UNIVERSITY OF TORONTO

Robotics Institute

Reinforcement learning for Robotics

Amey Pore

CSC415 Lecture 10

18th March 2026

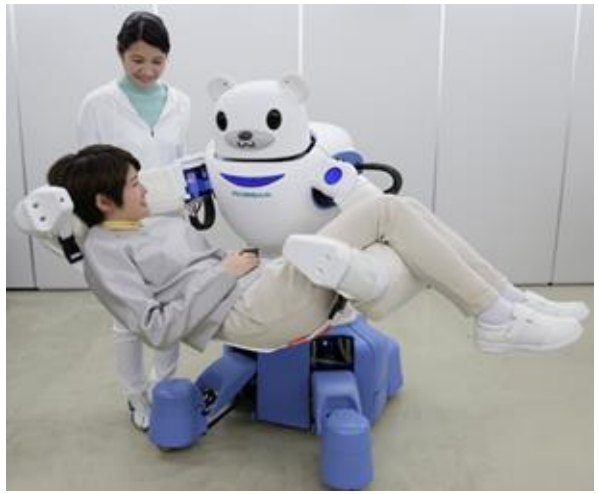
MEDCVR

MEDICAL COMPUTER VISION AND ROBOTICS

Why learning

Automation vs Autonomous

Real world robots to assist humans

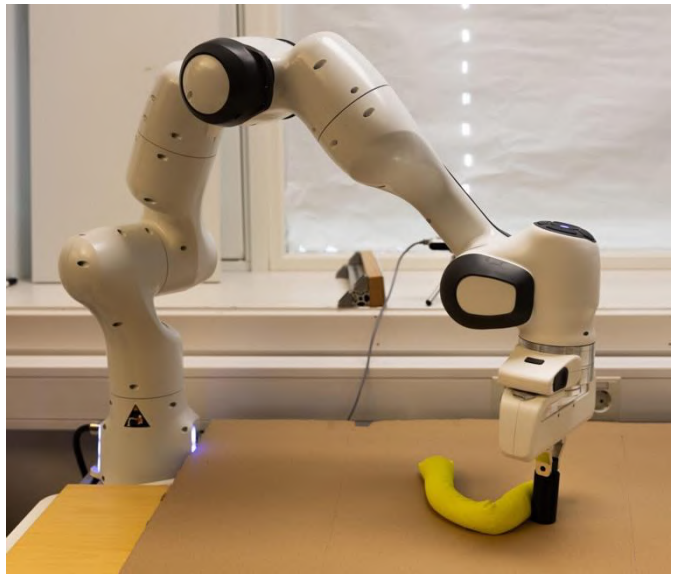


Deep RL success story ⟶ Real world environment

Goal: We want to learn manipulation and locomotion policies in the real world.

Rather than understanding the environment, simply collect a lot of experience and let the algorithm handle the rest

1. Deep RL is data hungry (millions/billions of iterations)
2. Robotic data is expensive: Robot cost, labelling
3. Safety



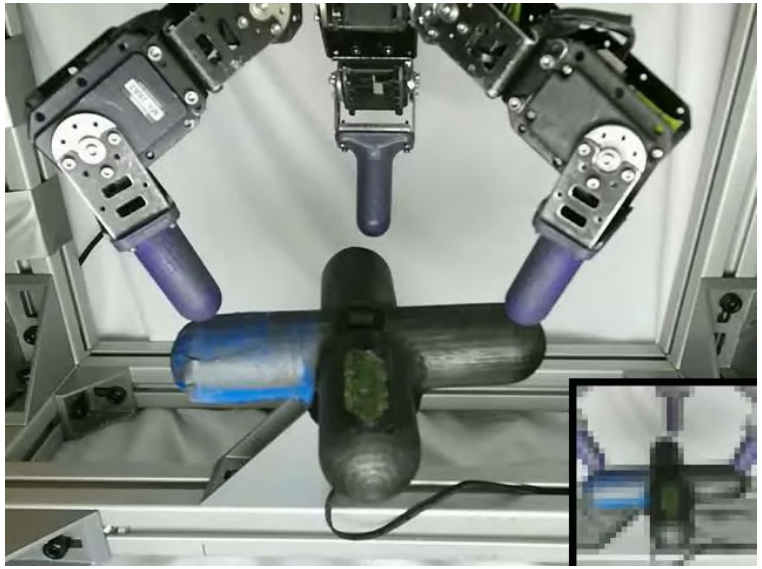
How can we get around the data availability problem?

Outline

1. **Train in real world directly**
2. Learn in Simulation and Sim2Real transfer
3. Use Human data: Imitation learning

The ingredients of sample-efficient real-world RL

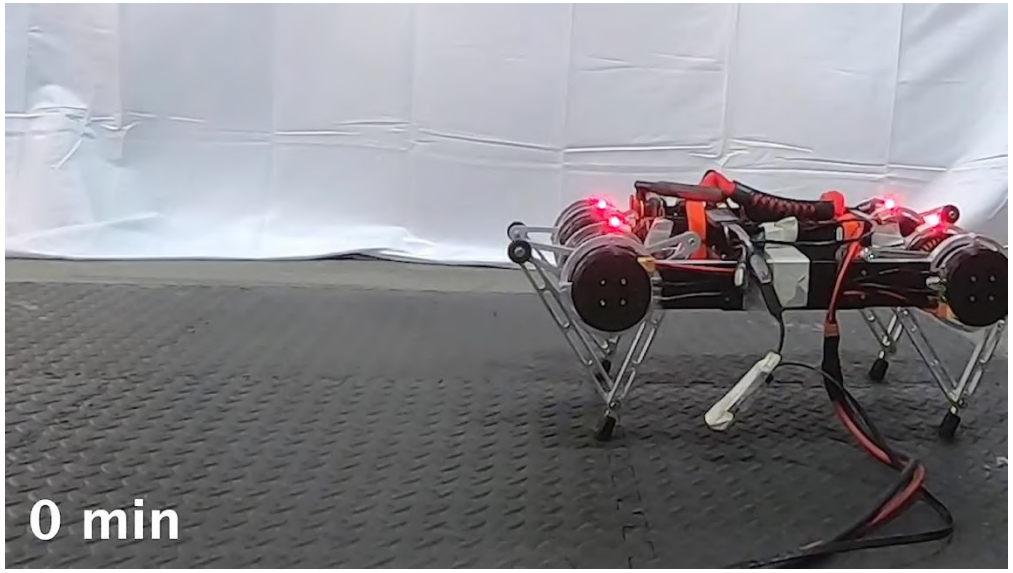
1. Sample efficient **off-policy algorithms** (eg: Soft actor critic).
2. Take as many gradient steps per timestep as possible
3. Need a way to reset
4. Avoid catastrophic failures wherever possible (eg: falling)



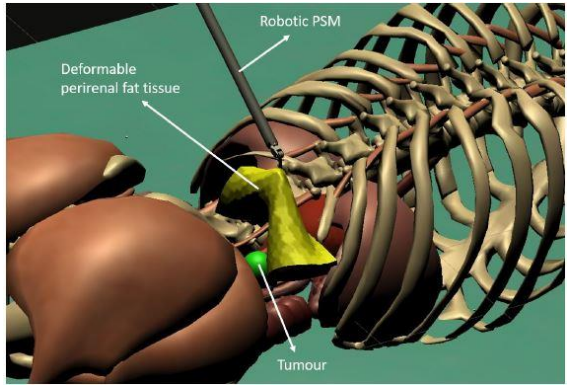
SAC (2018)

← 20 hrs of training time on real

2hrs of real world training →



Safety via constrained RL



[UnityFlexML: Simulation framework]

Objective:
Accomplish a RAMIS tissue retraction task without interacting with nearby tissue/organ

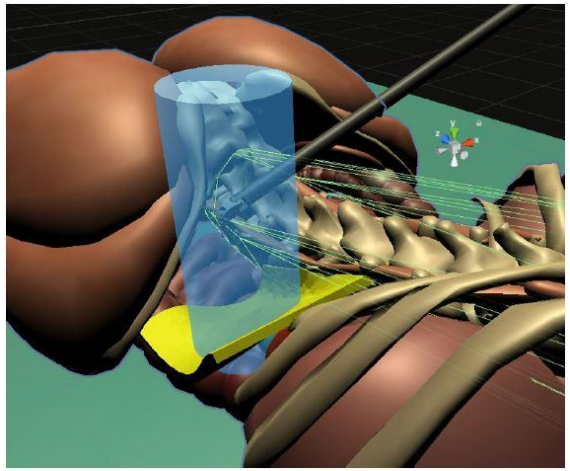
Lagrangian formulation for safety

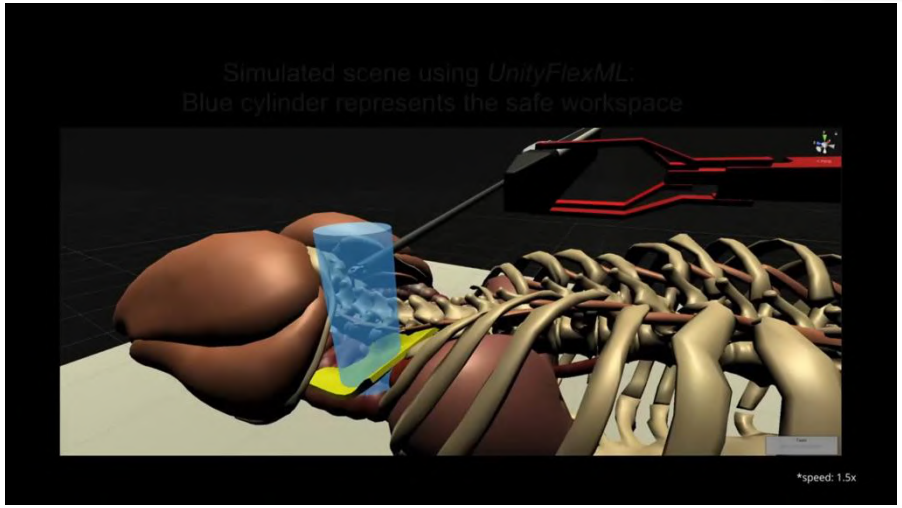
$$\max_{\pi_\theta \in \Pi} J_r(\pi_\theta) := \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$$

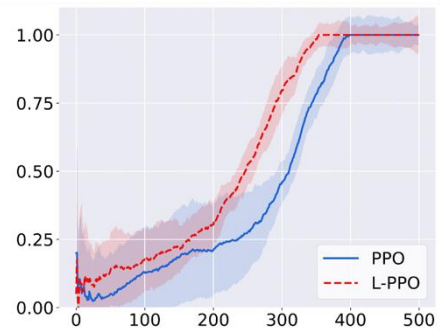
$$\max_{\pi_\theta} \quad J_r(\pi_\theta), \quad \text{s.t.} \quad J_C(\pi_\theta) \leq d$$

$$J(\theta) = \min_{\pi_\theta} \max_{\lambda \geq 0} \mathcal{L}(\pi_\theta, \lambda)$$

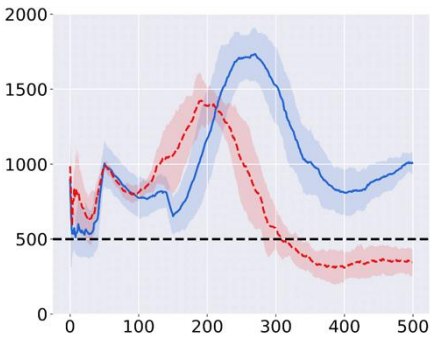
$$\mathcal{L}(\pi_\theta, \lambda) \;=\; J_r(\pi_\theta) - \lambda(J_C(\pi_\theta) - d)$$







Expected Reward



Cost

*[**Pore** et al, "Safe reinforcement learning using formal verification for tissue retraction in autonomous robotic-assisted surgery" , IROS 2021]*

Learning to Walk in the Real World with Minimal Human Effort

Sehoon Ha[12*], Peng Xu[2], Zhenyu Tan[2], Sergey Levine[23], and Jie Tan[2]

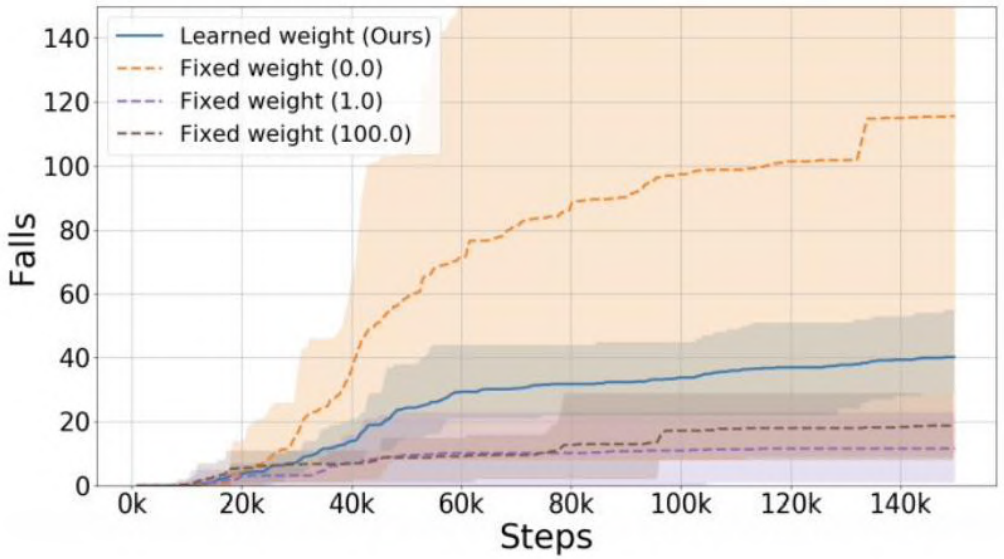
[1]Georgia Institute of Technology

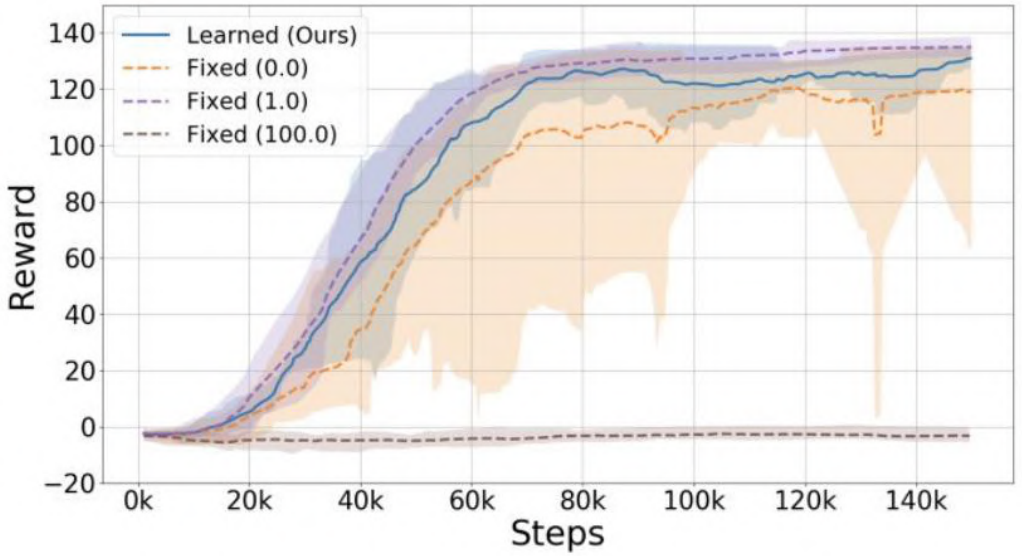
[2]Robotics at Google

[3]University of California, Berkeley

[*]The research was conducted when the author was at Google

Safety-Constrained SAC: Evaluation





How fast can we do real world RL?

[**Smith** et al, “Demonstrating a Walk in the Park: Learning to Walk in 20 Minutes With Model-Free Reinforcement Learning” , RSS 2023]

[**Smith** et al, “Grow Your Limits: Continuous Improvement with Real-World RL for Robotic Locomotion” , ICRA 2024]

Outline

1. Train in real world directly
2. **Learn in Simulation and Sim2Real transfer**
3. Use Human data: Imitation learning

Is simulated data the answer?

Real world

- Slow
- Unsafe
- Expensive
- Human supervision

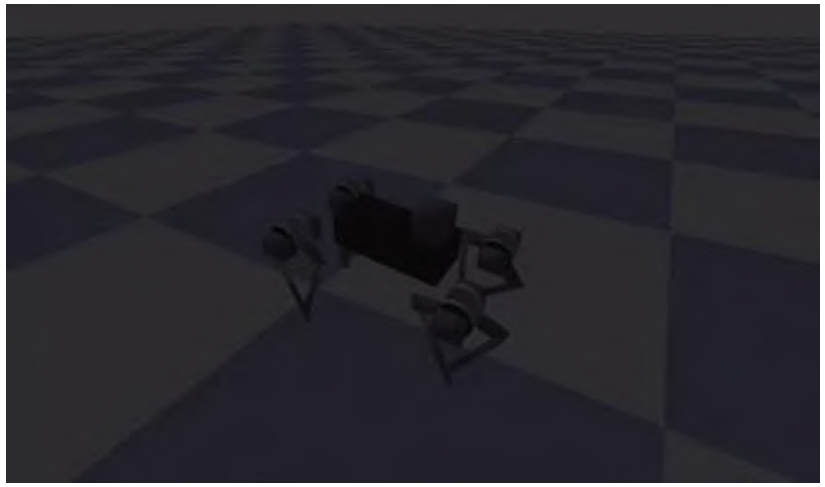
Simulation

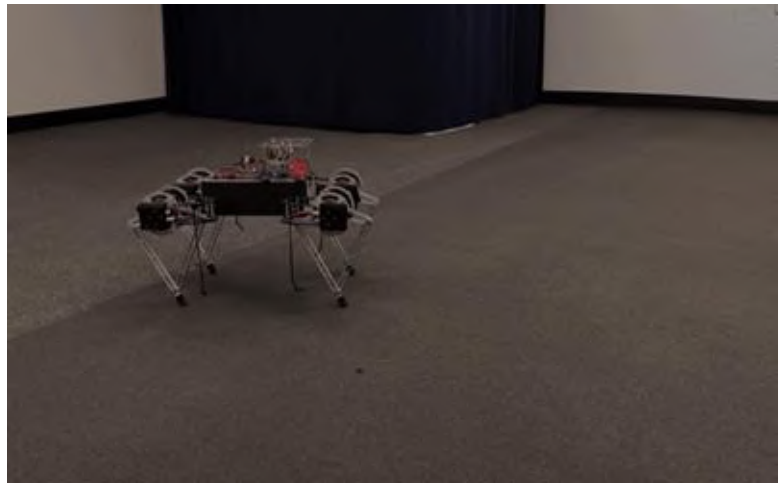
- Fast
- Safe
- Cheap
- Scalable

What's the sim-to-real gap?

Control policies developed in simulation usually do not work on robots

Dynamics

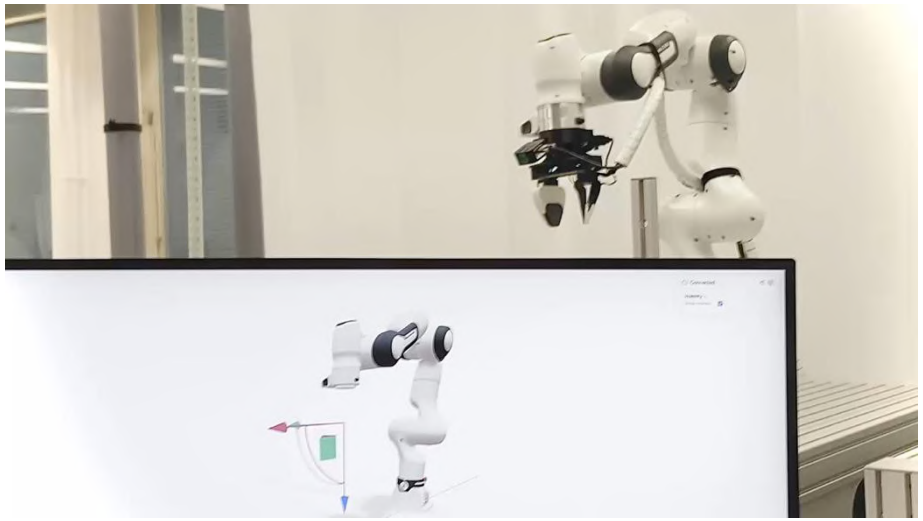




Perception

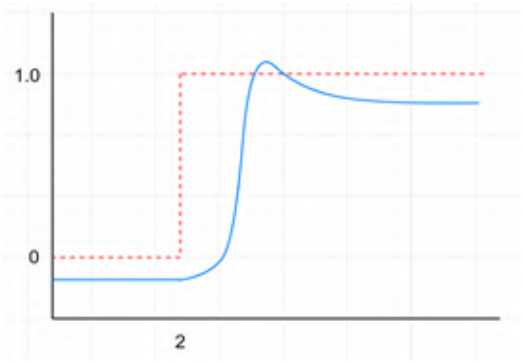
What are the causes of sim-to-real gap?

1. Unmodeled dynamics
2. Wrong simulation parameters
3. Inaccurate contact models
4. Latency
5. Actuator dynamics
6. Noise
7. Stochastic real environment



Compliant Robot control

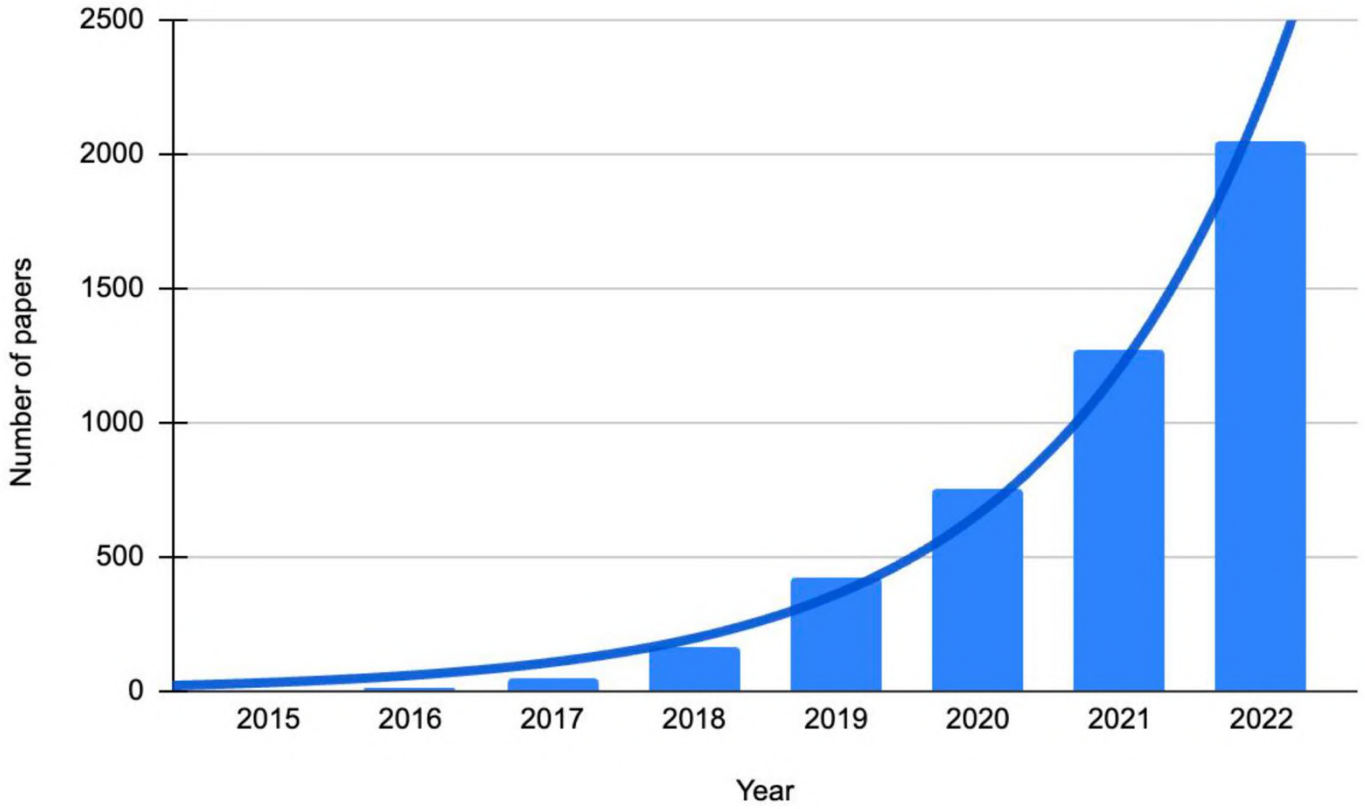
https://utiasdsl.github.io/crisp_controllers/examples_robot/



Red: Command position (p)
Blue: Actual robot position

[Figure credits: Radian Gondokaryono]

Trend on Sim2Real



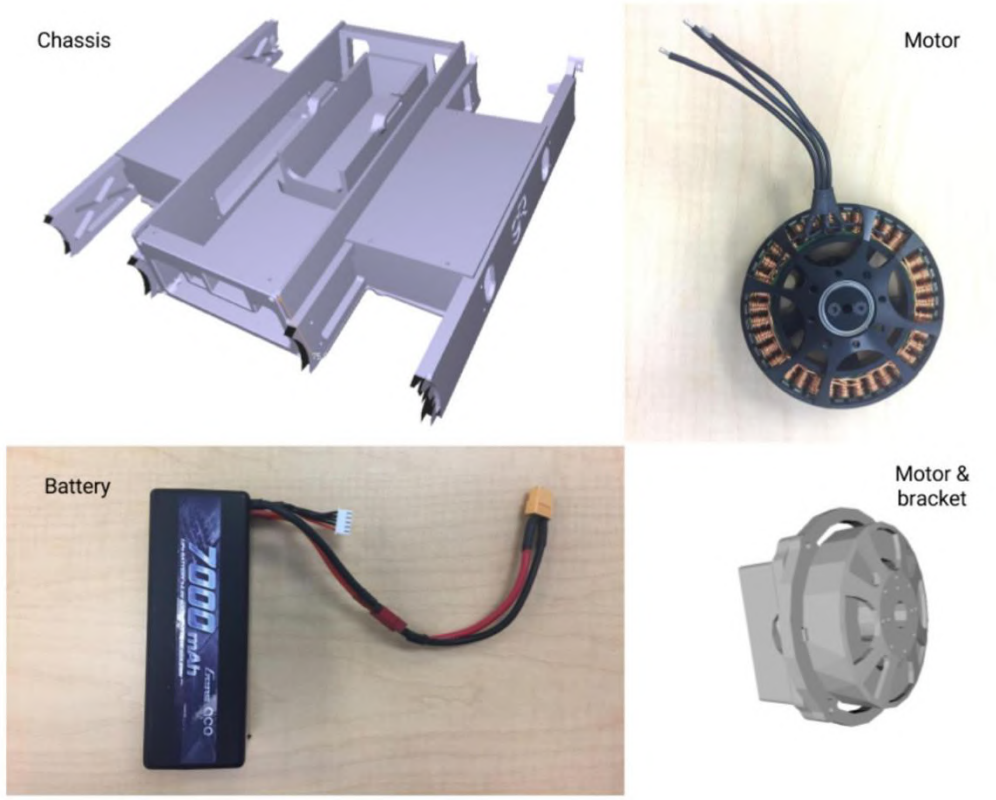
[Slide credits: Jie Tan]

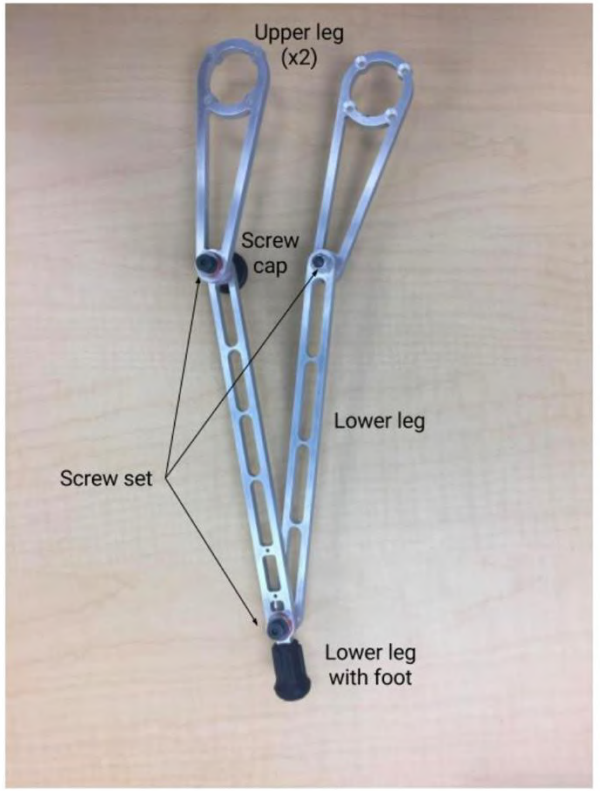
How to overcome Sim2Real Gap?

1. **Improve Simulation**
2. Improve Policy

- Domain Randomization
- Domain Adaptation

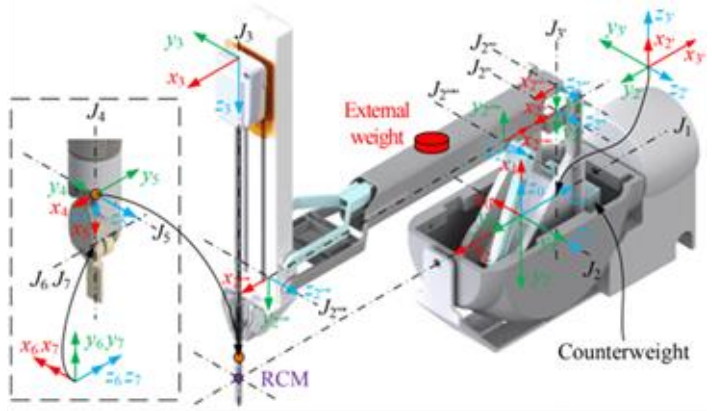
Simulations: System Identification



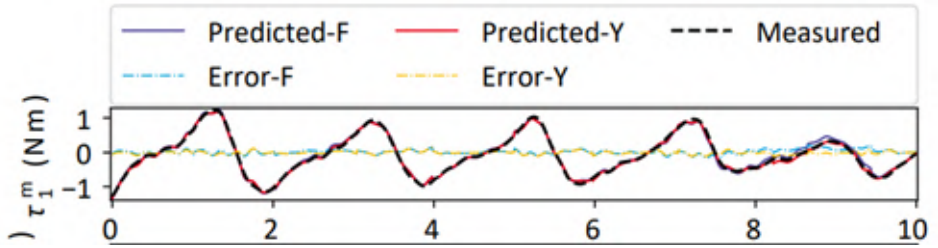


Simulations: System Identification

1. How to measure Mass?
2. How to measure Center of Mass?
3. How to measure Motor Damping (Viscous friction)
   - Spin the motor to a specific speed
   - Remove power
   - Record the data: motor speed vs. time
   - Fit the data based on physical equation about motor damping: $\tau d = k\omega$
   - Find out motor damping coefficient k



Wang, Yan, Gondokaryono, Radian, et al. "A Convex Optimization-Based Dynamic Model Identification Package ..." *RAL* 2019.



[Figure credits: Radian Gondokaryono]

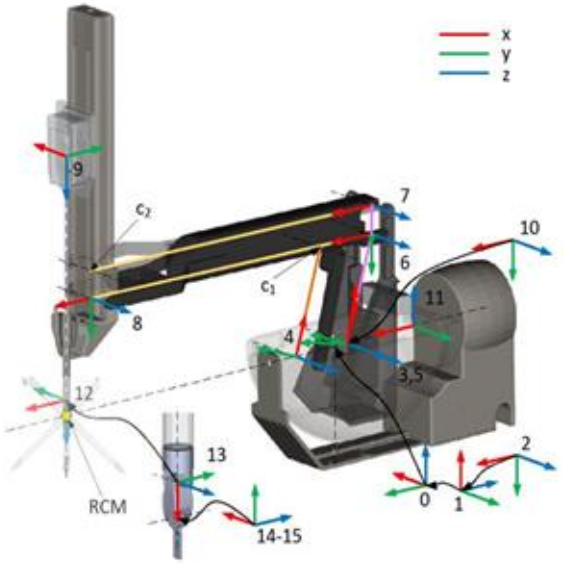
For every link, identify Mass & Inertia:

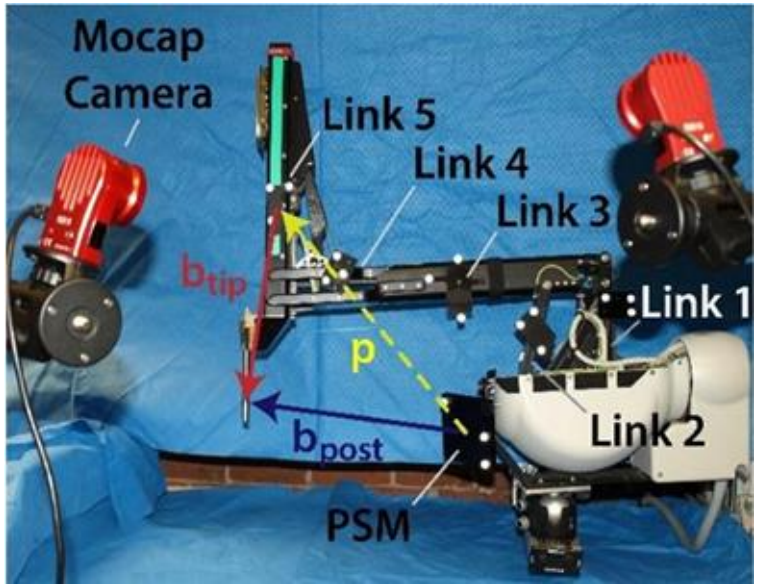
$$\boldsymbol{\delta}_{Lk} = [L_{kxx} \quad L_{kxy} \quad L_{kxz} \quad L_{kyy} \quad L_{kyz} \quad L_{kzz} \quad \boldsymbol{l}_k^\top \quad m_k]^\top \tag{7}$$

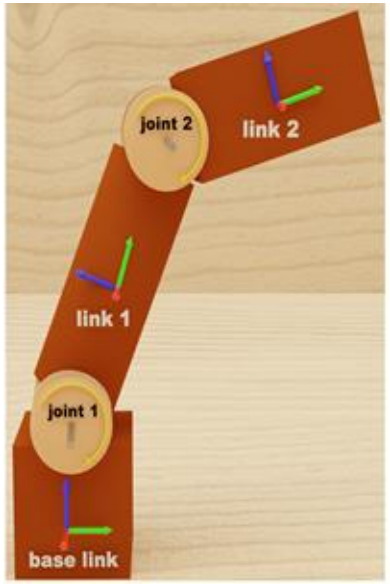
Frictions:

$$\boldsymbol{\delta}_{Ak} = [F_{vk} \quad F_{ck} \quad F_{ok} \quad I_{mk} \quad K_{sk}]^\top \tag{8}$$

Simulations: Robot Model in simulation



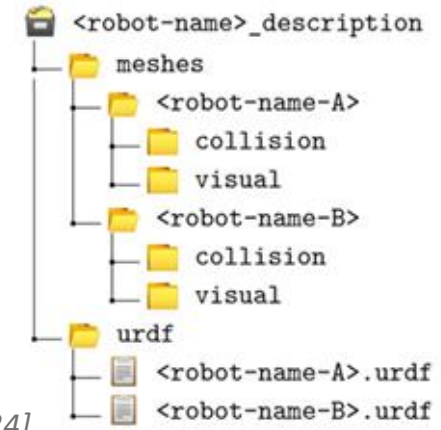





```
<?xml version="1.0" encoding="utf-8"?>
<robot name="2 DOF planar robot">
 <link name="base link">
  <visual>
   <origin xyz="0 0 0.25"/>
   <geometry>
    <box size="0.5 0.5 0.5"/>
   </geometry>
  </visual>
 </link>
...
 <joint name="joint 1" type="continuous">
  <parent link="base link" />
  <child link="link 1" />
  <axis xyz="0 1 0" />
  <origin xyz="0 0 0.5"/>
 </joint>
...
</robot>
```

Create Mesh Models: .stls

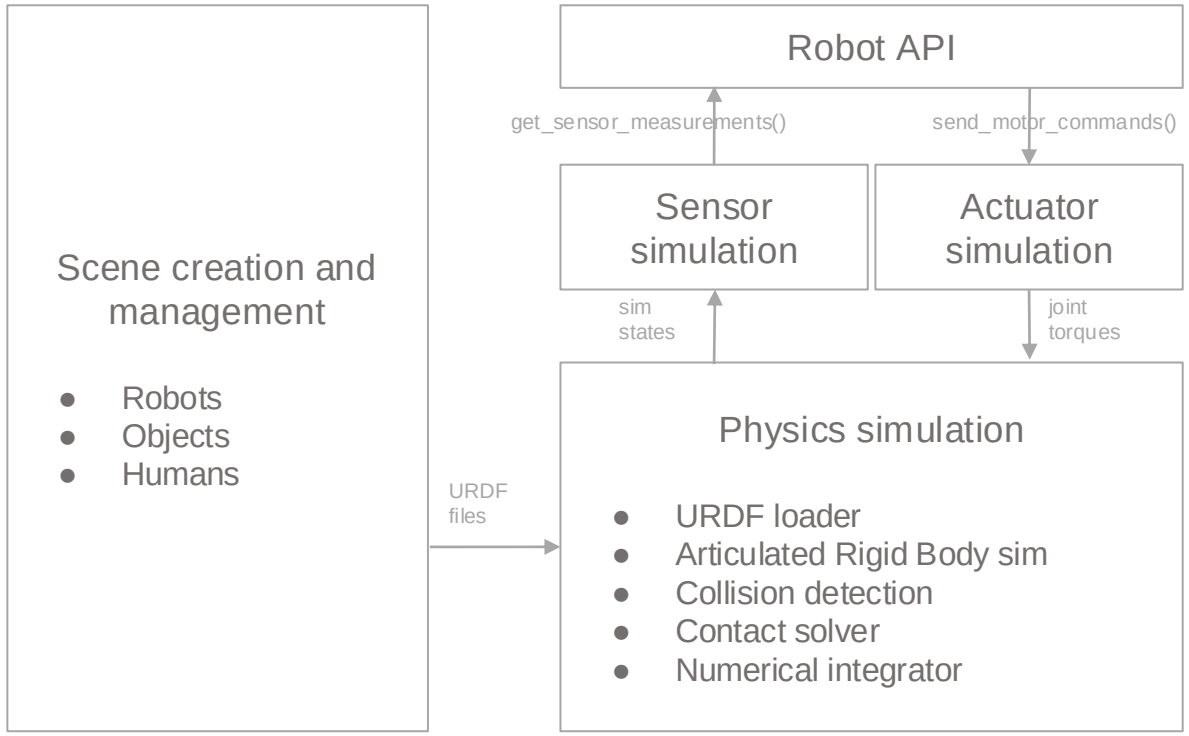
Obtain Link Lengths:
Frame locations

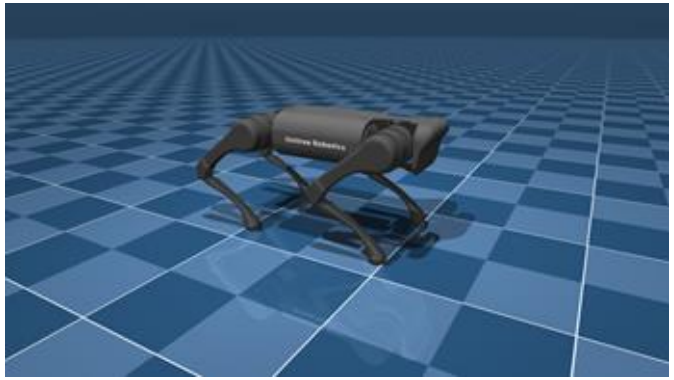


[Slide credits: Radian Gondokaryono]

[Tola et al., "Understanding urdf: A dataset and analysis", R-AL 2024]

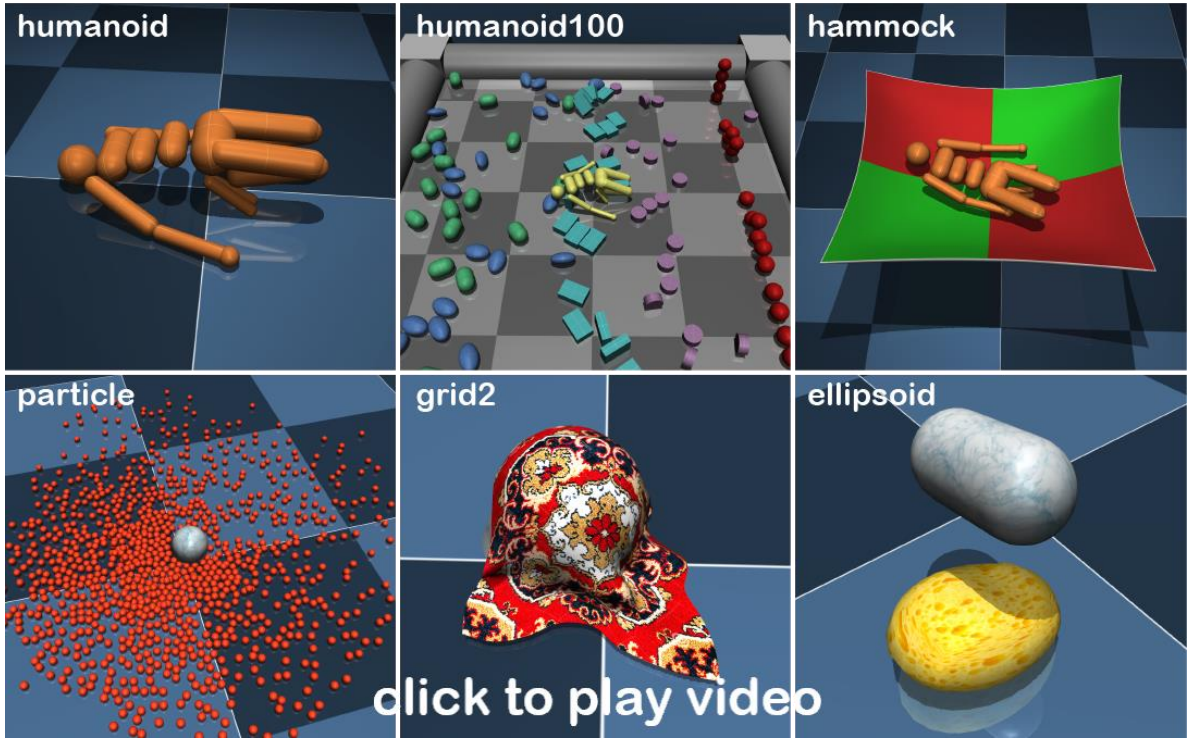
Simulations: URDF



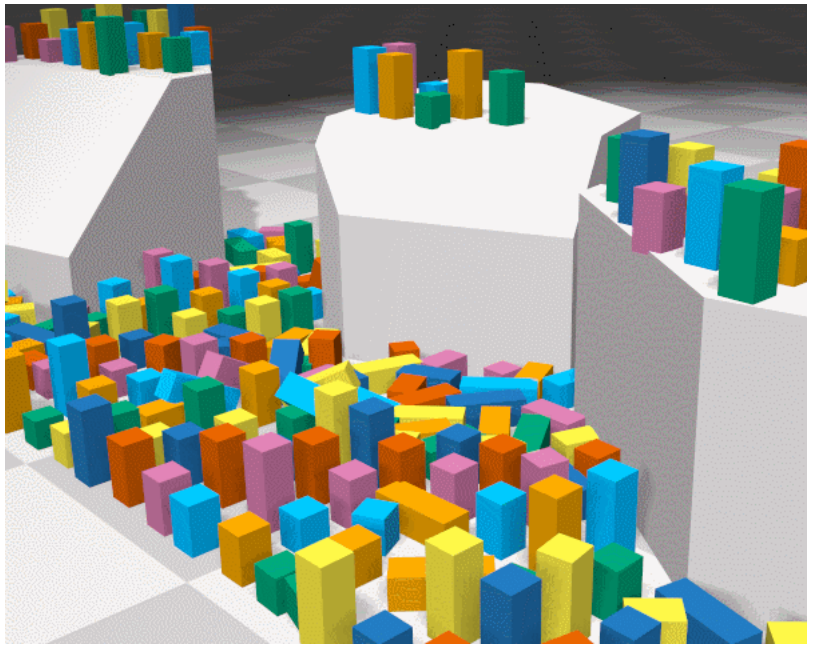


[Slide credits: Jie Tan]

Simulations: Mujoco (Popular)



[https://zalo.github.io/mujoco_wasm/]



FLEX: Real-time simulator on a GPU for both rigid and soft bodies, fluids and gas.

[https://www.youtube.com/watch?v=1o0Nuq71gI4]

Building a good simulation for sim-to-real

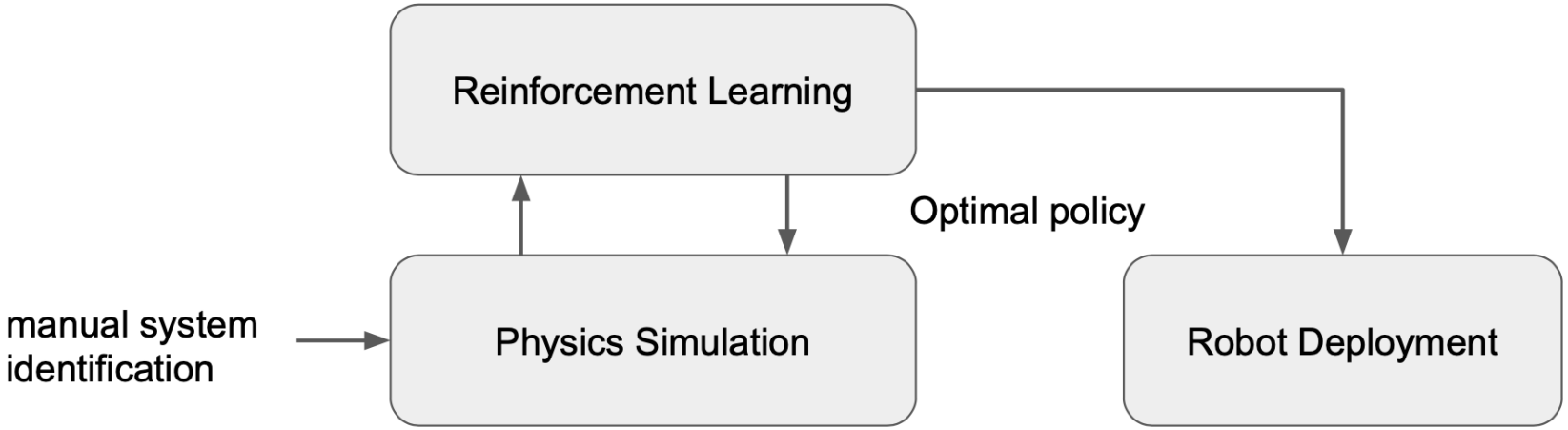
- Physics parameters vs visual rendering

Engine	Vectorized / massively scalable	Differentiable physics	Realistic dynamics	High fidelity visuals	Open source
pyBullet			■		■
MuJoCo			■		■
IsaacSim	■		■	■	■
Brax	■	■			■
Unity	■			■	

Chaos?

(Unreal)

Next steps after simulation

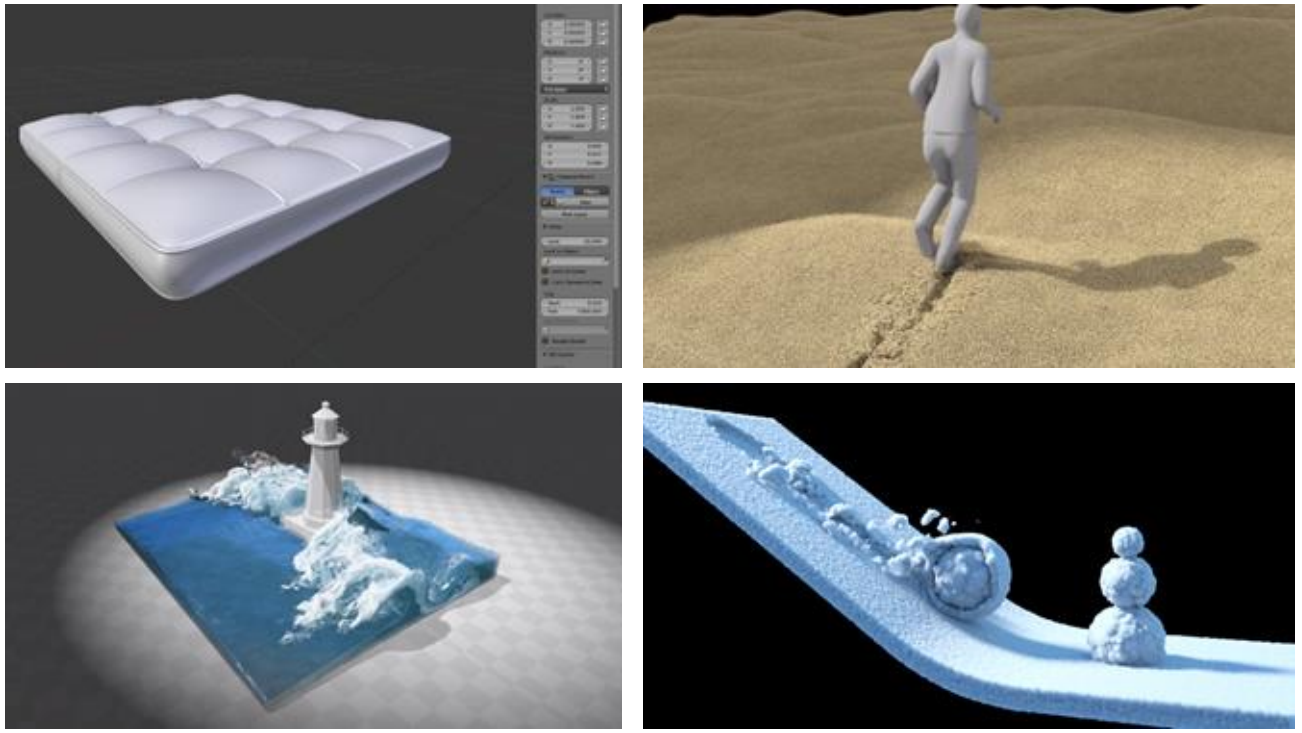


- Limitations
  - Disassemble the robot
  - Decide what parameters to identify
  - Design experiments for individual parameters
  - Lots of manual work

Several works try to do automatic system identification using trajectories. Out of scope!

What simulation challenges are remaining?

- Complex dynamics



What simulation challenges are remaining?

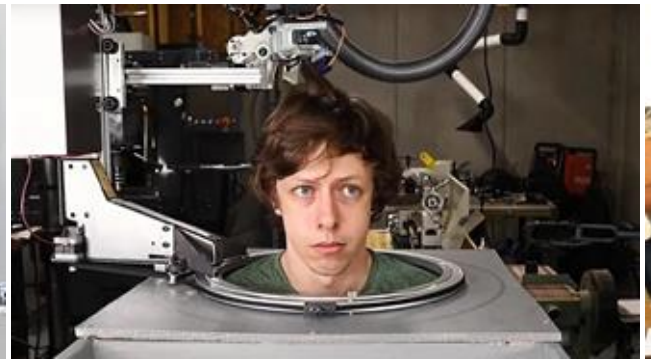
- Complex dynamics
- Realistic rendering

What simulation challenges are remaining?

- Complex dynamics
- Realistic rendering
- Scalable scene creation

What simulation challenges are remaining?

- Complex dynamics
- Realistic rendering
- Scalable scene creation
- Modeling humans


Remember: These are not RL problems!

How to overcome Sim2Real Gap?

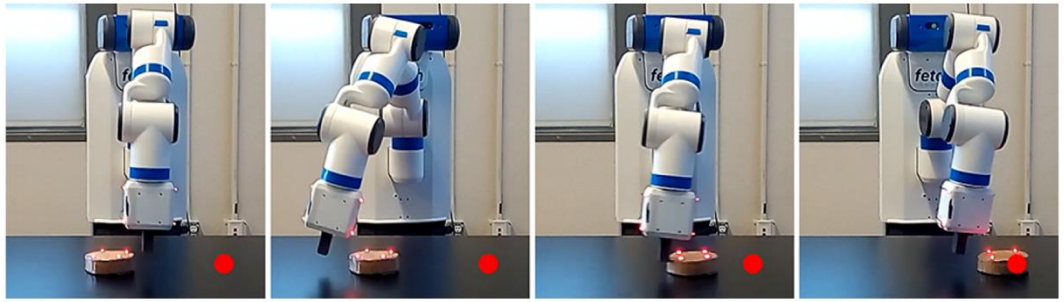
1. Improve Simulation
2. **Improve Policy**

- **Domain Randomization**
- Domain Adaptation

Domain Randomization: Dynamics

- Original objective: reward maximization

$$\mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} r(s_t, a_t) \right]$$



- New Objective with domain randomization

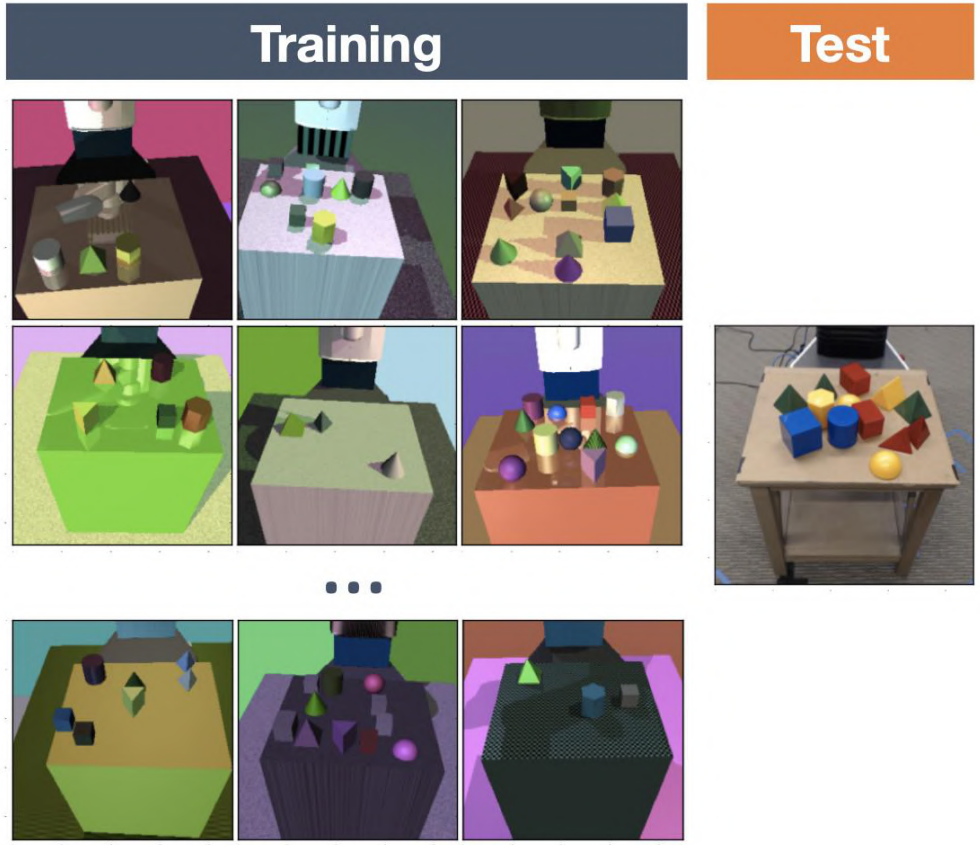
$$\mathop{\mathbb{E}}_{\mu \sim \rho_\mu} \left[\mathbb{E}_{\tau \sim p(\tau|\pi,\mu)} \left[\sum_{t=0}^{T-1} r(s_t, a_t) \right] \right]$$

Physical parameters

Parameter	Range
Link Mass	$[0.25, 4]\times$ default mass of each link
Joint Damping	$[0.2, 20]\times$ default damping of each joint
Puck Mass	$[0.1, 0.4]kg$
Puck Friction	$[0.1, 5]$
Puck Damping	$[0.01, 0.2]Ns/m$
Table Height	$[0.73, 0.77]m$
Controller Gains	$[0.5, 2]\times$ default gains
Action Timestep λ	$[125, 1000]s^{-1}$

[Peng et al., "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization", ICRA 2018]

Domain Randomization: Vision

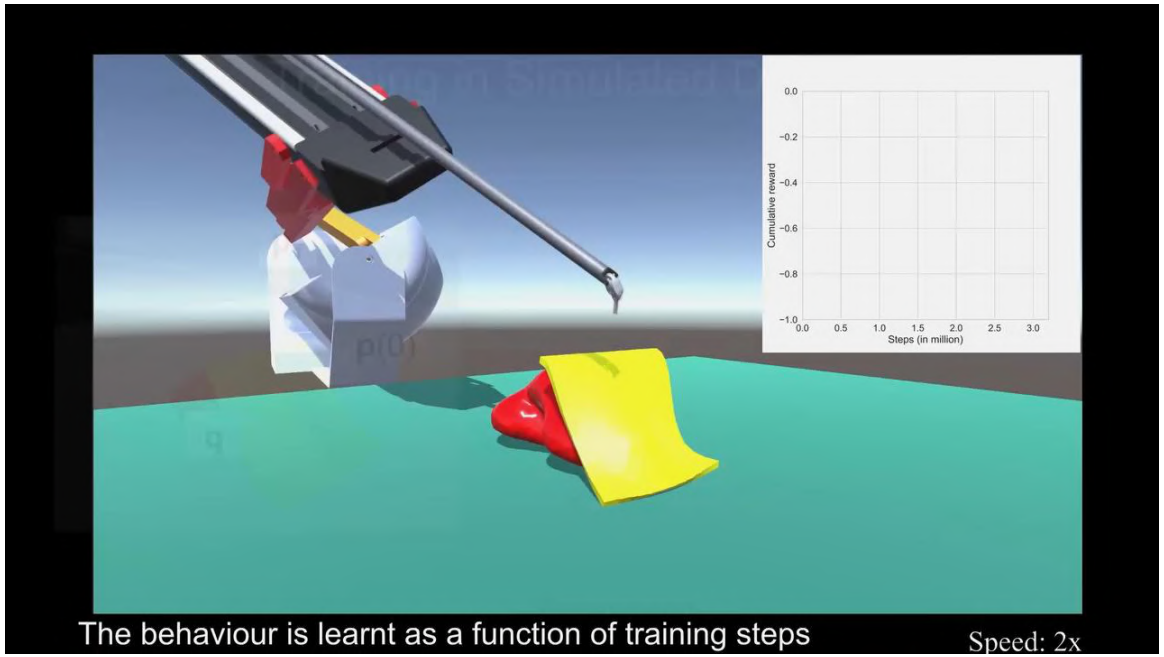


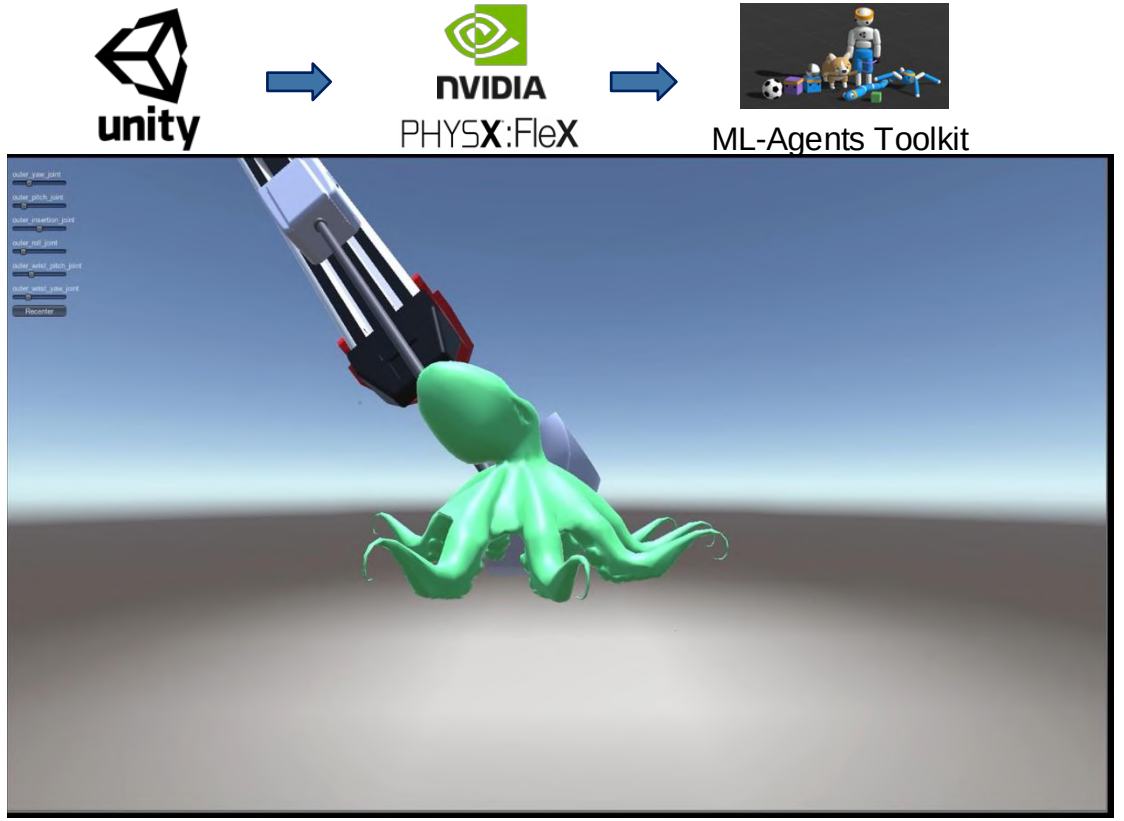
We create (automatically) tons of simulation environments by randomizing textures and camera viewpoints. We use the simulation data to train object detectors

[Tobin et al., "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World", IROS 2017]

UnityFlexML: Deformable Object Manipulation using RL

- Simulating deformable behavior

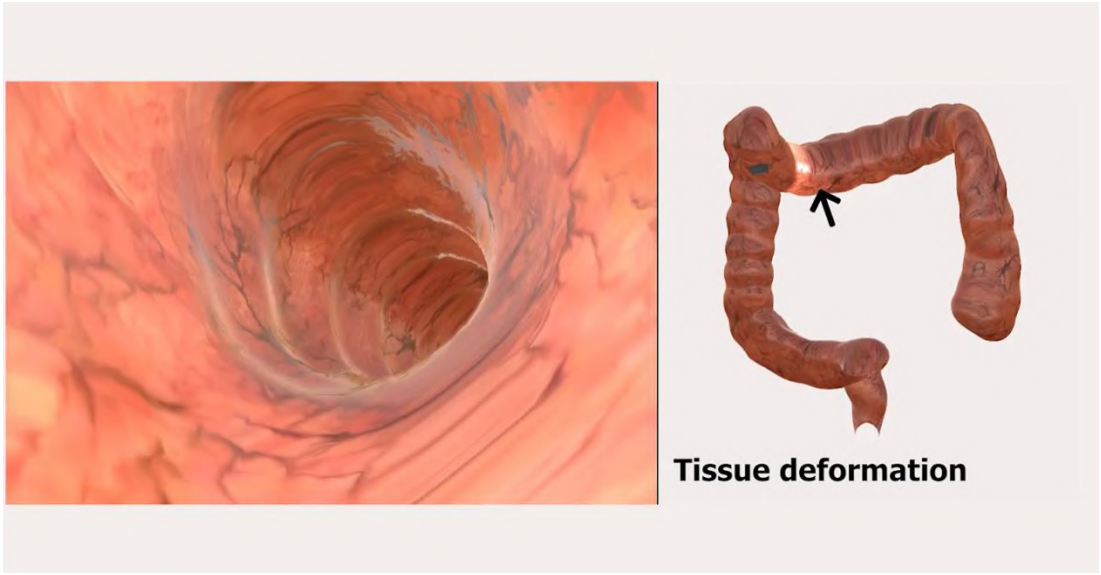


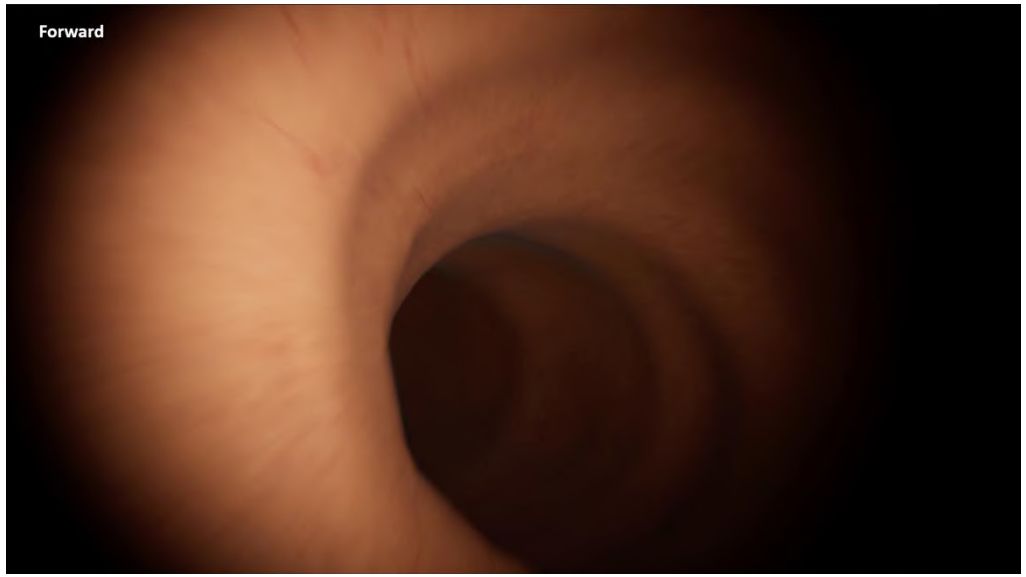


[Pore et al., "Soft Tissue Simulation Environment to Learn Manipulation Tasks in Autonomous Robotic Surgery", IROS 2020]

Colonoscopy Simulator

- Unity-engine based Simulator
- Realistic texture adopted from KVASIR dataset rendered in High-definition
- Biomechanical deformations using SOFA

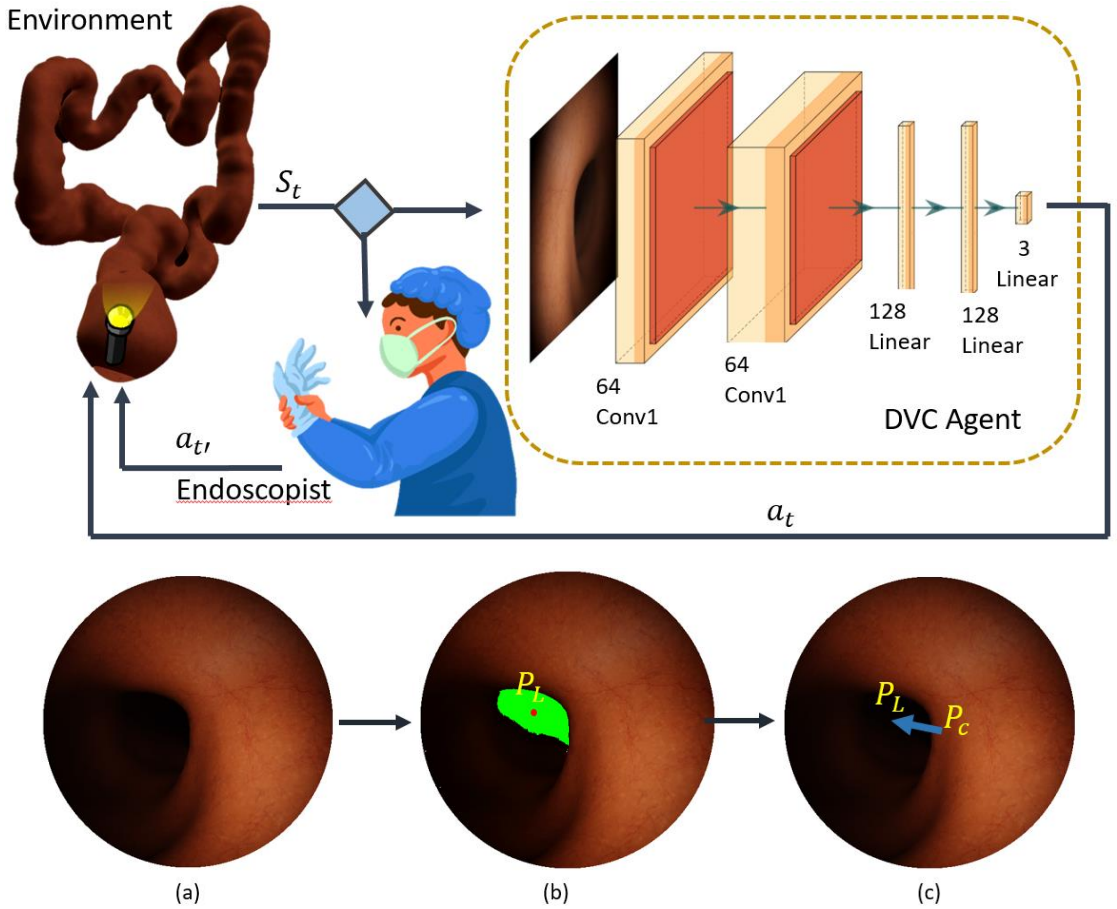




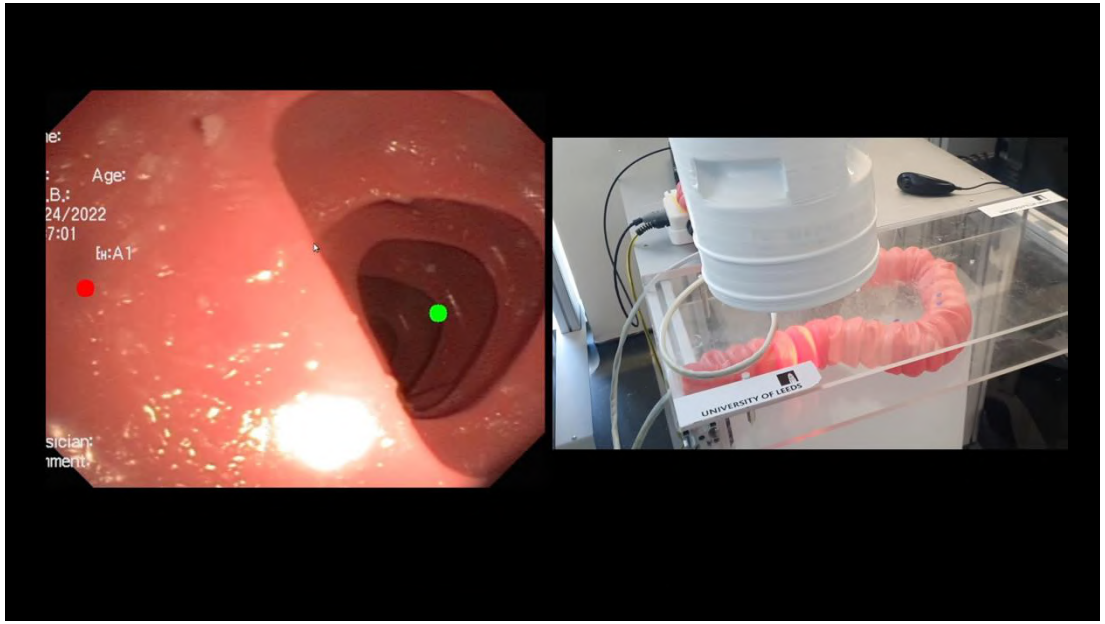
Modular: Textures, lighting conditions, polyps

[Pore et al., "Soft tissue simulation for AI driven colonoscopy navigation", Sofa Symposium 2022]

Deep Visuomotor Control (DVC)

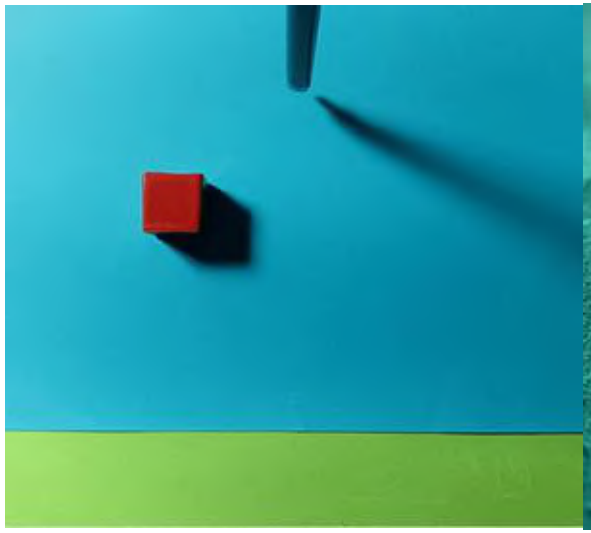


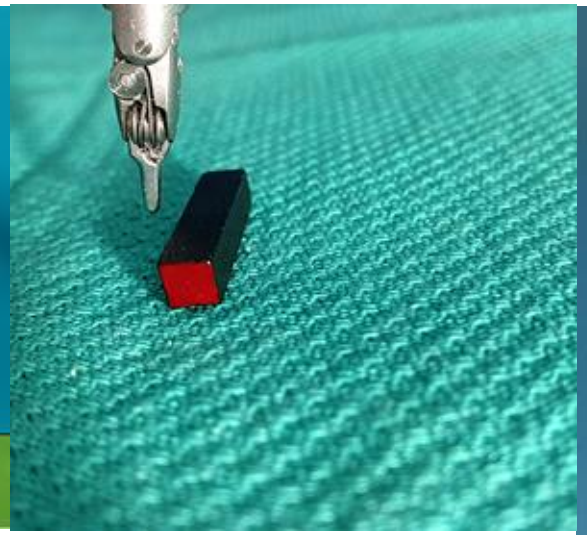
Deep Visuomotor Control (DVC) learns an end-to-end navigation policy to map the endoscopic images to the control signal of the endoscope.

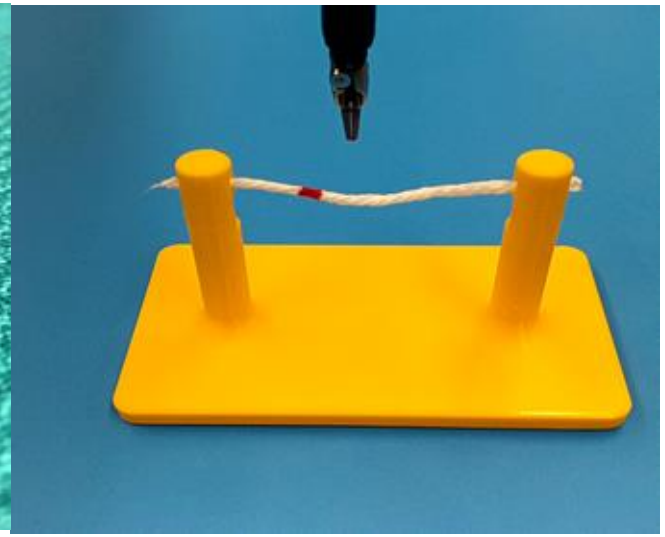


[Pore, et al. "Colonoscopy Navigation using End-to-End Deep Visuomotor Control: A User Study." IROS2022]

Scaling to other tasks





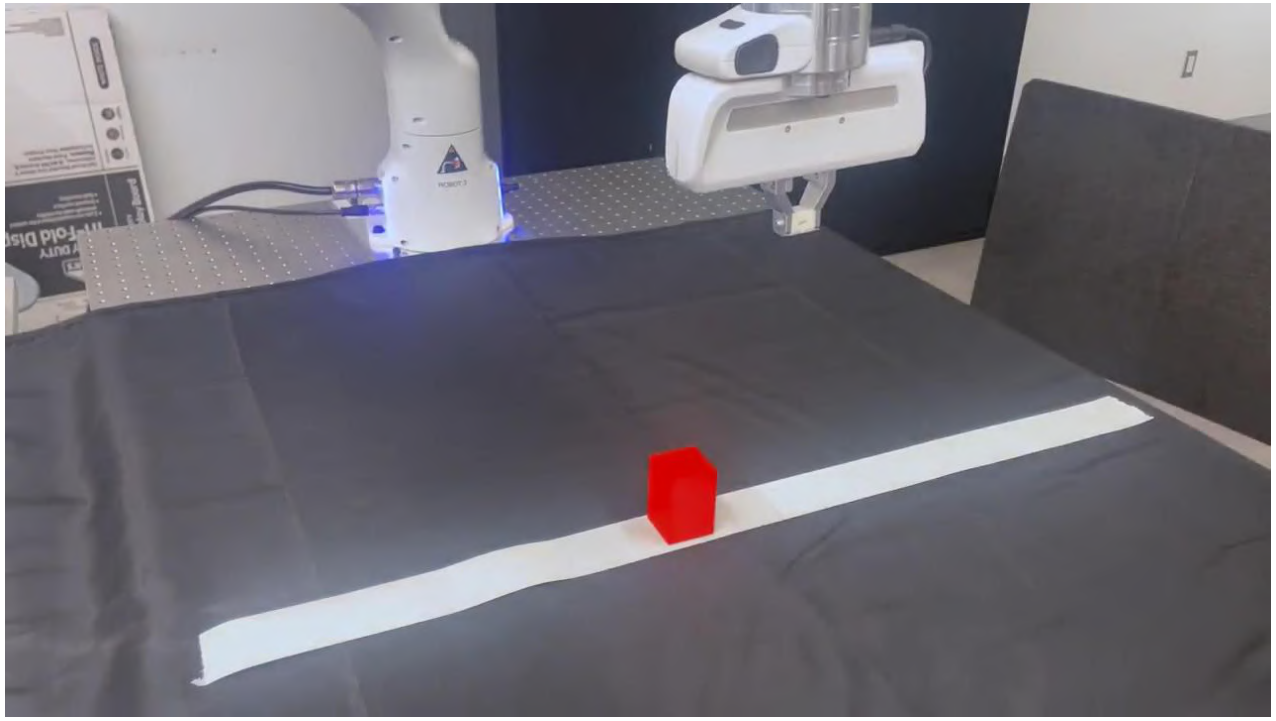


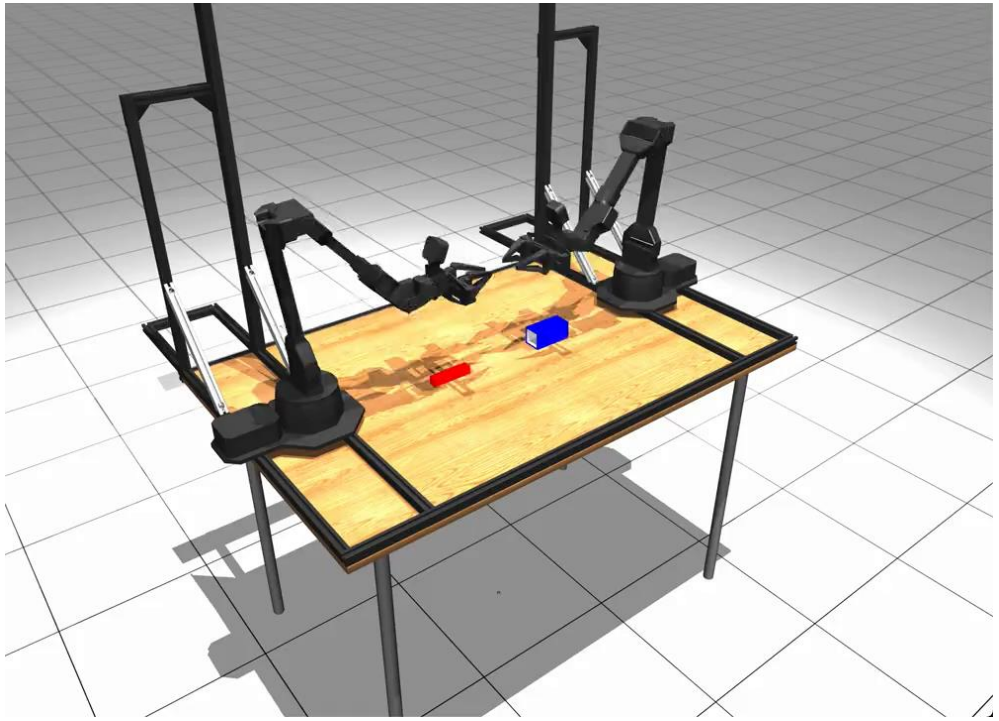
[Haiderbhai, et al. "Sim2Real Rope Cutting With a Surgical Robot Using Vision-Based Reinforcement Learning." TASE 2024

Gondokaryono, et al. "Learning Nonprehensile Dynamic Manipulation: Sim2real Vision-based Policy with a Surgical Robot." RA-L 2023

Haiderbhai, et al. "Robust sim2real transfer with the da vinci research kit: A study on camera, lighting, and physics domain randomization." IROS 2022.]

Accelerated Mujoco Simulation

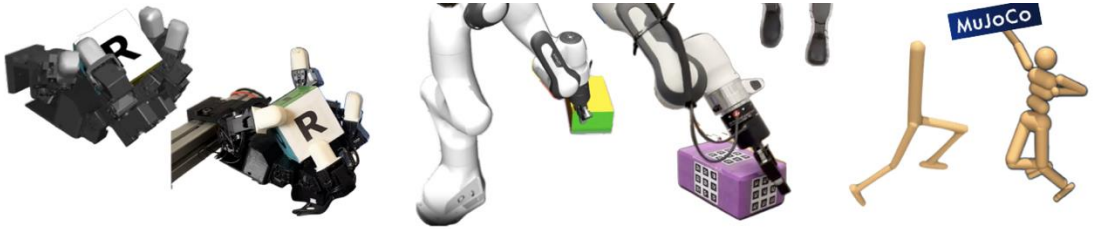




GPU based rendering and dynamics

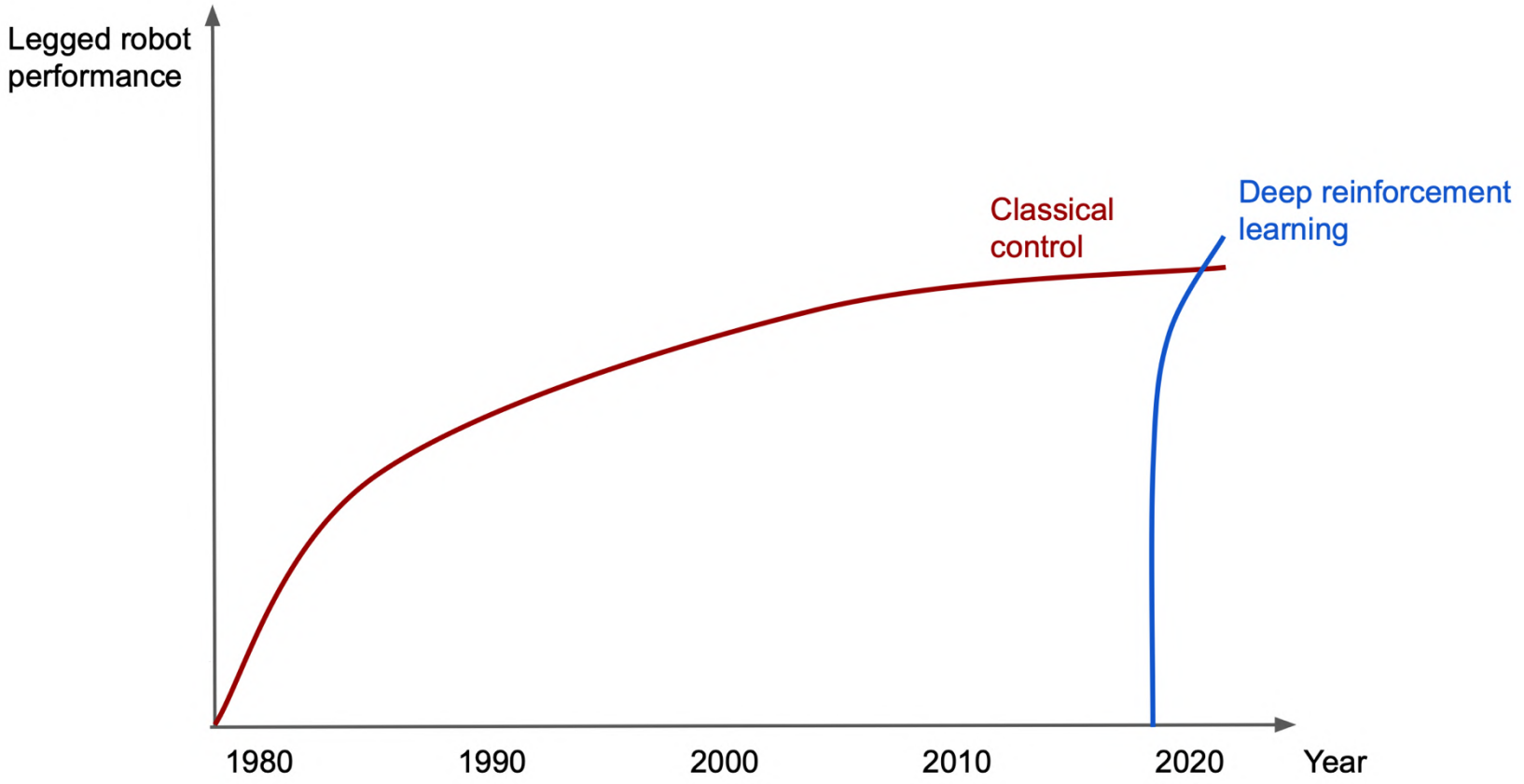
- Prior simulator train time ~6-10hrs
- Accelerated pipeline ~ 10mins

[Haiderbhai, et al. "Mujoco playground." RSS 2025]



Learning robust perceptive locomotion for quadrupedal robots in the wild, Science Robotics, 2022

Ma et al. “**Learning coordinated badminton skills for legged manipulators**” Science Robotics


Legged robot performance
Classical control
Deep reinforcement learning
1980
1990
2000
2010
2020
Year

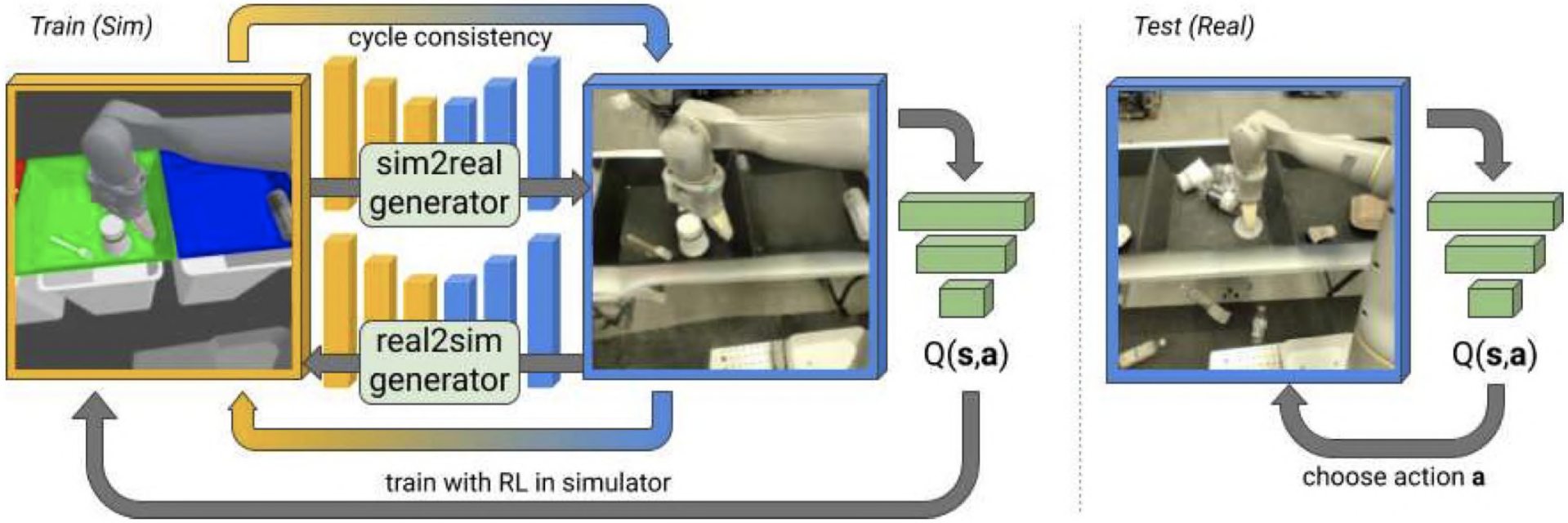
How to overcome Sim2Real Gap?

1. Improve Simulation
2. **Improve Policy**

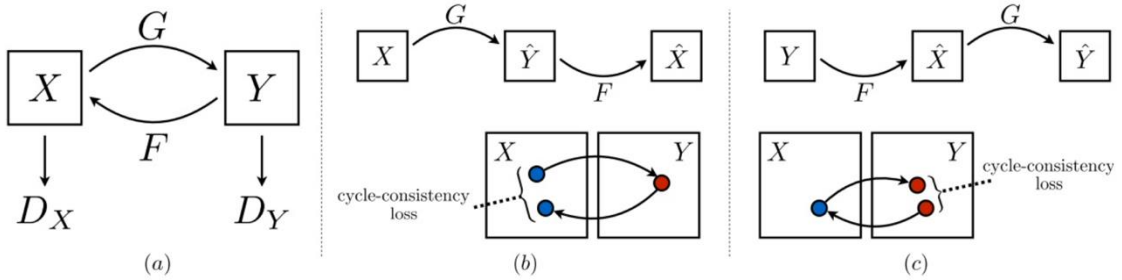
- Domain Randomization
- **Domain Adaptation**

Domain adaptation: Cycle-Gans

Learn to adapt the textures of the simulator to match the real domain.

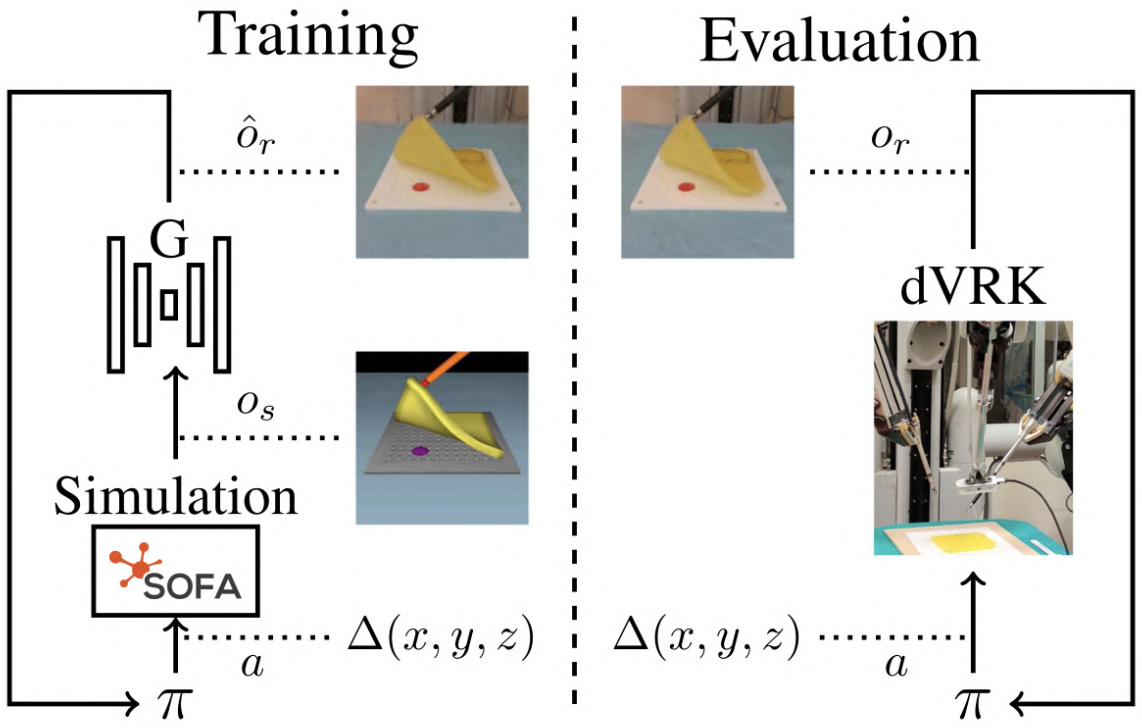


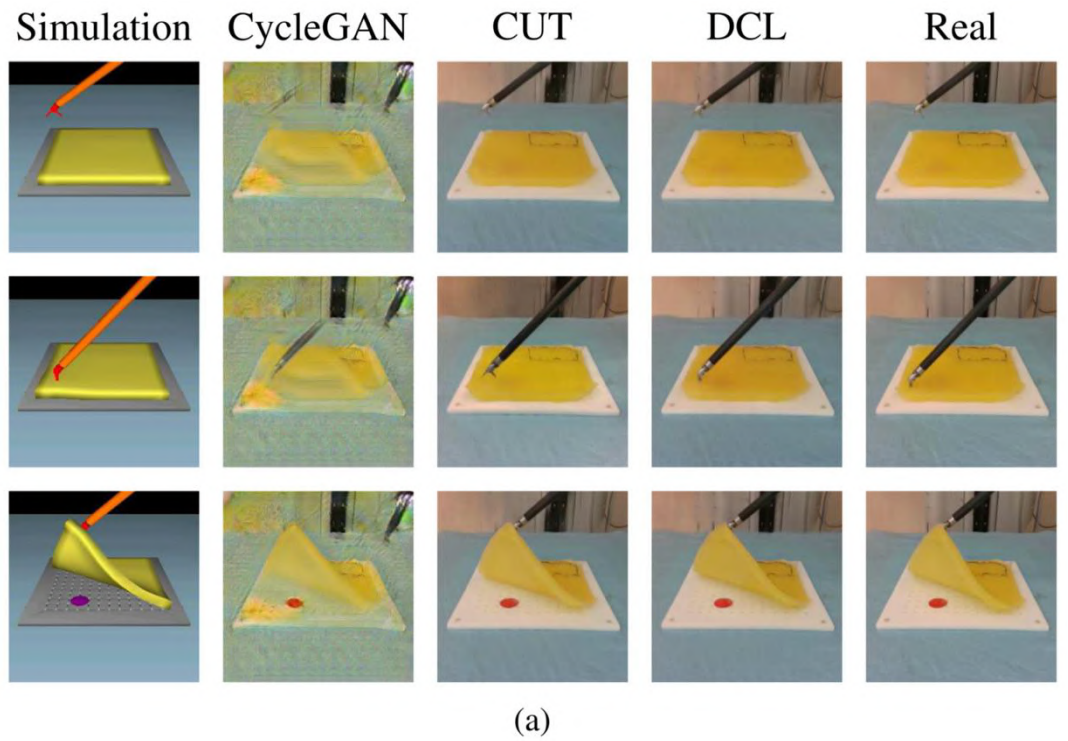
[Rao, et al. "RL-CycleGAN: Reinforcement Learning Aware Simulation-To-Real" CVPR 2020]



Cycle-gans

Domain adaptation





[https://taesung.me/ContrastiveUnpairedTranslation/]

[Tagliabue, et al. "Sim-to-Real Transfer for Visual Reinforcement Learning of Deformable Object Manipulation for Robot-Assisted Surgery" R-AL 2022]

Outline

1. Train in real world directly
2. Learn in Simulation and Sim2Real transfer
3. **Use Human data: Imitation learning**

Coming back to data availability

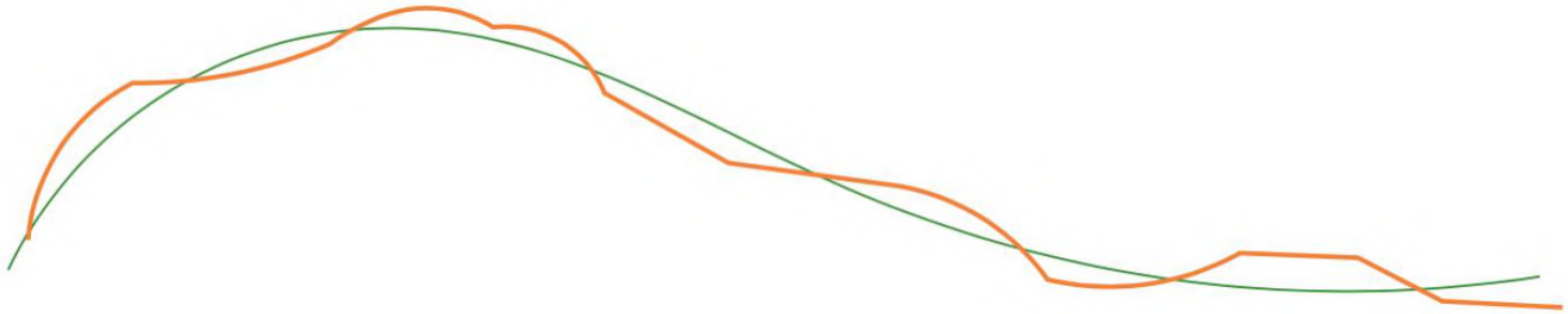
Can we train on real robot data? **Imitation learning**

Behaviour Cloning

- A **supervised learning** problem that maps state/action pairs to policy (State: feature, action: label)
- Works well in practice, especially when used with RNN.

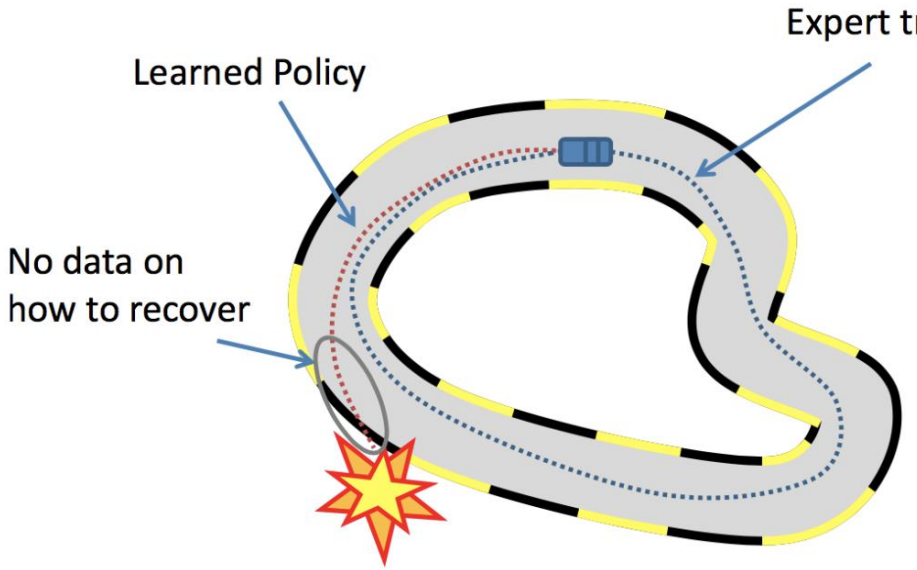
Behaviour Cloning Problem: Compounding error

- Requires a **large number of expert trajectories** (high sample complexity)
- Supervised learning assumes iid. (s,a) pairs and ignores temporal structure independent



- Error at time t with probability $\leq \in$
- E[Total Error] $\leq \in T$

Behaviour Cloning Problem: Compounding error



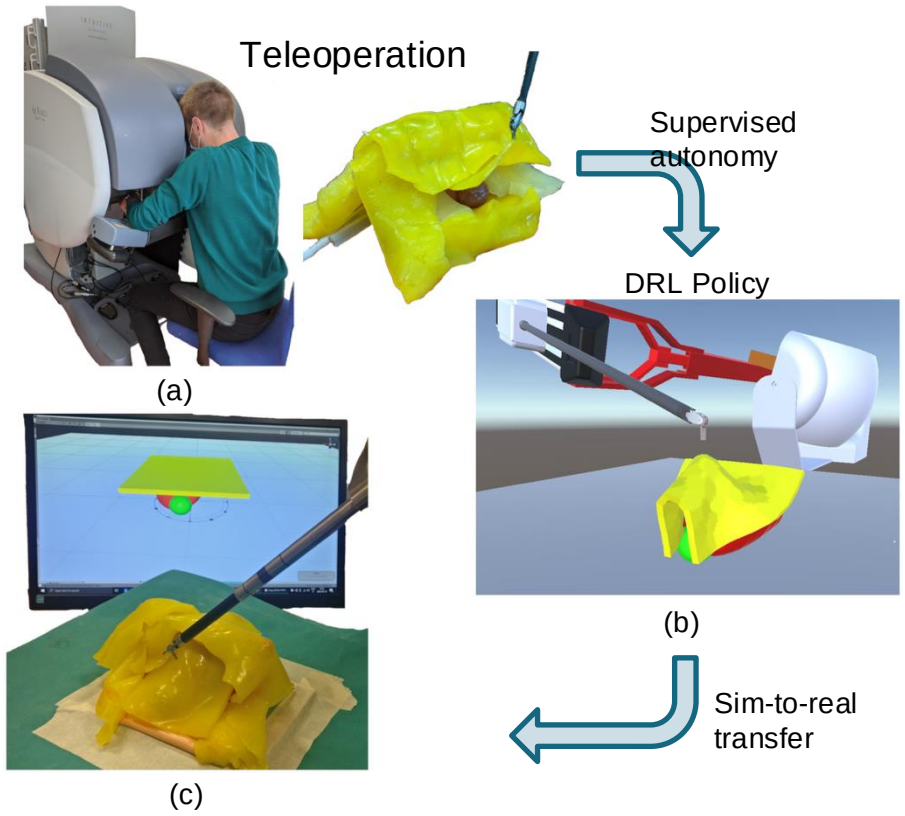
Data Distribution Mismatch!

In supervised learning, $(x, y) \sim D$ during train **and** test. In MDPs:

- Train: $s_t \sim D_{\pi^*}$
- Test: $s_t \sim D_{\pi_\theta}$

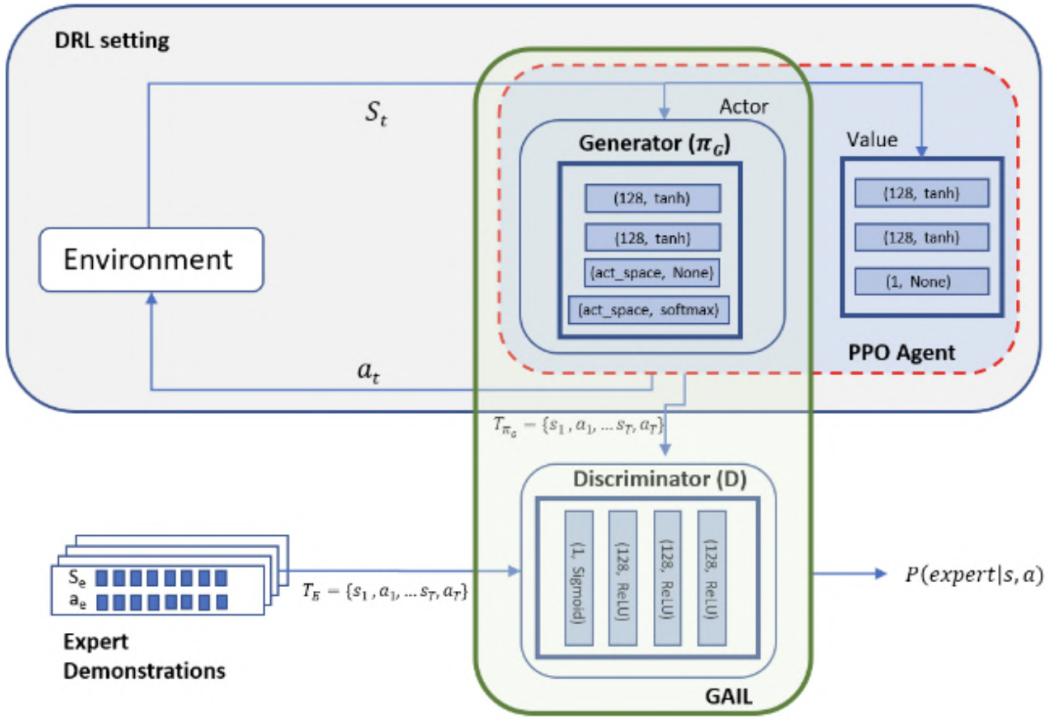
[Ross et al. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning], 2011

Generative Adversarial Imitation learning (GAIL)



[Pore et al. "Learning from demonstrations for autonomous soft-tissue retraction." ISMR 2021.]

[Ho and Ermon. "Generative adversarial imitation learning." NeurIPS 2016]

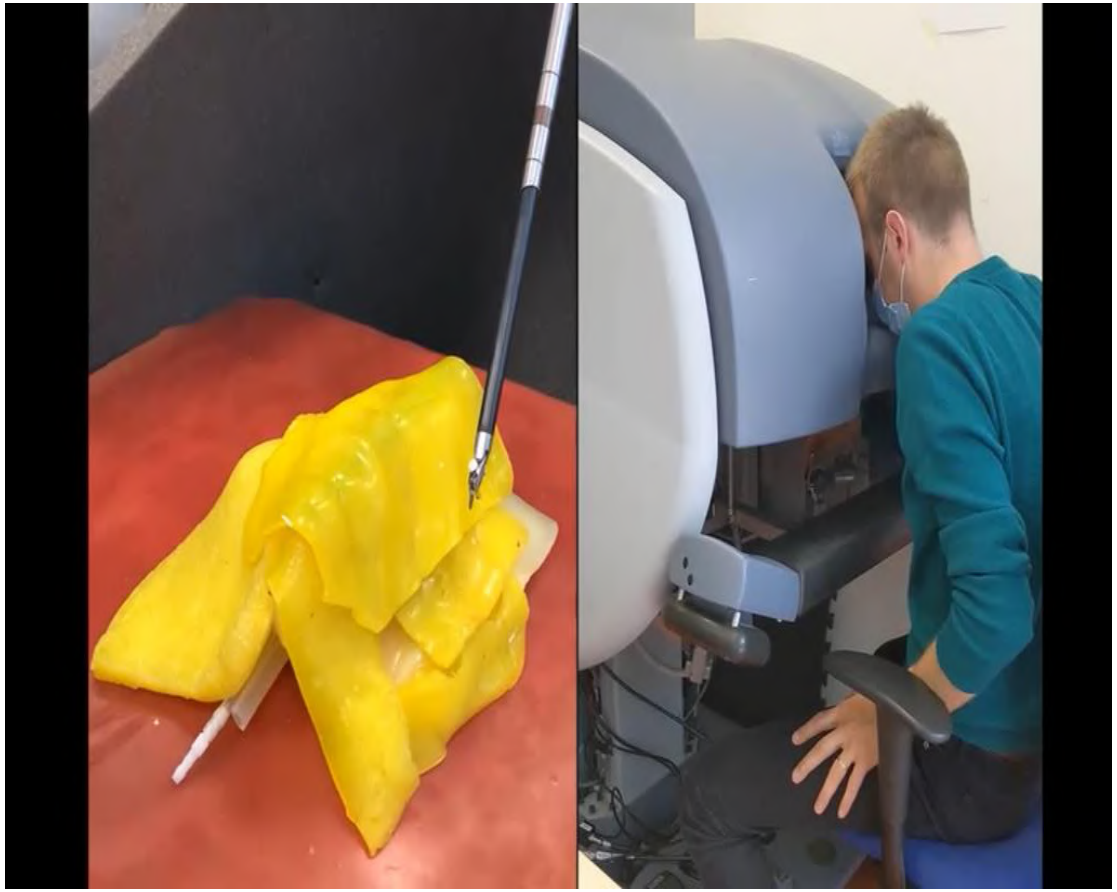


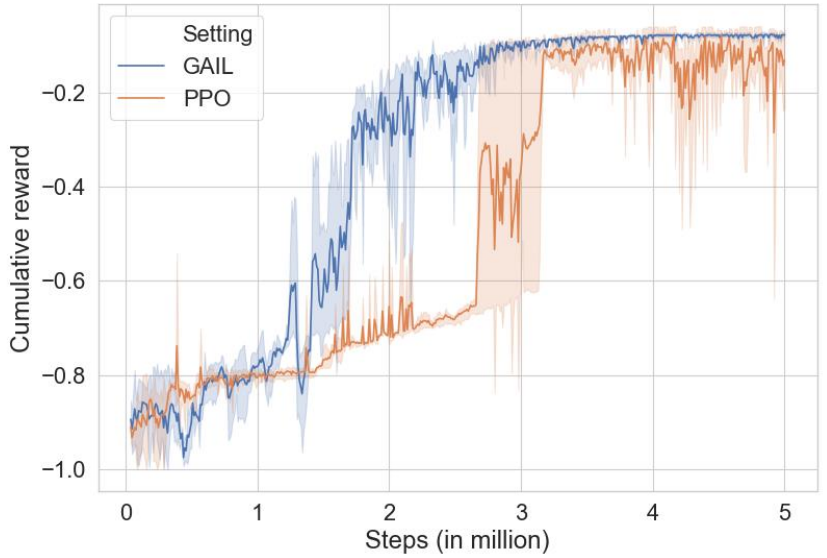
$$L_{GAIL} = E_{\tau_\phi}[\log(D_{\phi'}(s_t, a))] + E_{\tau_E}[\log(1 - D_{\phi'}(s_t, a))]$$

LfD loss

$$L_{Total} = \alpha L_{DRL} + \beta L_{GAIL}$$

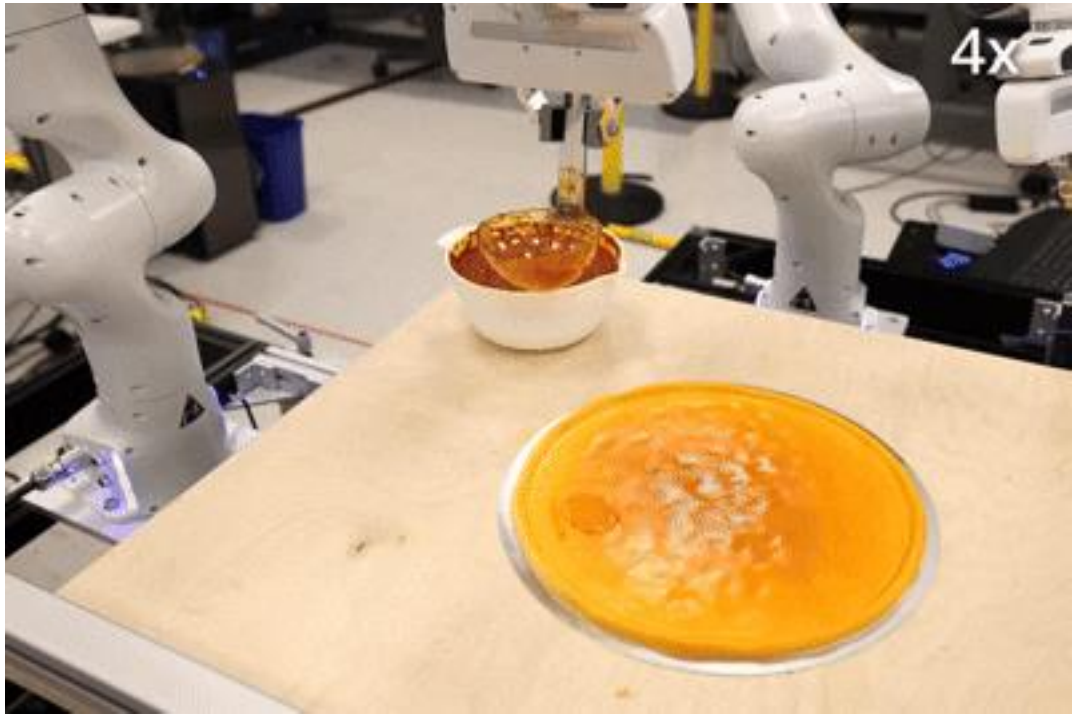
GAIL

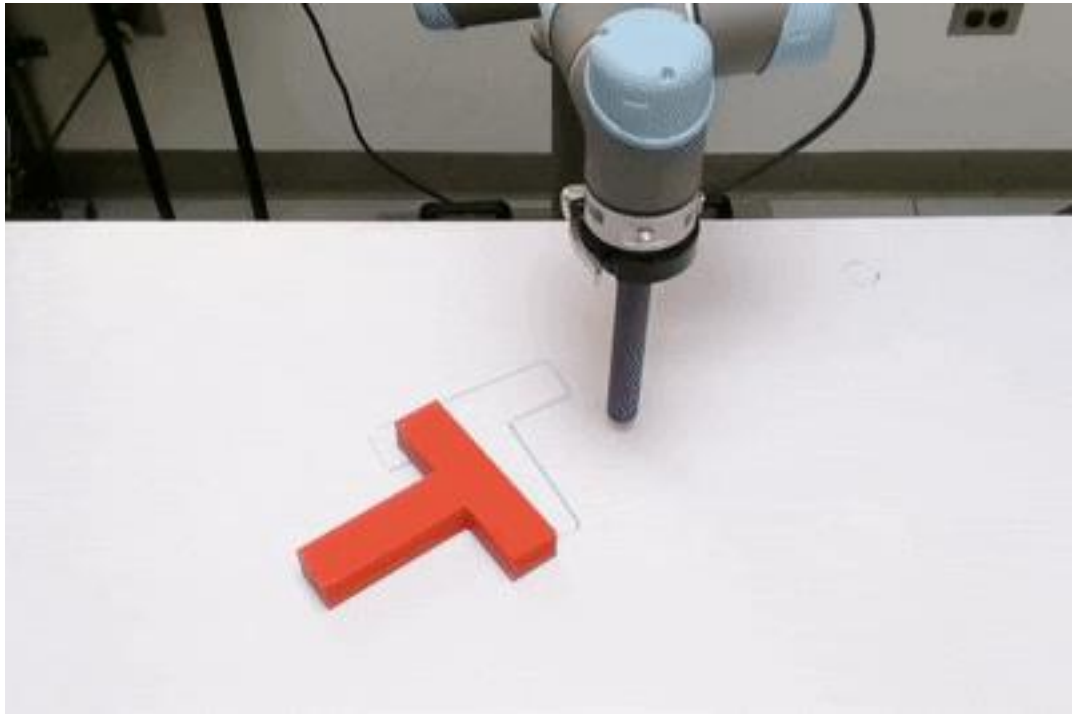




[Pore et al. "Learning from demonstrations for autonomous soft-tissue retraction." ISMR 2021.]

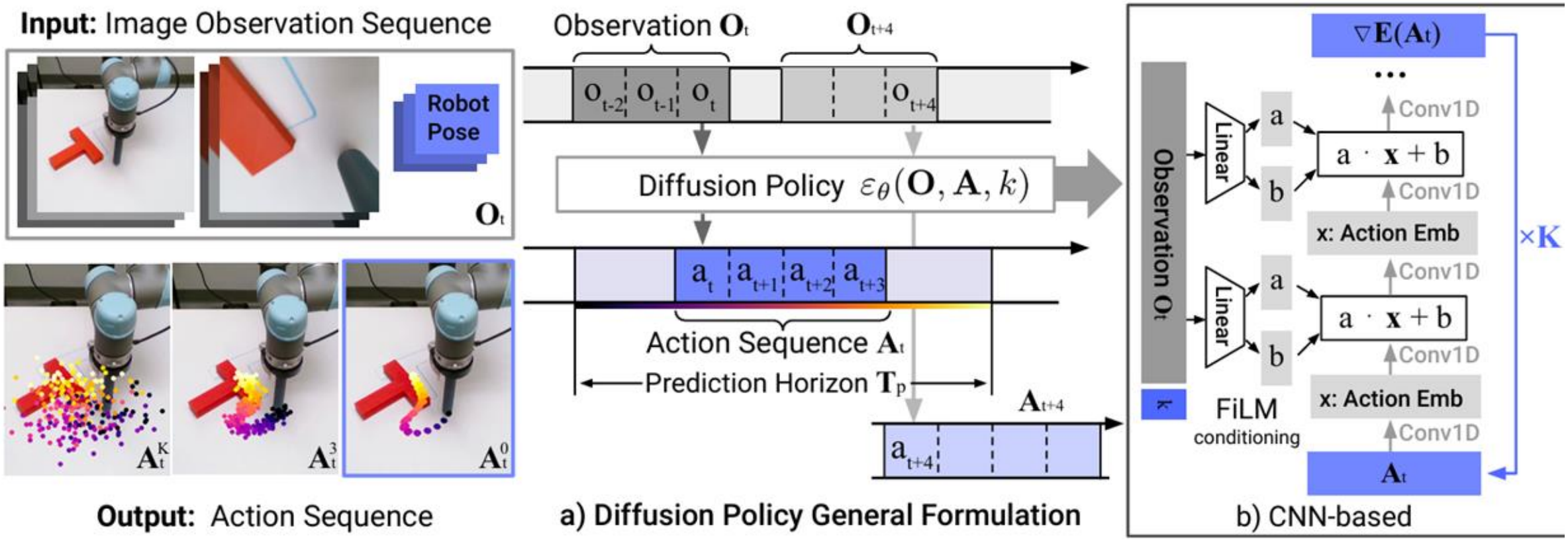
Diffusion Policy





https://diffusion-policy.cs.columbia.edu/

Diffusion Policy



https://diffusion-policy.cs.columbia.edu/

Data collection

Teleoperation

30 Hz

MedRCT

30 Hz

Franka Robot

Controller

Robot pose:
p xyz, **R** xyzw: 50Hz

Image: 30 Hz

Static Camera

Dataset

10 Hz

Synchronizer

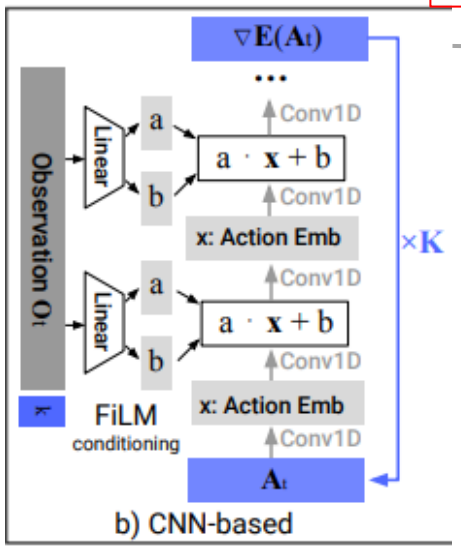
Image: 20 Hz

Wrist Camera

Slide Credits: Radian Gondokaryono

Inference Pipeline

1-5 Hz!



MedRCT

1-5 Hz

RL Model

30-50 Hz

Franka Robot

Controller

Robot pose:
p xyz, **R** xyzw: 50Hz

Image: 30 Hz

Static Camera

Image: 20 Hz

Wrist Camera

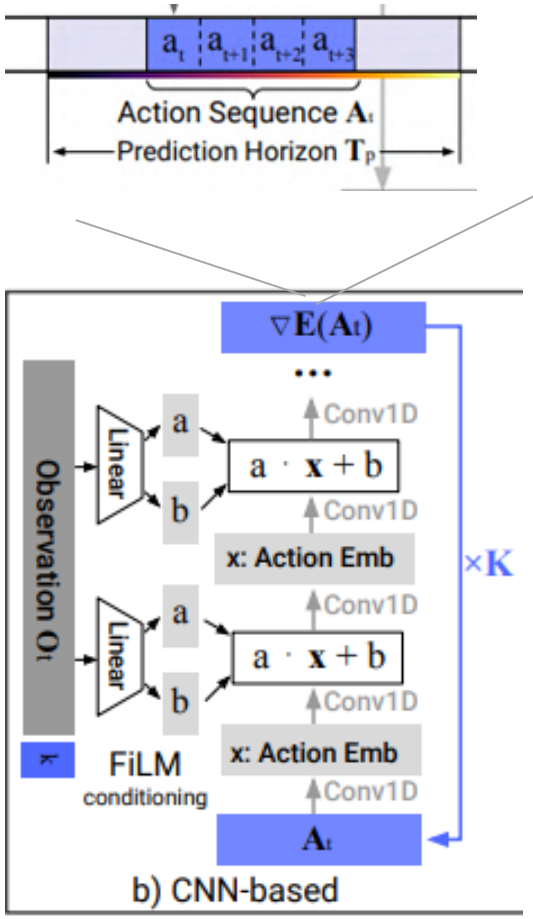
IL Model

10 Hz

Synchronizer

Slide Credits: Radian Gondokaryono

Diffusion speed



1-5 Hz!

Teleoperation

MedRCT

1-5 Hz

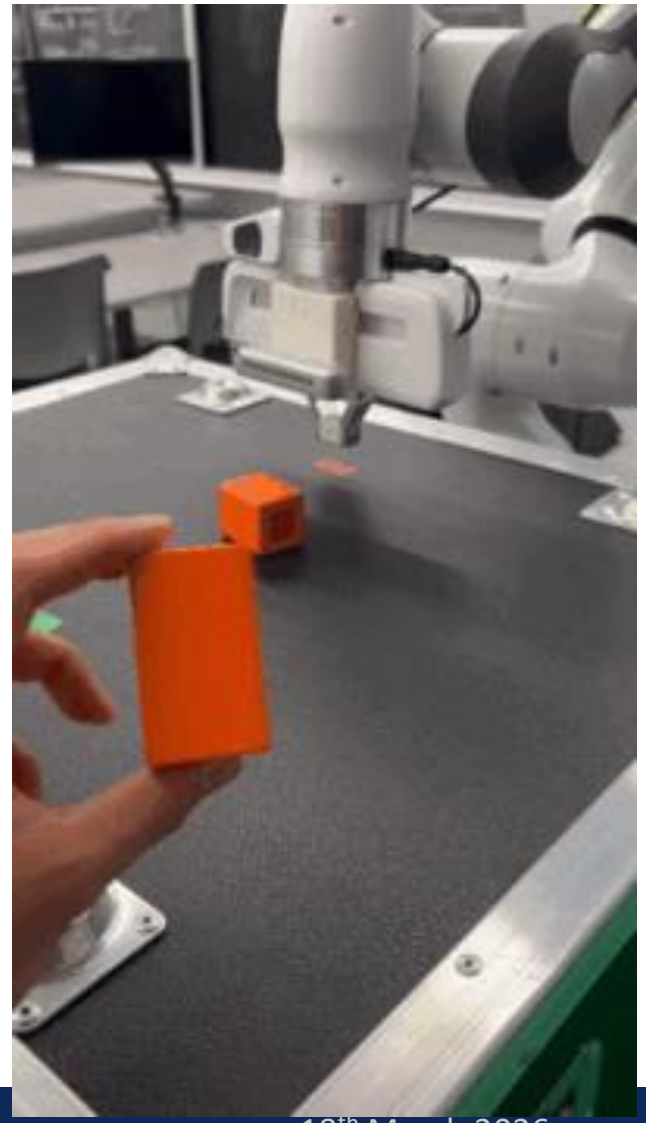
K : [10 - 100] iterations !

Recently solved with Flow Matching

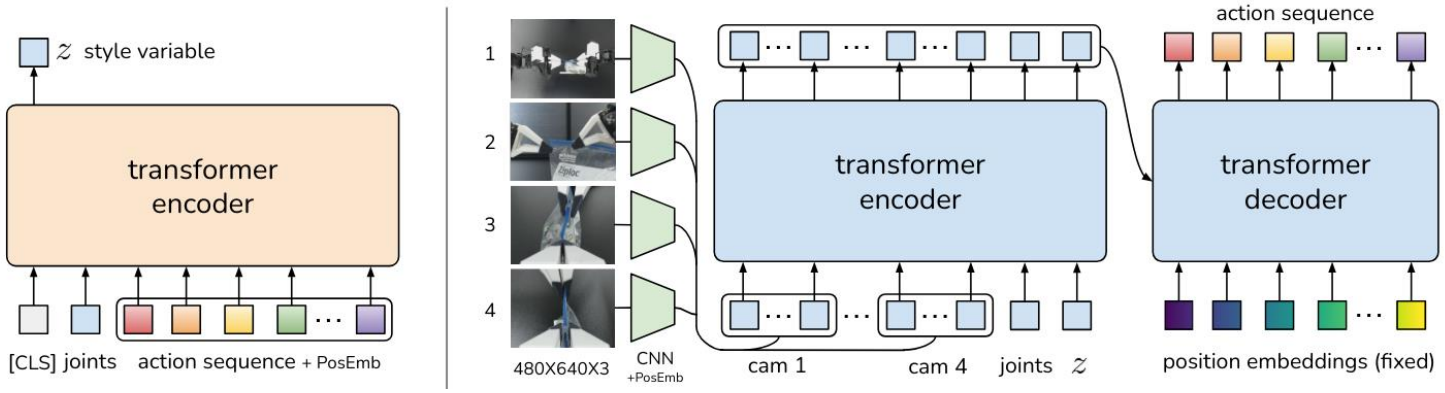
Lipman, Yaron, et al. "Flow matching for generative modeling." arXiv preprint arXiv:2210.02747 (2022).

Slide Credits: Radian Gondokaryono

Diffusion Policy: Results from our lab

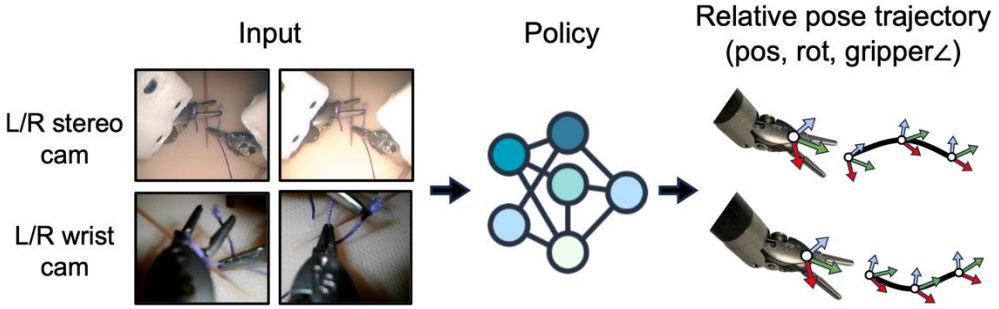


Action Chunking Transformer

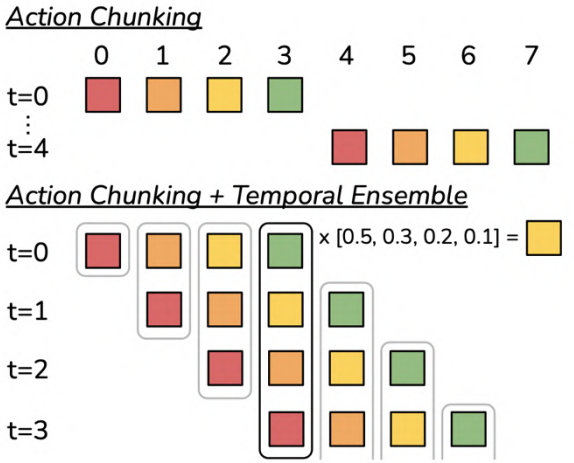


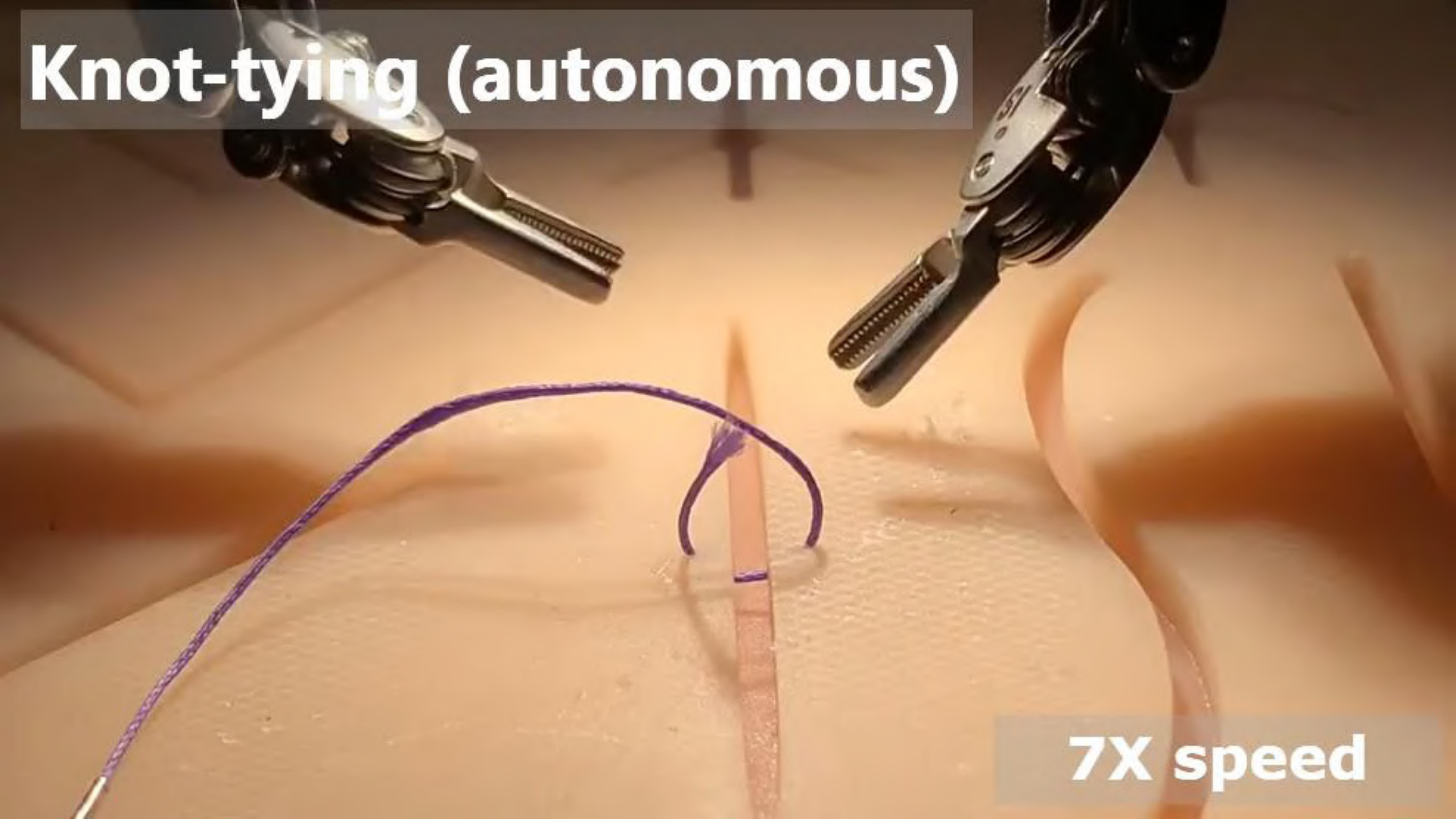
Action Chunking Transformer (ACT)

[Zhao et al. "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware", RSS 2023]

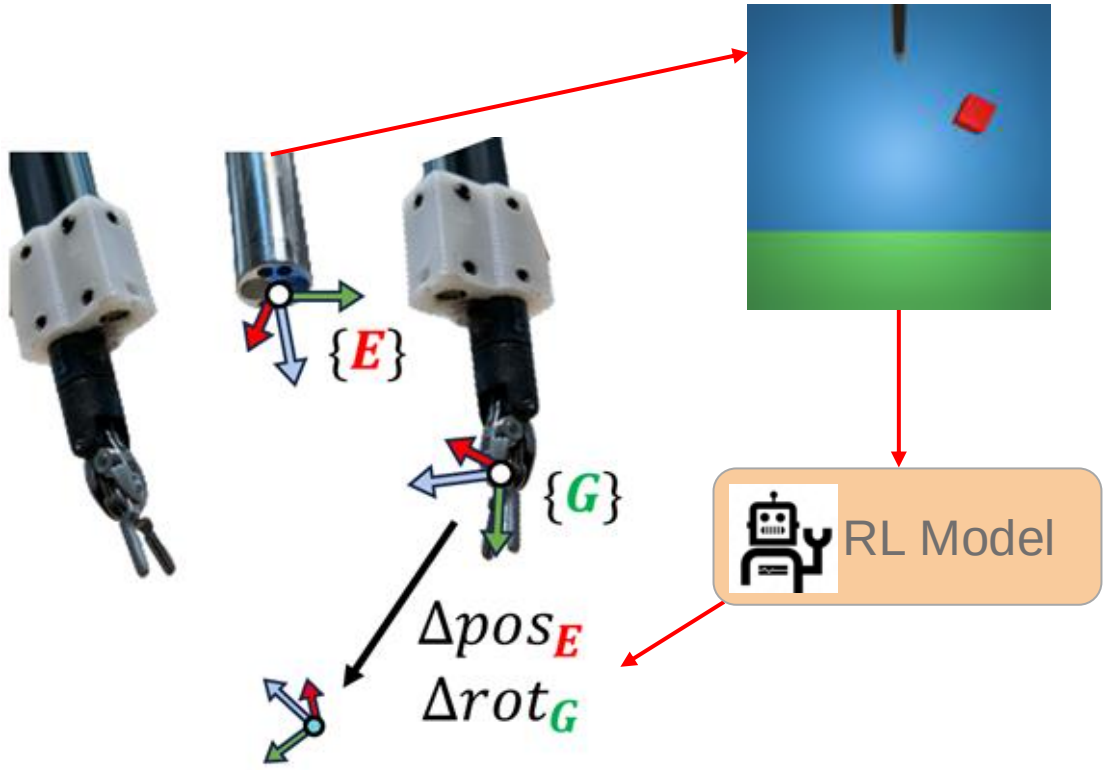


[Kim, Ji Woong, et al. "Surgical robot transformer (srt): Imitation learning for surgical tasks." CoRL 2025.]



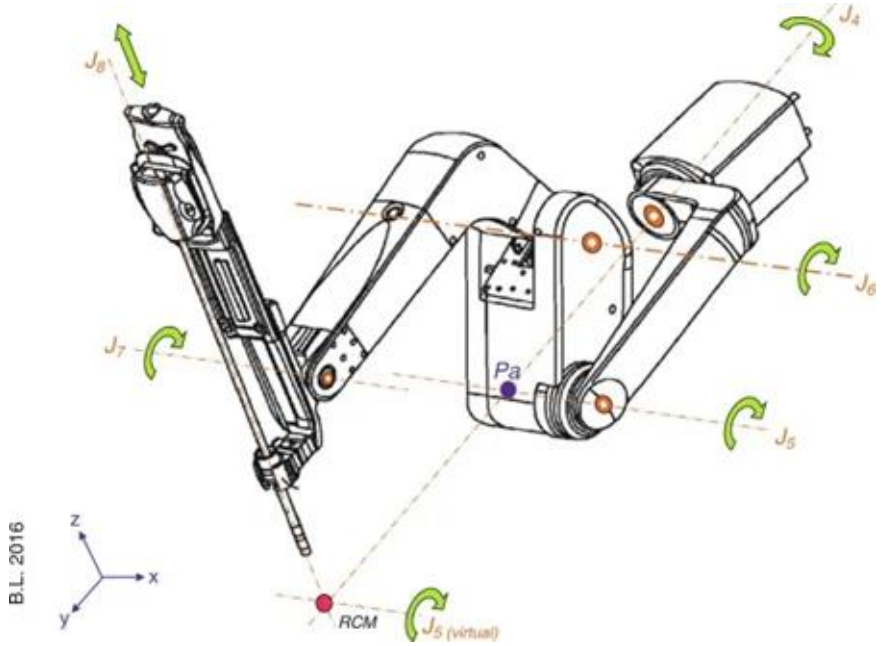

Knot-tying (autonomous)
7X speed

Visual motor Control



https://surgical-robot-transformer.github.io/

https://entokey.com/the-da-vinci-system-technology-and-surgical-analysis/



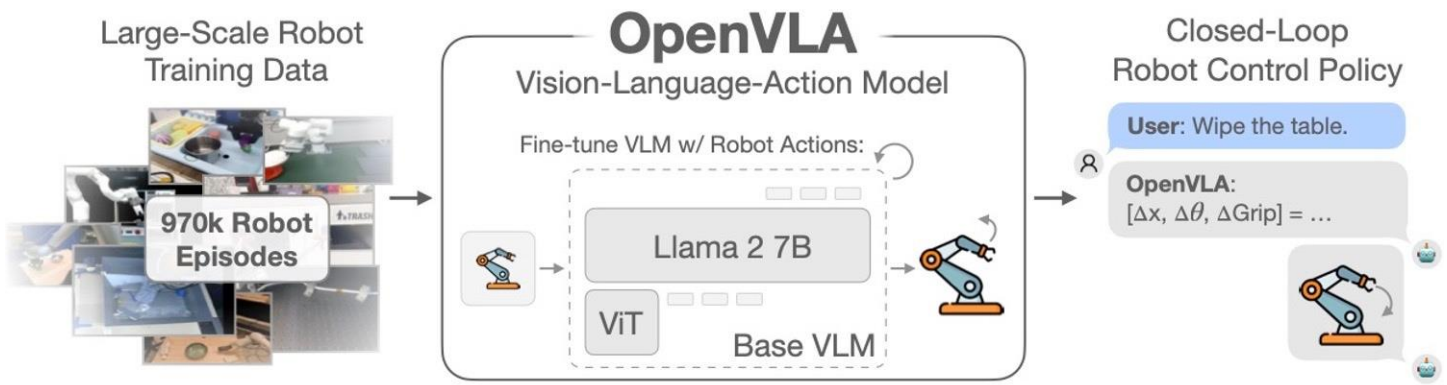
Cartesian Control: Fine Manipulation
Sample Efficient

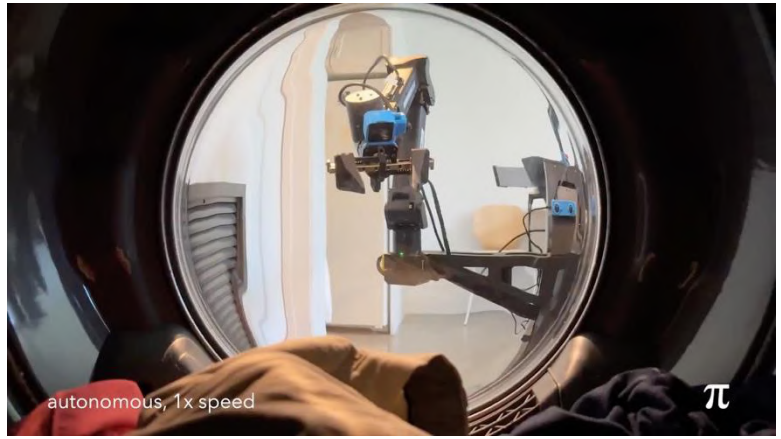
Slide Credits: Radian Gondokaryono

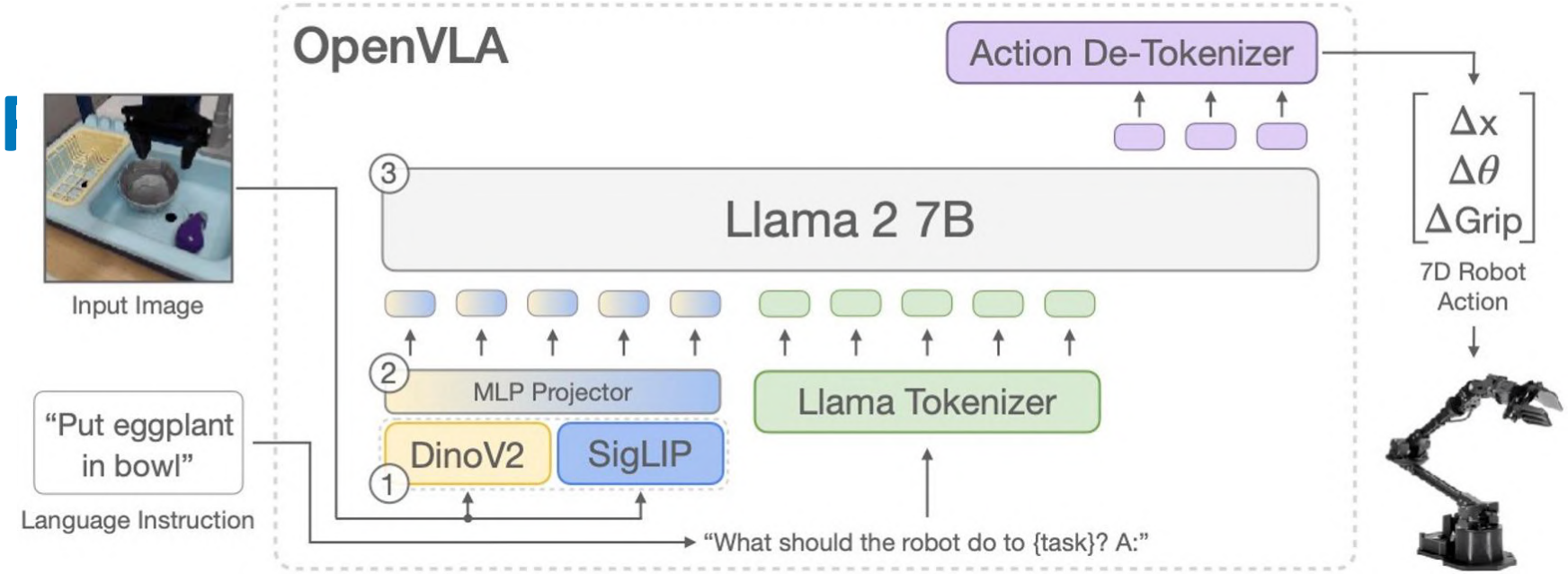
Joint Control

Dynamics Identification

Vision Language Action (VLA)







Kim, et al. "Openvla: An open-source vision-language-action model." arXiv preprint arXiv:2406.09246 (2024).

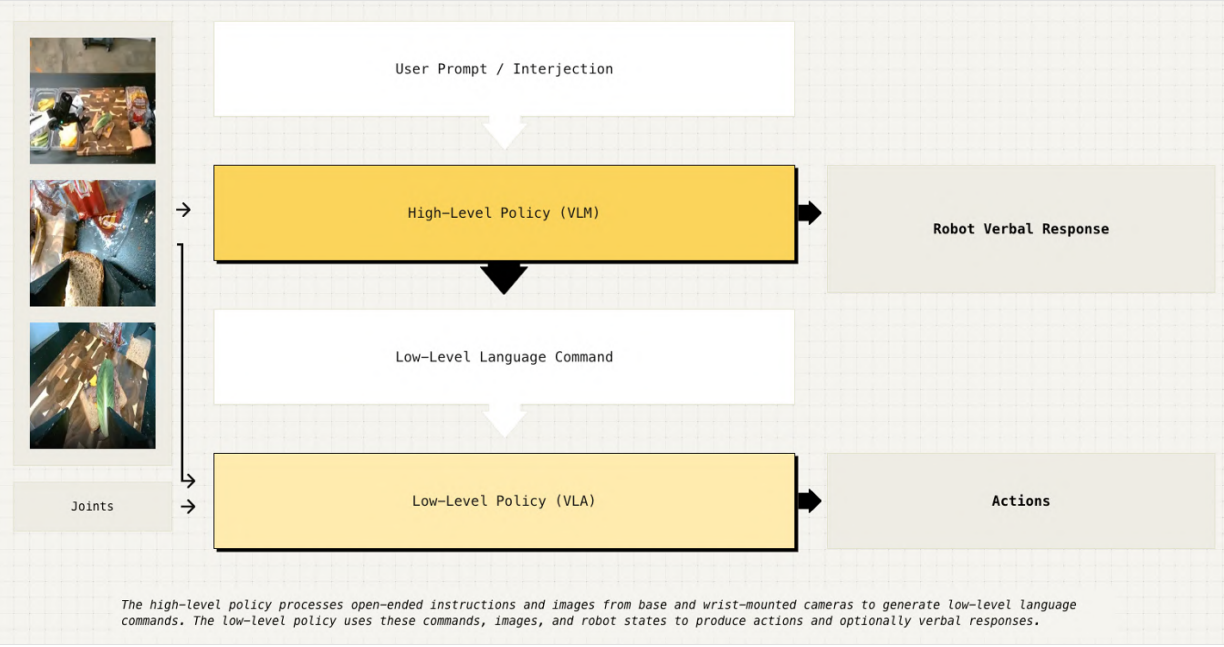
Black, Kevin, et al. "π_0: A Vision-Language-Action Flow Model for General Robot Control." arXiv preprint arXiv:2410.24164 (2024).

Parameter Efficient Fine-Tuning

Strategy	Success Rate	Train Params ($\times 10^6$)	VRAM (batch 16)
Full FT	**69.7 ± 7.2 %**	7,188.1	163.3 GB*
Last layer only	30.3 ± 6.1 %	465.1	51.4 GB
Frozen vision	47.0 ± 6.9 %	6,760.4	156.2 GB*
Sandwich	62.1 ± 7.9 %	914.2	64.0 GB
LoRA, rank=32	**68.2 ± 7.5%**	**97.6**	**59.7 GB**
rank=64	**68.2 ± 7.8%**	195.2	60.5 GB

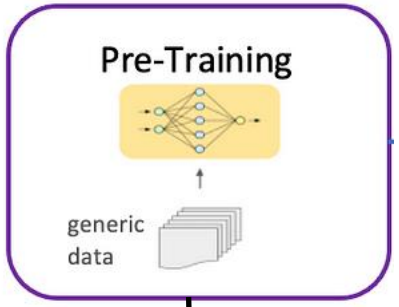
Infrastructure

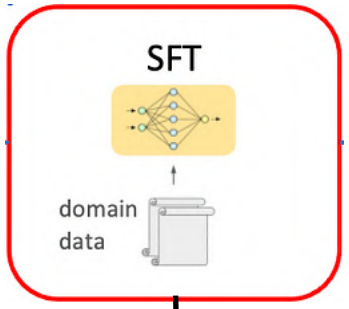
- **Training:** 64 A100 GPU for 14 days: 21500 hrs
- **Full Fine-tuning:** 8 A100 GPU for 5-15 hrs
- **Inference:** 15GB GPU RTX 4090 memory. Runs at 6Hz.

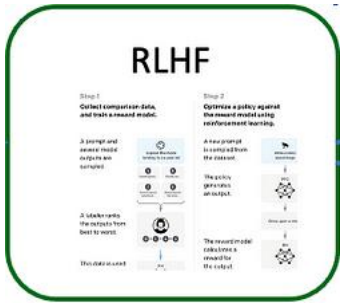


Hierarchical structure for long horizon tasks

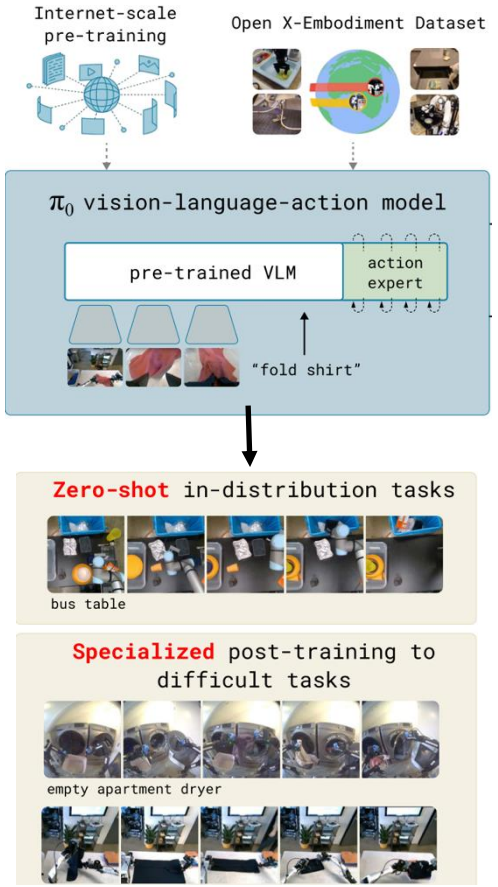
Foundational Models for Robotics







Data is no longer a bottleneck



Pre-training

SFT

Large scale robotic data is the bottleneck

RL?

We are here.

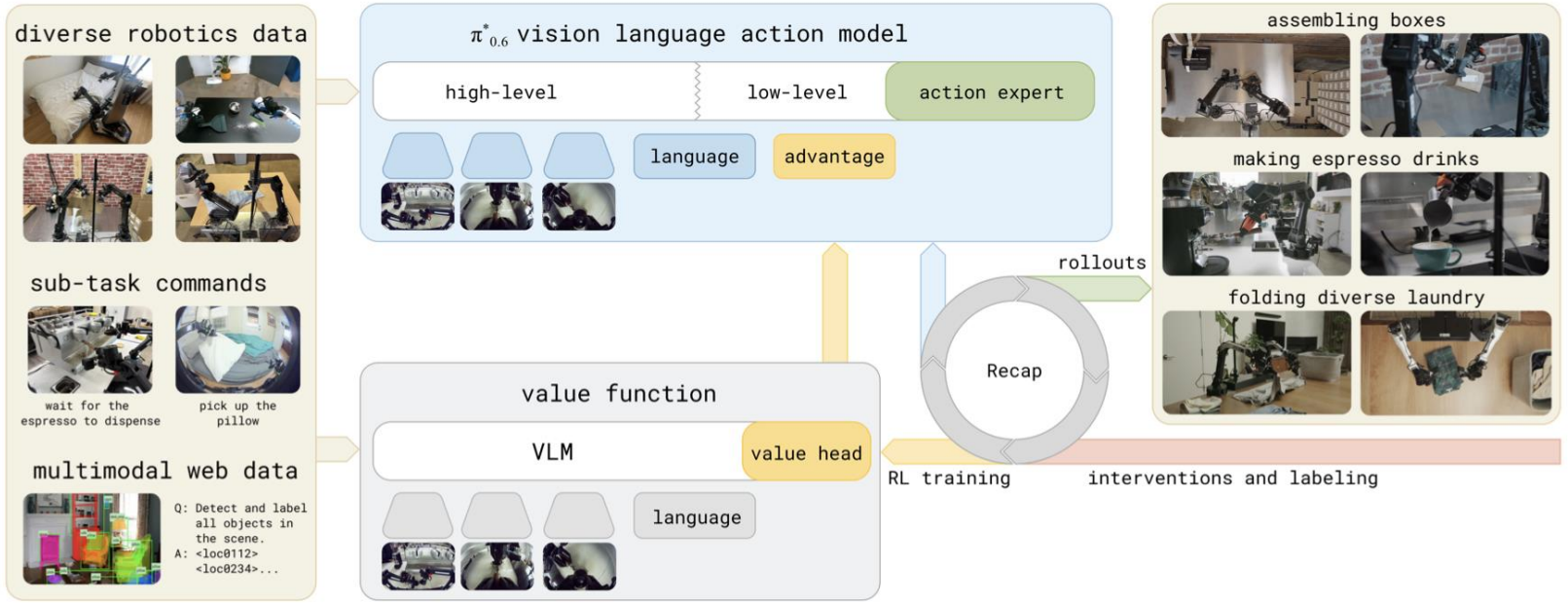
Upcoming Opportunities: RL for VLA

$\pi^*_{0.6}$: a VLA That Learns From Experience

Physical Intelligence

Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, Zhiyuan Zhou

https://pi.website/blog/pistar06



2026 seems to be about RL for VLA!!

UNIVERSITY OF TORONTO

Robotics Institute

Thank you!