 2: Human demonstrations perform tasks in different ways, leading to **multimodal data distribution**.

Naive policy training achieves 0% success.

	Slide Ziploc (real)			**Slot Battery** (real)		
	Grasp	Pinch	Open	Grasp	Place	Insert
BC-ConvMLP	0	0	0	0	0	0
BeT	8	0	0	4	0	0
RT-1	4	0	0	4	0	0
VINN	28	0	0	20	0	0

Success rates of carefully-tuned prior IL methods.

Imitation Learning System

Train neural network policy to map from images to target joint positions.

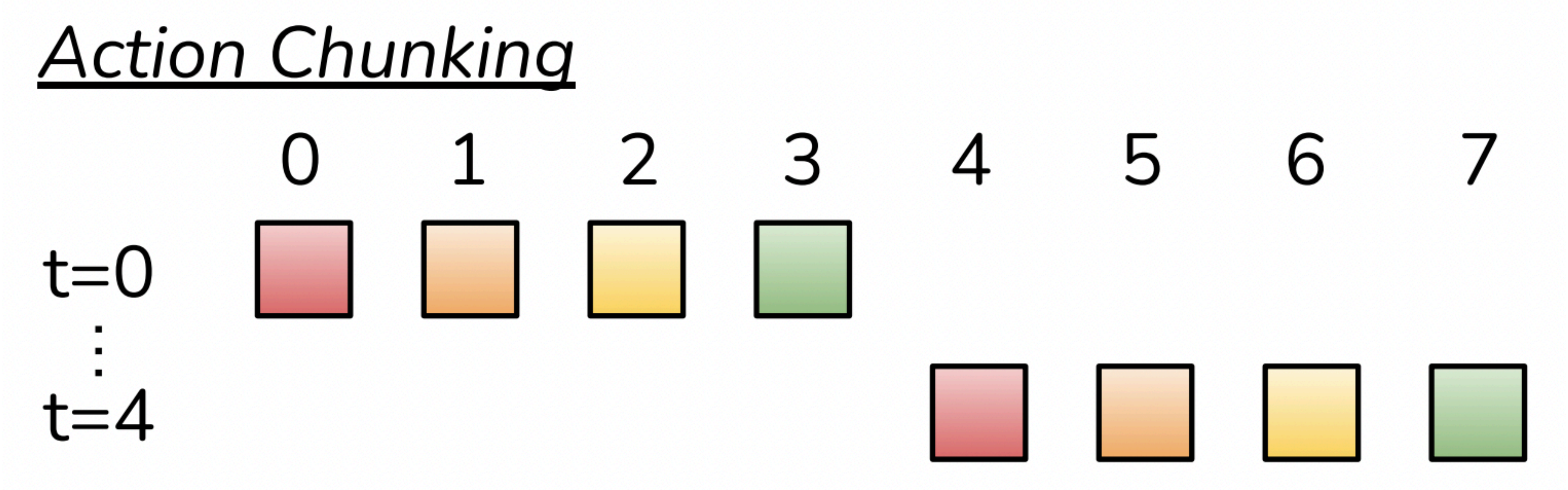
Challenge 1: Supervised imitation learning struggles with **compounding errors**, particularly at 50 Hz.

Challenge 2: Human demonstrations perform tasks in different ways, leading to **multimodal data distribution**.

Naive policy training achieves 0% success.

Solutions for #1:

- Policy predicts chunks of ~60 actions open-loop (closed-loop at ~0.8 Hz, rather than making new decision every timestep) } trade-off drift & open-loop



Imitation Learning System

Train neural network policy to map from images to target joint positions.

Challenge 1: Supervised imitation learning struggles with **compounding errors**, particularly at 50 Hz.

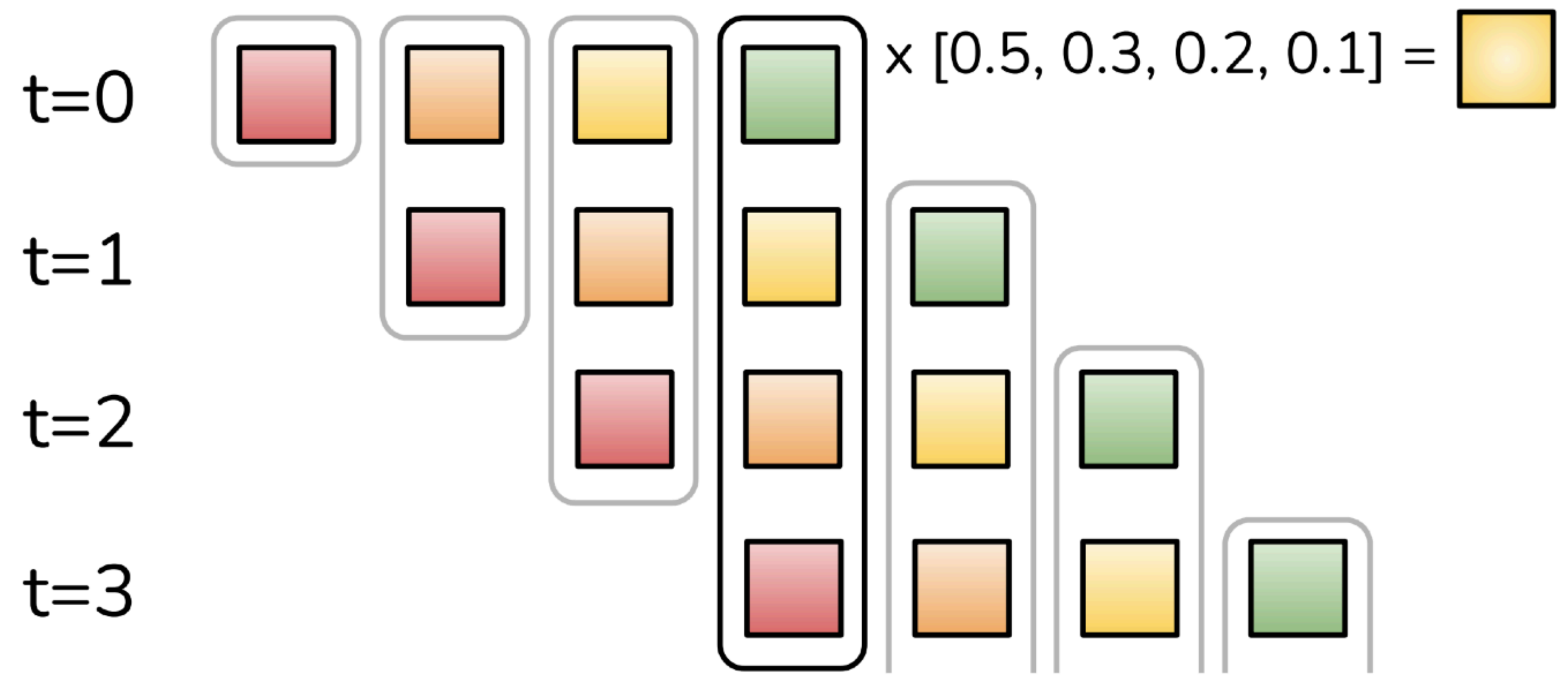
Challenge 2: Human demonstrations perform tasks in different ways, leading to **multimodal data distribution**.

Naive policy training achieves 0% success.

Solutions for #1:

- Policy predicts chunks of ~60 actions open-loop
  (closed-loop at ~0.8 Hz, rather than making new decision every timestep)
- Weighted average over predicted actions for that timestep

} trade-off drift & open-loop



Imitation Learning System

Train neural network policy to map from images to target joint positions.

Challenge 1: Supervised imitation learning struggles with **compounding errors**, particularly at 50 Hz.

Challenge 2: Human demonstrations perform tasks in different ways, leading to **multimodal data distribution**.

Naive policy training achieves 0% success.

Solutions for #1:

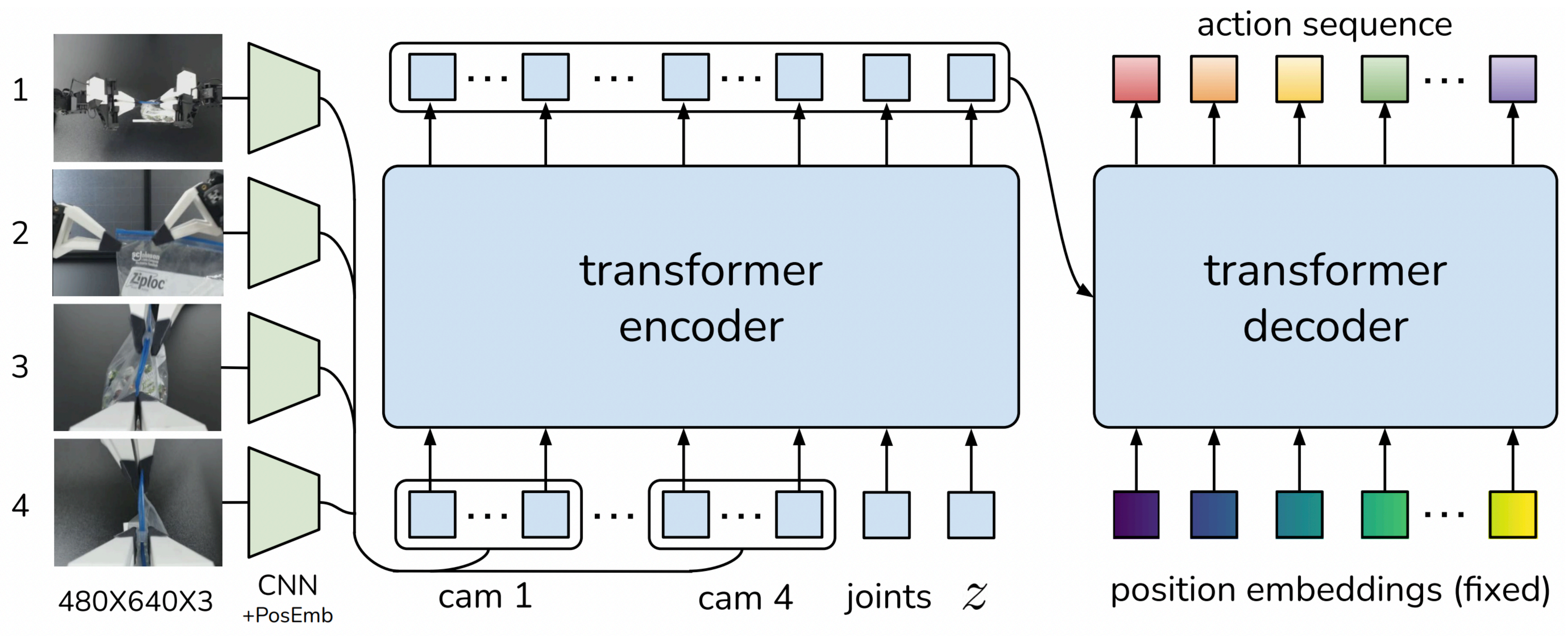
- Policy predicts chunks of ~60 actions open-loop (closed-loop at ~0.8 Hz, rather than making new decision every timestep) } trade-off drift & open-loop
- Weighted average over predicted actions for that timestep
- Transformer-based policy architecture
- Actions correspond to target absolute joint positions (rather than relative joint positions)

Solution for #2:

- Use variational auto-encoder (VAE) to model multimodality

Imitation Learning System

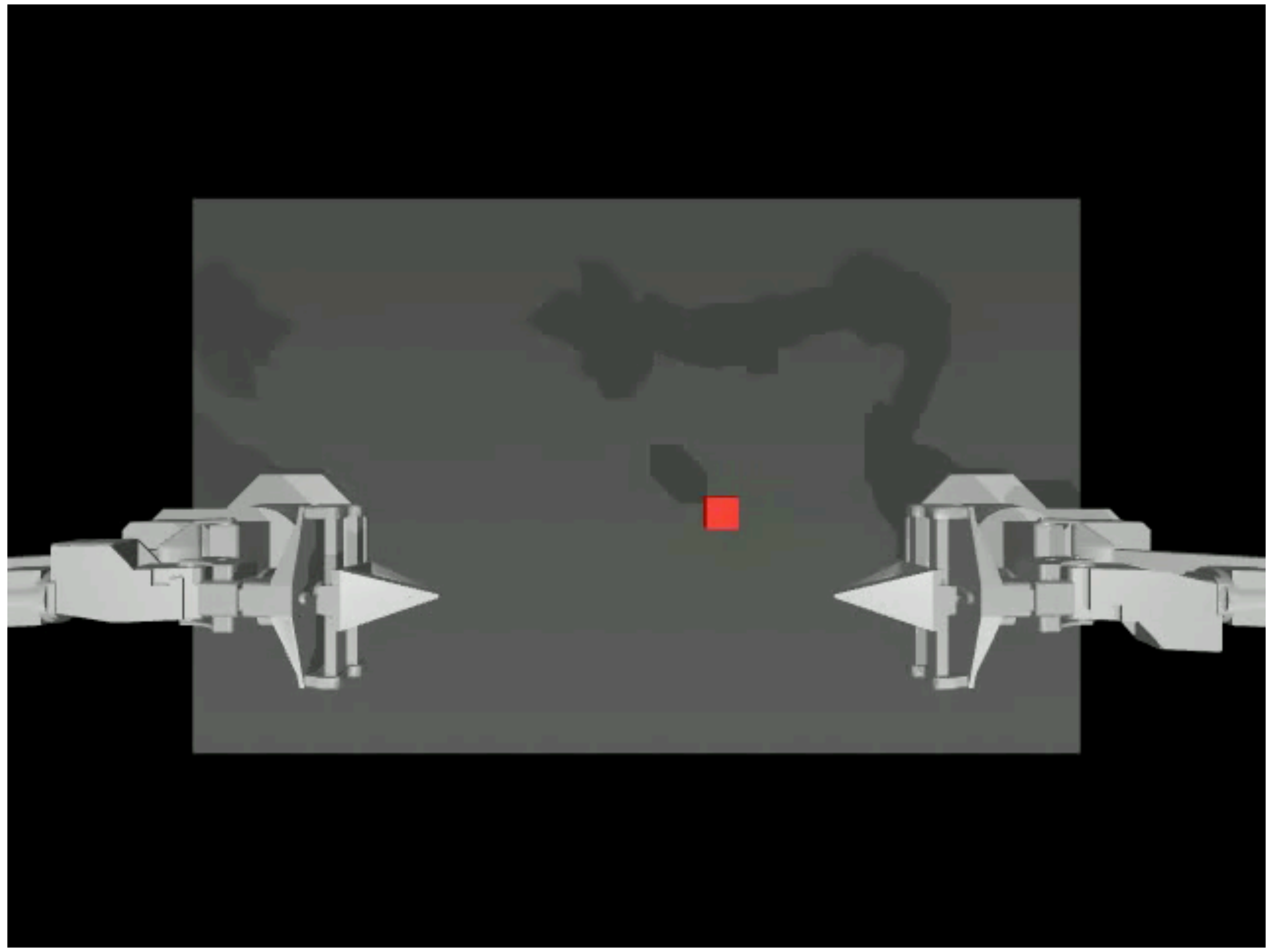
Policy architecture



Action chunking with transformers (ACT)

Simulated Results

Grasp & transfer from image observations



How does ACT compare to prior methods?

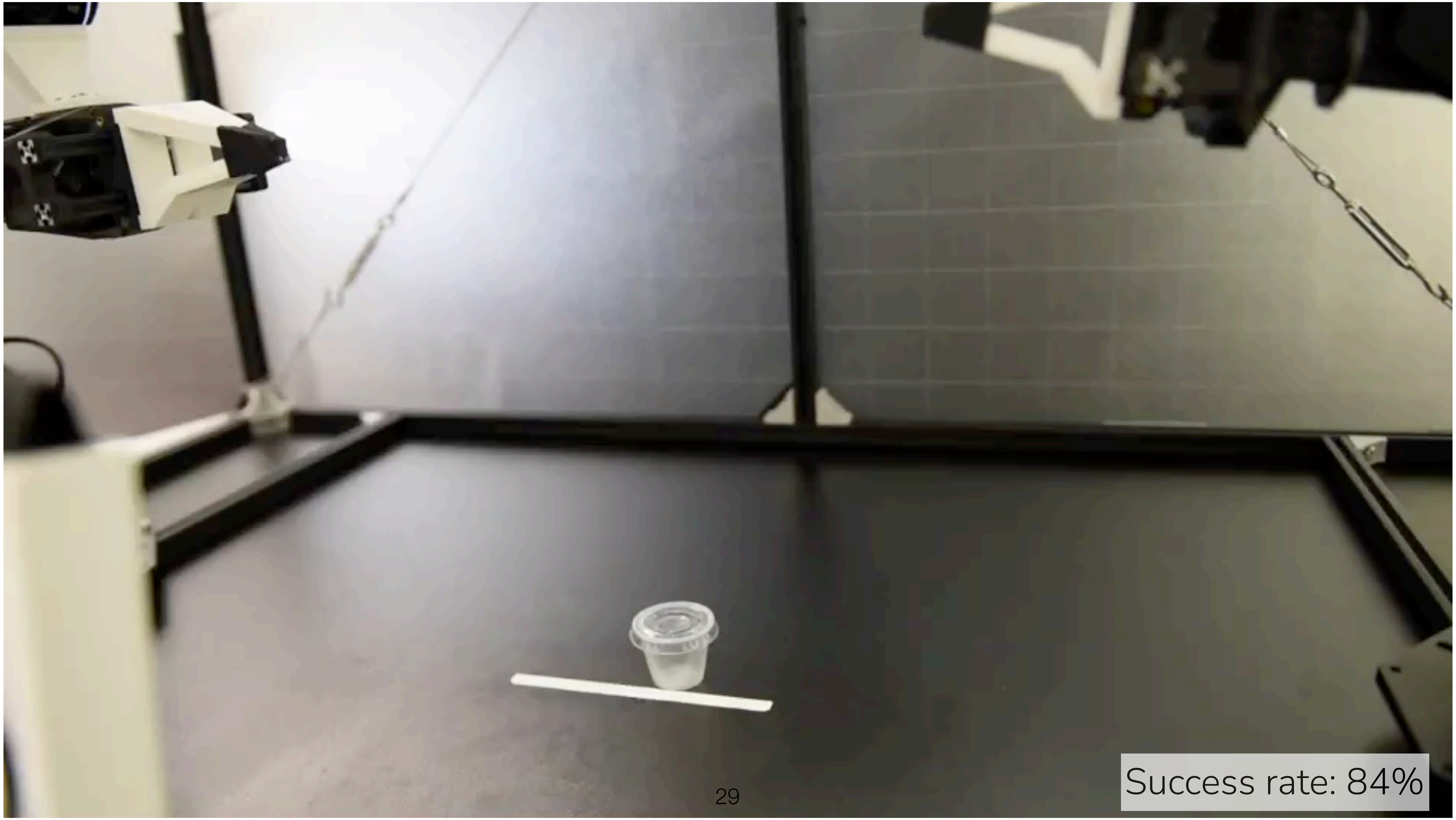
	Success Rate
MLP policy	1%
Behavior Transformer (BeT)	27%
Visual Imitation Nearest Neighbors (VINN)	3%
RT-1	2%
ACT	**86%**

Is action chunking important?

ACT, no action chunking	0%

Real Robot Results

Collect 50 demonstrations, randomize object location along white line.

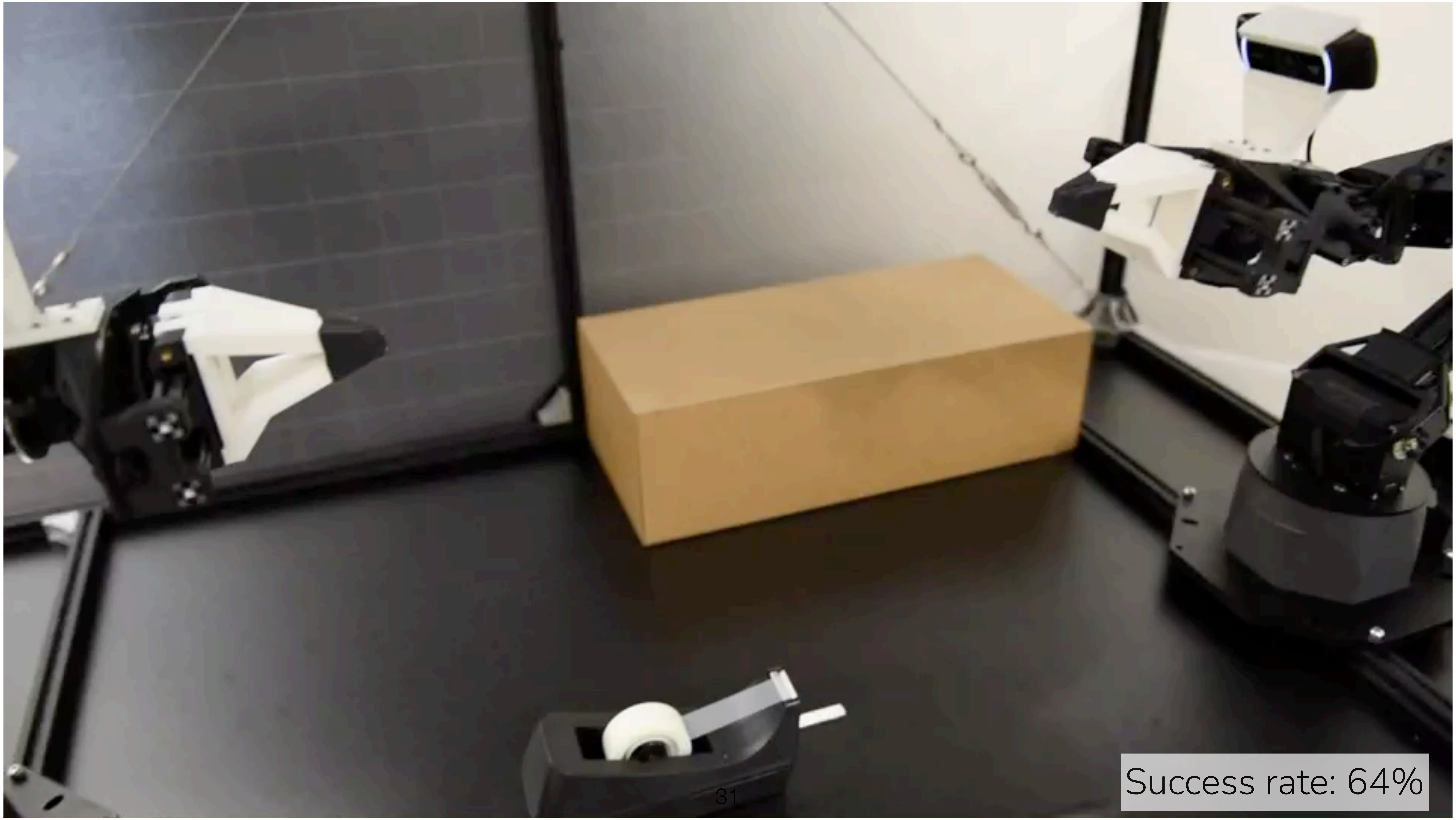


Real Robot Results

Collect 50 demonstrations, randomize object location along white line.

Real Robot Results

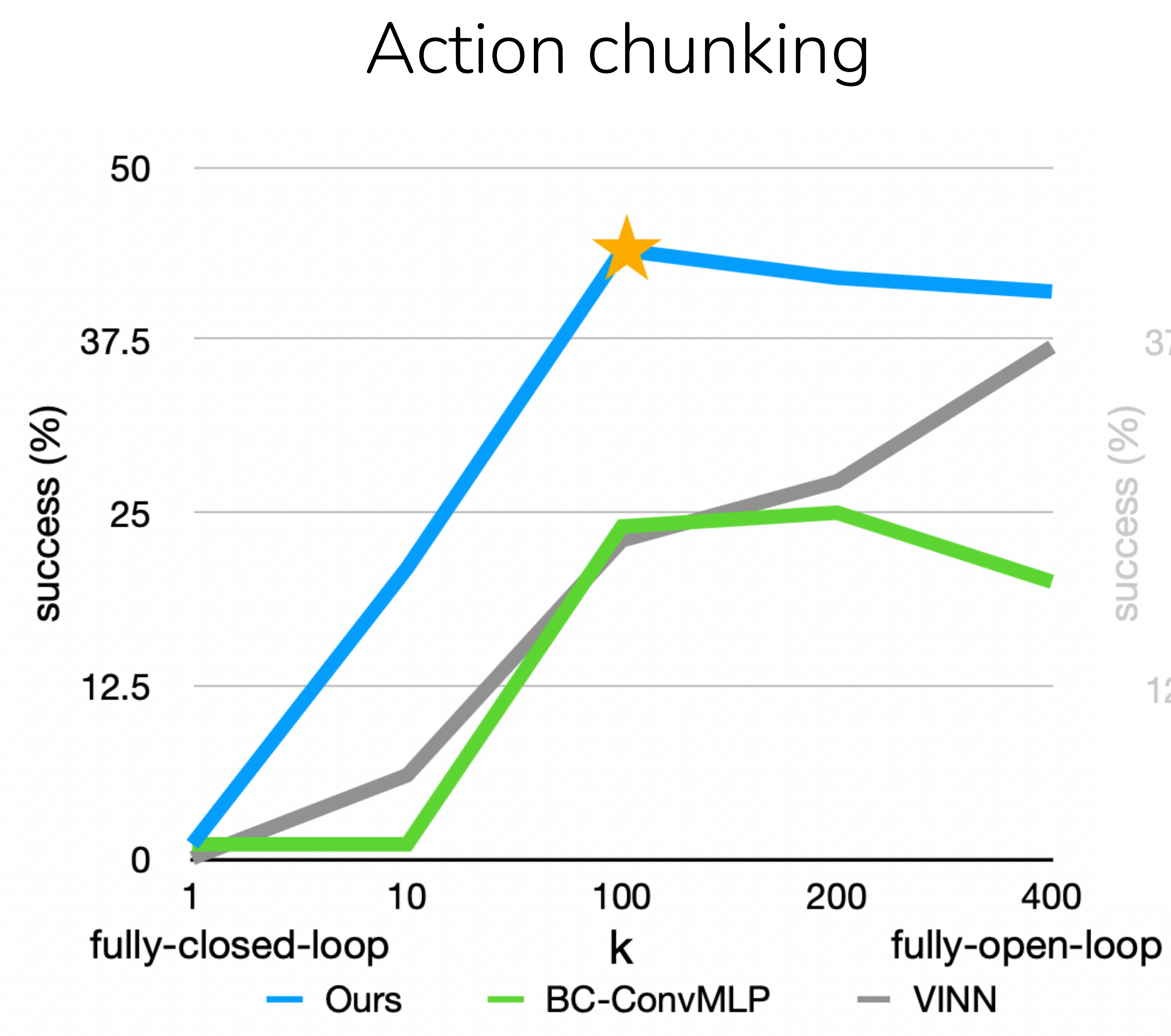
Collect 50 demonstrations, randomize object location along white line.

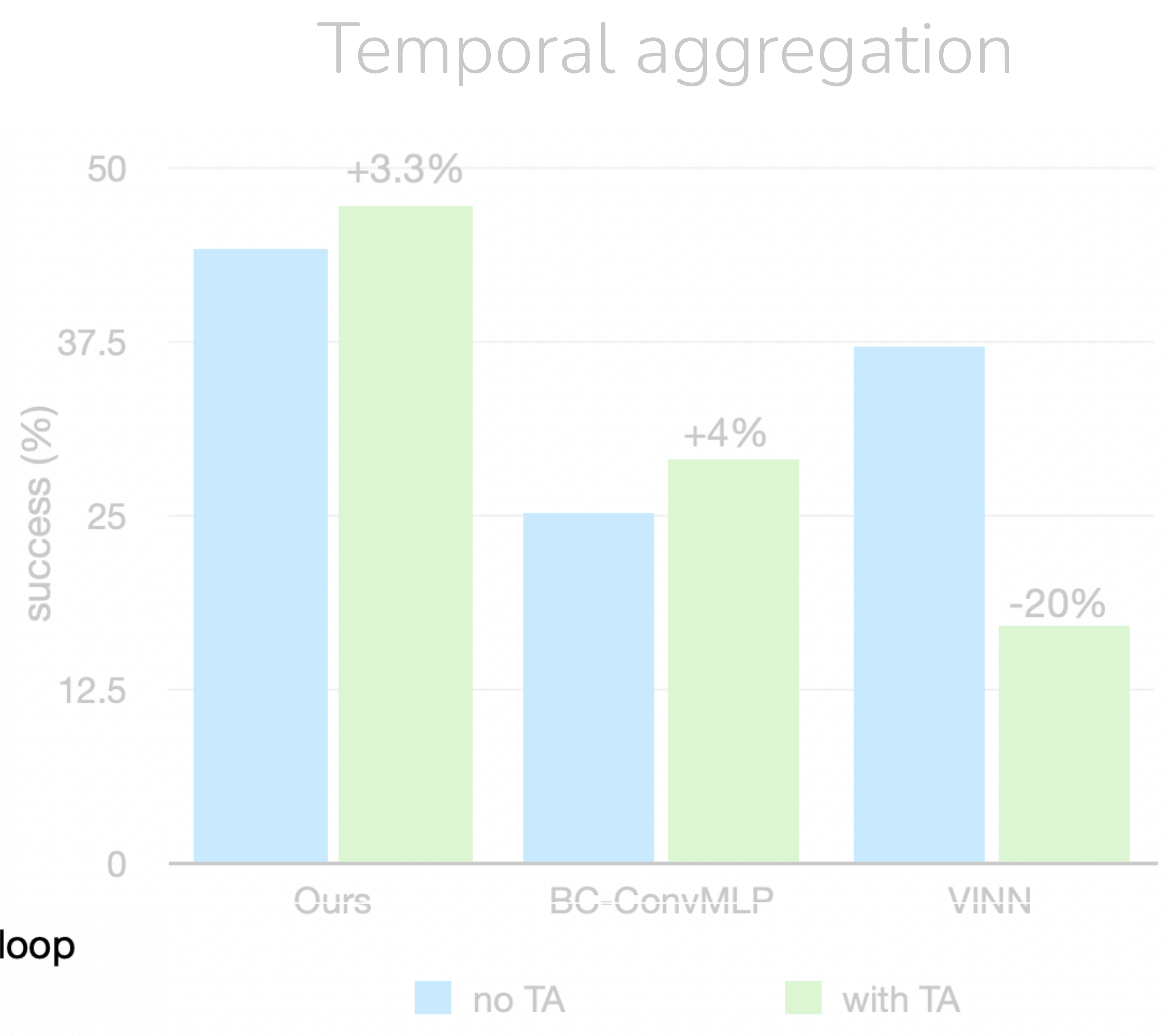


Real Robot Results

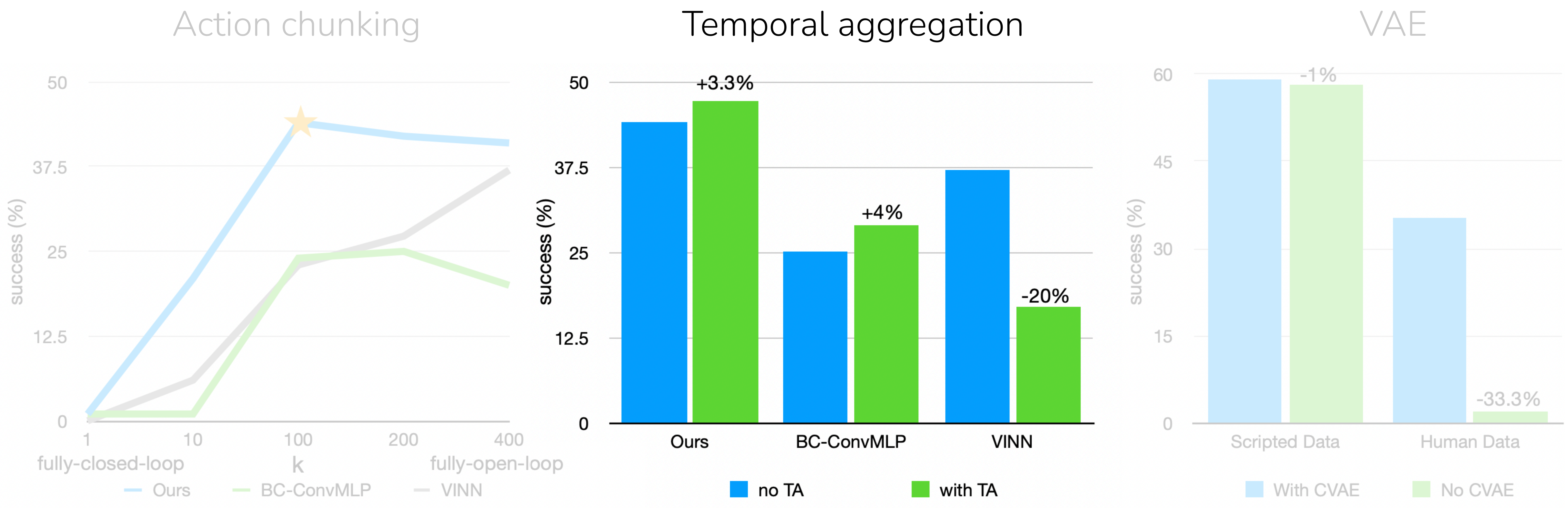
Collect 50 demonstrations, randomize object location along white line.

Simulated Ablations

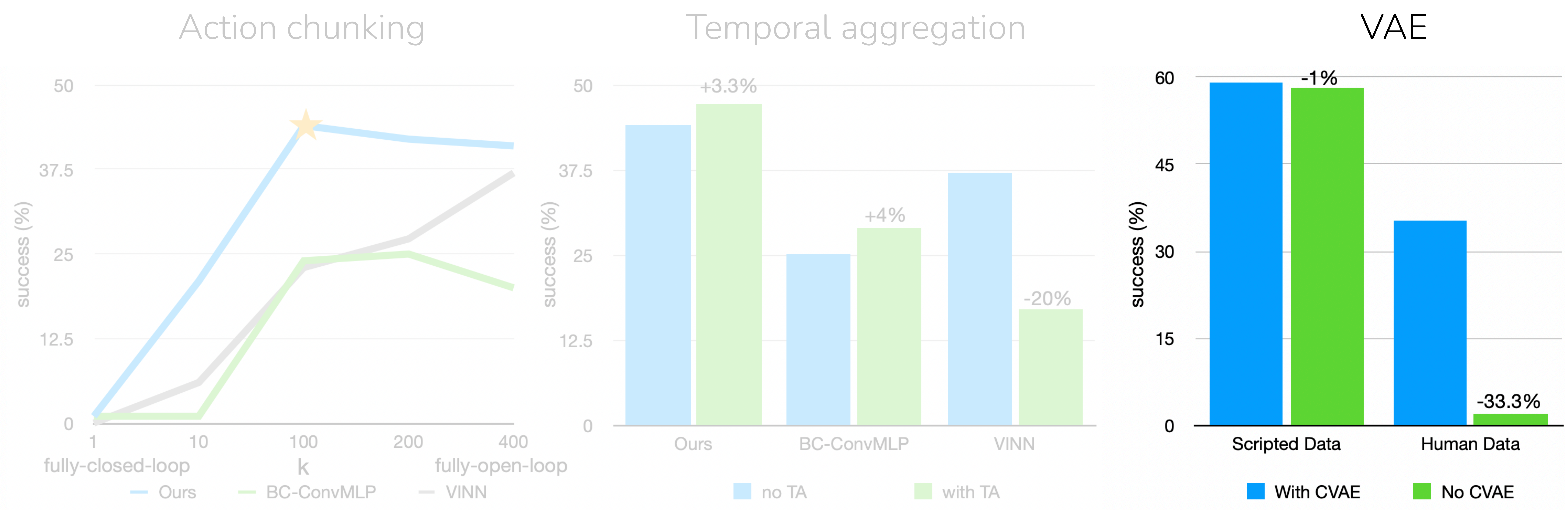




Simulated Ablations



Simulated Ablations



Recap

Imitation Learning

1. Where does the data come from?
2. What can go wrong?
3. Learning from online interventions
4. Case study in fine robotic manipulation

Key learning goals:

- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

Recap

Key learning goals:

- the **basic mechanics** of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

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: $\min_\theta -\mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathscr{D}}[\log \pi_\theta(\mathbf{a} \mid \mathbf{s})]$

Recap

Key learning goals:

- the basic mechanics of imitation learning & how to implement it
- the most **common challenges** & **latest solutions** for addressing them

Common Challenges:

1. Compounding errors
2. Multimodal demonstration data
3. Mismatch in observability

Some Solutions:

-> more data, online interventions

-> use more expressive distributions

-> provide more context, or collect data with less context

Is Imitation Learning All You Need?

A simple & powerful framework for learning behavior!

But:

- **Collecting expert demonstrations** can be difficult or impossible in some scenarios
  - Learned behavior will never be *better* than expert
- Does not provide a framework for **learning from experience, indirect feedback**
  - Can agents learn autonomously, from their own mistakes?

Next time: Start of *reinforcement learning algorithms*

We'll revisit imitation learning in week 4.

Course reminders

- Start forming **final project groups** (survey due Mon April 17)
- **Homework 1** out today, due Weds April 19
- Fill out **AWS form** with account ID by this Friday April 7

News

- Thursday PyTorch tutorial (4:30 pm) moved to **Skilling Auditorium**
- Up to **2% extra credit** for providing TA-endorsed answers on Ed