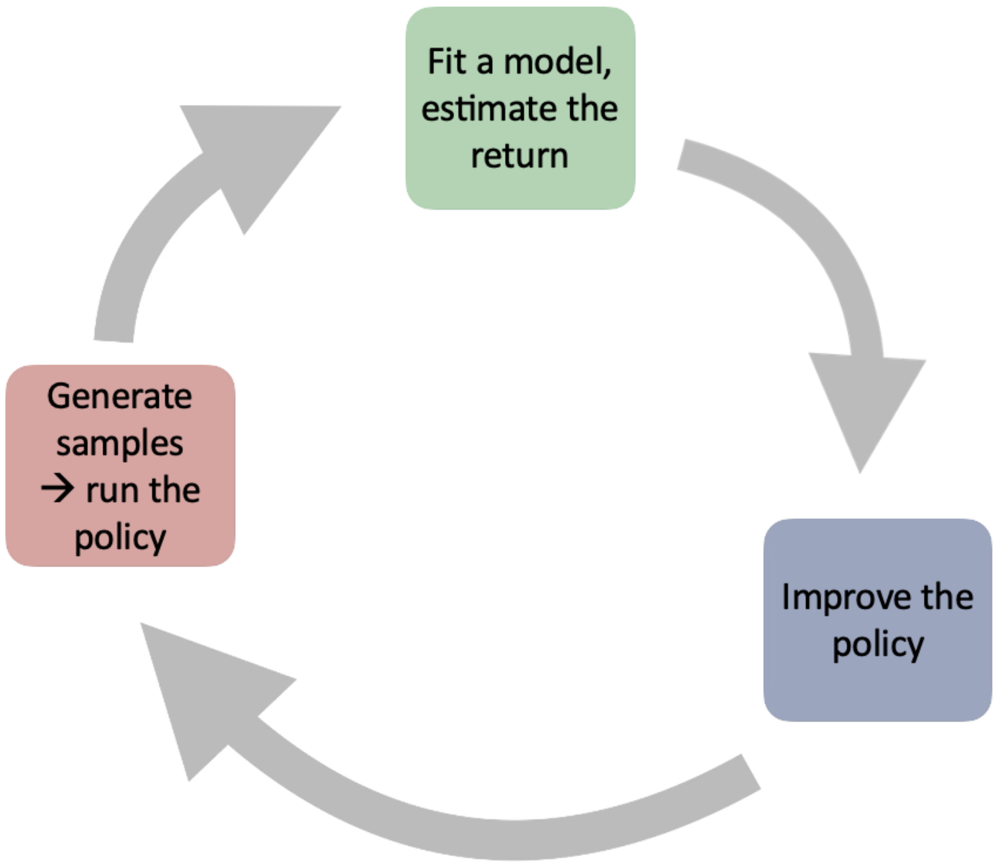
Behavioral Cloning

ESE 6510
Antonio Loquercio

Philadelphia,
1957

What we have seen up to now

- Learning from interaction with the environment:
  - Formalization of trial and error
- Very elegant way of learning:
  - Behaviors are emergent, not hardcoded
  - (In theory) Scalable



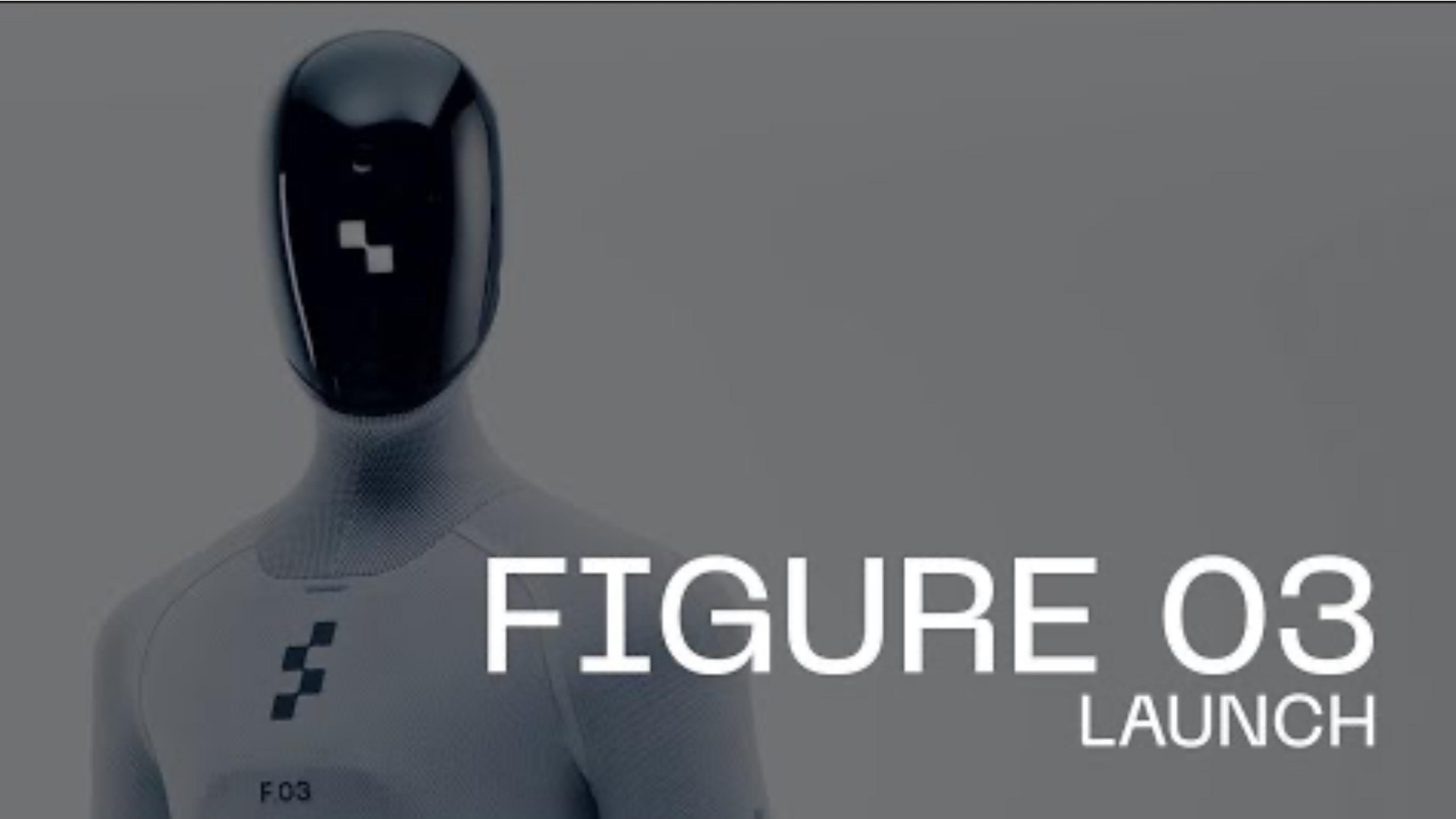


FIGURE 03
LAUNCH
F.03

The Challenges of Learning These Behaviors with RL

- What algorithm (on-policy, off-policy, policy gradient, Q-learning, etc.)?
- What reward?
- Interaction with a real or simulated environment?
  - If real, potentially unsafe and slow.
  - If simulated, how to build this environment?
- A lot of inductive biases sneak in from the back door to overcome these challenges.
  - Pure RL is not truly scalable in practice as of today (but it might be in the future!)

An Alternative Approach: Behavioral Cloning

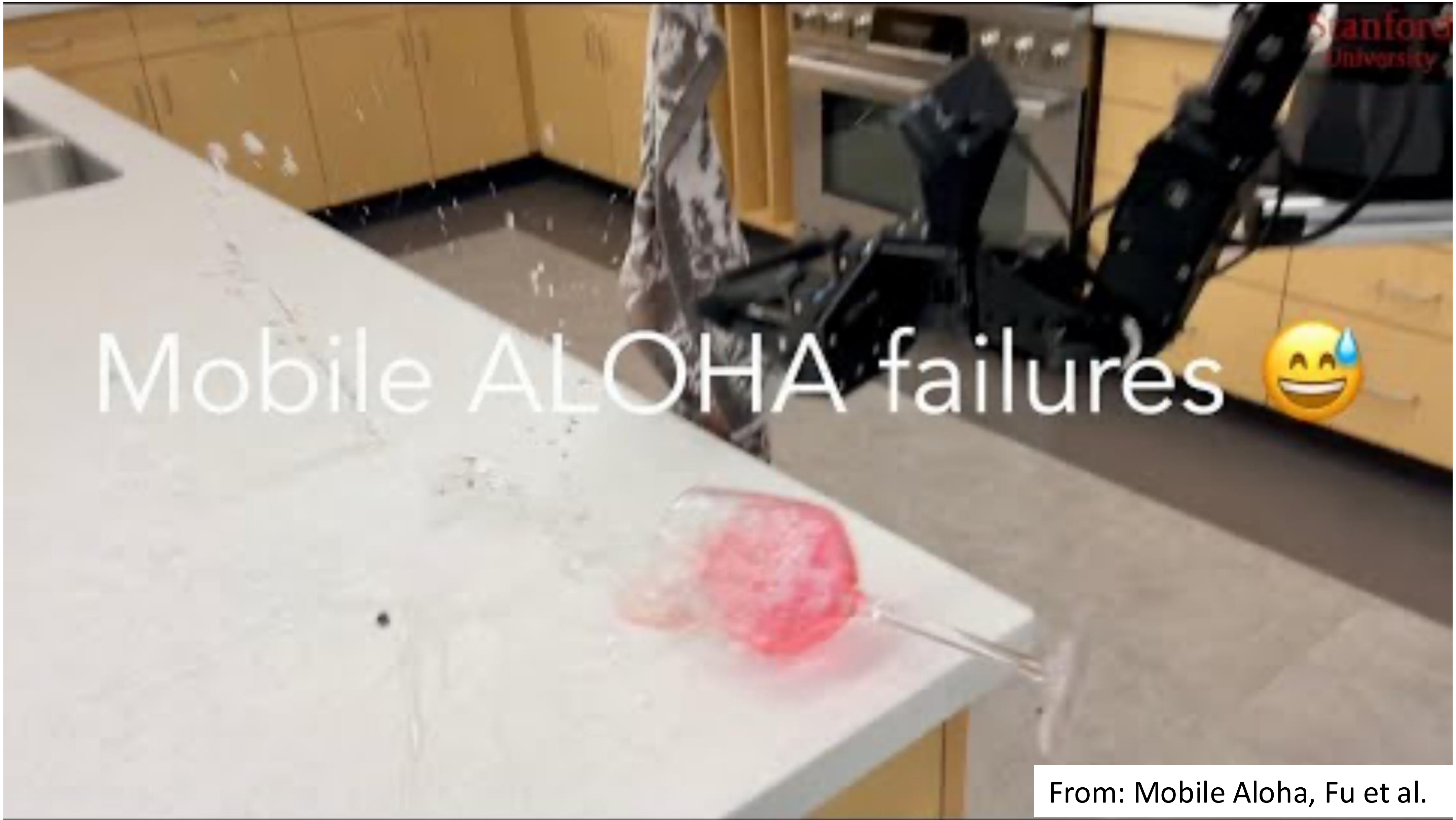
From: Mobile Aloha, Fu et al.

Philadelphia,
1957

An Alternative Approach: Behavioral Cloning

- No rewards.
- No environment creation. Can train directly in the real world.
- Potentially Scalable:
  - For tasks where you can teleoperate
  - If you assume that data collection is a scalable procedure
- (Easier to do a fancy demo)

Not as easy as it sounds



From: Mobile Aloha, Fu et al.

Behavioral Cloning: Agenda

- Theoretical Foundations
- Tools for Data Collection
- Algorithms
- Leveraging foundation models
- Challenges

Supervised Learning 101

- Step 1:
  Collect a dataset

Inputs



Labels

Cat

Glasses

Horse

- Step 2:
  Train a network

Fancy NN

Horse

- Step 3:
  Inference on
  data from same
  distribution

Fancy NN

Cat

Supervised Learning 101

- Random variables x (input) and y (label).
- Dataset of realizations: $\{(x_i, y_i)\}^D$.
- Stochastic network $\pi_\theta(y|x)$ to model the conditional $P(y|x)$.
- Objective:

$$\theta^* = argmax_\theta \sum_i \pi_\theta(y_i|x_i) = \operatorname{argmax}_\theta \sum_i \log \pi_\theta(y_i|x_i)$$

Sequence Labeling

- Sequence of observation and labels

Which object is picked (if any)?

Input x

Label y None None Mozzarella

- Can I optimize the same objective as before?

$$\theta^* = \operatorname{argmax}_\theta \sum_i \log \pi_\theta(y_i|x_i)$$

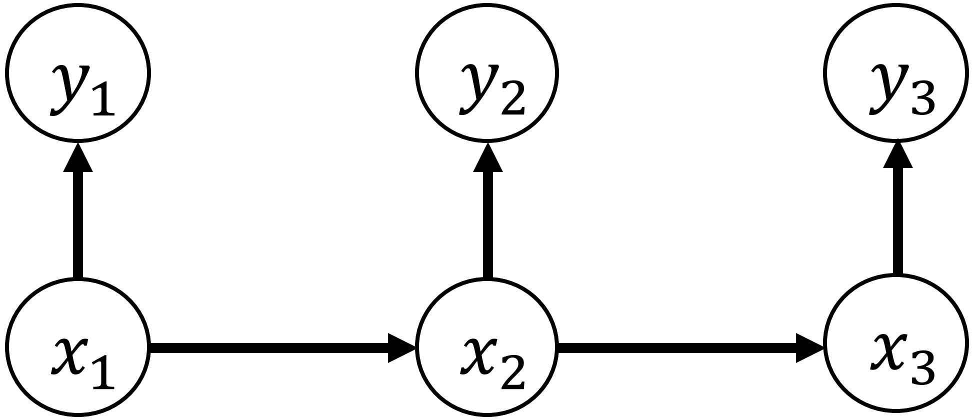
Sequence Labeling

- Can I optimize the same objective as before?

$$\theta^* = \mathrm{argmax}_\theta \sum_i \log \pi_\theta(y_i|x_i)$$

- Yes, but only if the labels are independent!

$$P(y_t|x_{0:t}, y_{0:t-1}) = P(y_t|x_t)$$



Behavioral Cloning

- Connection to sequence labeling: $x_t = s_t, y_t = a_t$

Input s

Label a

Straight

Left

Right

- Can we optimize the same objective as before?

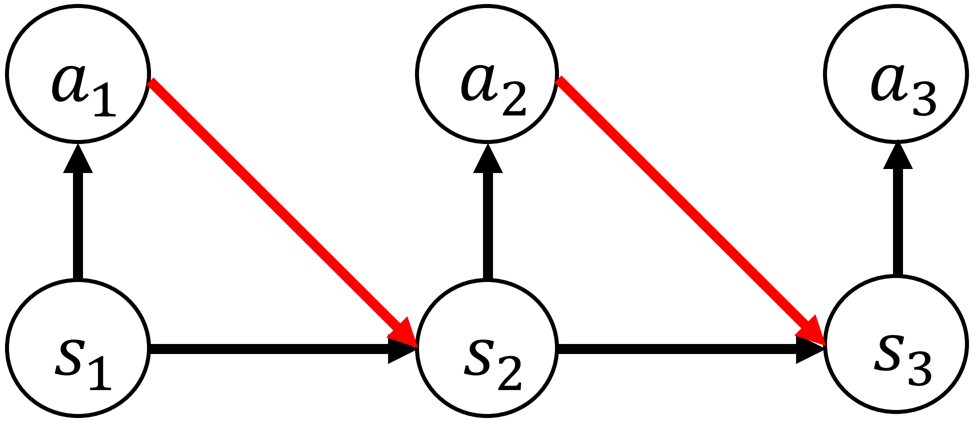
$$\theta^* = \text{argmax}_\theta \sum_i \log \pi_\theta(a_i | s_i)$$

Behavioral Cloning

- Can we optimize the same objective as before?

$$\theta^* = \text{argmax}_\theta \sum_i \log \pi_\theta(a_i|s_i)$$

- Technically no, because distribution is different.



Behavioral Cloning

- What we can optimize (ρ_{π^*} is the trajectory distribution of the expert):

$$\theta^* = \text{argmax}_\theta \sum_{a_i, s_i \sim \rho_{\pi^*}} \log \pi_\theta(a_i | s_i)$$

- What we should to optimize (ρ_{π_θ} is the trajectory distribution of the student):

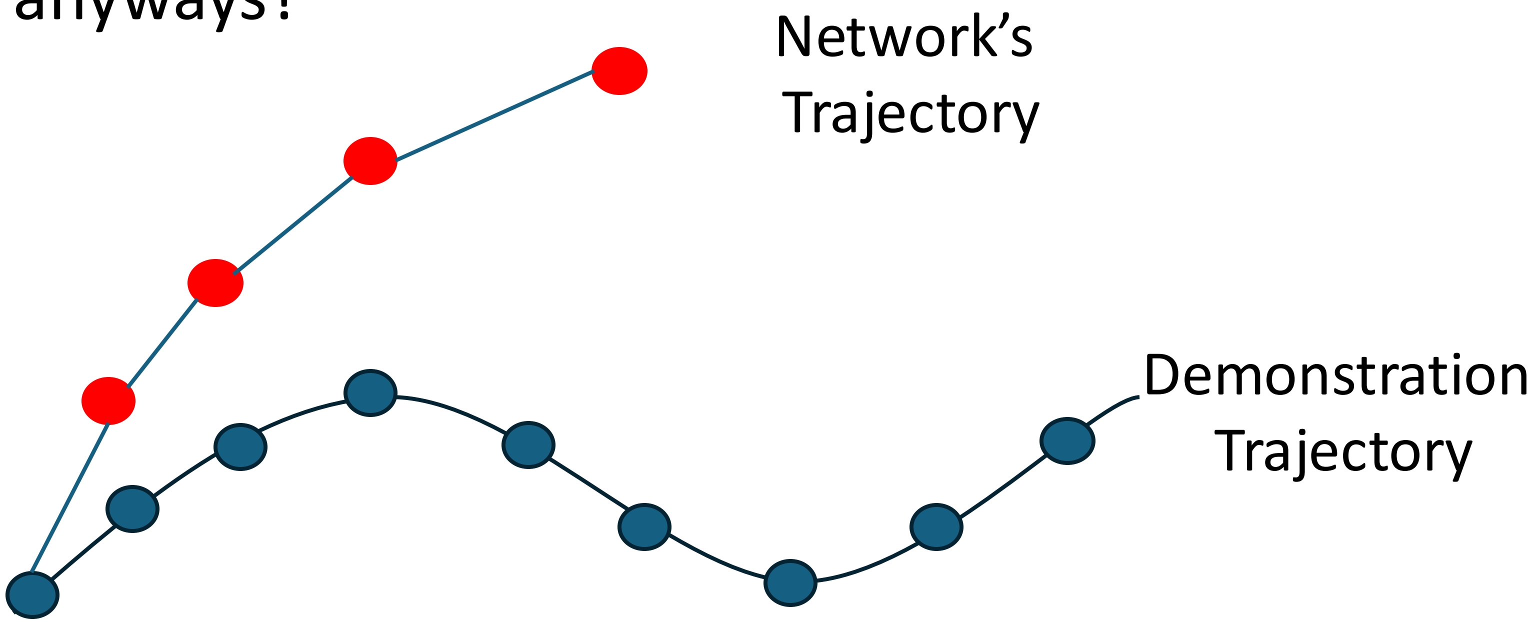
$$\theta^* = \text{argmax}_\theta \sum_{a_i, s_i \sim \rho_{\pi_\theta}} \log \pi_\theta(a_i | s_i)$$

Behavioral Cloning

- Can we optimize the same objective as before?

$$\theta^* = \operatorname{argmax}_\theta \sum_i \log \pi_\theta(a_i | s_i)$$

- What if I do it anyways?



Behavioral Cloning

- What if I do it anyways?
- Define errors as: $c(s_t, a_t) = \begin{cases} 0 \text{ if } a_t = \pi^*(s_t) \\ 1 \text{ otherwise} \end{cases}$

- You can formally prove that $\sum_t \mathrm{E}_{s_t \sim \rho_\pi}[c(s_t, \pi(s_t))] < O(\epsilon T^2)$, where:

- T is the episode length
- ϵ is the average training error, i.e. $\mathrm{E}_{s_t \sim \rho_{\pi^*}}[|\pi(s_t) - \pi^*(s_t)|]$

Detailed analysis in: A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning

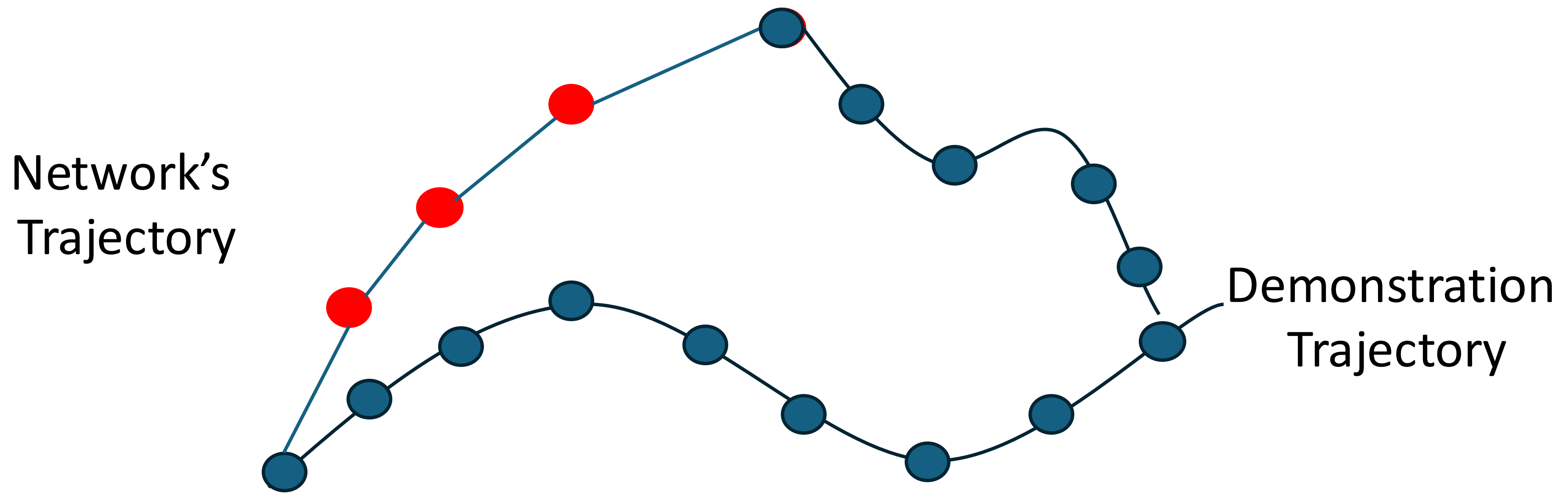
Behavioral Cloning

- Hand-wavy proof that $\sum_t \mathrm{E}_{s_{t \sim \rho_\pi}}[c\big(s_t, \pi(s_t)\big)] < O(\epsilon T^2)$
- Define d_t as the expected "distance" between the agent and the expert's trajectory at time t, then $\mathrm{d}_{\mathrm{t+1}} \leq d_t + \epsilon$
- This implies that $d_t \leq t\epsilon$
- Therefore, all possible states where the agent can get (and therefore make mistakes) is:

$$\sum_t d_t \leq \sum_t t\epsilon = O(\epsilon T^2)$$

Behavioral Cloning

- Performance is expected to decrease quadratically with the episode length.
- But we have seen examples of autonomous policies going on for minutes. How is this possible?
- Collect correction behaviors!



Behavioral Cloning

- Performance is expected to decrease quadratically with the episode length.
- But we have seen examples of autonomous policies going on for minutes. How is this possible?
- Collect correction behaviors!
- A lot of effort in imitation learning today is spent on finding ways to collect as few correction behaviors as possible:
    - Use powerful backbones that can automatically figure out how to correct.
    - Input preprocessing schemes to decrease the dim of the state space (e.g., 3D)
    - Use an algorithm to find out failure cases and collect corrections (Dagger).

Behavioral Cloning: Agenda

- Theoretical Foundations
- **Tools for Data Collection**
- Algorithms
- Leveraging foundation models
- Challenges

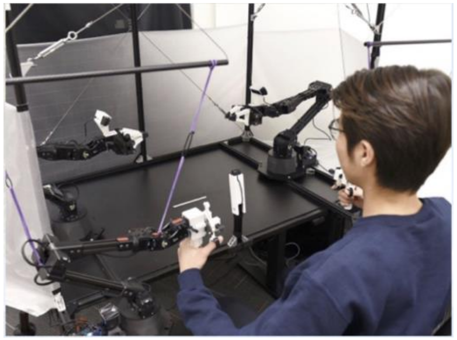
Collecting Demonstrations

- **General Goal**: collect the history of expert actions (e.g., joint position) and observations (e.g., camera views) to train BC policies.

- Two main categories of data collection:

Kinesthetic Teaching

Direct guidance



Puppeteering

Teleoperation

VR Controller

Spacemouse

Images from:
Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware
The Impact of Demonstration Modality on Imitation Learning

Kinesthetic Teaching via Direct Guidance

- Advantages:

- Can directly feel the robot joint limits.
- No need for external devices.

- Disadvantages:

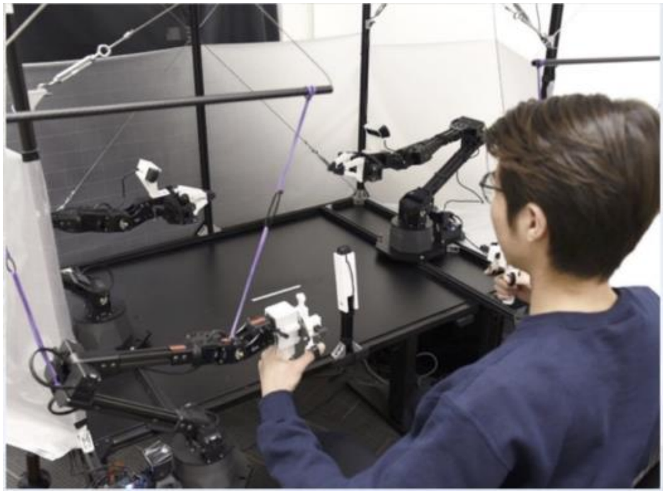
- You don’t collect actions but only a sequence of joint positions. Need tricks to recover actions.
- Slow and troublesome (the human can potentially occlude sensors).

- Not very much used in practice.

Kinesthetic Teaching via Puppeteering

- Advantages:

- Can directly feel the robot joint limits.
- Directly recovers actions and observations.
- Can perform very precise tasks.



- Disadvantages:

- Doubling hardware requirements (and costs). Robot-specific.
- Slow and tiring (controlling all joints requires a lot of attention/training).

- Used a lot in both industry and academia. Painful to use/scale.

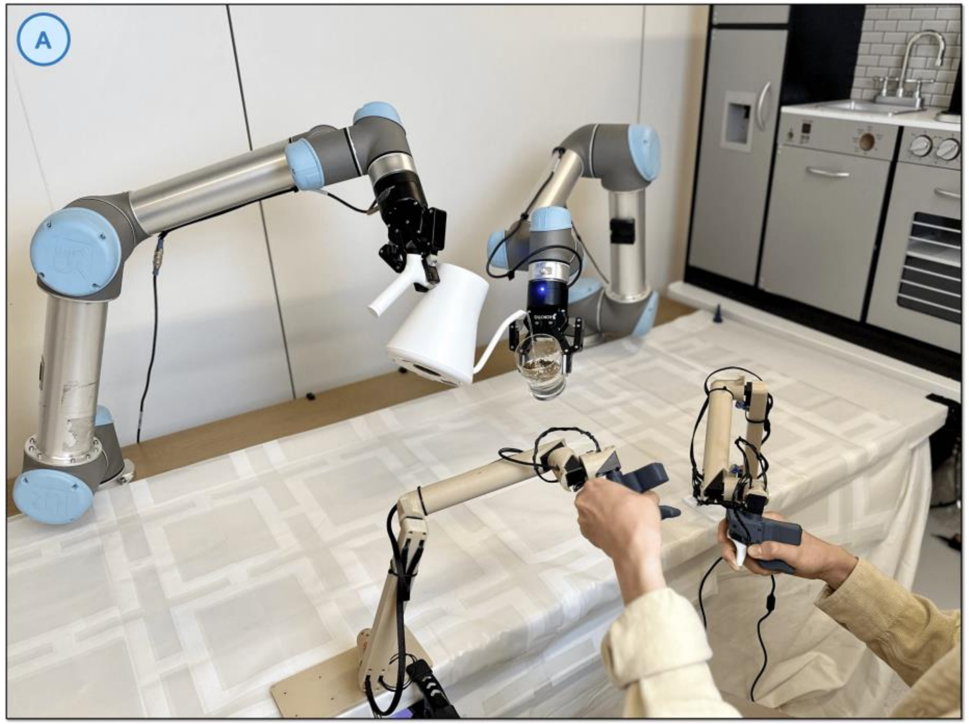
Kinesthetic Teaching via Puppeteering

From: Mobile Aloha, Fu et al.

Kinesthetic Teaching via Puppeteering: Devices

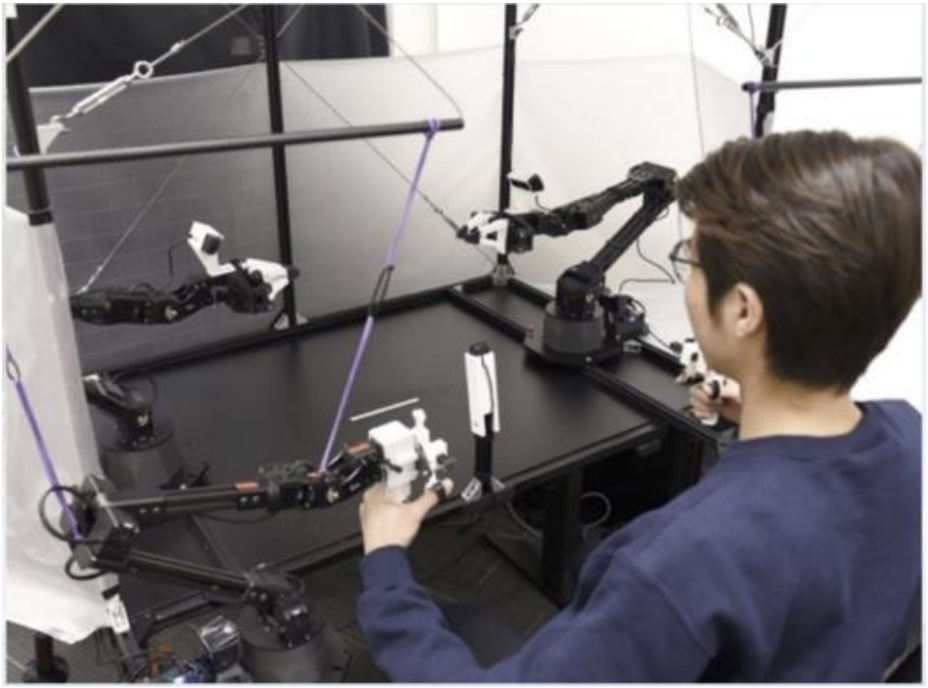
- Many options with vast differences in price.

Lower end (a few hundred $)



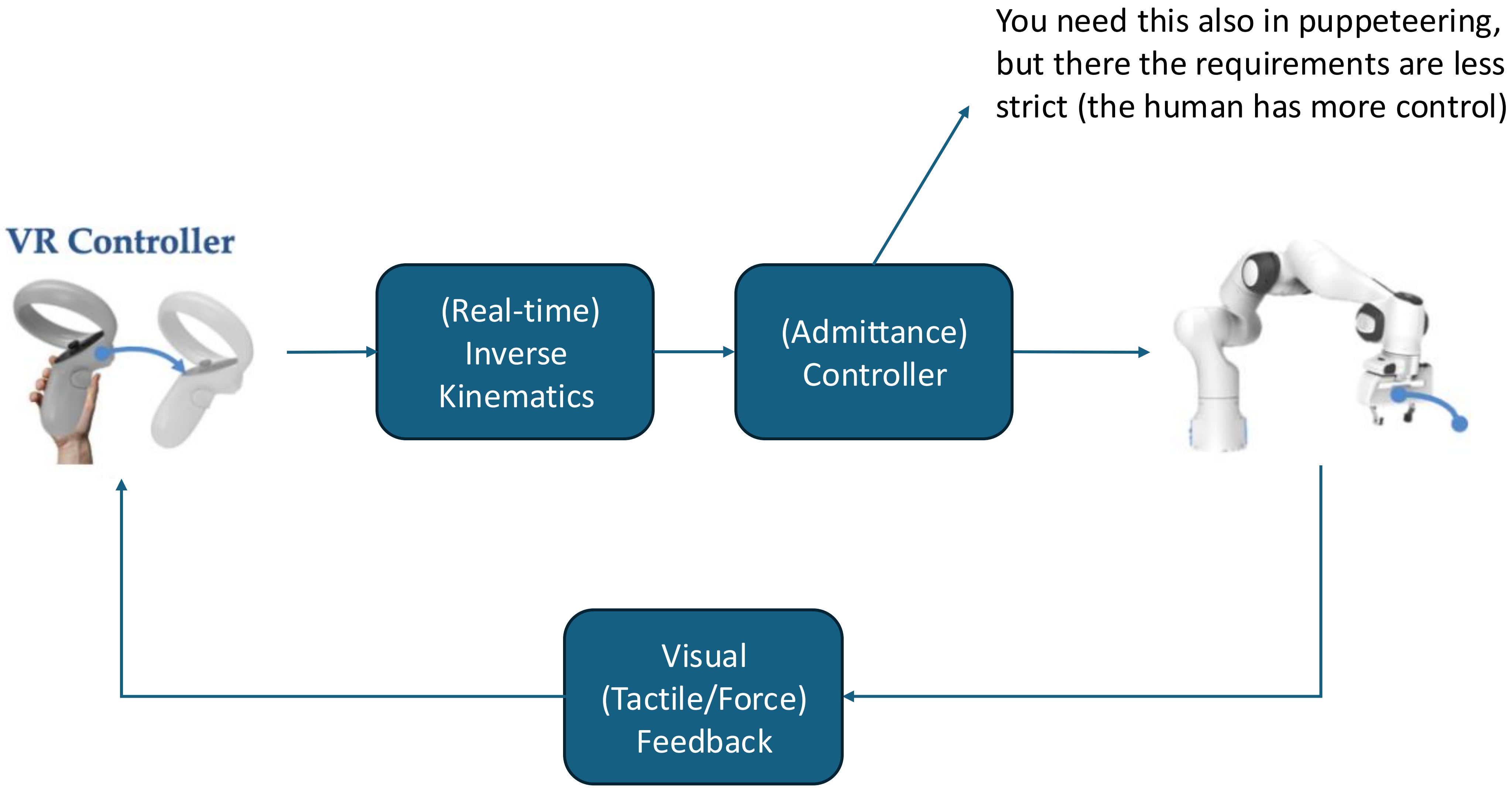
GELLO: A General, Low-Cost, and Intuitive Teleoperation Framework for Robot Manipulators

Higher end (robot cost * 2)



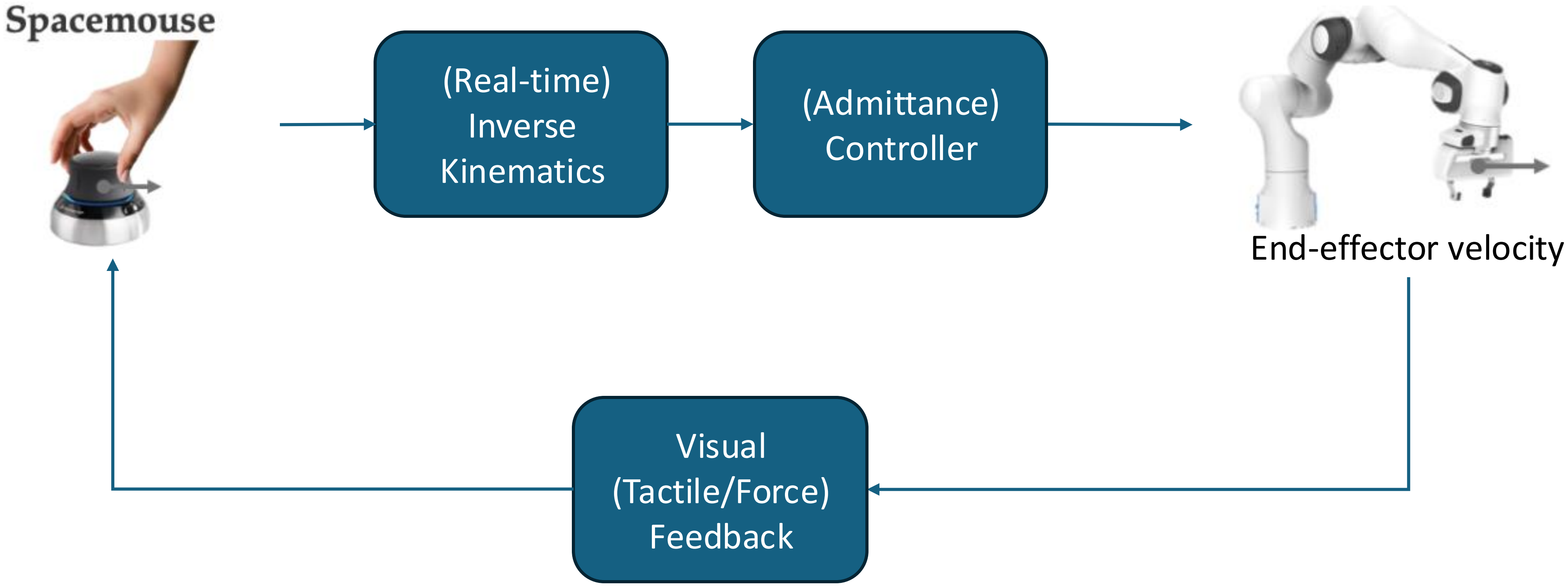
Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware

Teleoperation



Teleoperation

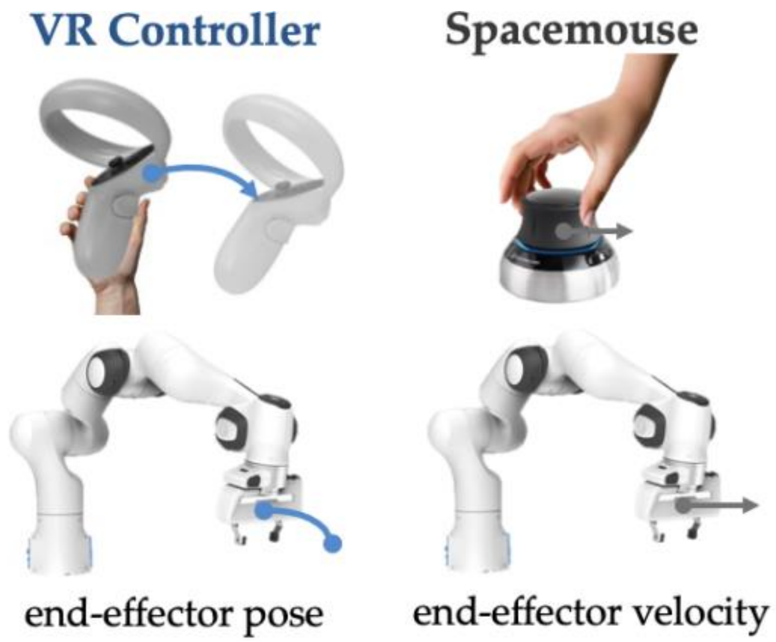
- The robot can be either controlled in position space or in velocity space, depending on the device.



Teleoperation

- Advantages:

- Accessible and low-cost.
- Generalizes across robots.
- Less tiring than other mechanisms.

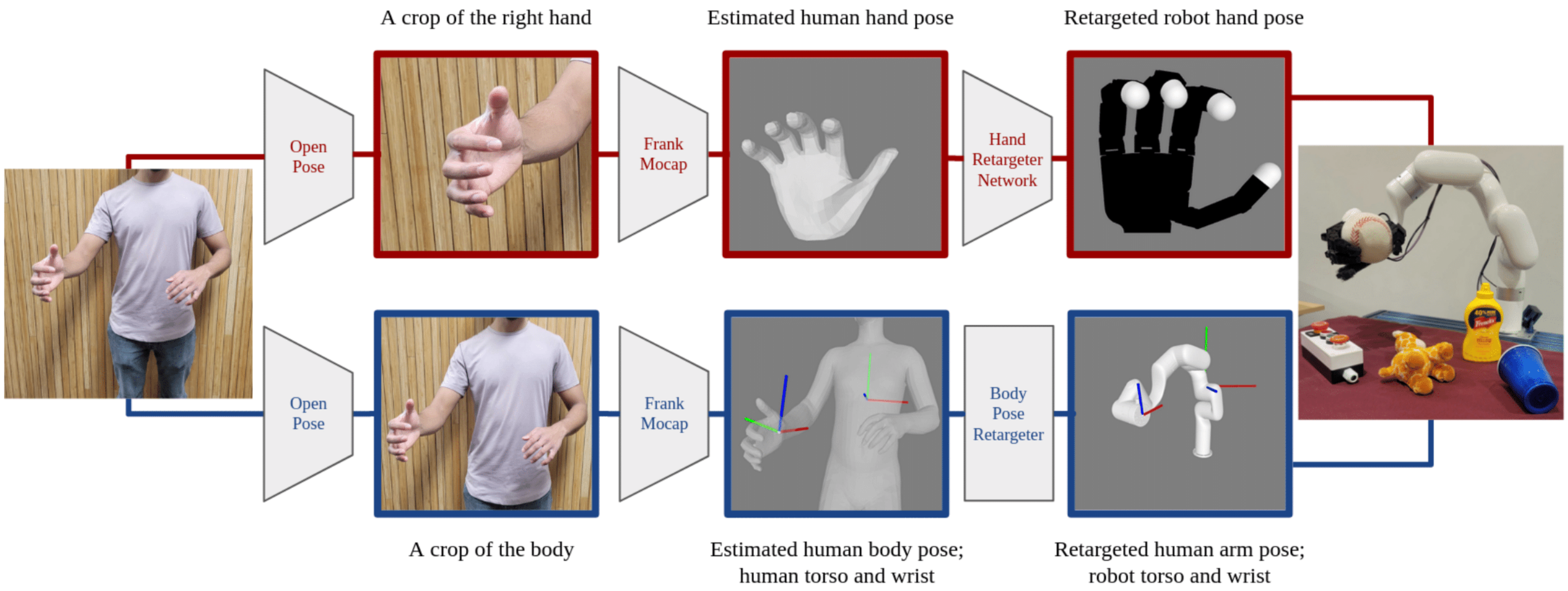


- Disadvantages:

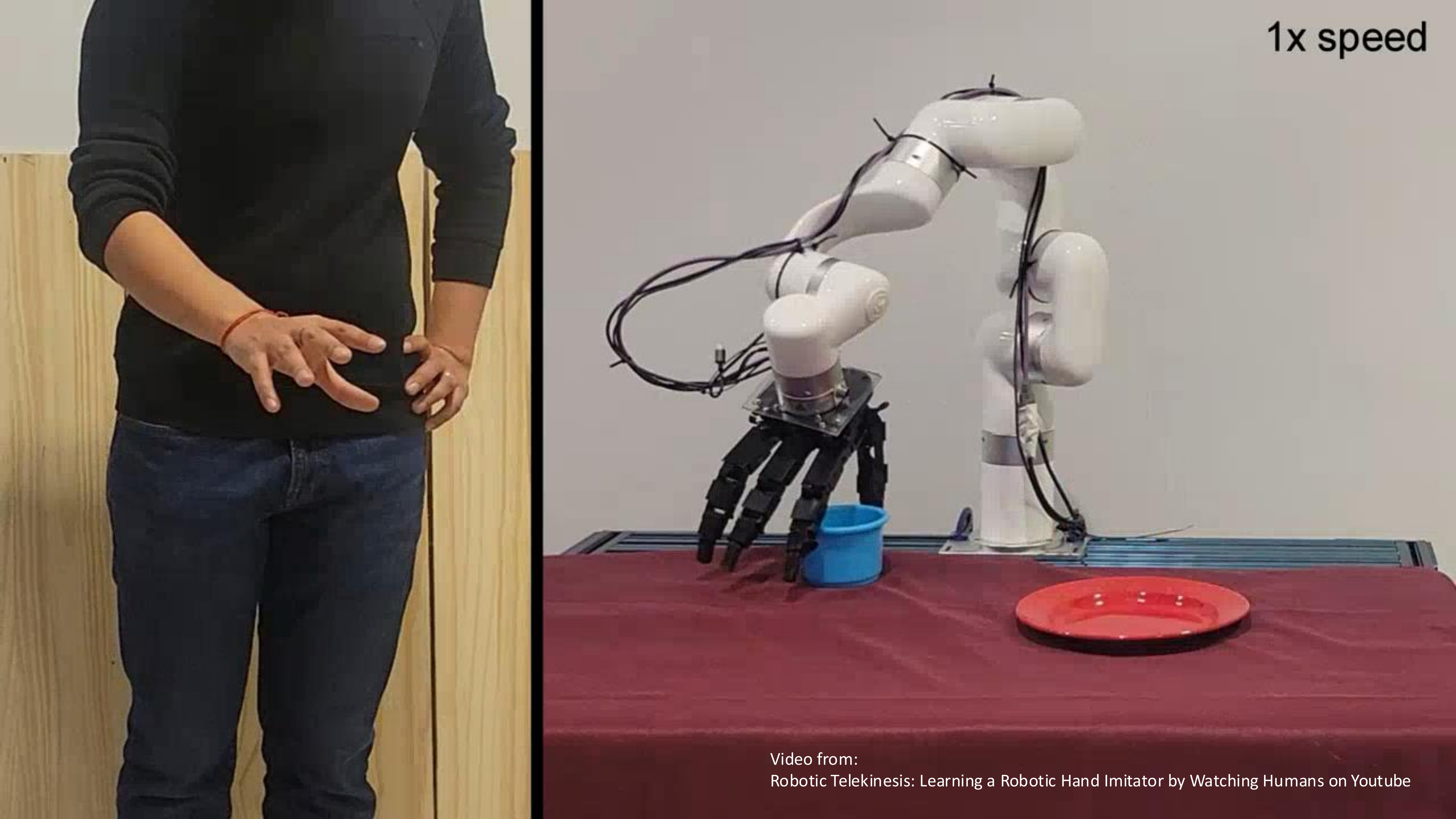
- Delays from IK and the controller increase latency.
- Small errors in positioning/velocity make precise teleoperation hard.
- Less control of the robot (e.g., only end-effector and not the whole body).

Teleoperation: Devices

- Many options with vast differences in price.
- Cheapest option: Vision-based estimation. No external devices.



Images from:
Robotic Telekinesis: Learning a Robotic Hand Imitator by Watching Humans on Youtube



Video from:
Robotic Telekinesis: Learning a Robotic Hand Imitator by Watching Humans on Youtube

Teleoperation: Devices

- Many options with vast differences in price.

Lower end (a few hundred $)

Higher end (up to 90K per side)

Benefits

- ✓ Large workspace to work at scale1
- ✓ High force
- ✓ Great measurement resolution
- ✓ Telerobotic ergonomic handle
  - 4 user buttons
  - 1 led indicator
  - 1 analog finger gripper 0 - 100% (7^{th} DOF)
  - Hand presence detection feature
  - Tool changer via a connector, no tools required
- ✓ Static and active gravity compensation capability.
- ✓ Professional real-time Ethernet/UDP or EtherCat communication up to 1000 Hz
- ✓ Compact form factor: 12 kg

Virtuose 6D TAO

Industrial grade force-feedback device designed for robotics applications

Teleoperation: Devices

- Many options with vast differences in price.
- Cost is roughly proportional to the amount of feedback that the operator gets back from the robot (e.g., force feedback).
- Need a lot of extra knobs to implement specific maneuvers (stop, go to origin, rotate, etc.)

Teleoperation: Mobile Robots

Quite popular in industry

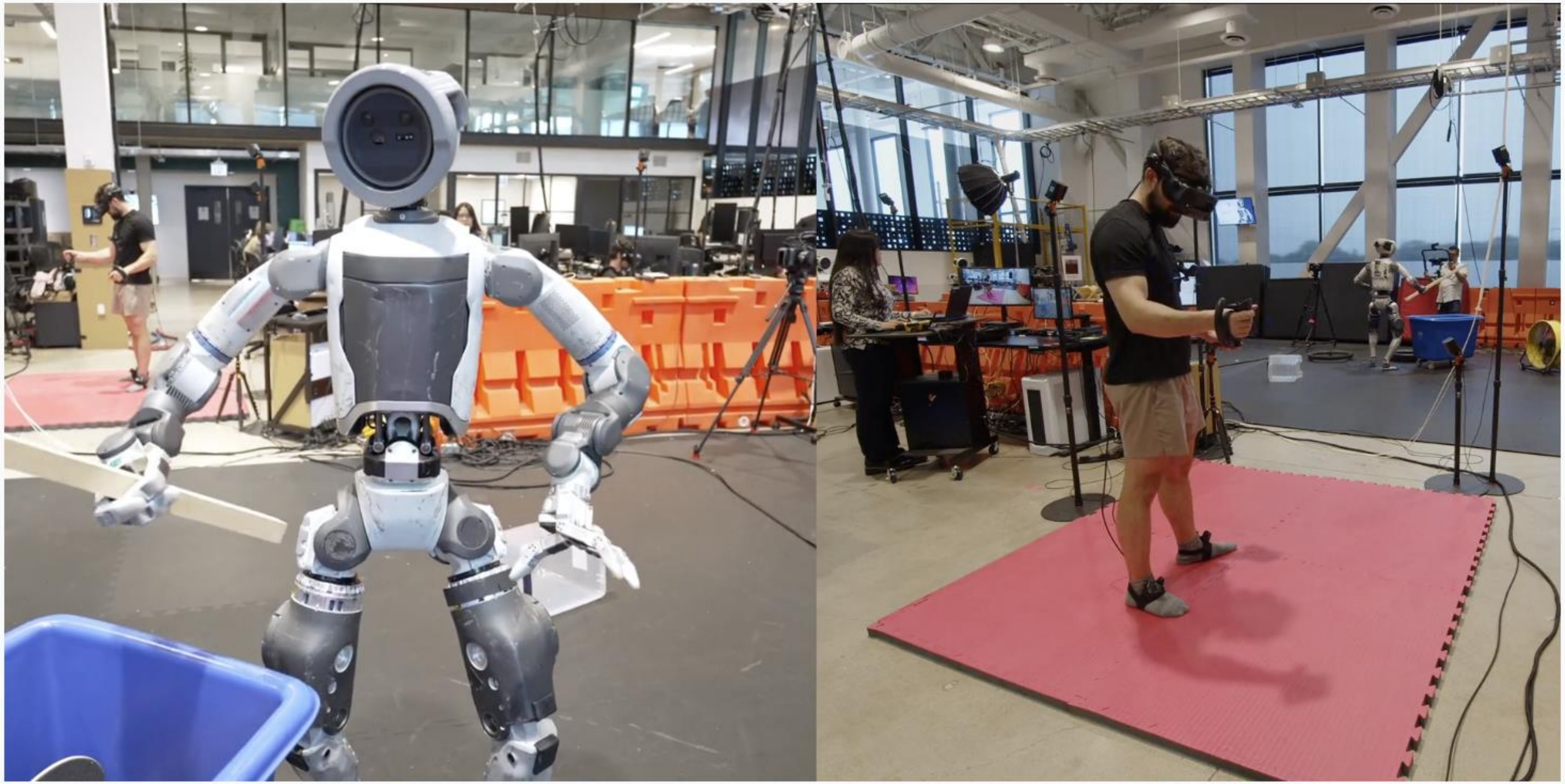
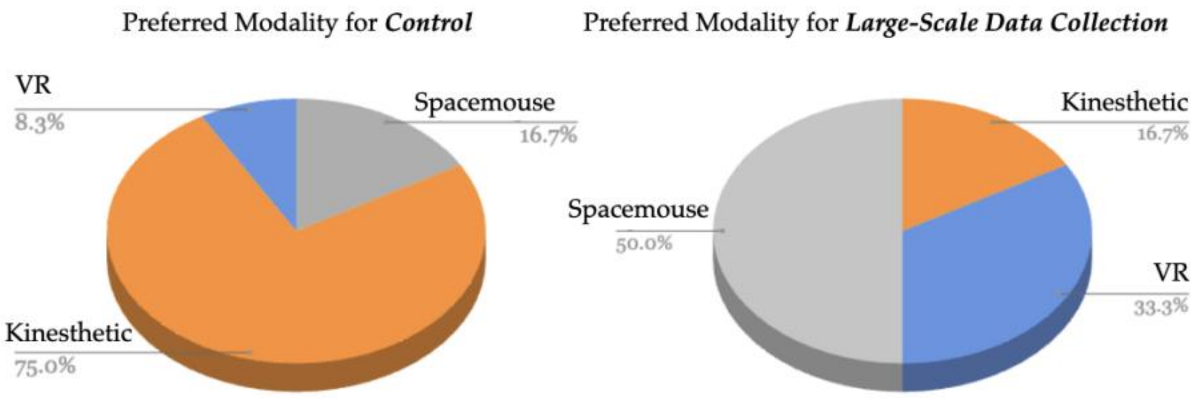
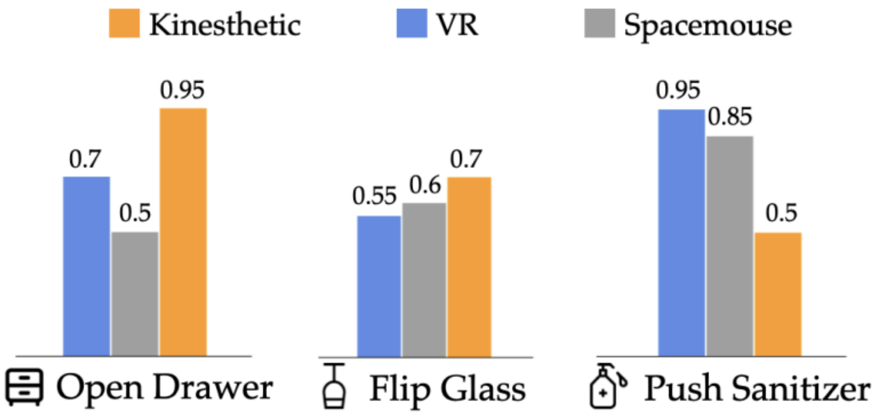


Image from:
BD-AI Research Blog

Data Collection Tools: Summary

- Kinesthetic Teaching
  - Direct Control
  - Puppeteering
- Teleoperation
  - Position space/velocity space end effector control





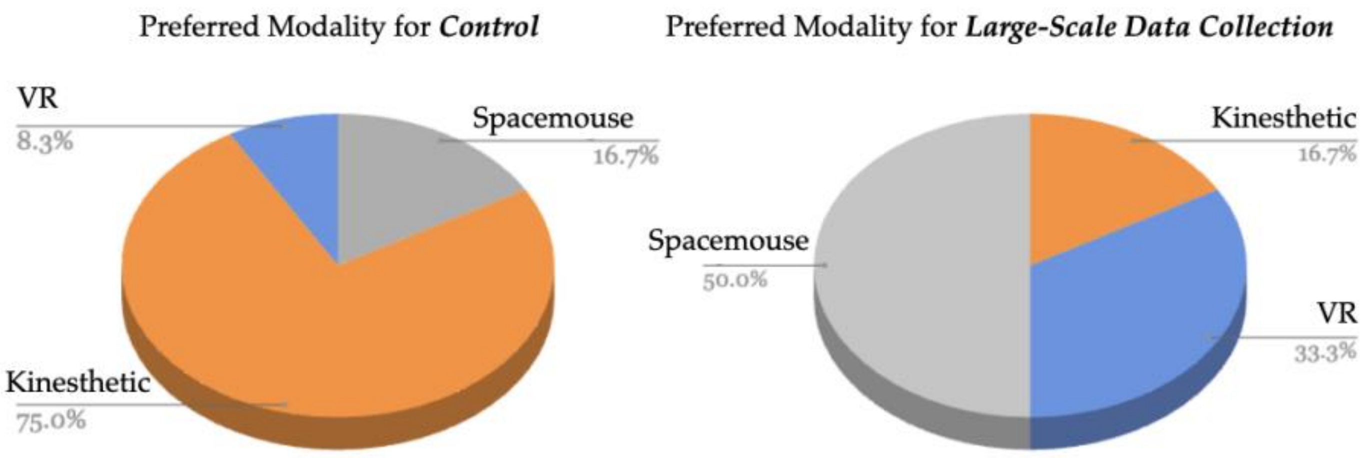
Final Policy Performance

Images from:
The Impact of Demonstration Modality on Imitation Learning

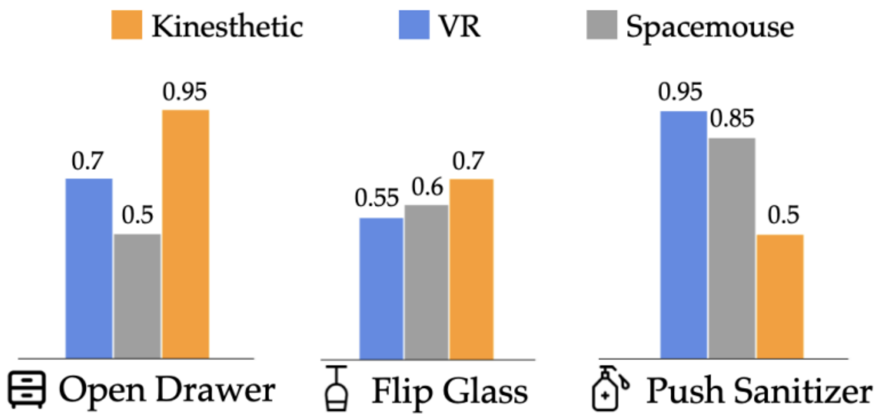
Data Collection Tools: Summary

- The two methods have different trade-offs.

Users Preference

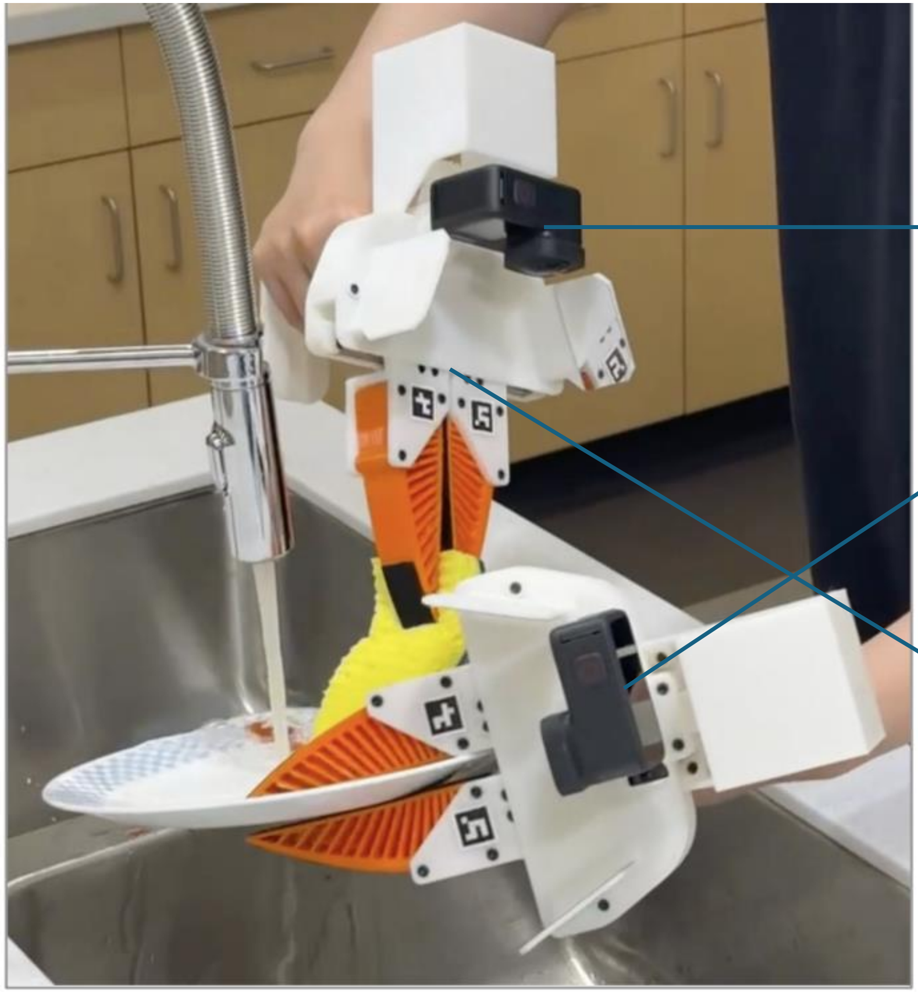


Final Policy Performance



Images from:
The Impact of Demonstration Modality on Imitation Learning

Hybrid Systems: “Wearing” a Robot



Universal Manipulation Interface:
In-The-Wild Robot Teaching Without In-The-Wild Robots

Estimate camera position using SLAM.

IK + Control to follow the same trajectory with the robot’s end effector.

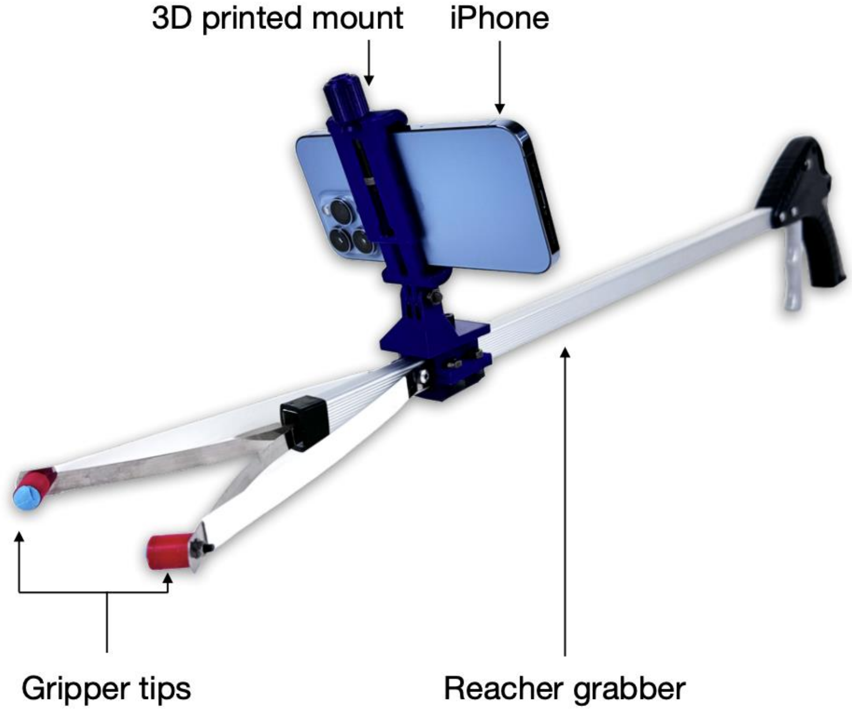
Record Joint States (like puppeteering)

Hybrid Systems: “Wearing” a Robot

Universal Manipulation Interface:
In-The-Wild Robot Teaching Without In-The-Wild Robots

Hybrid Systems: “Wearing” a Robot

Cheaper version

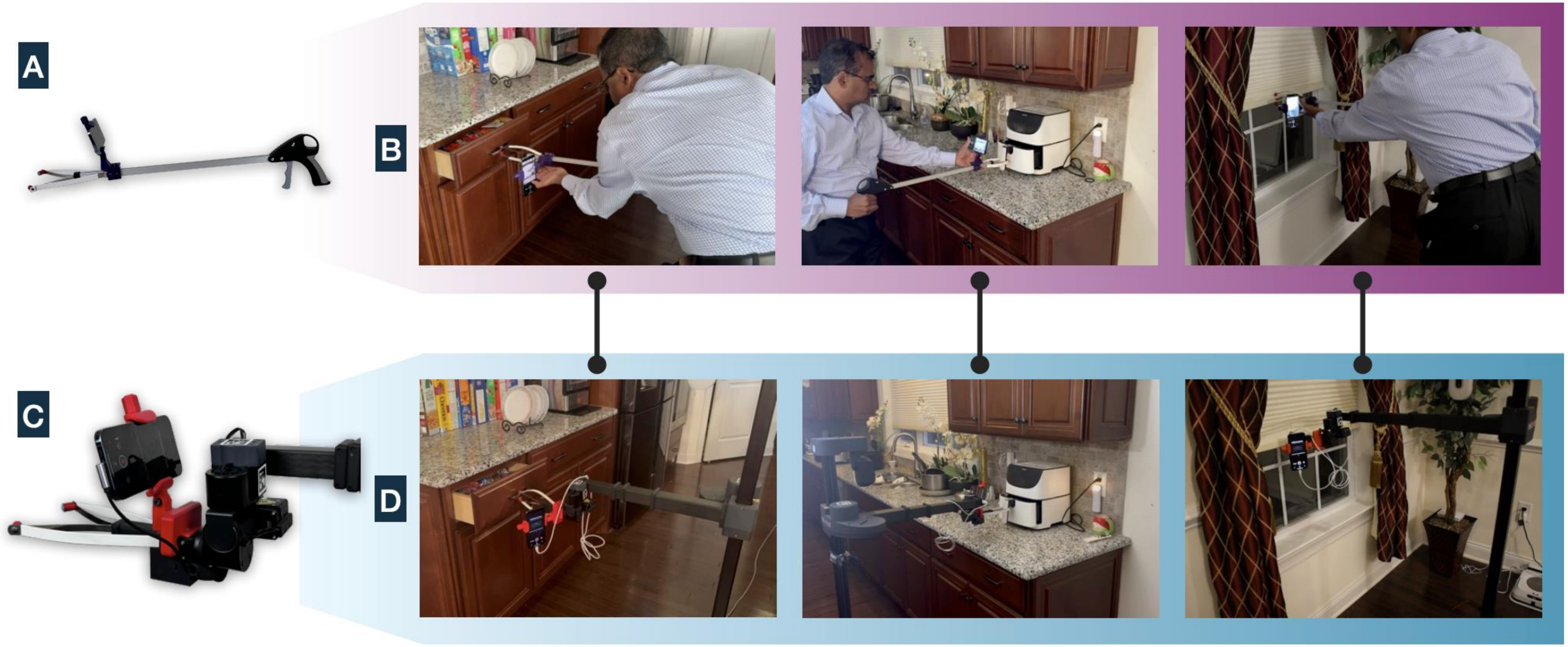


(A) The Stick

Images from:
On Bringing Robots Home

Hybrid Systems: “Wearing” a Robot

Cheaper version



Images from:
On Bringing Robots Home

Hybrid Systems: “Wearing” a Robot

Maybe used at Tesla?

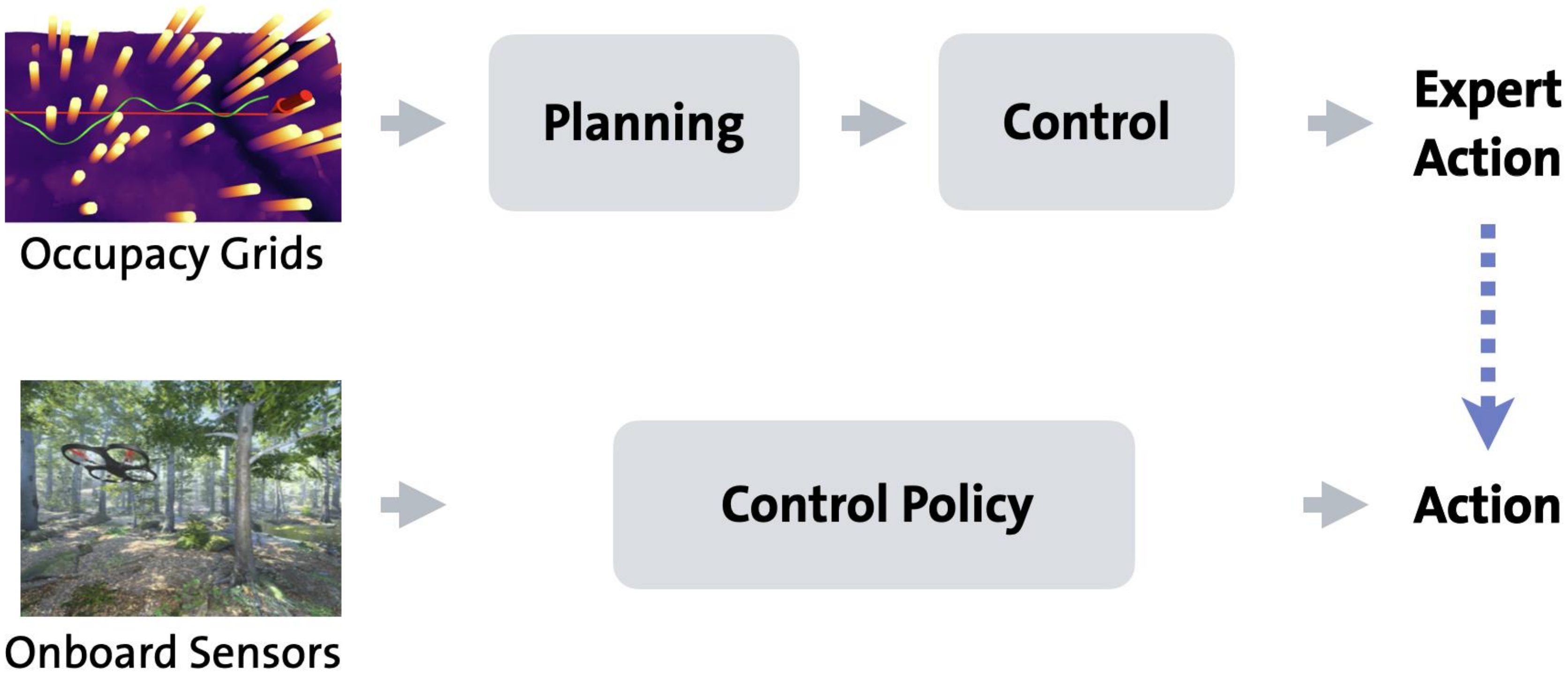
Image from:
Tesla X’s account

Data Collection Tools: Summary

- Kinesthetic Teaching
  - Direct Control
  - Puppeteering
- Teleoperation
  - Position space/velocity space end effector control
- Hybrid Systems
  - Aim to combine the best of both worlds

Collecting Demonstration with Algorithmic Experts

- Requires access to privileged information at training time (available, for example, in simulators).



Images from:
Deep Drone Acrobatics

Collecting Demonstration with Algorithmic Experts

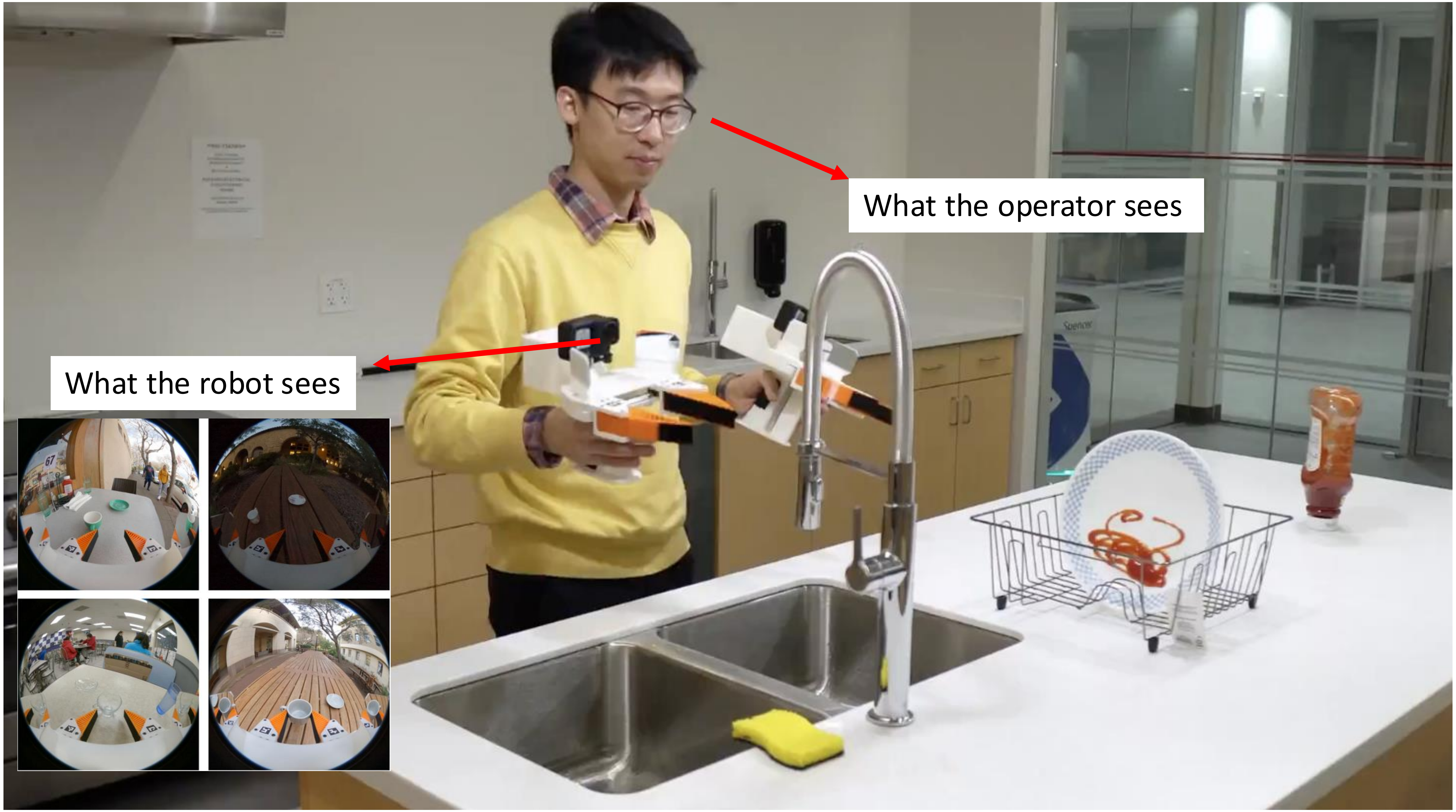
- Advantages:
- No human required.
- An algorithm’s actions are potentially easier to predict than those of humans.

- Disadvantages:
- Only possible when we have privileged information at training time and we know how to build an expert.

Data Collection Tools: Summary

- Kinesthetic Teaching
  - Direct Control
  - Puppeteering
- Teleoperation
  - Position space/velocity space end effector control
- Hybrid Systems
  - Aim to combine the best of both worlds
- Algorithmic Experts
  - Only feasible in very specific applications

A Potential Failure Mode: Difference in Observation



A Potential Failure Mode: Difference in Observation

- If the collected observations do not contain enough information for predicting the expert actions, there is nothing you can do.

- This is something challenging to get used to as an operator.

- Possible solutions:
  - Add many external cameras.
  - Immersive devices.

Immersive Devices for Data Collection

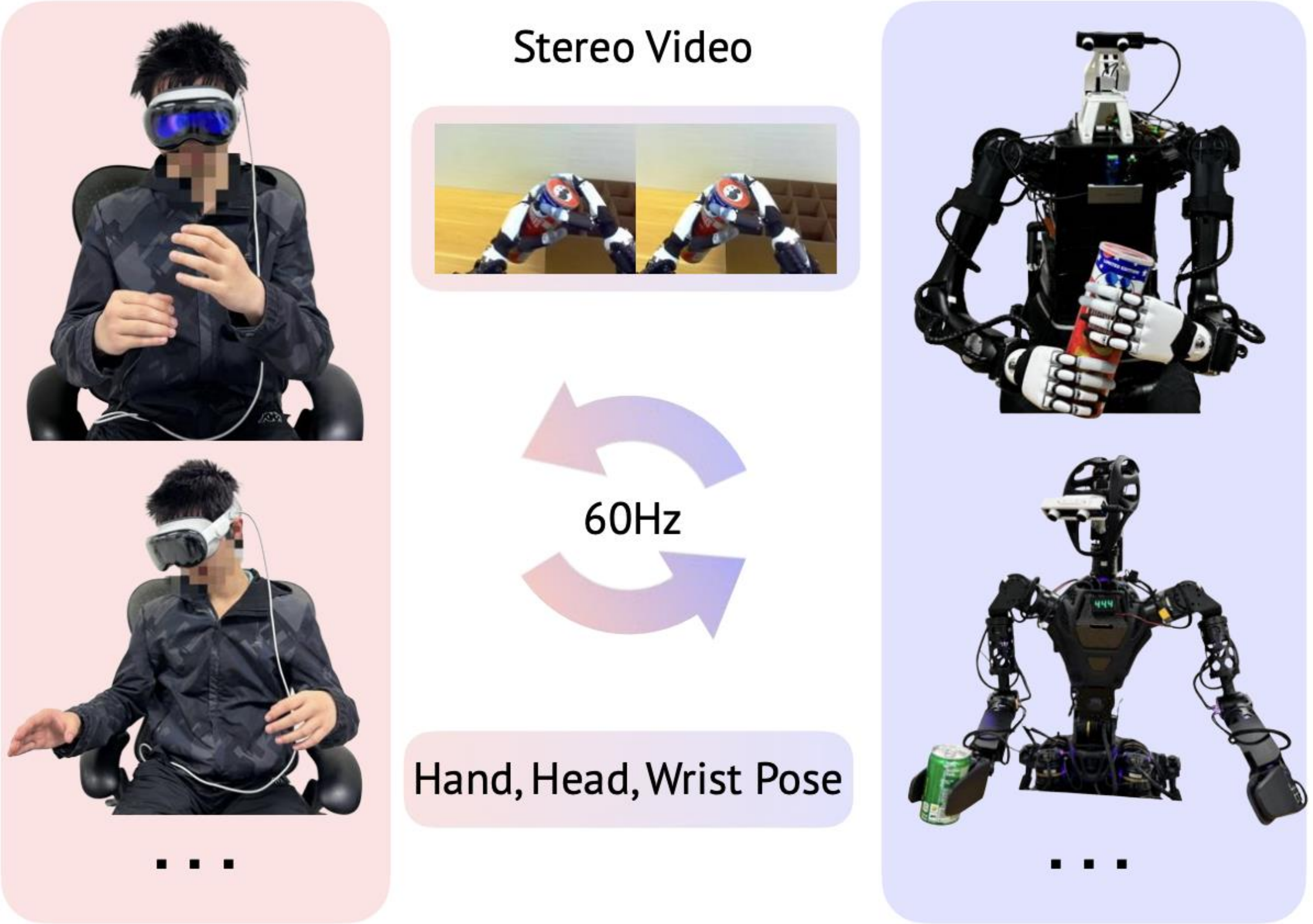
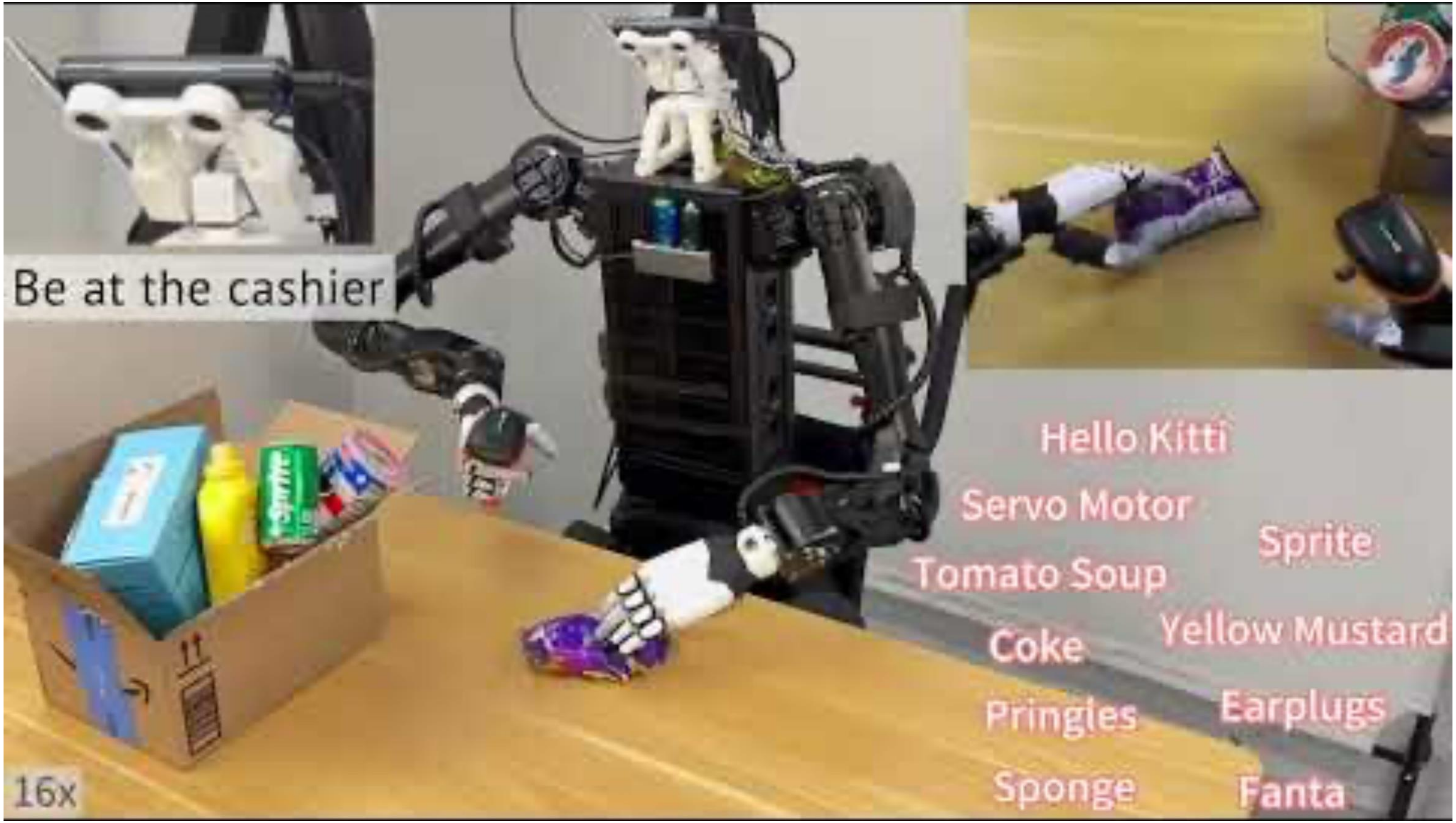


Image from:
Open-TeleVision: Teleoperation with Immersive Active Visual Feedback

Immersive Devices for Data Collection



Very slow

Open-TeleVision: Teleoperation with Immersive Active Visual Feedback

A Potential Failure Mode: Difference in Observation

- If the collected observations do not contain enough information for predicting the expert actions, there is nothing you can do.

- This is something challenging to get used to as an operator.

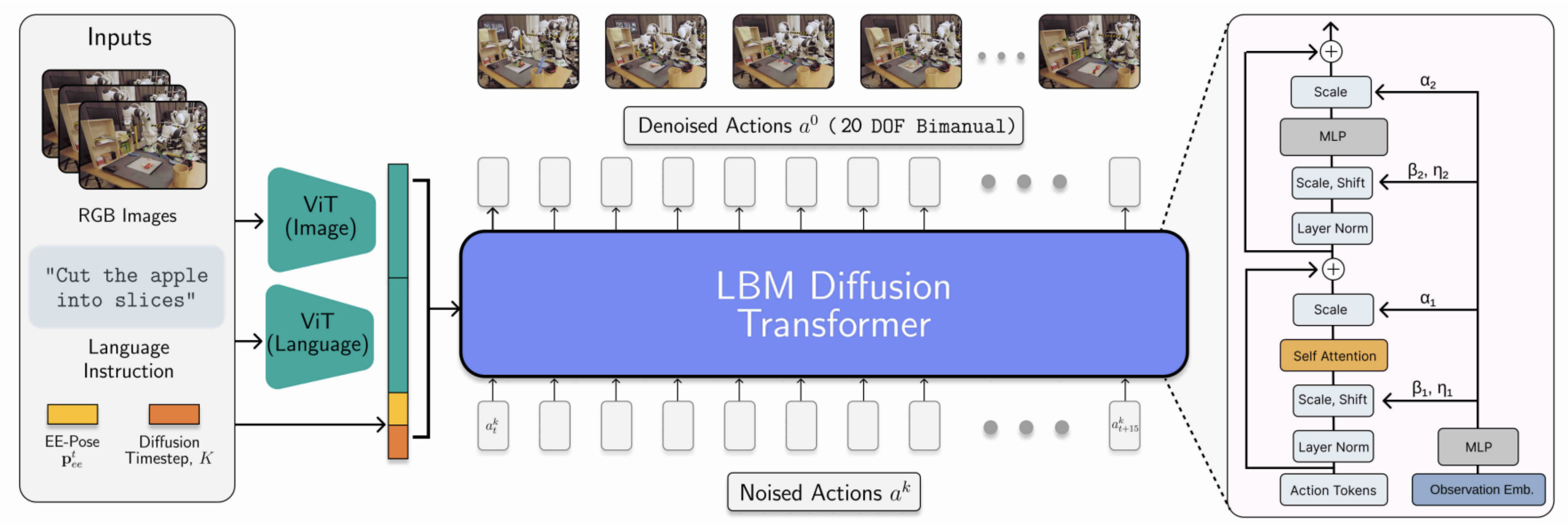
- Possible solutions:
  - Add many external cameras.
  - Immersive devices.

- Problem is still not solved if the operator relies on other sensory inputs, e.g., tactile.

A Potential Failure Mode: Non-Markovian Behavior

- Often, control policies are trained with very short observation histories (mainly to decrease the computational burden).

- Example: LBM from Toyota uses a history length of 2.



A Potential Failure Mode: Non-Markovian Behavior

- Often, control policies are trained with very short observation histories (mainly to decrease the computational burden).

- Modeling the controller as $\pi_\theta(a_t|o_t)$ implies that given the same observation, the action is the same, independently on what happened before.

- **It is hard for operators to behave in this way**. We can't help using our memory. Therefore, the operator's policy might not be markovian, i.e., $\pi_e(a_t|o_t, o_{t-1}, \dots)$.

- Using a larger history and asking operators to behave in a Markovian fashion helps, but it's not really a solid solution.

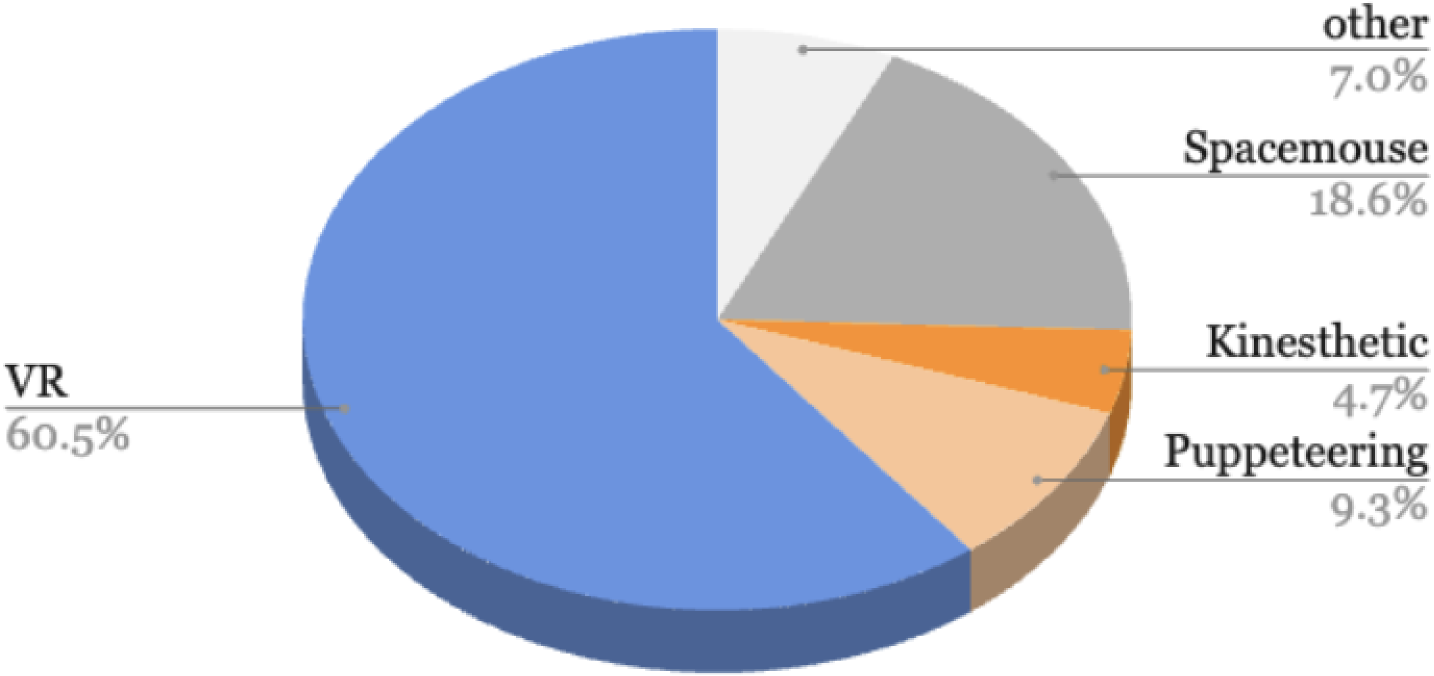
Summary

- Behavioral Cloning appears to be an elegant and scalable way of training robot policies.

- However, significant engineering is required to build effective data collection tools and data collection strategies.

- No perfect tool exists; they all come with advantages and limitations. Some tasks are impossible with any of the current tools (more on this later in the class).

Popularity of Data Collection Methods (in Academia)



Composition of human demonstration modalities present in the OpenXE dataset

Image from:
The Impact of Demonstration Modality on Imitation Learning

Behavioral Cloning: Agenda

- Theoretical Foundations
- Tools for Data Collection
- **Algorithms**
- **Leveraging foundation models**
- Challenges