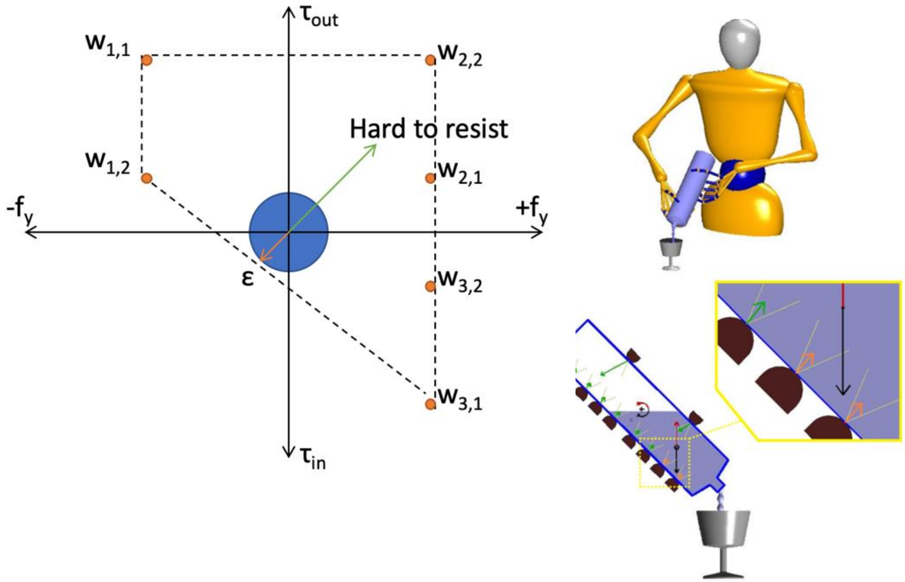
Principles of Robot Autonomy II

Learning-based Approaches to **Grasping**

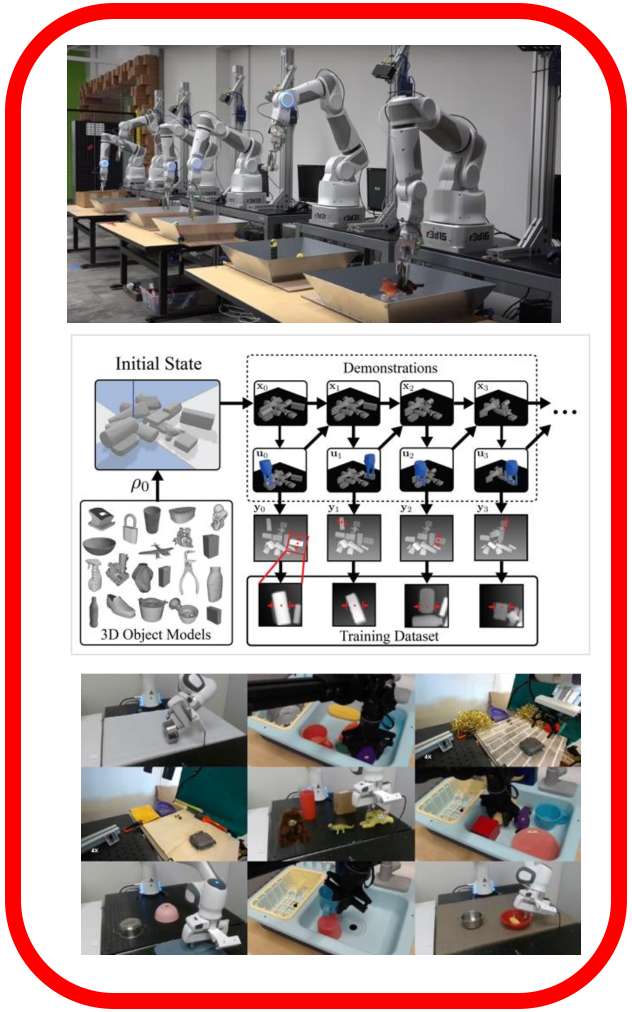
Jeannette Bohg

Stanford University

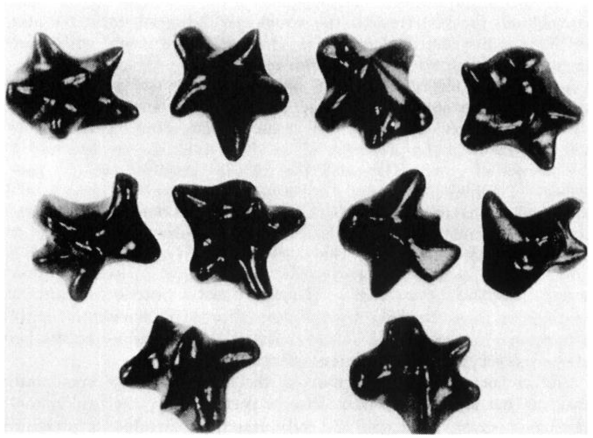
Learning Outcome for next five Lectures



Modeling and Evaluating Grasping and Manipulation



Learning-based Grasping and Manipulation



Use Manipulation to Perceive better

Modeling Planar Pushing

Friction limit surface: describes friction forces occurring when part slides over support.

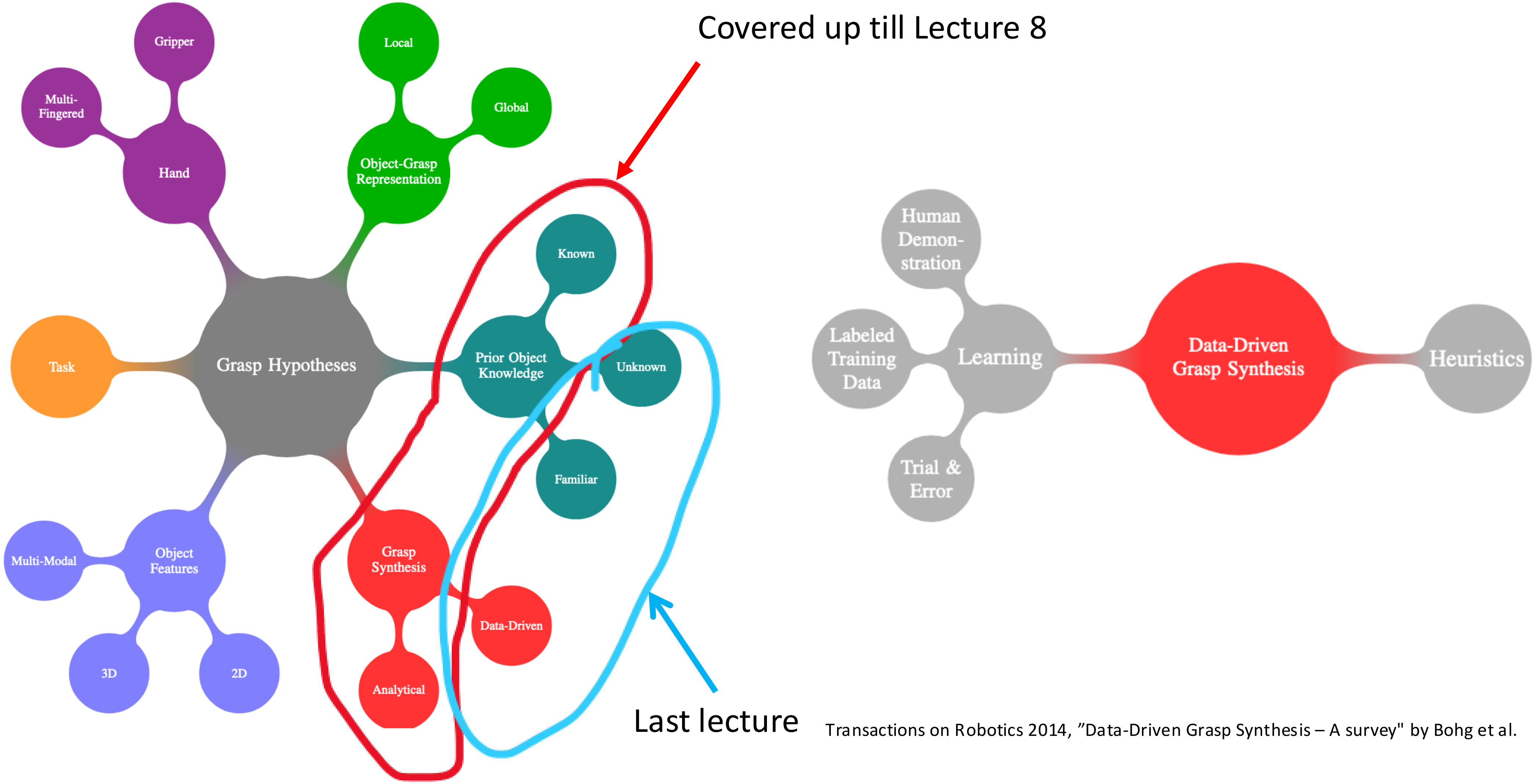
When pushed with a wrench within the limit surface: **no motion.**

For **quasi-static pushing**: wrench on the limit surface; object twist normal to limit surface where **twist** = linear and angular velocity: $t_i = (v_x^i, v_y^i, \omega_z^i)$

Coulomb friction follows the **maximum dissipation principle**: for a given applied wrench, the system will move in a way that maximizes power dissipation due to friction. This implies that the velocity (twist) should be in the direction where the friction force does the most work.

Relation between wrench cone, limit surface and unit twist sphere. Adopted from Chapter 37, Fig 37.10 in Springer Handbook of Robotics.

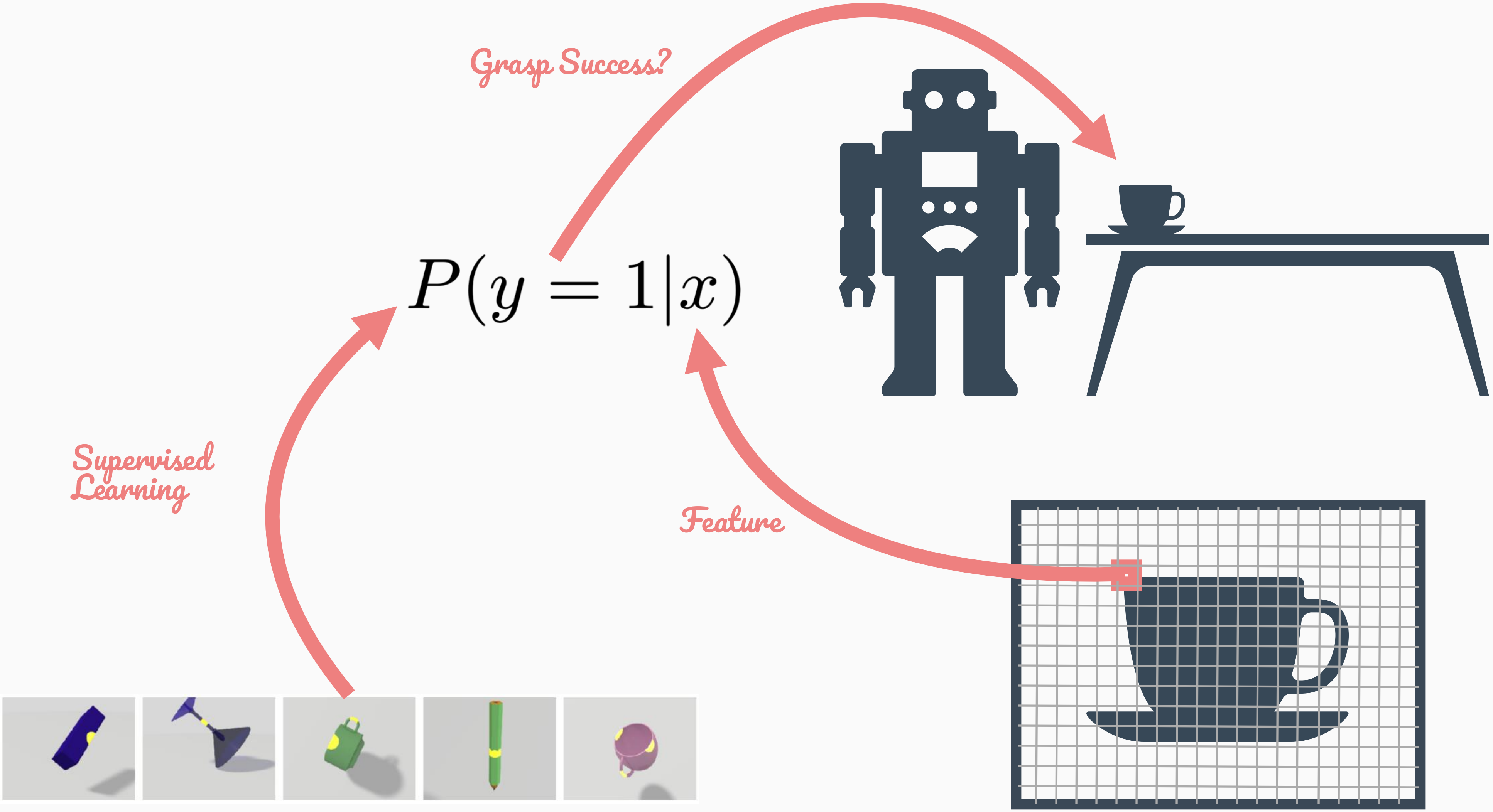
Data-Driven Approaches to Grasping



Transactions on Robotics 2014, "Data-Driven Grasp Synthesis – A survey" by Bohg et al.

Detecting 2D Grasping Points

Grasp Point Detection as a Classification Problem

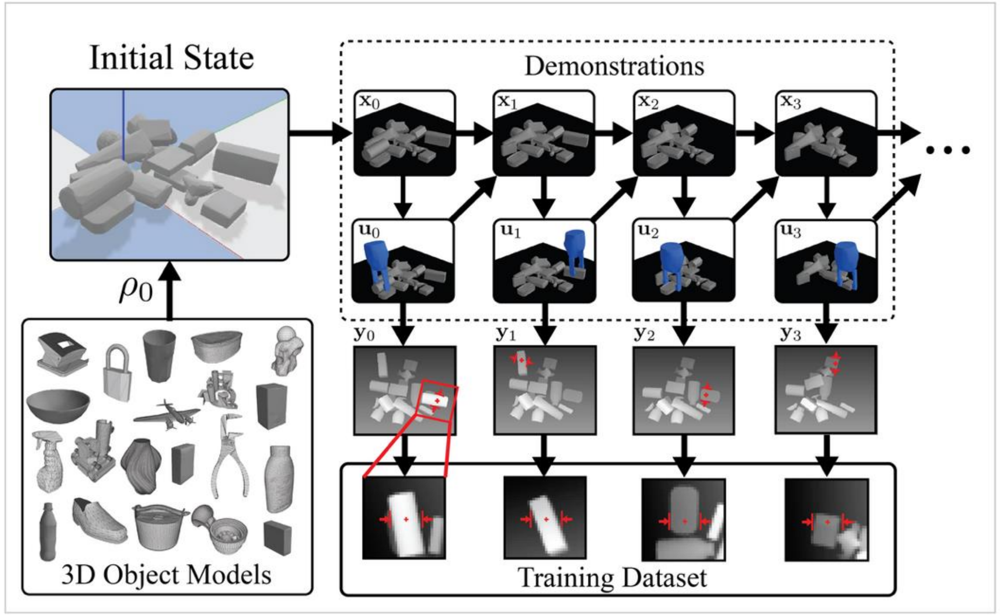


Saxena et al. *Robotic Grasping of Novel Objects.* NeurIPS 2006

Using more sensing modalities and data to learn features and grasp policies

- DexNet 1.0 – 4.0 – Berkeley – AutoLab
- Google Arm Farm

"**Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection**" by Levine et al. IJRR 2017.



"**Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences**" by Mahler and Goldberg. CORL 2017. https://berkeleyautomation.github.io/dex-net

Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection

Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, Deirdre Quillen

Problem Statement

End-to-end learn to grasp a **wide variety** of household objects in **clutter** using **real hardware**.

Assumptions

- ~~3D Model of Object~~
  - ~~Depth Sensing~~
  - ~~Wrist Mounted Camera~~
- ~~Specific Representation of Geometry~~
- ~~Contact Model~~
- ~~Simulated Data~~
- ~~Hand-Annotations~~
- ~~Hand-Designed Path Planner~~

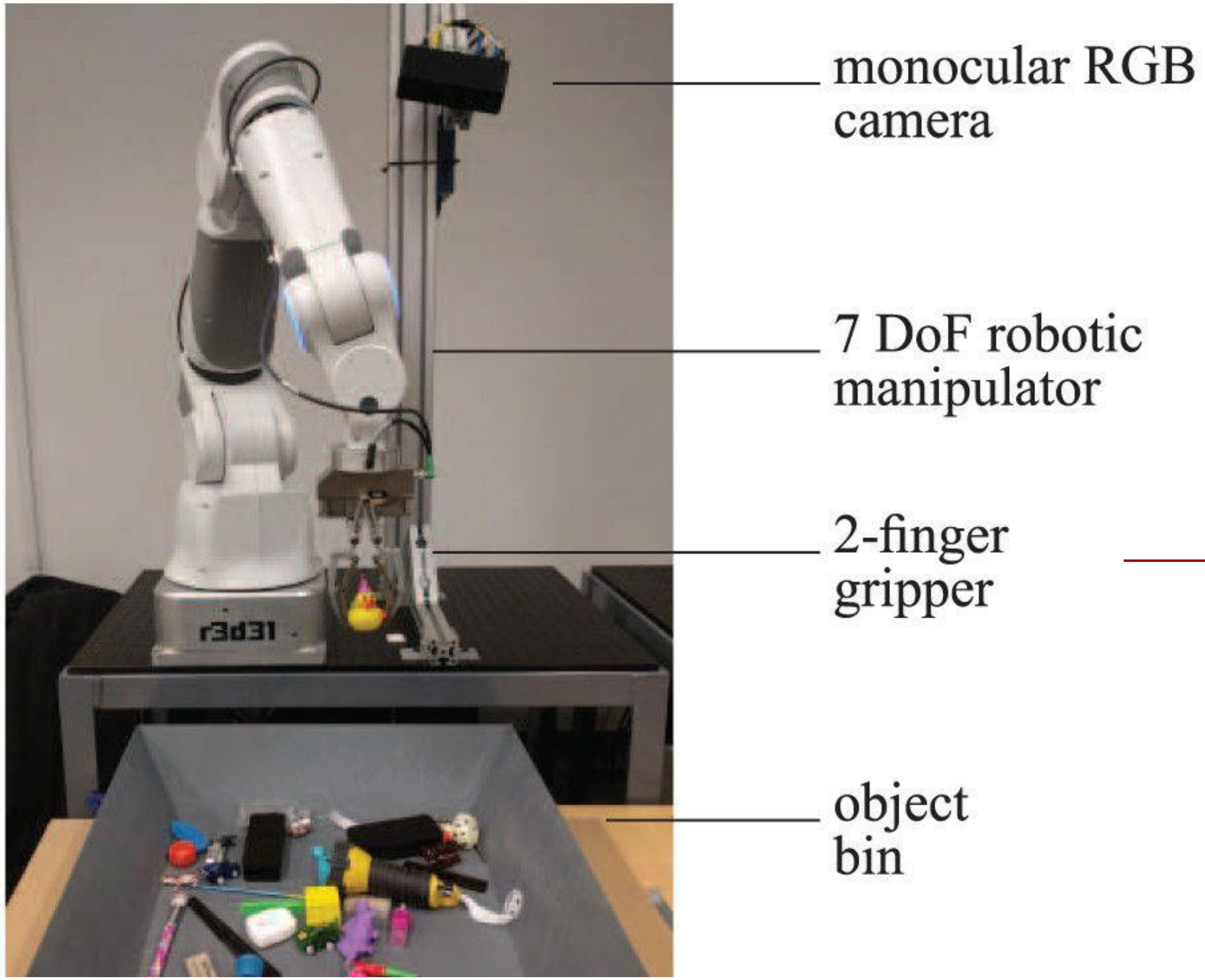
→

RGB Camera

Mounted Over-the-Shoulder

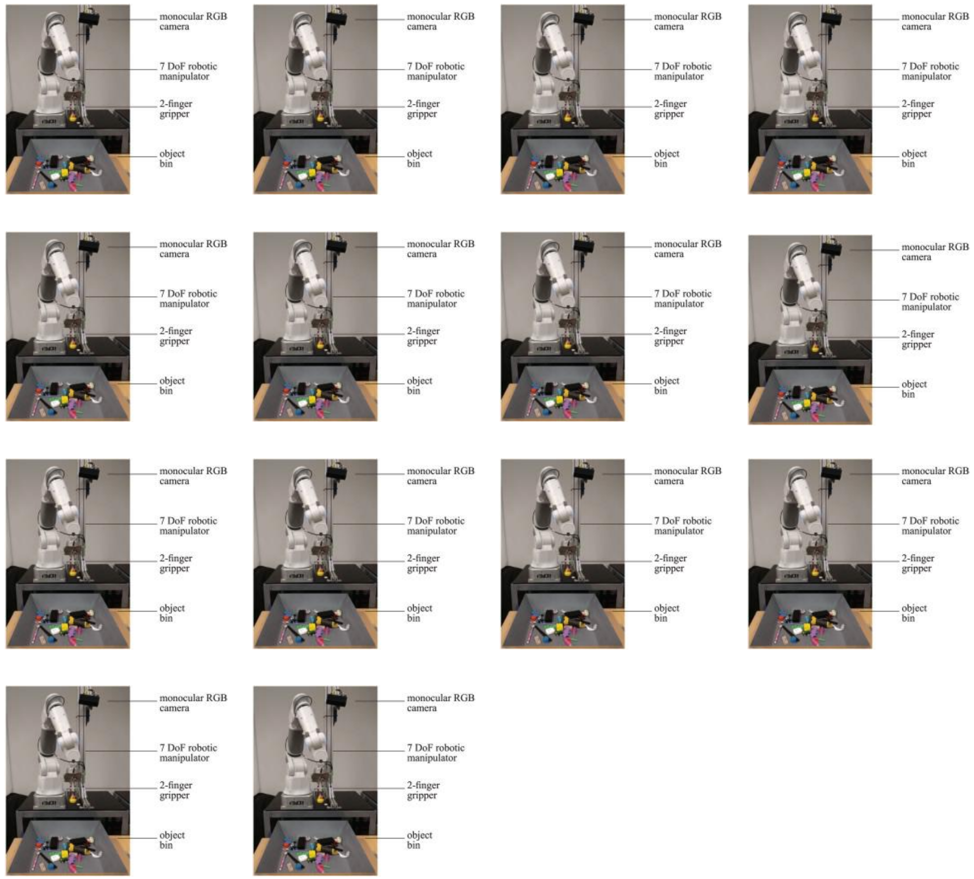
- ~~Camera to Base Calibration~~

So what do we have?



Underactuated to conform to object geometry

So what do we have?



+ Time

Goal

"Examine to what degree a grasping method **based entirely on learning** from raw autonomously collected data can scale to complex and diverse grasp scenarios"

Uncertainty

- Using real hardware leads to a ton of uncertainty
  - Object
    - Geometry & Pose
    - Material Properties
      - weight, frictional properties, deformability
  - Robot
    - End-Effector Pose
    - Wear and Tear
- Accentuated by lack of explicit hand-eye-coordination

Dataset

Data Point Format

(Image, Motor Command, Label)

$$(p_T - p_t)$$

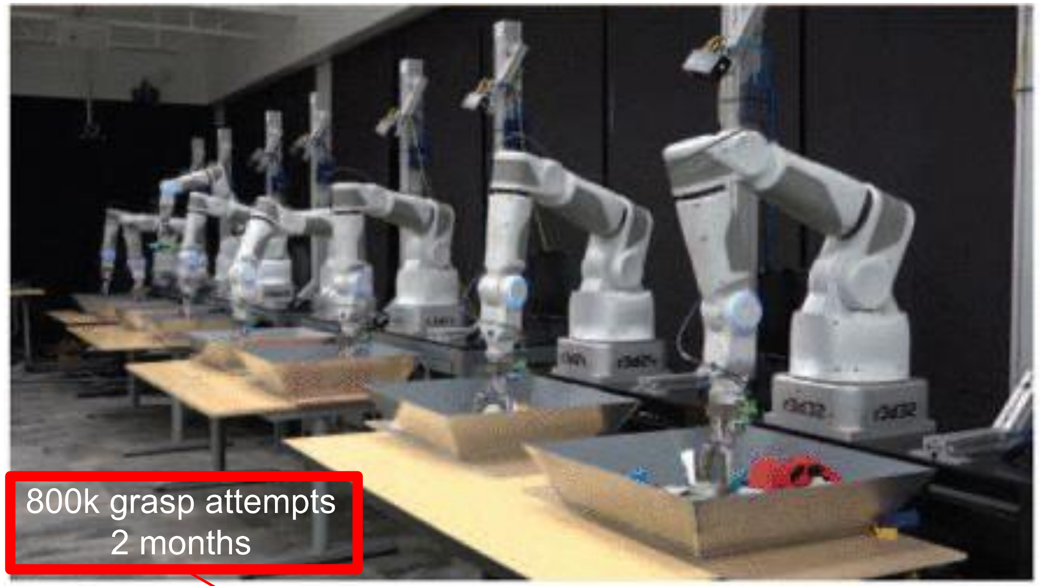
Success

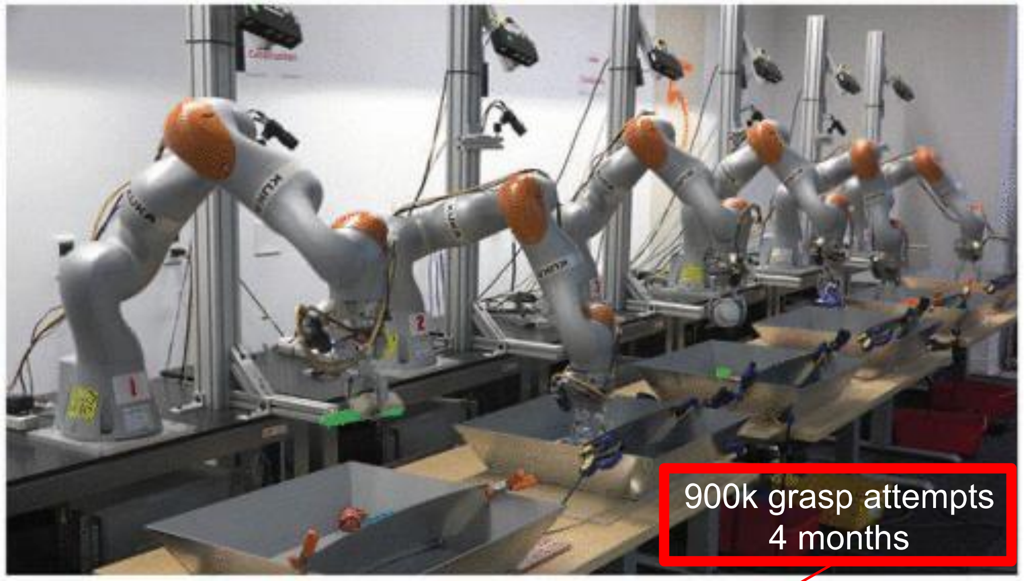
or

Failure

Dataset

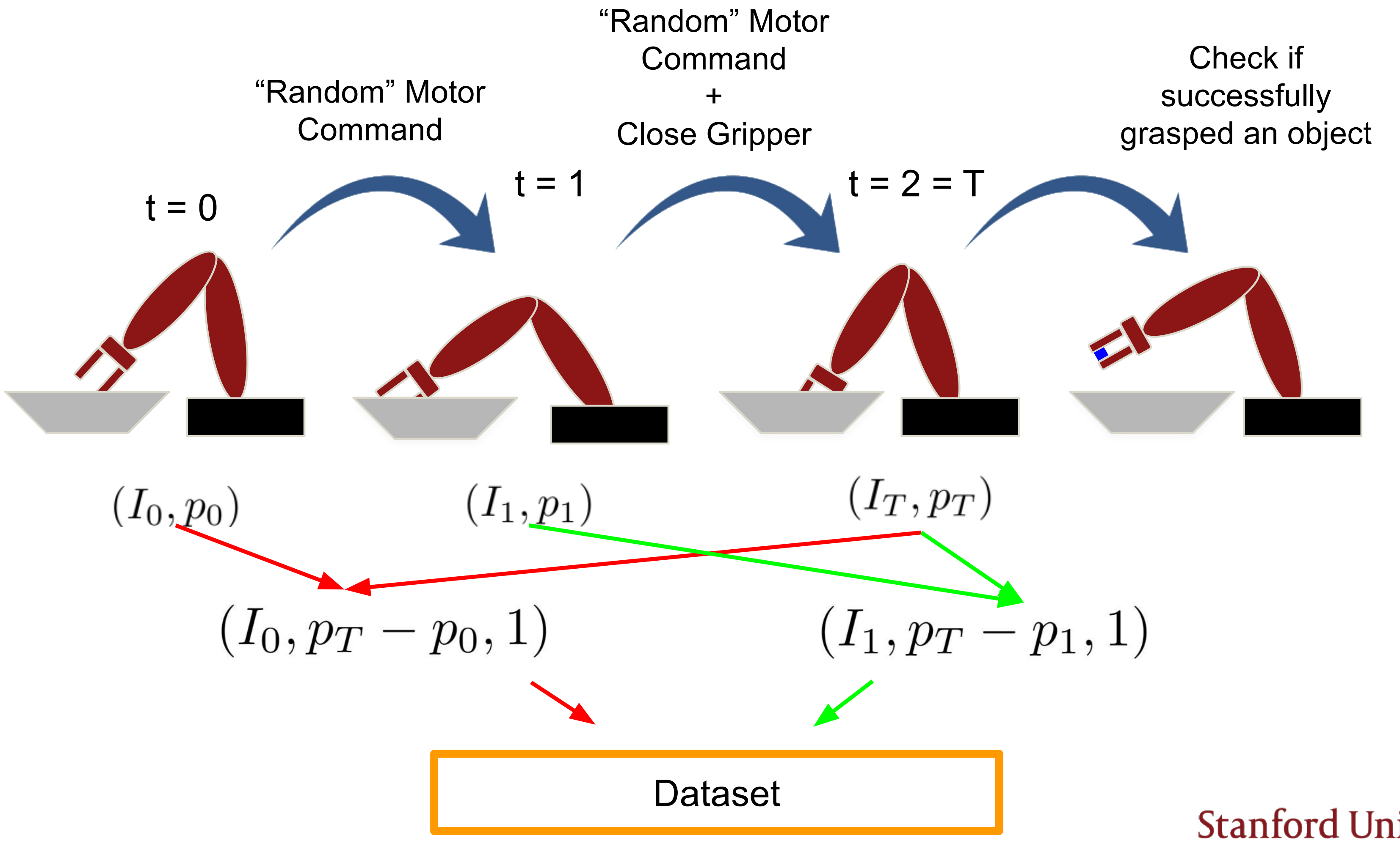
Two Rounds of Self-Supervised Data Collection





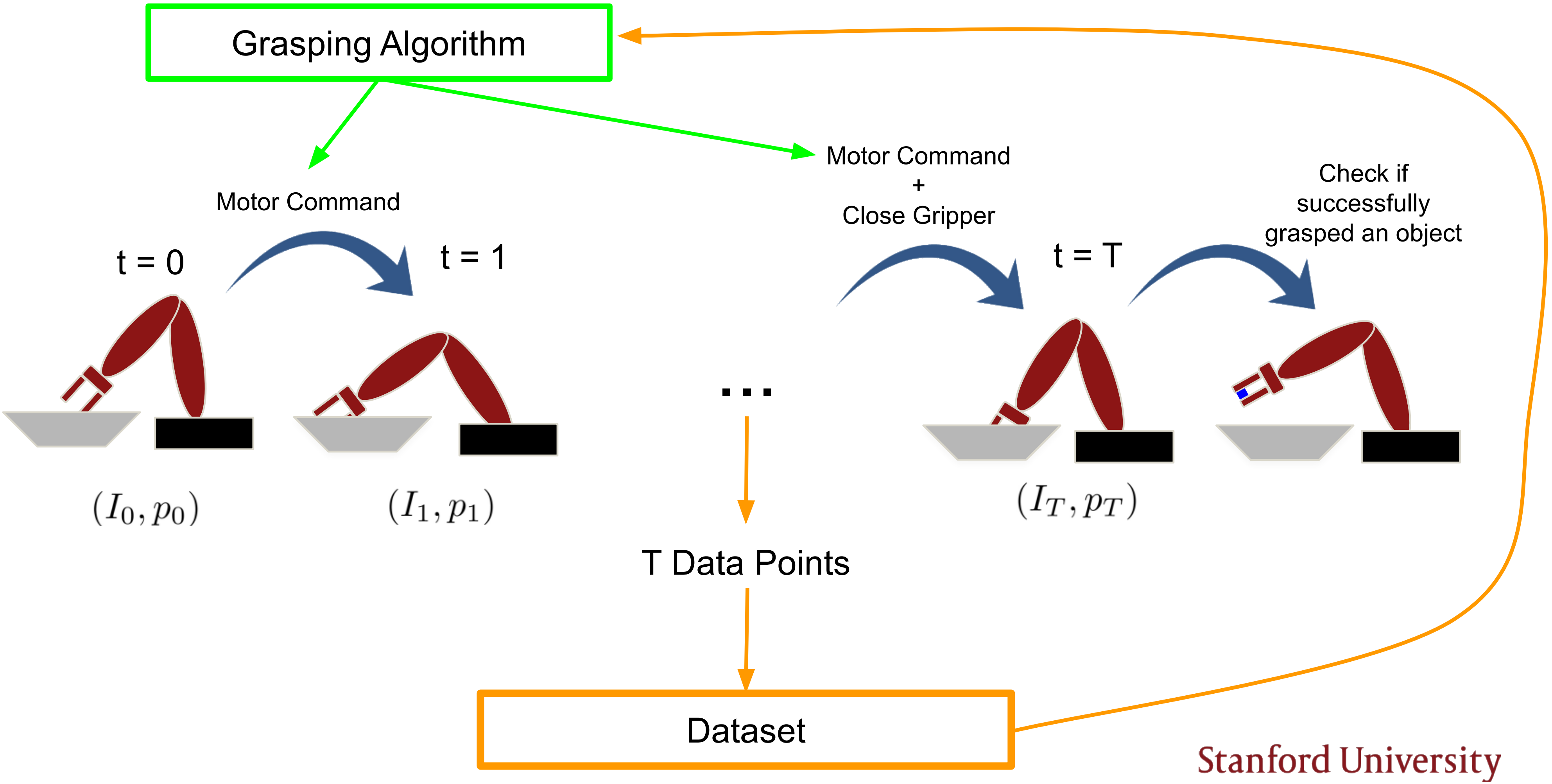
1.7M Grasp Attempts

Self-Supervised Data Collection: Phase 1



Stanford University

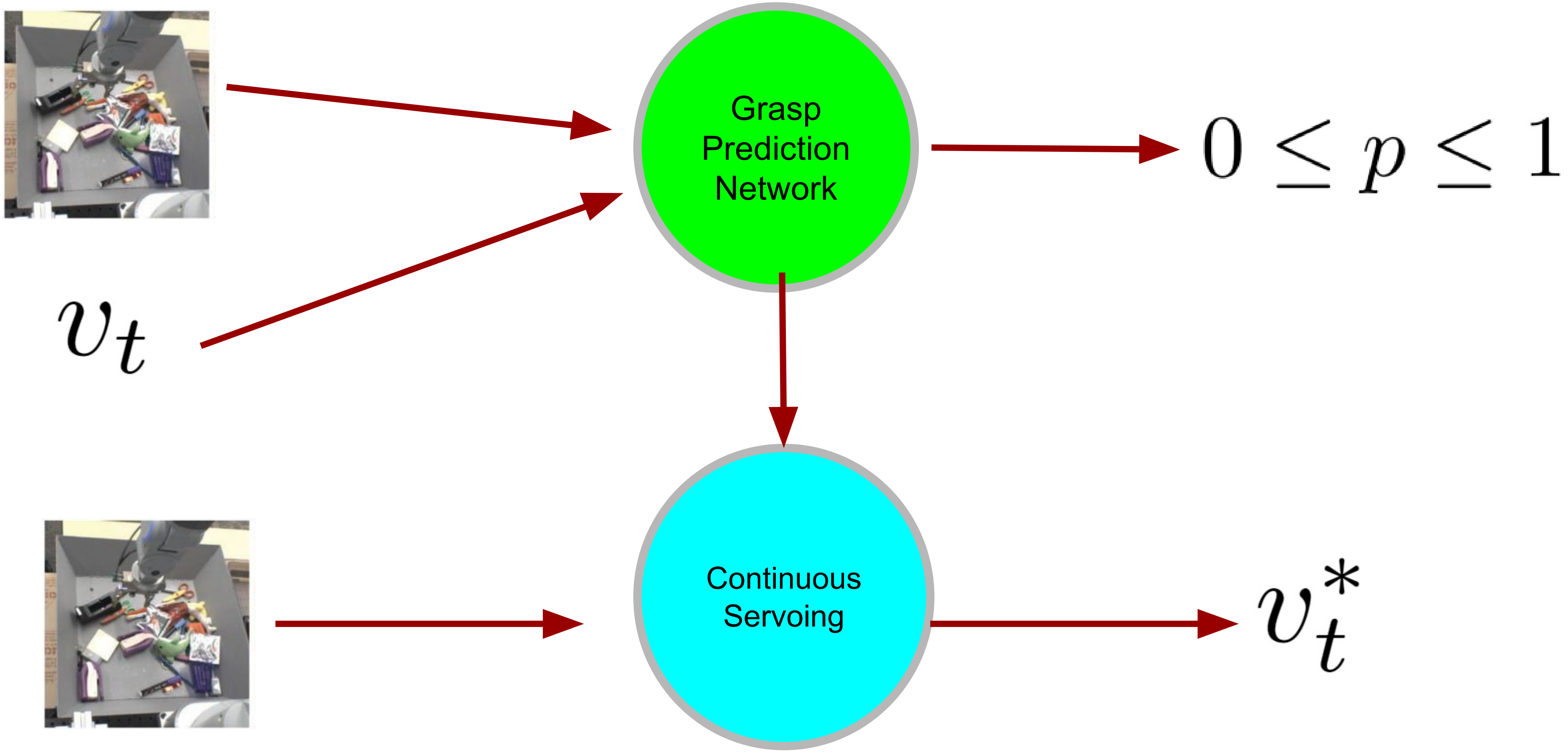
Self-Supervised Data Collection: Phase 2



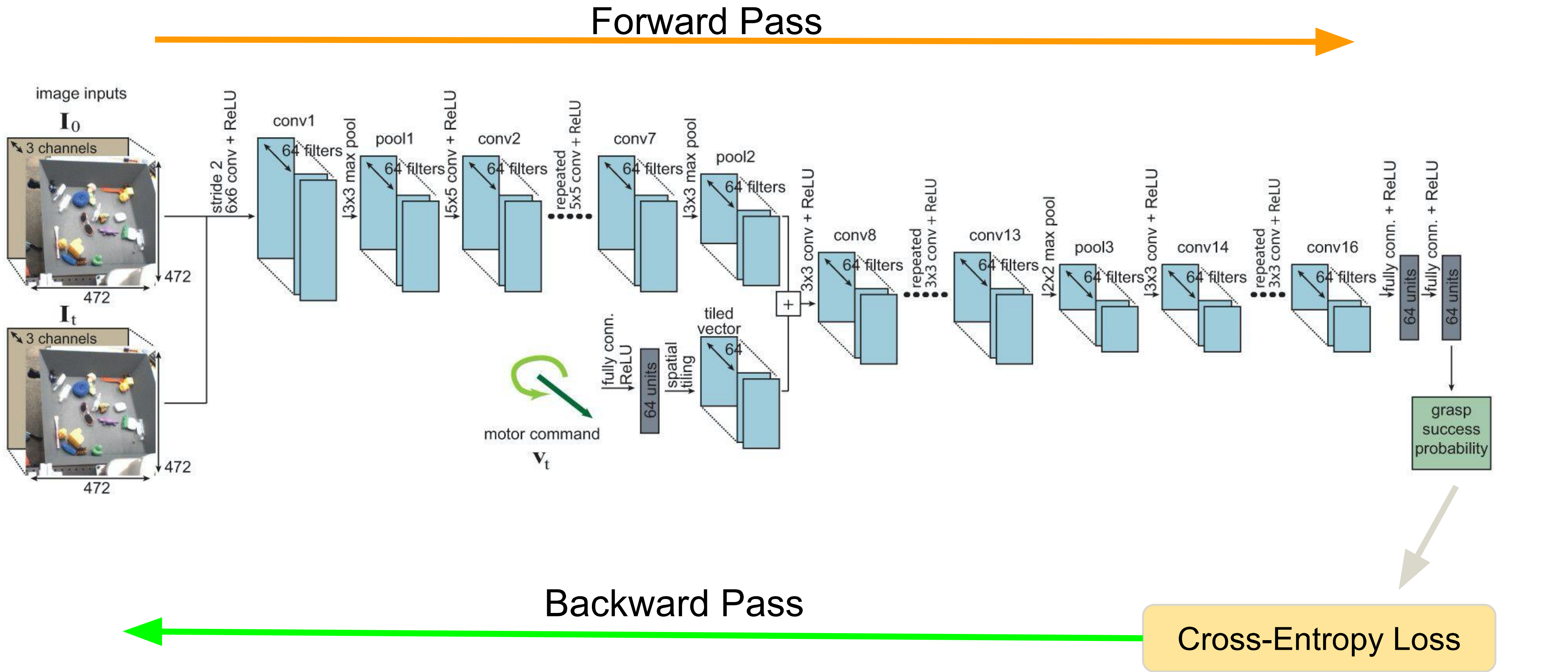
Stanford University

Grasping Algorithm

v_t Task space motion command

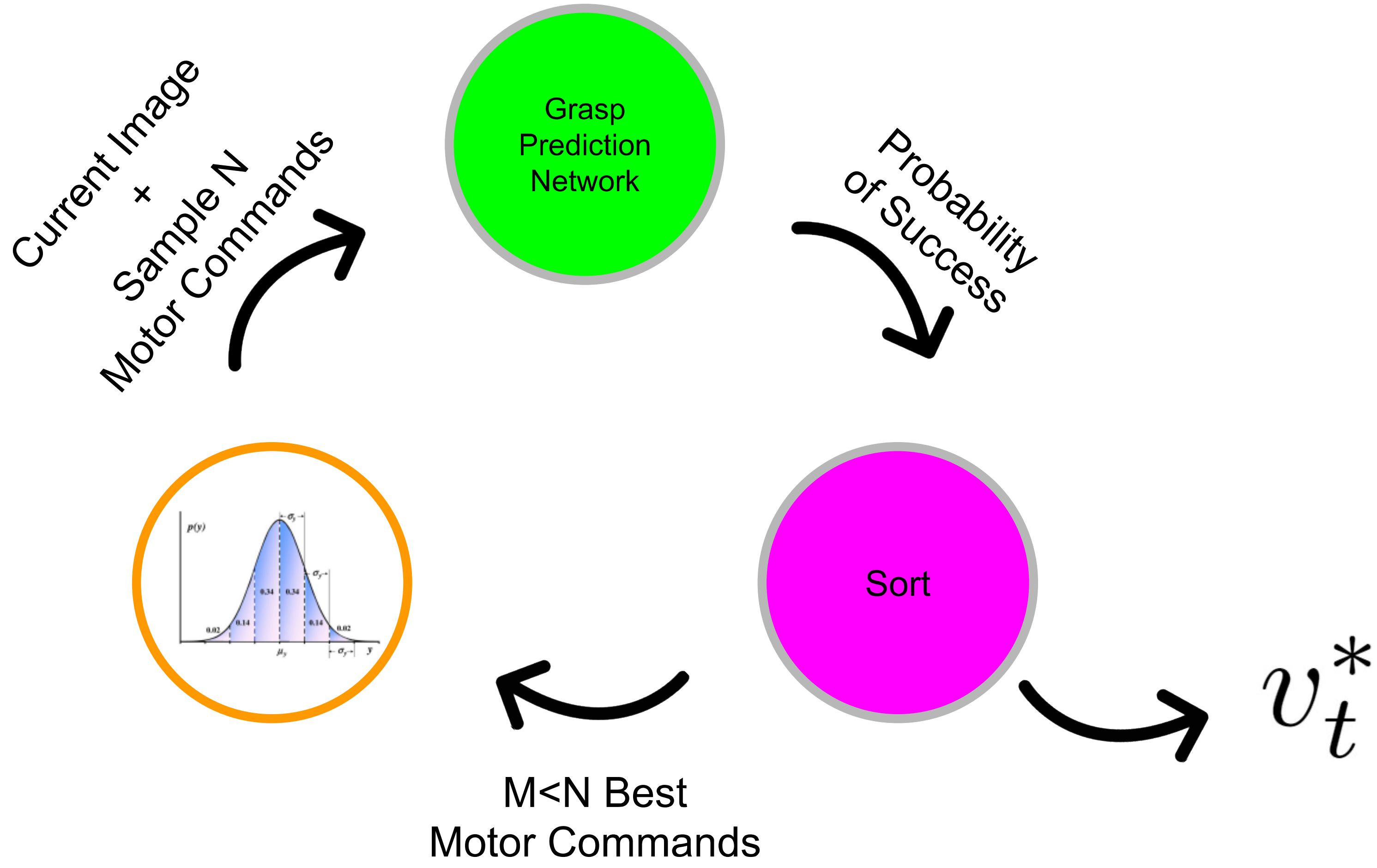


Grasp Prediction Network

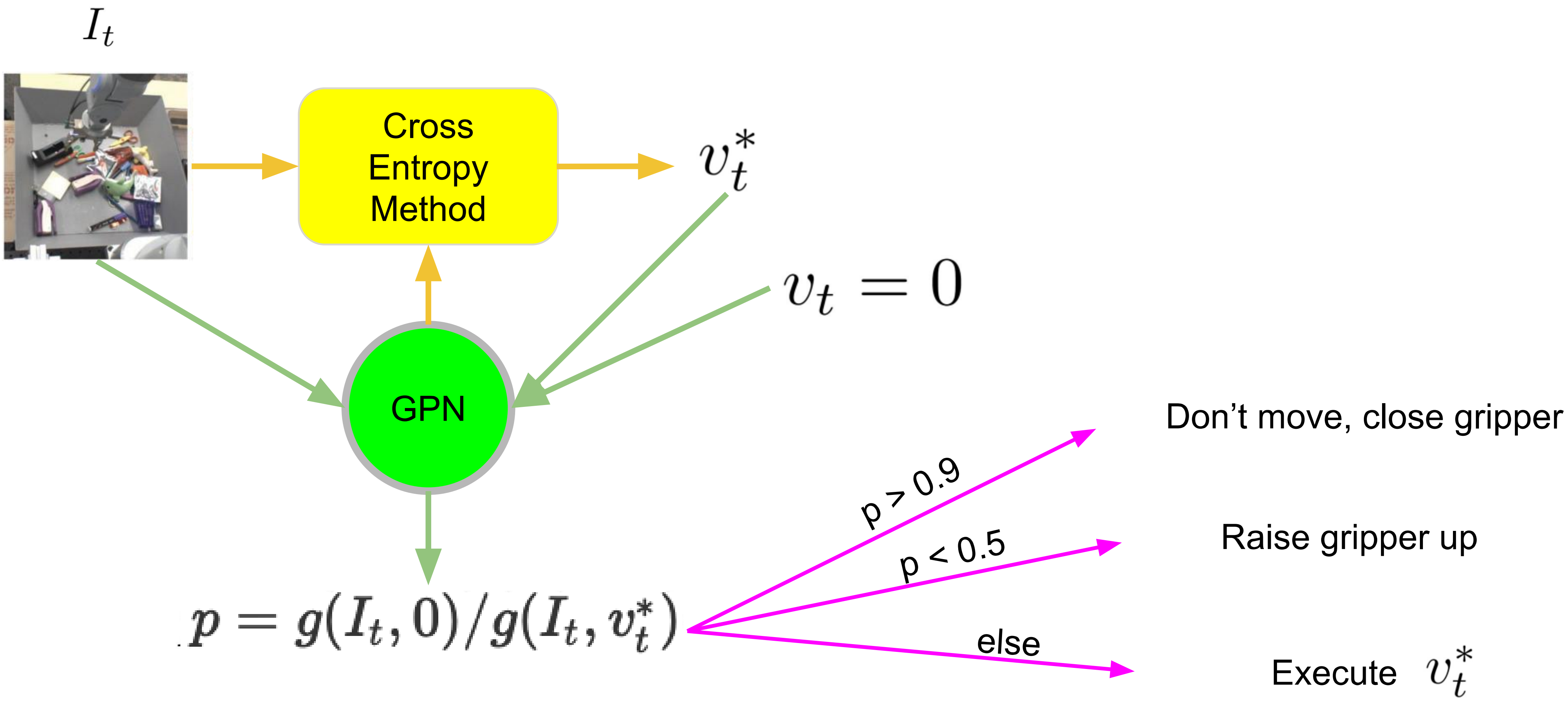


$$-(y\log(p) + (1-y)\log(1-p))$$

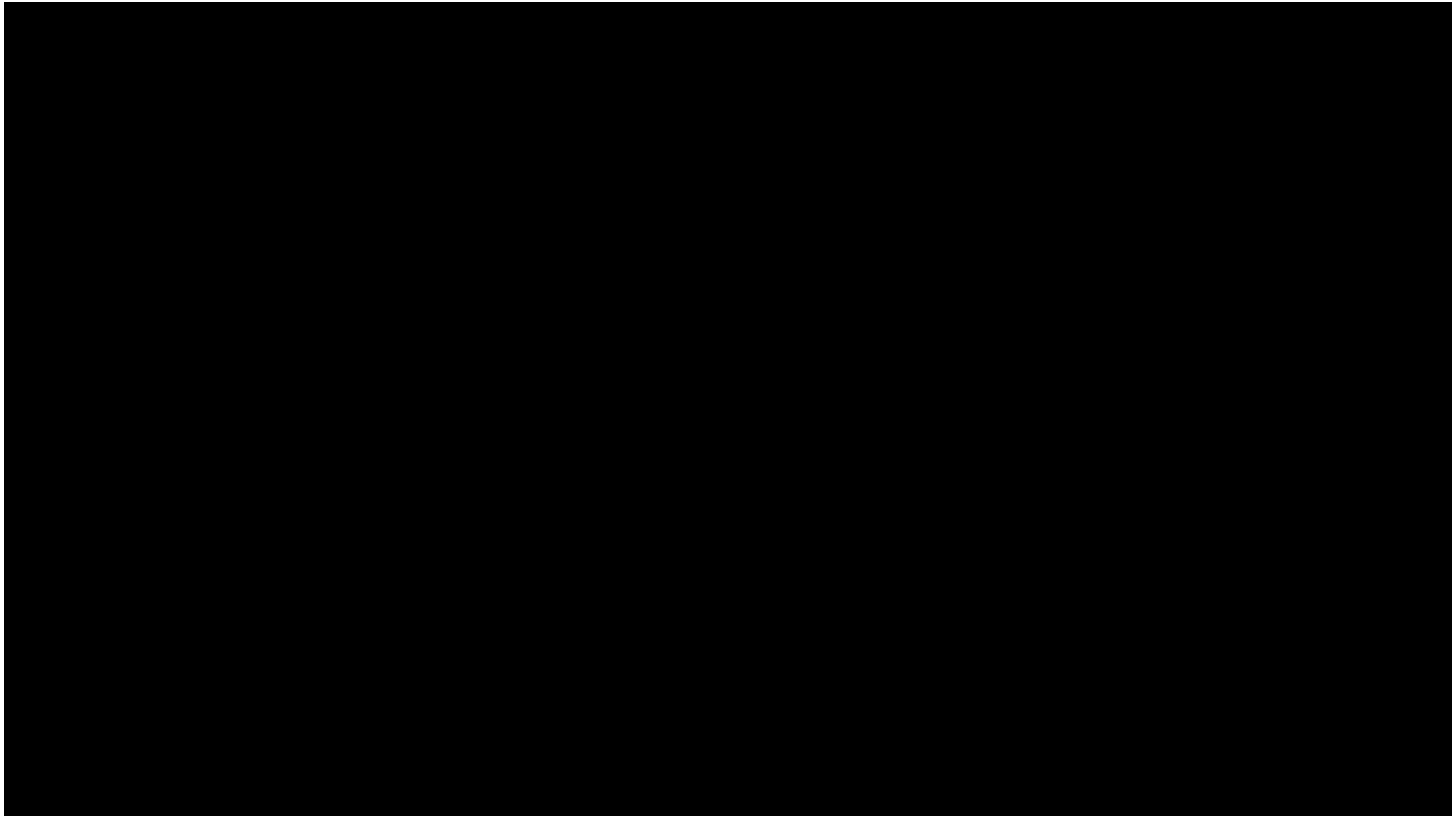
Continuous Servoing: Cross-Entropy Method



Continuous Servoing



Video



Overall Performance: Failure Rate Results

Table 1. Failure rates of each method for each evaluation condition. When evaluating without replacement, we report the failure rate on the first 10, 20, and 30 grasp attempts, averaged over 4 repetitions of the experiment. N indicates the number of grasps used to compute each value. The experiments without replacement were repeated four times.

Without replacement	First 10 ($N = 40$)	First 20 ($N = 80$)	First 30 ($N = 120$)
Random	67.5%	70.0%	72.5%
Hand-designed	32.5%	35.0%	50.8%
Open loop	27.5%	38.7%	33.7%
Our method	**10.0%**	**17.5%**	**17.5%**

With replacement	Failure rate ($N = 100$)
Random	69%
Hand-designed	35%
Open loop	43%
Our method	**20%**

Struggled with clutter (Hand-designed)

Unable to react to objects moving (Open loop)

Performs better and requires fewer assumptions (Our method)

Discussion

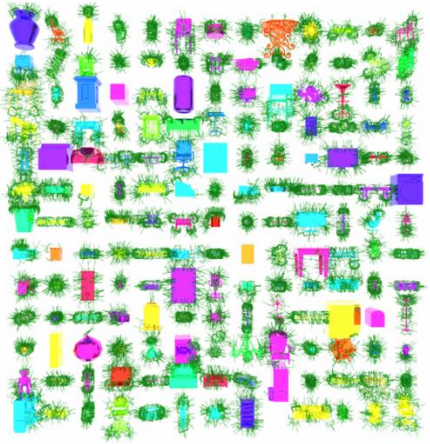
- **End-to-end learning** can achieve good results with **few assumptions**
- It requires **a lot of data** to achieve good performance
  - More tolerable the more **generalizable**
    - Variation in hardware was **small-scale**

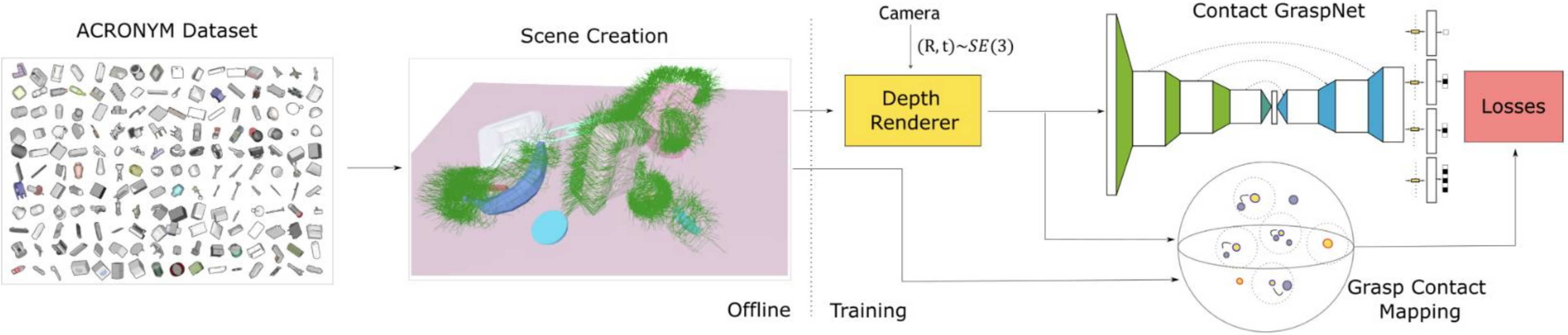
Conclusion: Two Approaches

	Dex-Net	**Arm Farm**
Setup	Single object in simulation	Bin of objects in real world
Number Data Points	13,000 objects, 2.5M grasps	1,100 objects, 1.7M grasps
Data Point	(object, grasp, label = probability of success)	(Image, motor command, label = ground truth success)
Diversity of Objects	Rigid, Opaque	Rigid & deformable, opaque & translucent
Object Representation	3D Mesh Model	None
Data Collection Method	Generated in simulation	Self-supervised on real hardware
Type of Learning	Deep learning, reinforcement learning	End-to-end deep learning

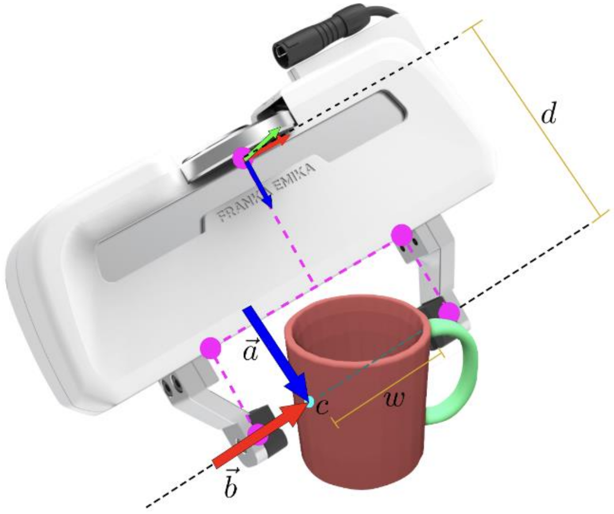
Today

Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes. Sundermeyer et al. ICRA 2021



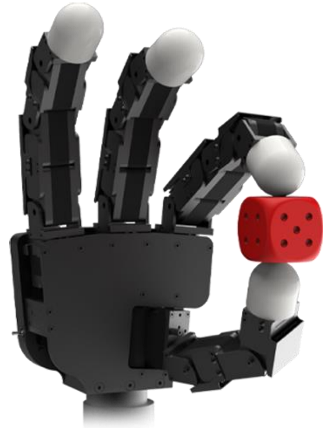


Simulation data is dominating



Today

AnyGrasp: Robust and Efficient Grasp Perception in Spatial and Temporal Domains. By Hao-Shu Fang et al. TRO 2023.



Less work on dexterous grasping

Suggested Reading

- **AnyGrasp: Robust and Efficient Grasp Perception in Spatial and Temporal Domains.** By Hao-Shu Fang et al. TRO 2023.
- **Deep Learning Approaches to Grasp Synthesis: a Review.** By Rhys Newbury et al. TRO 2023.
- **Data-Driven Grasp Synthesis – A survey** by Bohg et al. TRO 2014
- **Robotic Grasping of Novel Objects** by Saxena et al. NeurIPS 2006.
- **Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics** by Mahler et al.. RSS 2017. https://berkeleyautomation.github.io/dex-net
- **Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection** by Levine et al. IJRR 2017.
- **Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes.** Sundermeyer et al. ICRA 2021

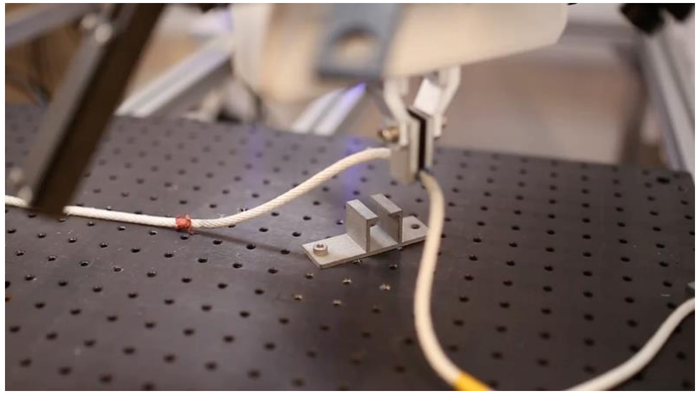
Principles of Robot Autonomy II

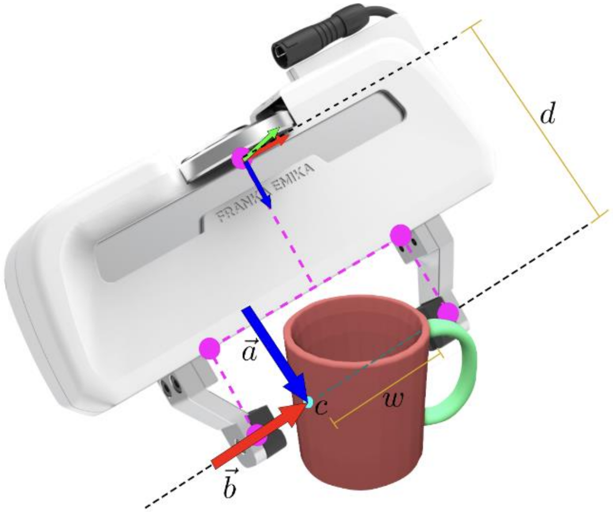
Learning-based Approaches to **Manipulation**

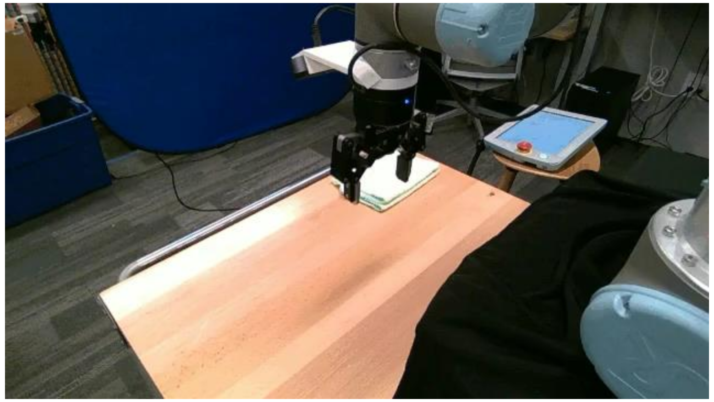
Jeannette Bohg

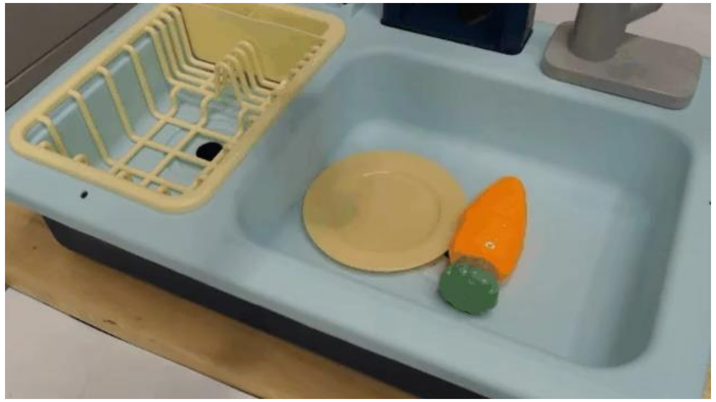
Stanford University

Learning-Based Grasping vs Manipulation

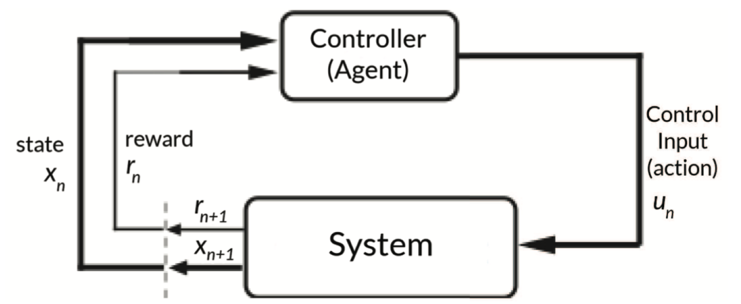


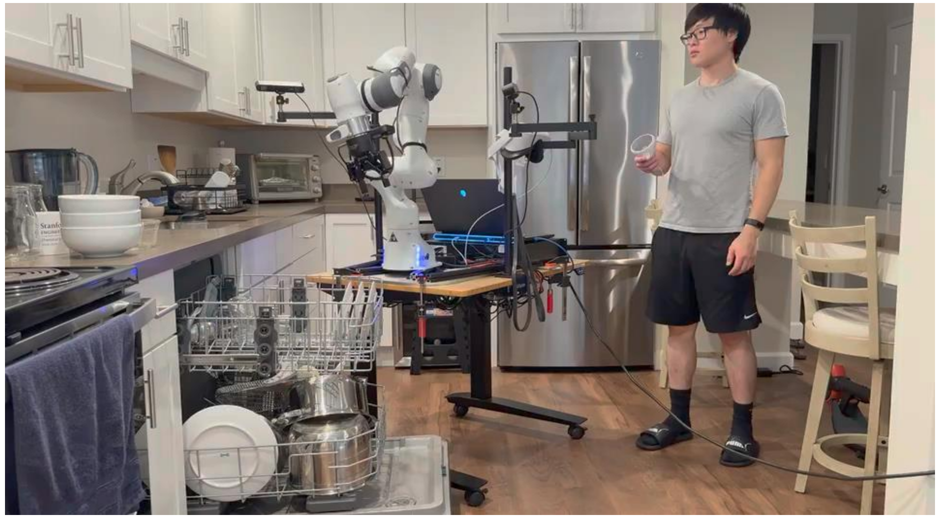






Main Technical Approaches to Learning Manipulation





Reinforcement Learning

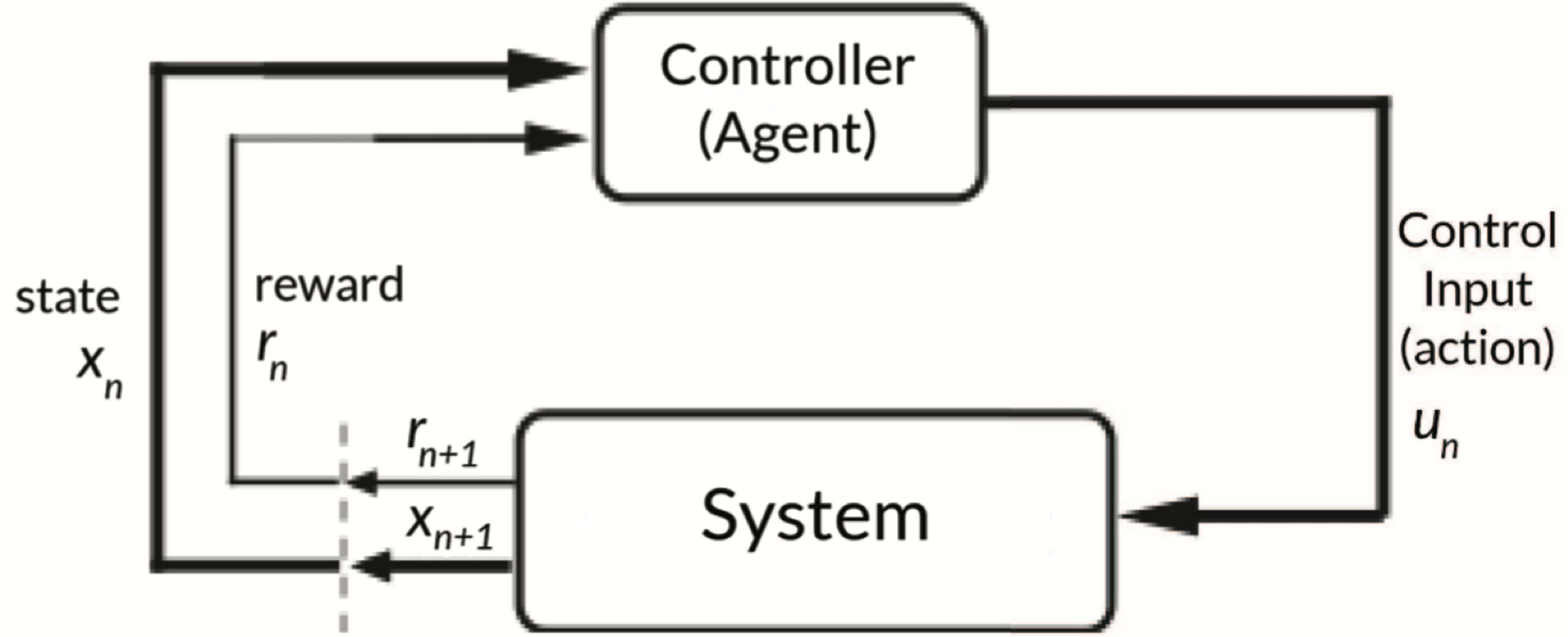
Imitation Learning

State and Action Representations for Manipulation

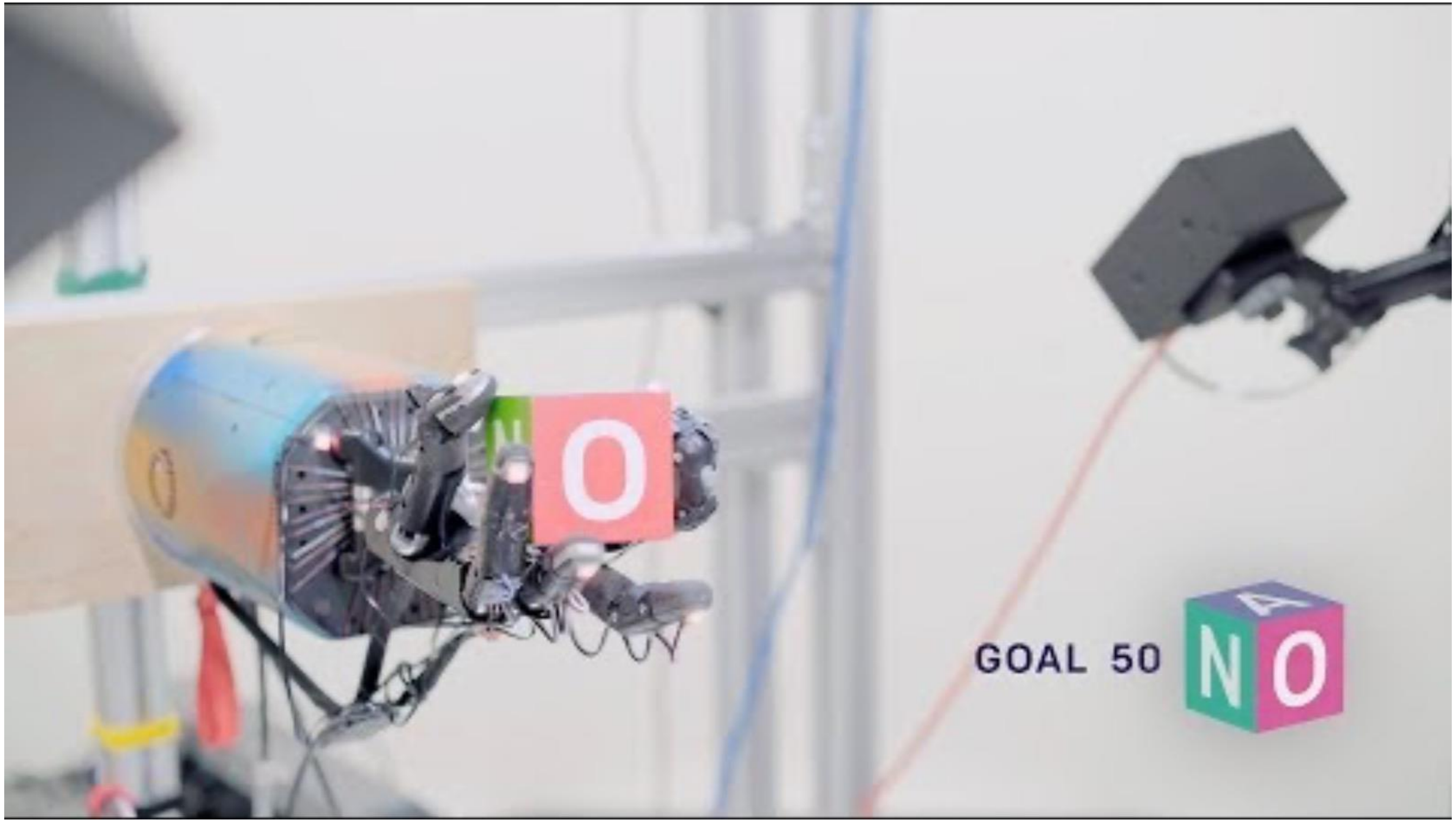
- State Representations
  - Low-dimensional and interpretable
    - Object pose
    - EE pose + finger configuration
    - Robot joint angles, velocities, torques
    - EE forces/Torques
  - High-dimensional, not interpretable
    - Image/Point Cloud encoding in latent space
    - Tactile feedback

- Action Representations
  - EE pose, velocities, or position deltas
  - Trajectories or a set of waypoints
  - Joint angles, velocities, torques

Reinforcement Learning for Manipulation

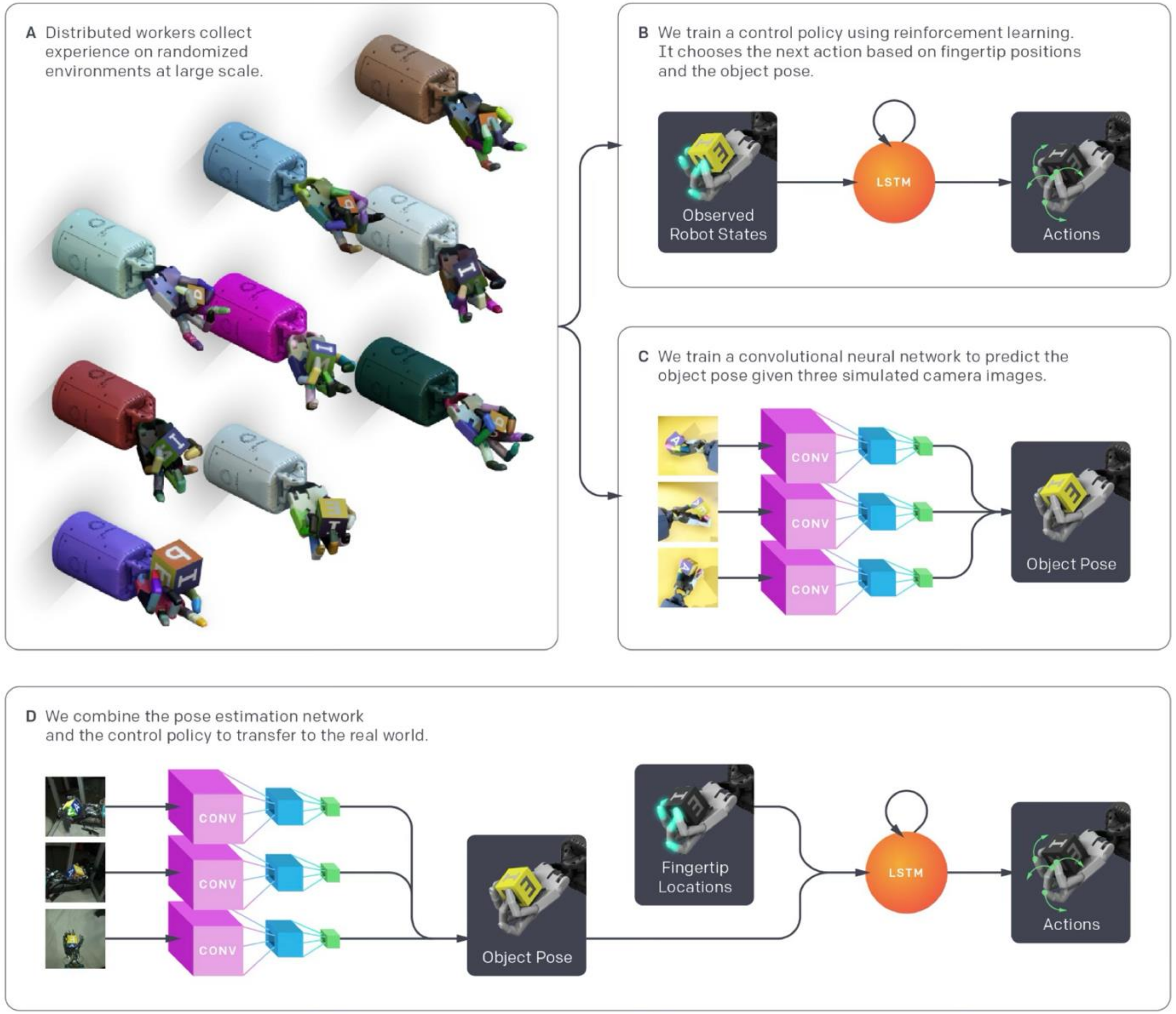


Learning Dexterous In-hand Manipulation OpenAI, IJRR, 2020.



System overview

Training



Domain Randomisation (rendering and physics)
Distributed PPO
Learned object pose estimator
MoCap for fingertips

What stuck: Domain Randomization

Table 1: Ranges of physics parameter randomizations.

Parameter	**Scaling factor range**	**Additive term range**
object dimensions	$\text{uniform}([0.95, 1.05])$	
object and robot link masses	$\text{uniform}([0.5, 1.5])$	
surface friction coefficients	$\text{uniform}([0.7, 1.3])$	
robot joint damping coefficients	$\text{loguniform}([0.3, 3.0])$	
actuator force gains (P term)	$\text{loguniform}([0.75, 1.5])$	
joint limits		$\mathcal{N}(0, 0.15)$ rad
gravity vector (each coordinate)		$\mathcal{N}(0, 0.4)$ m/s^2

Figure 4: Simulations with different randomized visual appearances. Rows correspond to the renderings from the same camera, and columns correspond to renderings from 3 separate cameras which are simultaneously fed into the neural network.

Results

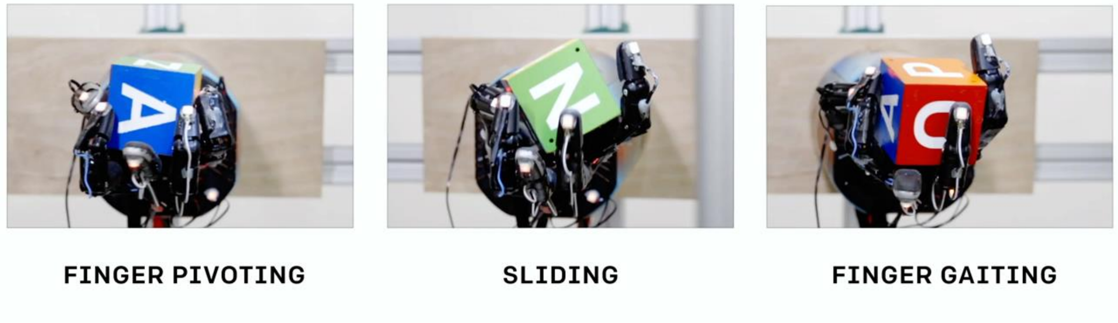
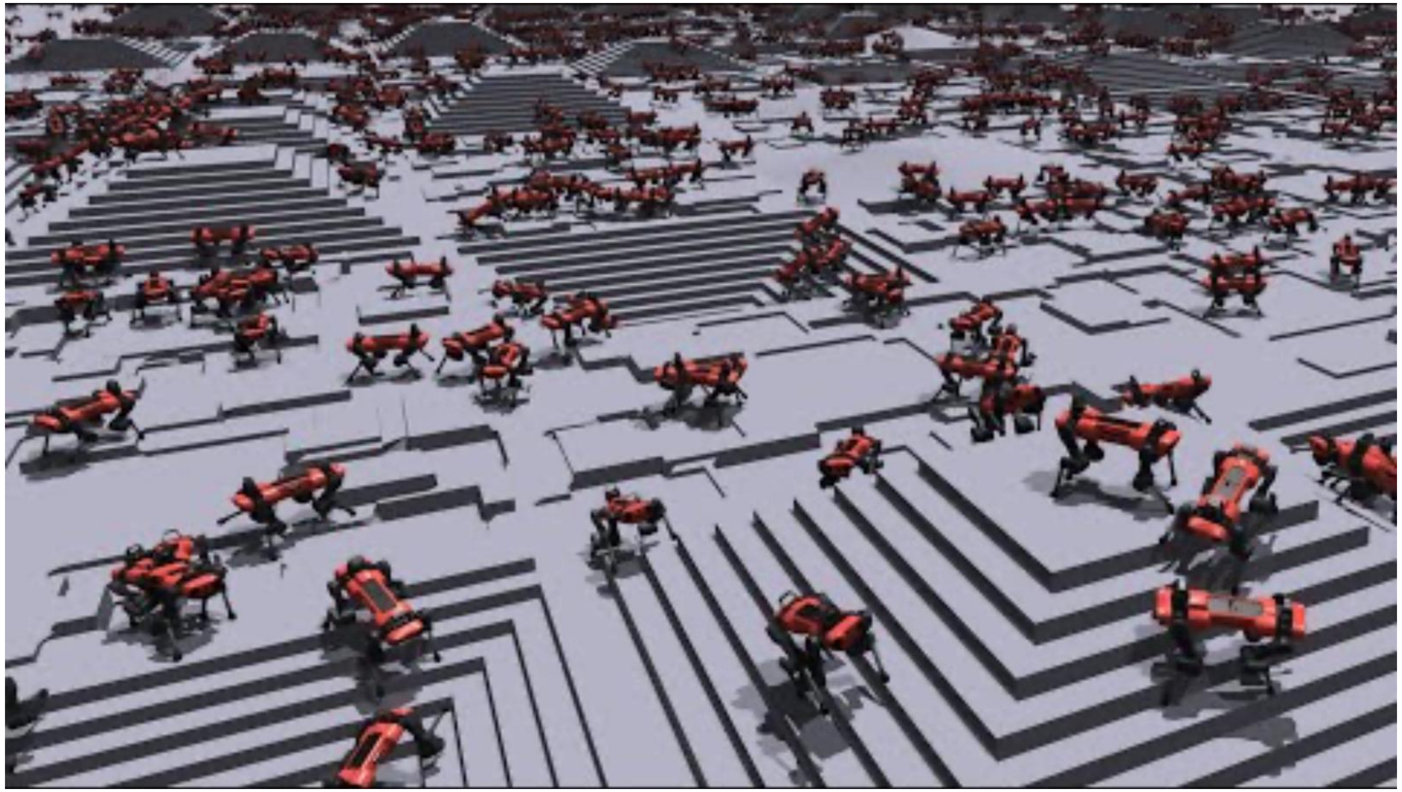


Table 3: The number of successful consecutive rotations in simulation and on the physical robot. All policies were trained on environments with all randomizations enabled. We performed 100 trials in simulation and 10 trails per policy on the physical robot. Each trial terminates when the object is dropped, 50 rotations are achieved or a timeout is reached. For physical trials, results were taken at different times on the physical robot.

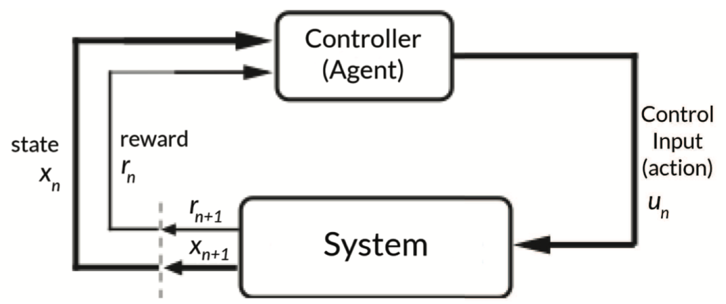
Simulated task	**Mean**	**Median**	**Individual trials (sorted)**
Block (state)	43.4 ± 13.8	50	-
Block (state, locked wrist)	44.2 ± 13.4	50	-
Block (vision)	30.0 ± 10.3	33	-
Octagonal prism (state)	29.0 ± 19.7	30	-
Physical task			
Block (state)	18.8 ± 17.1	13	50, 41, 29, 27, 14, 12, 6, 4, 4, 1
Block (state, locked wrist)	26.4 ± 13.4	28.5	50, 43, 32, 29, 29, 28, 19, 13, 12, 9
Block (vision)	15.2 ± 14.3	11.5	46, 28, 26, 15, 13, 10, 8, 3, 2, 1
Octagonal prism (state)	7.8 ± 7.8	5	27, 15, 8, 8, 5, 5, 4, 3, 2, 1

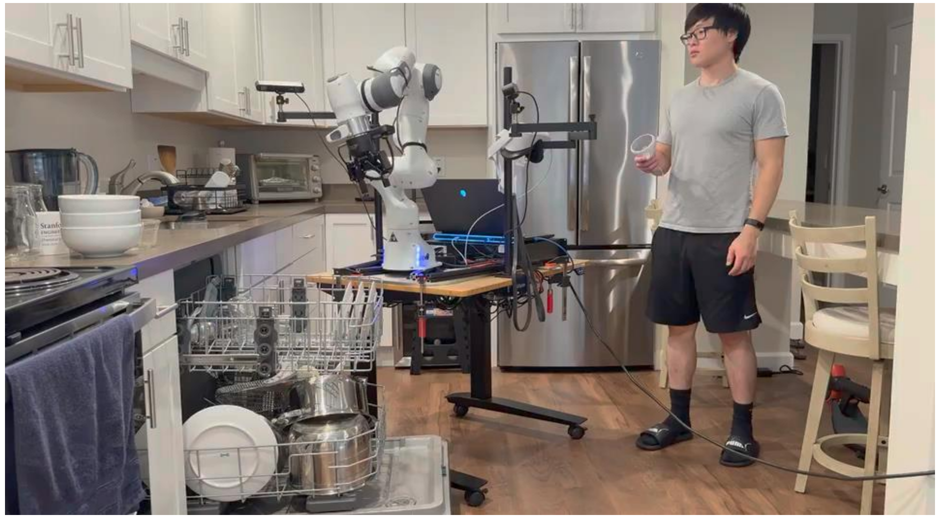
The impact on Locomotion

- Successful Sim2Real RL in quadruped and more recently humanoid locomotion
- Learning to walk in minutes using Massively Parallel Deep RL. Rudin et al. CoRL'21



Main Technical Approaches to Learning Manipulation



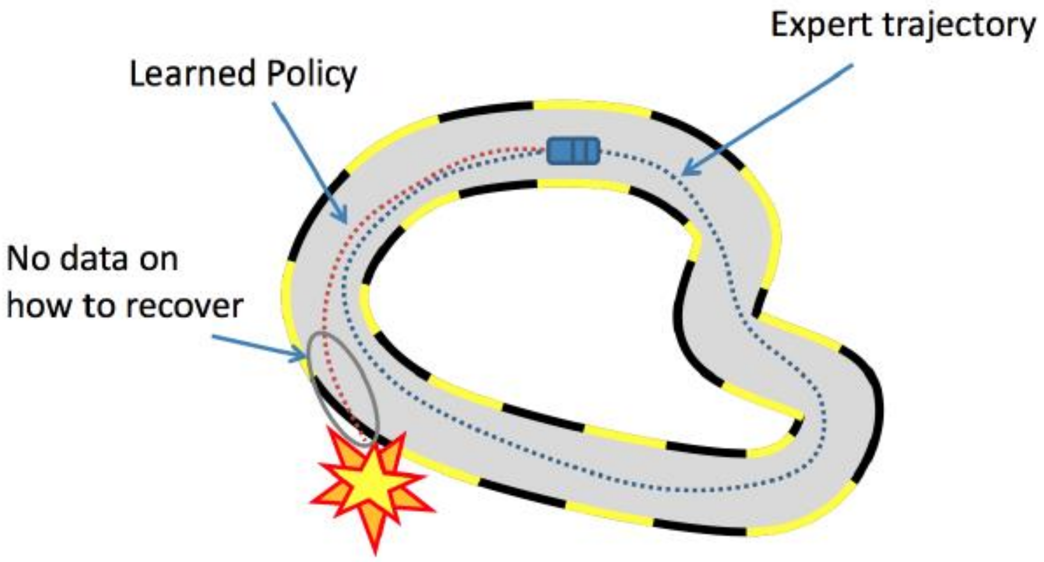


Reinforcement Learning

Imitation Learning

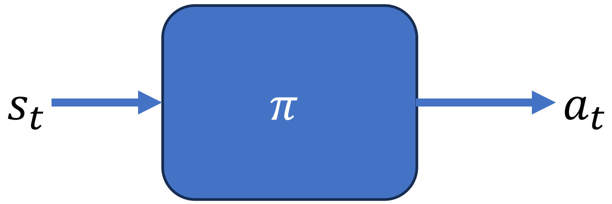
Imitation learning

- Turning the learning problem into a supervised problem
- Estimate a policy from training examples (s_0, a_0), (s_1, a_1) (s_2, a_2)

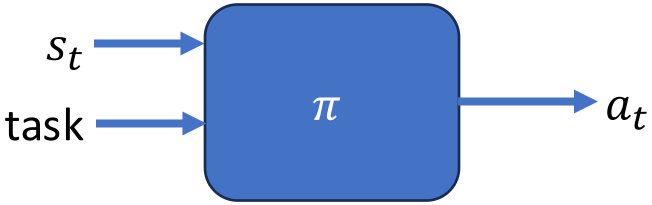


From CS237: Reinforcement Learning

Two flavors of imitation learning

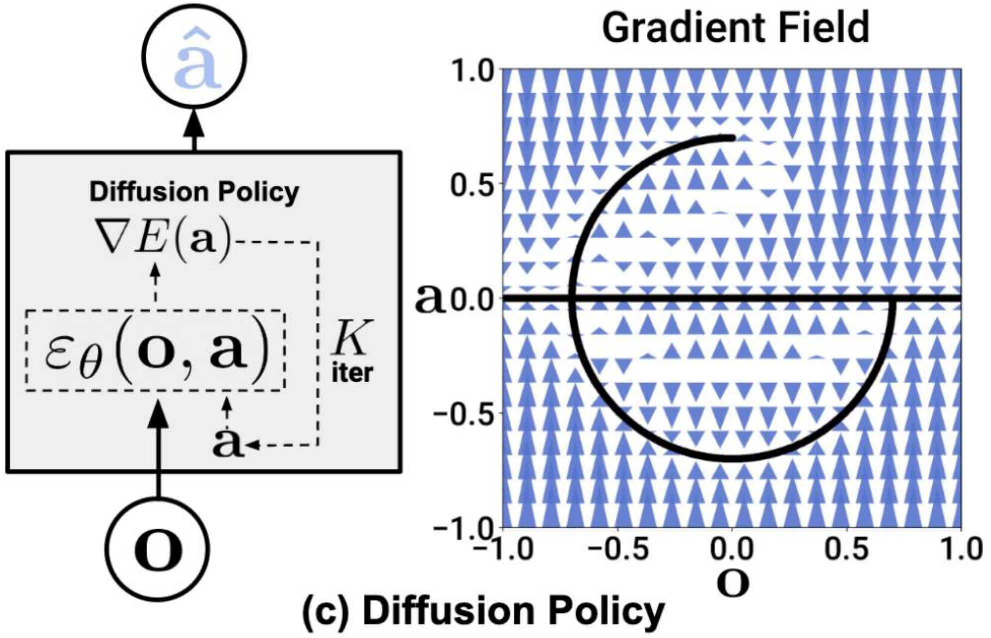


Single task policy

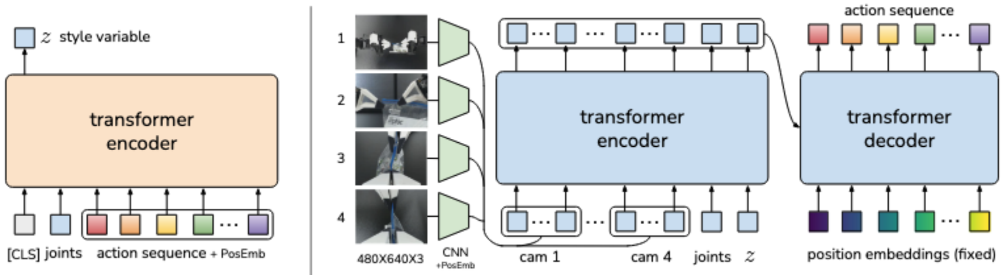


Multi-task (Generalist) policy

Single-task policies

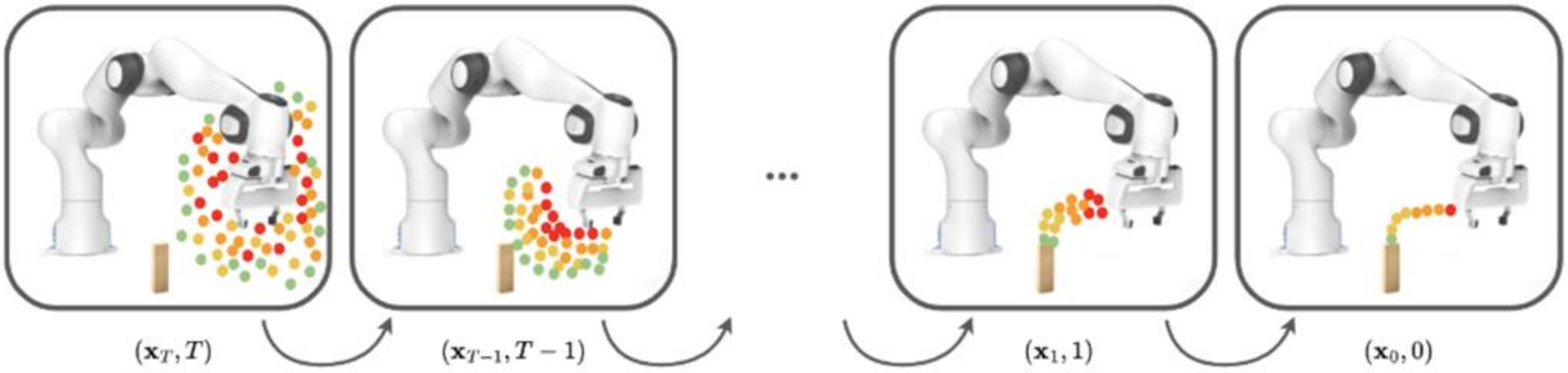


Diffusion Policies

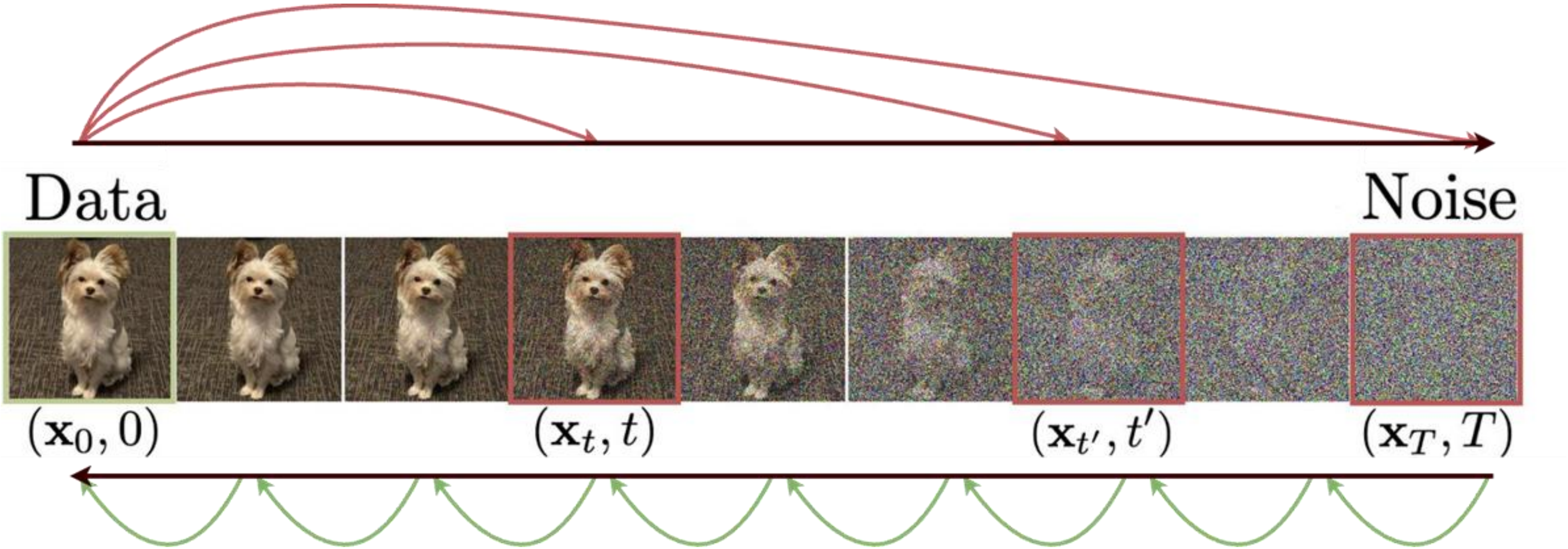


Action Chunking with Transformers

Diffusion policy

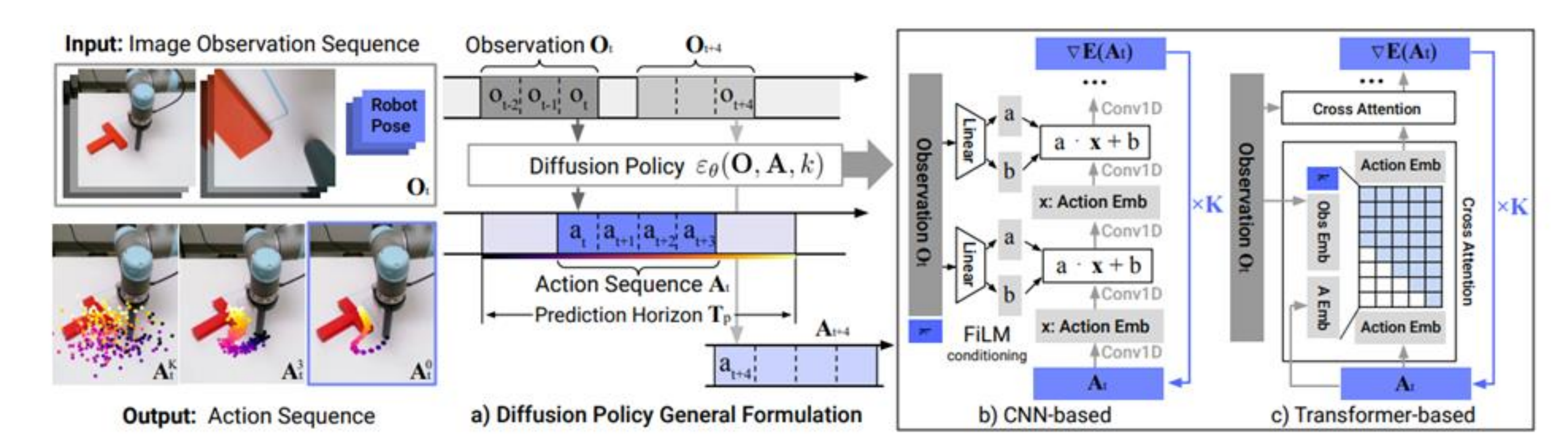


Idea behind Diffusion from Computer Vision



Diffusion Policy

1. Diffuse through actions, condition on observations
2. Chunk observations and actions for smoother control
3. Pro: SOTA success rates, continuous action space, multimodality
4. Con: Inference takes a lot of time b/c of denoising process



Next time

- Wrapping up Learning-based Manipulation
- Interactive Perception