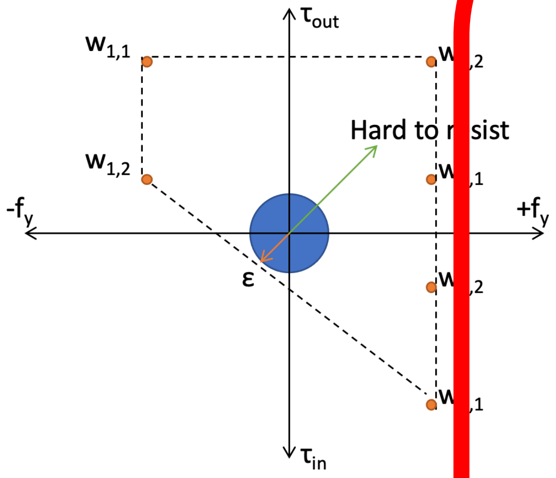
Principles of Robot Autonomy II

Learning-based Approaches to **Grasping and Manipulation**

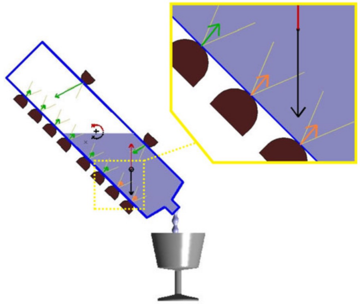
Jeannette Bohg

Stanford University

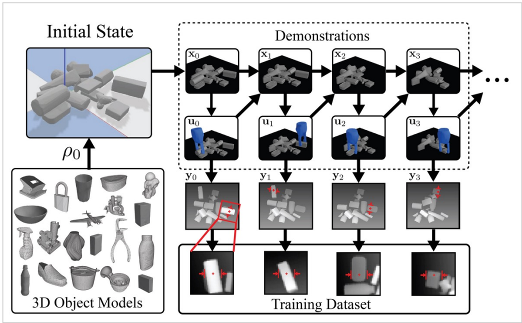
Learning Outcome for next four Lectures



Modeling and Evaluating Grasps



Modeling and Executing Manipulation



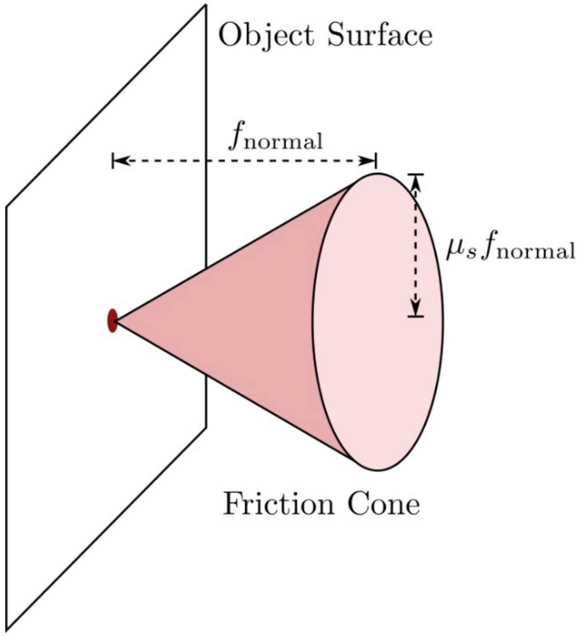
Apply Learning to Grasping and Manipulation

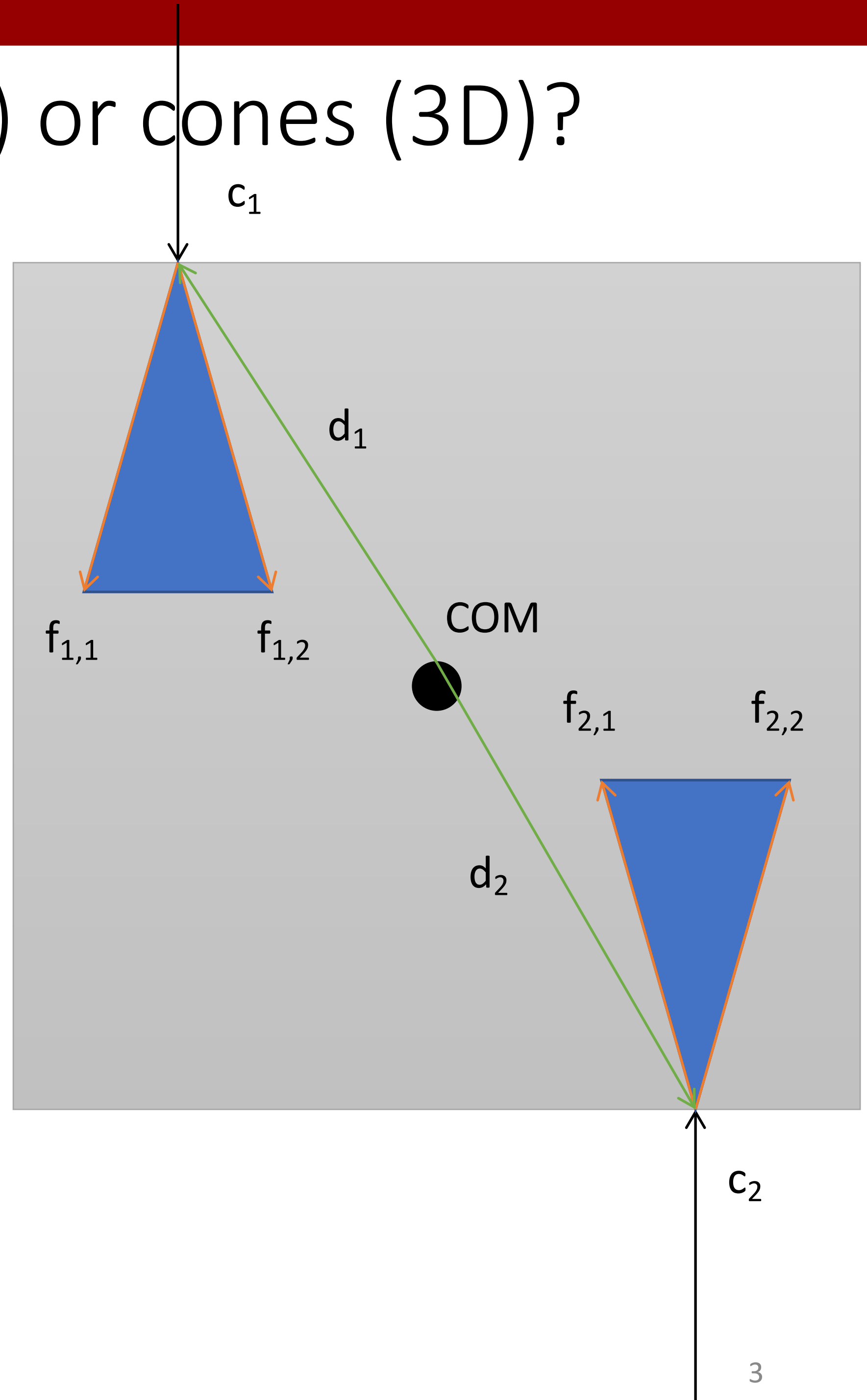
Use Manipulation to Perceive better

Why are friction cones triangles (2D) or cones (3D)?

$$\boldsymbol{f} = \boldsymbol{f}_{\text{normal}} + \boldsymbol{f}_{\text{tangent}},$$

$$\mathcal{F} = \{\boldsymbol{f} \mid \|\boldsymbol{f}_{\text{tangent}}\| \leq \mu_s \|\boldsymbol{f}_{\text{normal}}\|, \quad f_z \geq 0\}.$$





Grasp Force Optimization

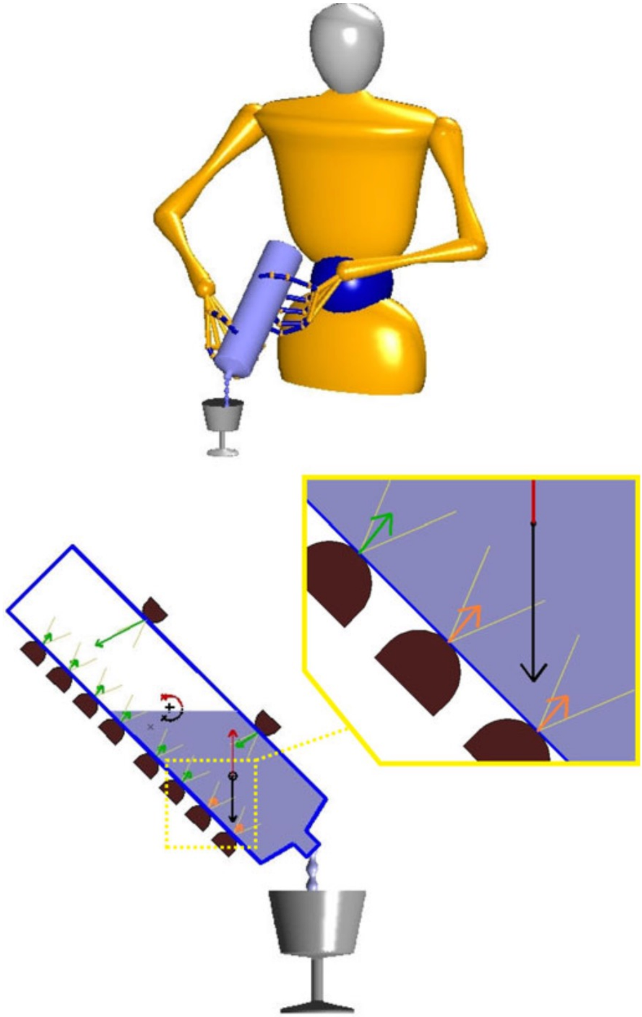


Fig. 3. Sequence of significant configurations of the bottle and of the forces during task execution with $n = 10$.

Figure adapted from *A Grasping Force Optimization Algorithm for Multiarm Robots With Multifingered Hands*. Lipiello et al. Transactions on Robotics. 2013

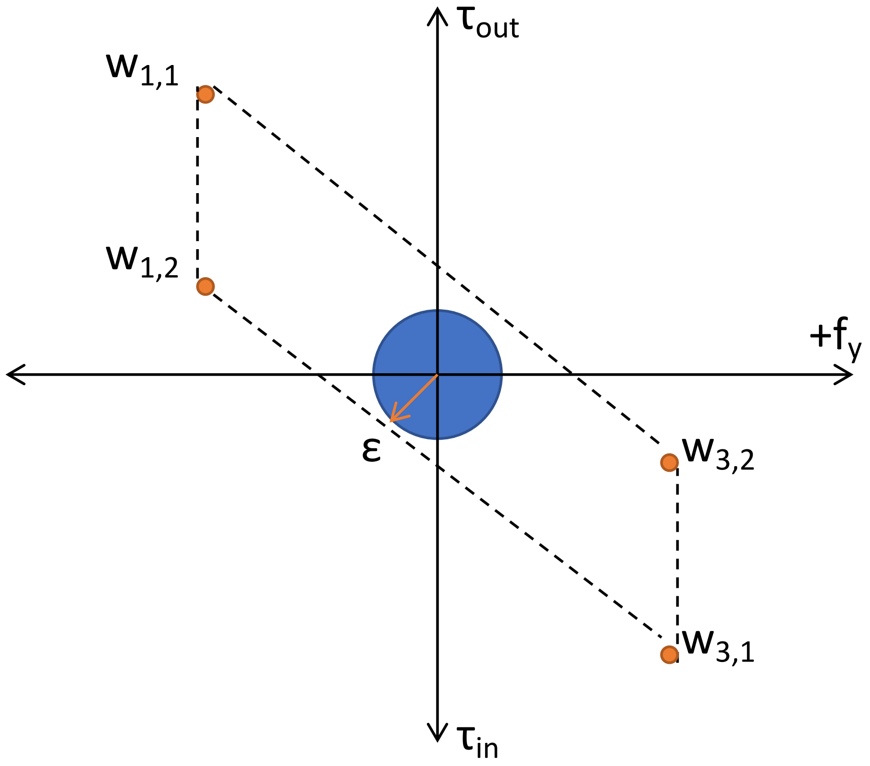
Equilibrium Constraints – Force Closure

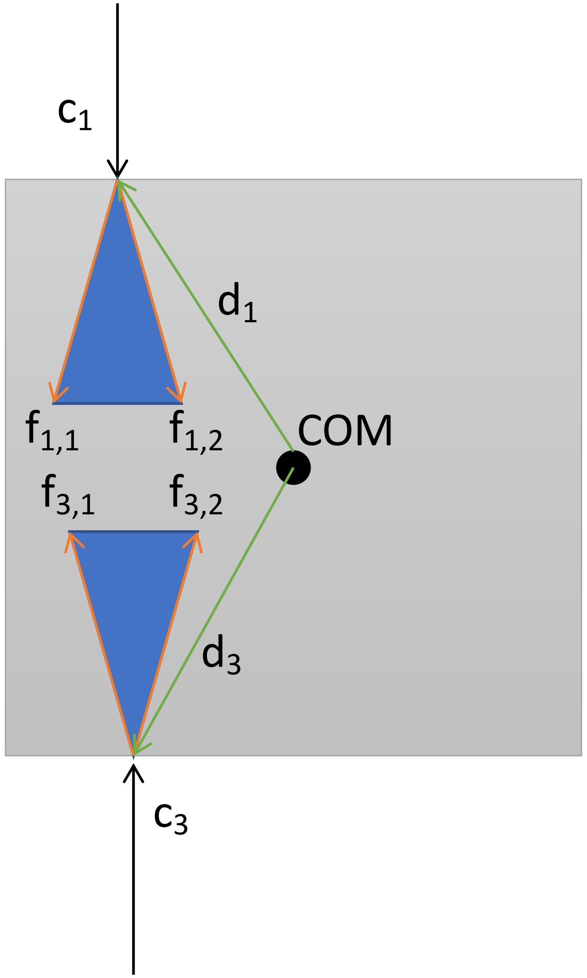
Compact notation

- Contact force vector $f \in \boldsymbol{R}^{3M}$

$$f = (f^{(1)}, \ldots, f^{(M)})$$

- Contact Matrices $G_i \in \boldsymbol{R}^{6x3}$
  - $G_i = \begin{matrix} Q^{(i)} \\ S^{(i)}Q^{(i)} \end{matrix}, i = 1 \ \ldots M$
- Grasp matrix
  - $G = [G_1, \ldots, G_M] \in \boldsymbol{R}^{6x3M}$
- External Wrench $\omega^{ext} = (f^{ext}, \tau^{ext})$
- Equilibrium conditions
  - $Gf + \omega^{ext} = 0$





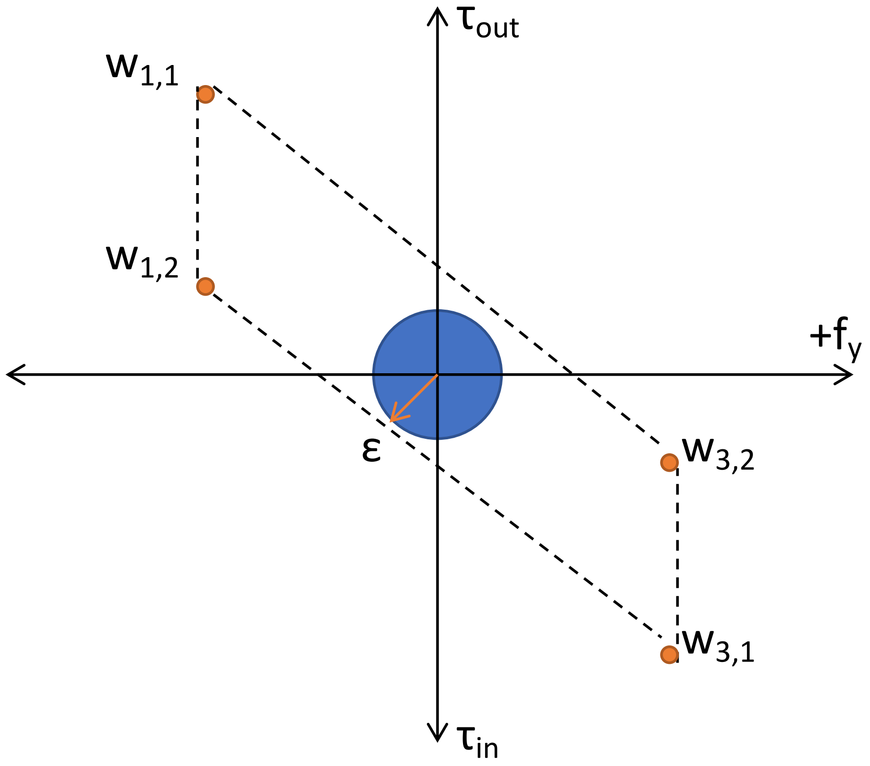
Following Approach in *Fast Computation of Optimal Contact Forces* by Stephen P. Boyd and Ben Wegbreit. Transactions on Robotics. 2007.

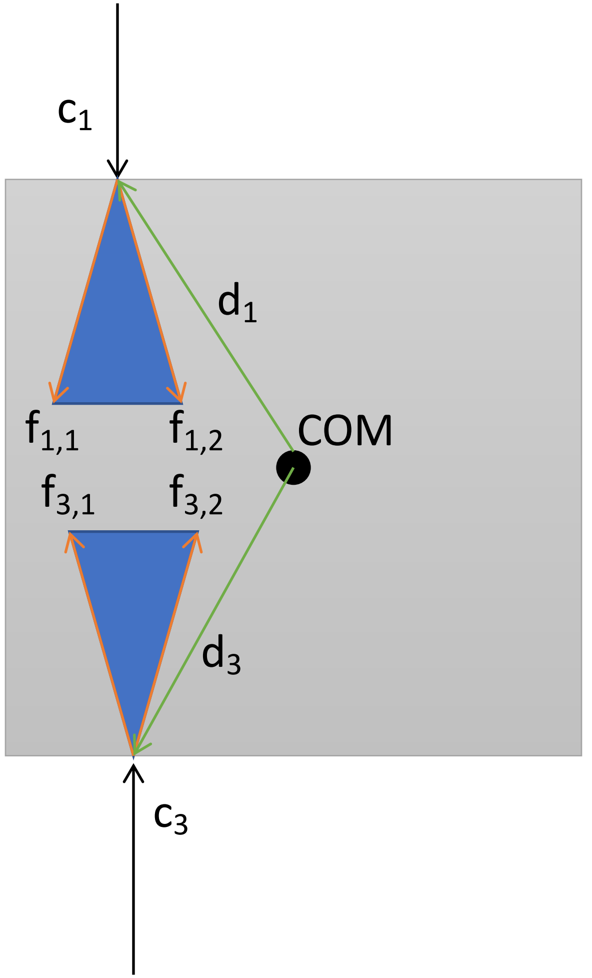
Convex Optimization Problem

- Second-order cone program because friction cones are quadratic.
- Objective function:

$$F^{\max} = \max\{\|f^{(1)}\|, \ldots, \|f^{(M)}\|\}$$
$$= \max_{i=1,\ldots,M} \sqrt{f_x^{(i)2} + f_y^{(i)2} + f_z^{(i)2}}$$

- Optimization problem:
  - *minimize* F^{max}
  - *subject to* $f^{(i)} \in K_i, i = 1 \ldots M$
  - $Gf + \omega^{ext} = 0$





Following Approach in *Fast Computation of Optimal Contact Forces* by Stephen P. Boyd and Ben Wegbreit. Transactions on Robotics. 2007.

Today's itinerary

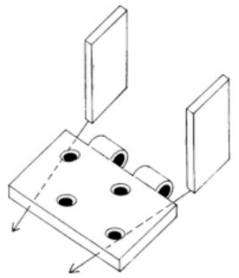
- Modeling Push/Non-Prehensile Manipulation
- Learning-based Approaches to
  - Grasping
  - Planar Pushing
  - Manipulation (Guest Lecture Feb 21 by Quan Vuong from Google DeepMind)

For a Deeper Dive into Grasping and Manipulation

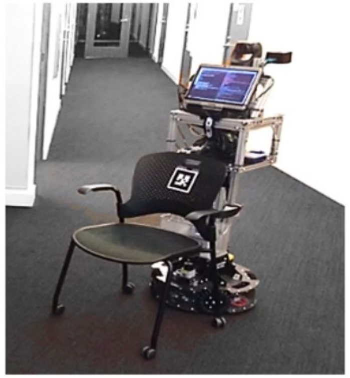
- CS326 – Topics in Advanced Robotic Manipulation – Fall 2024

Case Study – Planar Pushing

$$\arg\min_{u(t)} \quad h(x(T)) + \int_0^T g(x(t), u(t))\, dt$$

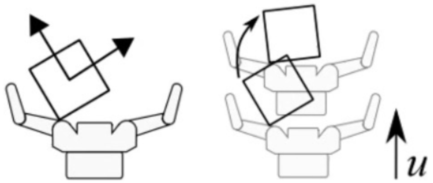


Reorient parts
- Mason 1986



Transport **large objects**
- Meriçli 2015

Push-grasp under **clutter**
- Dogar 2010



Track object **pose**
- Koval 2015

Stable Pushes to manoeuvre an object around obstacles. Adopted from Chapter 37, Fig 37.11 in Springer Handbook of Robotics.

Modeling Planar Pushing

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

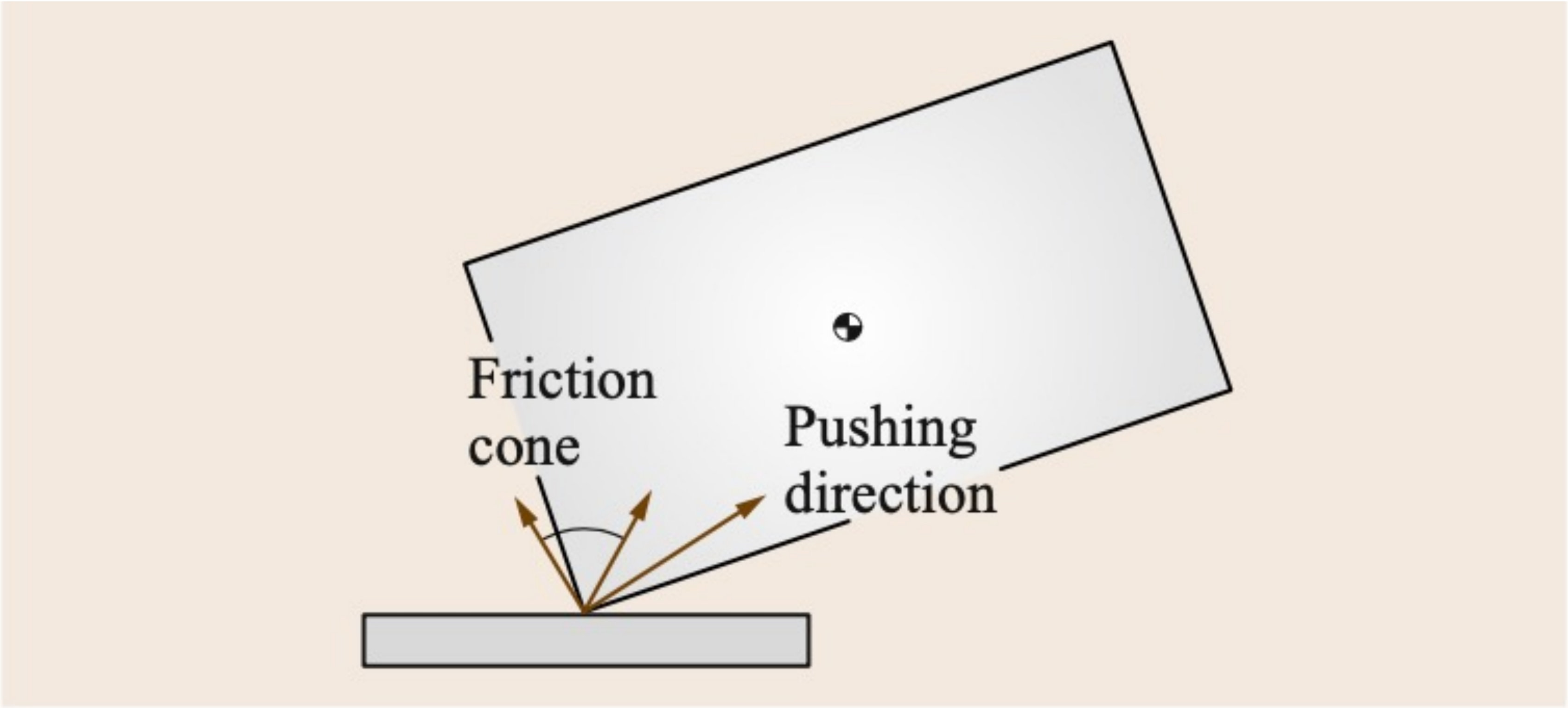
When pushed with a wrench within the limit surface: **no motion.**

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

If **object translates without rotation** the friction force magnitude $\mu m g$ where μ = friction coefficient, m = object mass, g = gravitational acceleration

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

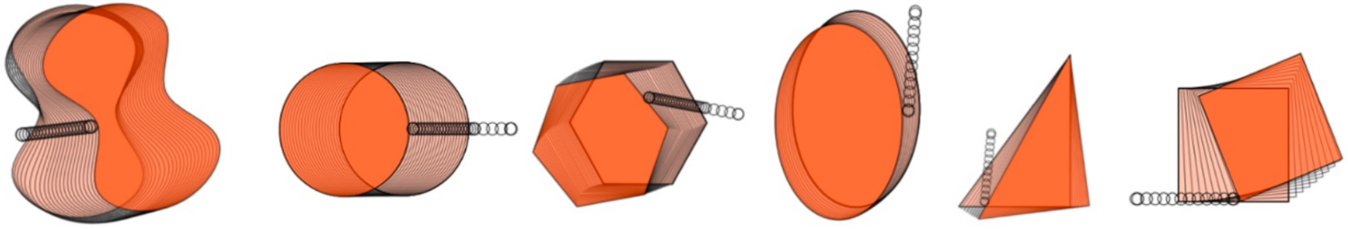
Modeling Planar Pushing –Voting theorem

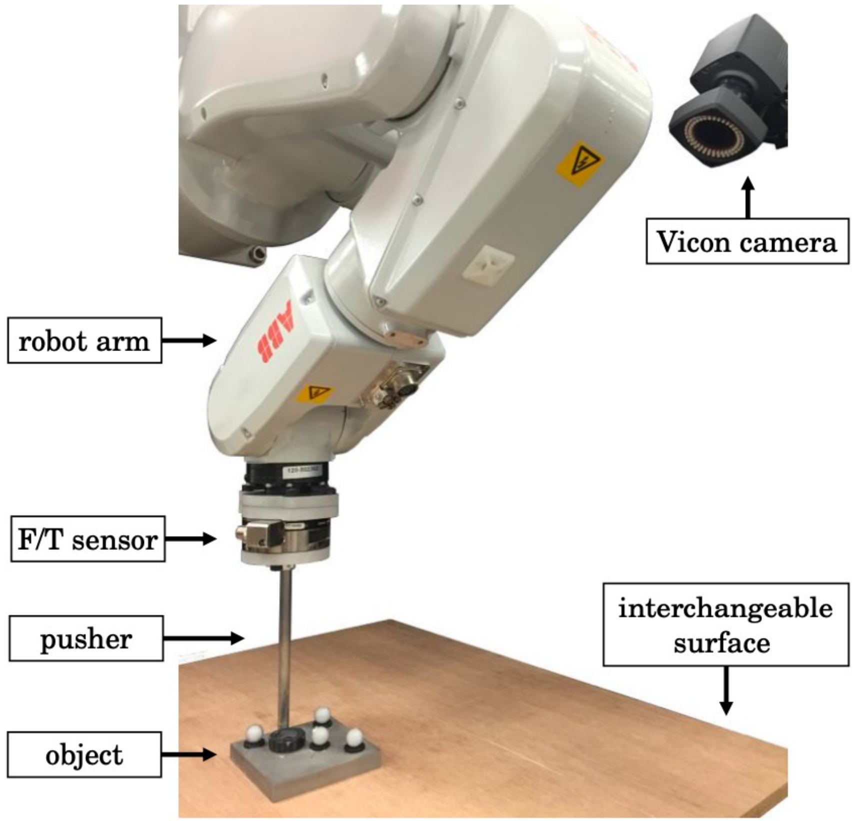


How will the object rotate? Adopted from Chapter 37, Fig 37.12 in Springer Handbook of Robotics.

Combining learned and analytical models for predicting action effects from sensory data . Kloss et al. 2020. IJRR 2020.
K. M. Lynch, H. Maekawa, and K. Tanie, “Manipulation and active sensing by pushing using tactile feedback.” in *IROS*, 1992.

Validating Models for Planar Pushing



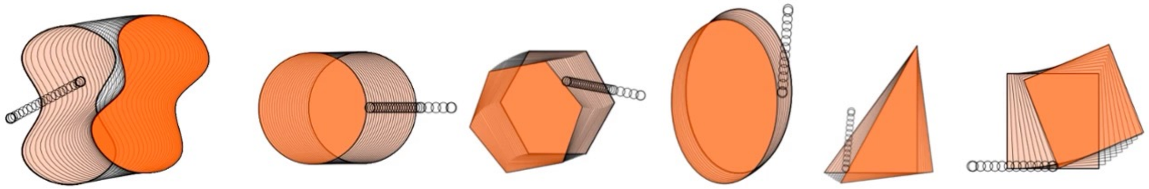


IROS 2016, "More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing" by Peter Yu, Maria Bauza et al.

Validating Models for Planar Pushing

More than a Million Ways to Be Pushed.

A High-Fidelity Experimental Dataset of Planar Pushing



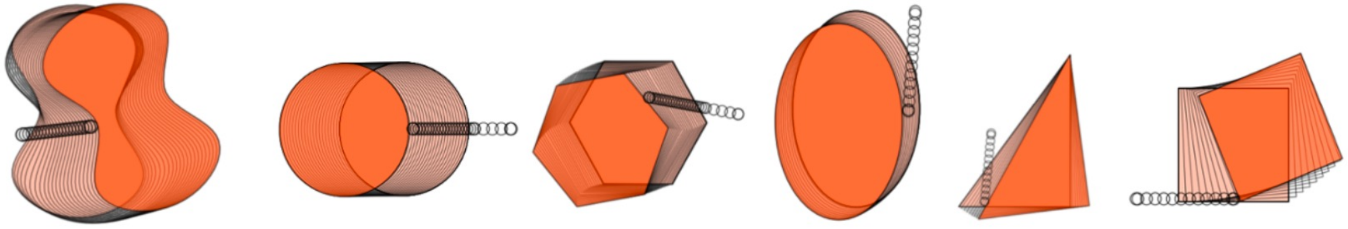
Kuan-Ting Yu, Maria Bauza, Nima Fazeli, and Alberto Rodriguez

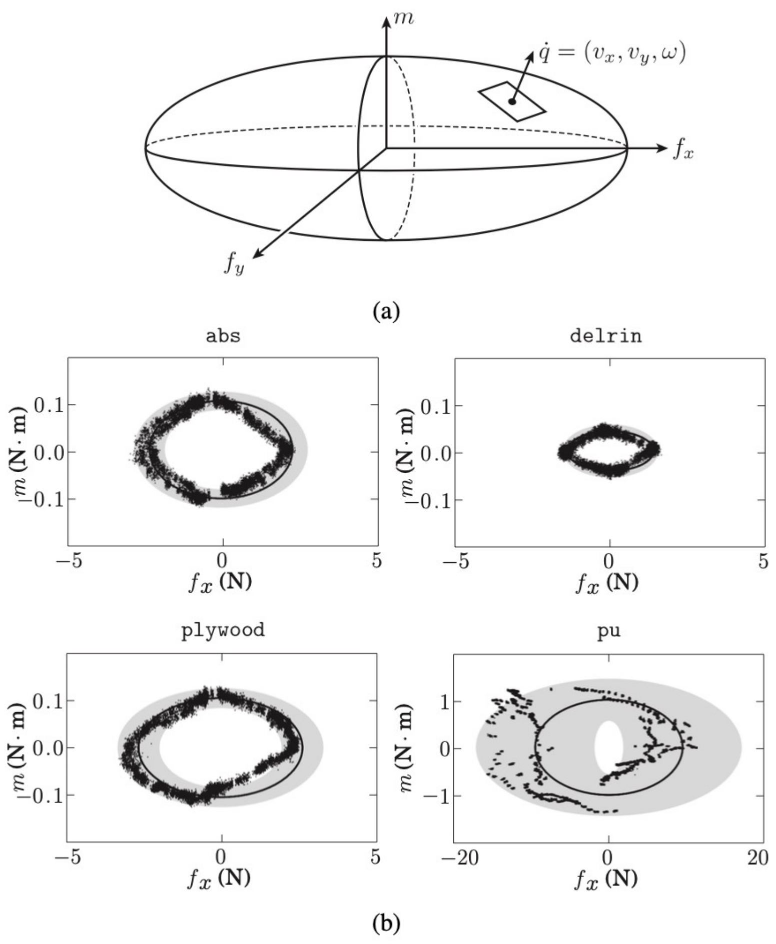
Computer Science and Artificial Intelligence Lab &
Mechanical Engineering Department, MIT

MCube Lab

IROS 2016, "More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing" by Peter Yu, Maria Bauza et al.

Validating Models for Planar Pushing



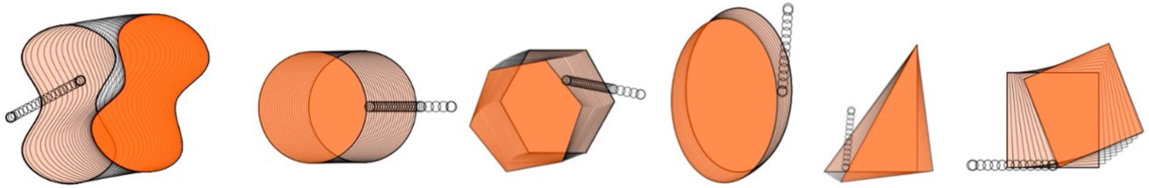


IROS 2016, "More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing" by Peter Yu, Maria Bauza et al.

Validating Models for Planar Pushing

More than a Million Ways to Be Pushed.

A High-Fidelity Experimental Dataset of Planar Pushing



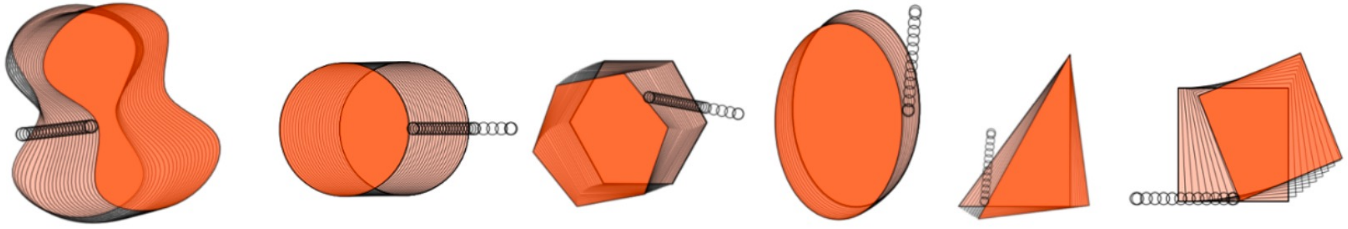
Kuan-Ting Yu, Maria Bauza, Nima Fazeli, and Alberto Rodriguez

Computer Science and Artificial Intelligence Lab &
Mechanical Engineering Department, MIT

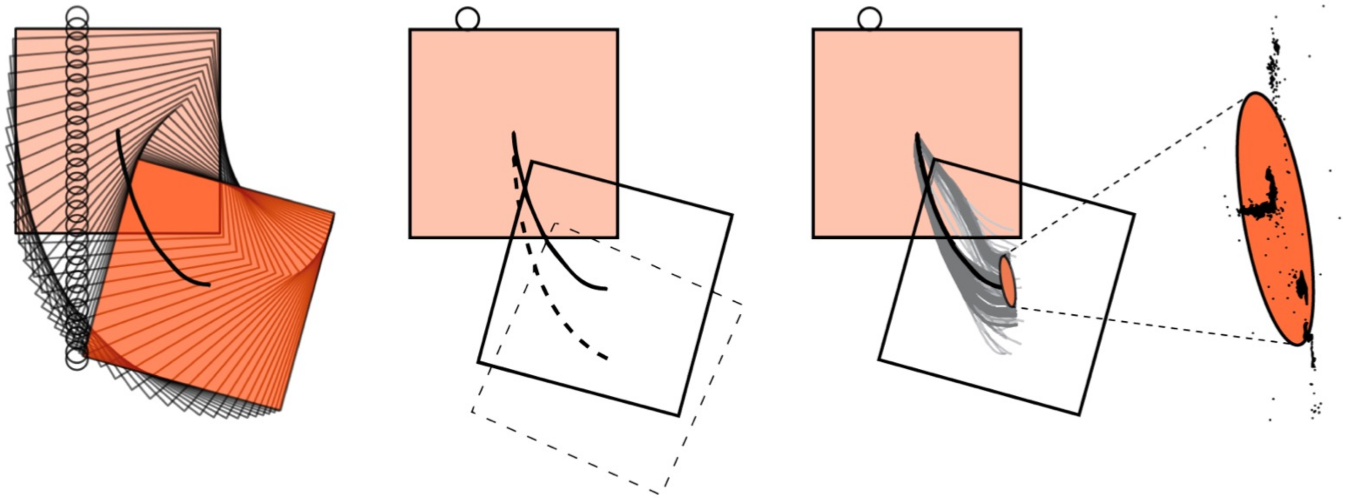
MCube Lab

IROS 2016, "More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing" by Peter Yu, Maria Bauza et al.

Validating Models for Planar Pushing



$$\underset{u(t)}{\arg\min} \quad h(x(T)) + \int_0^T g(x(t), u(t))\, dt$$



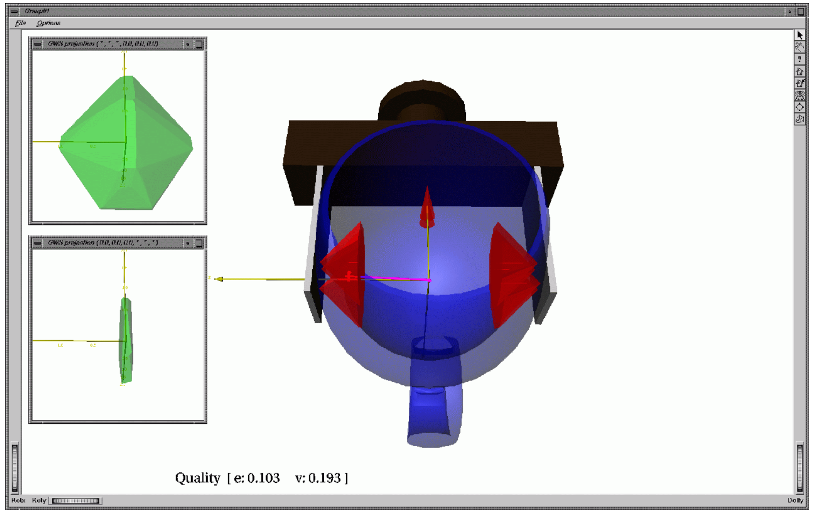
IROS 2016, "More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing" by Peter Yu, Maria Bauza et al.

Suggested Reading

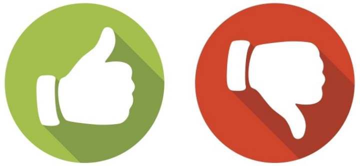
- More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing by Peter Yu, Maria Bauza et al. IROS 2016.
- Maria Bauza and Alberto Rodriguez. A probabilistic data-driven model for planar pushing. ICRA 2017

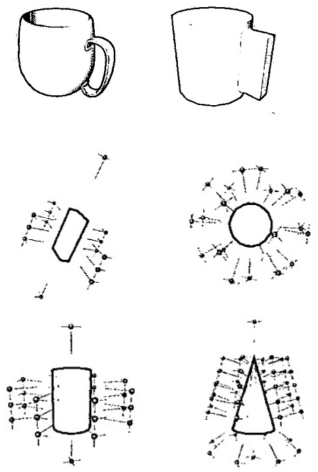
What are common assumptions?

How do we generate a grasp?

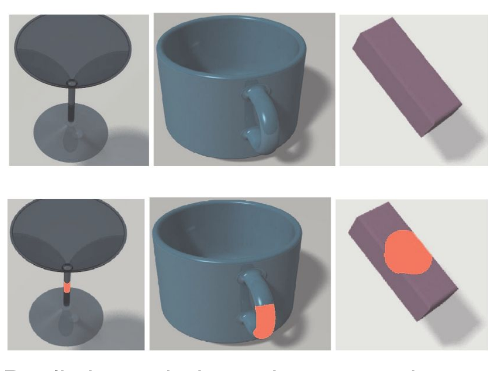


Grasp Evaluation

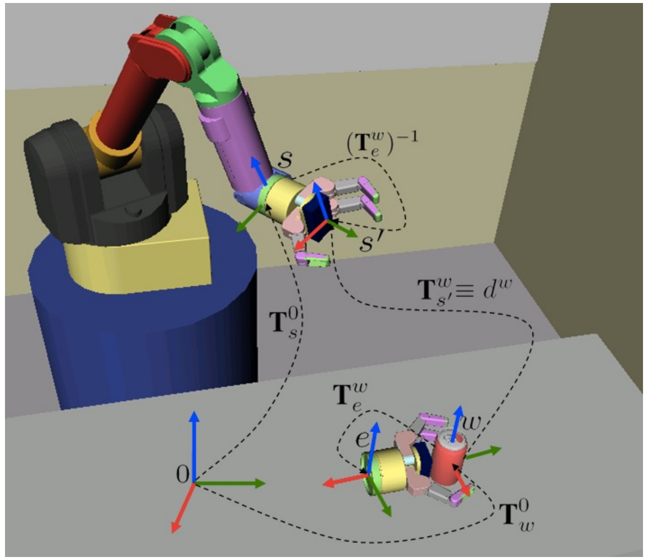




Offline database with grasps linked to 3D objects



Perception

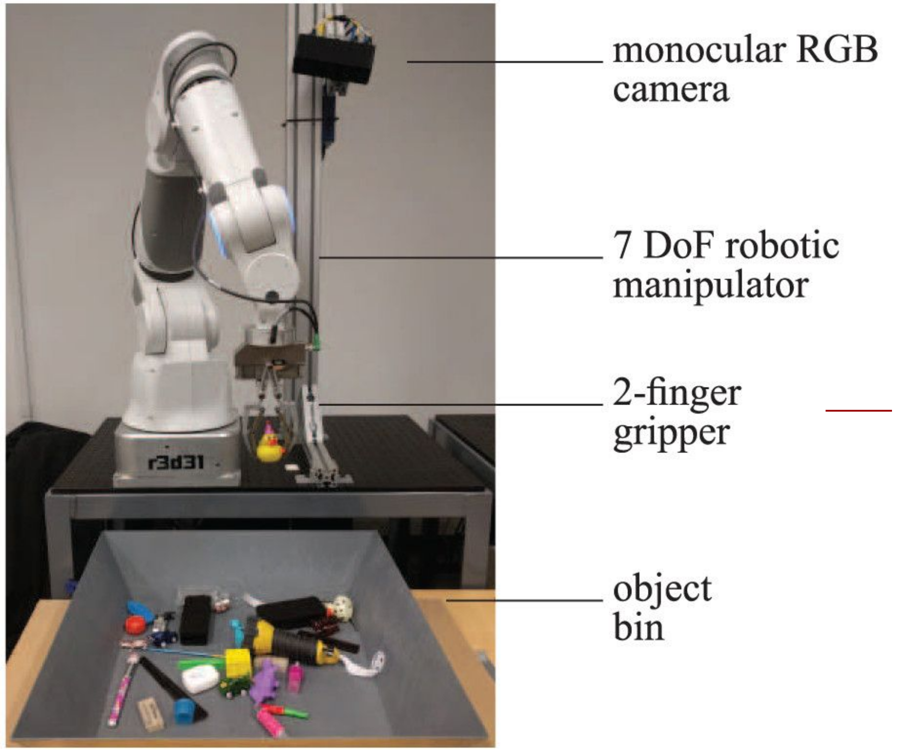


Motion Planning

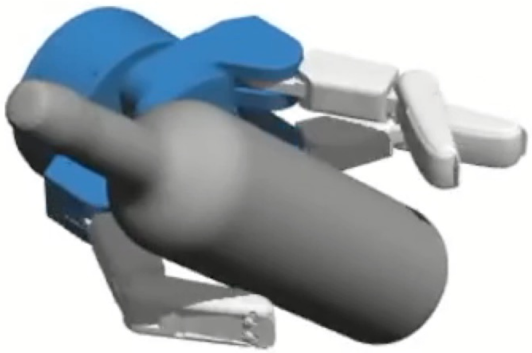
Offline

Online

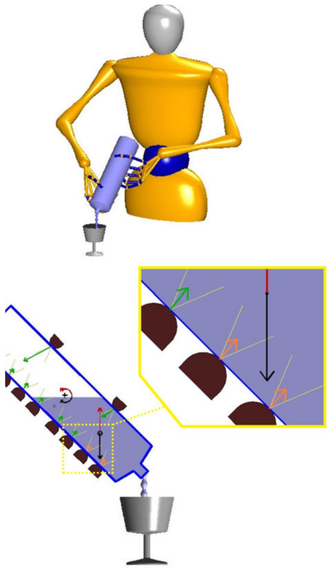
How do we execute a grasp?



Top-Down & Open-Loop

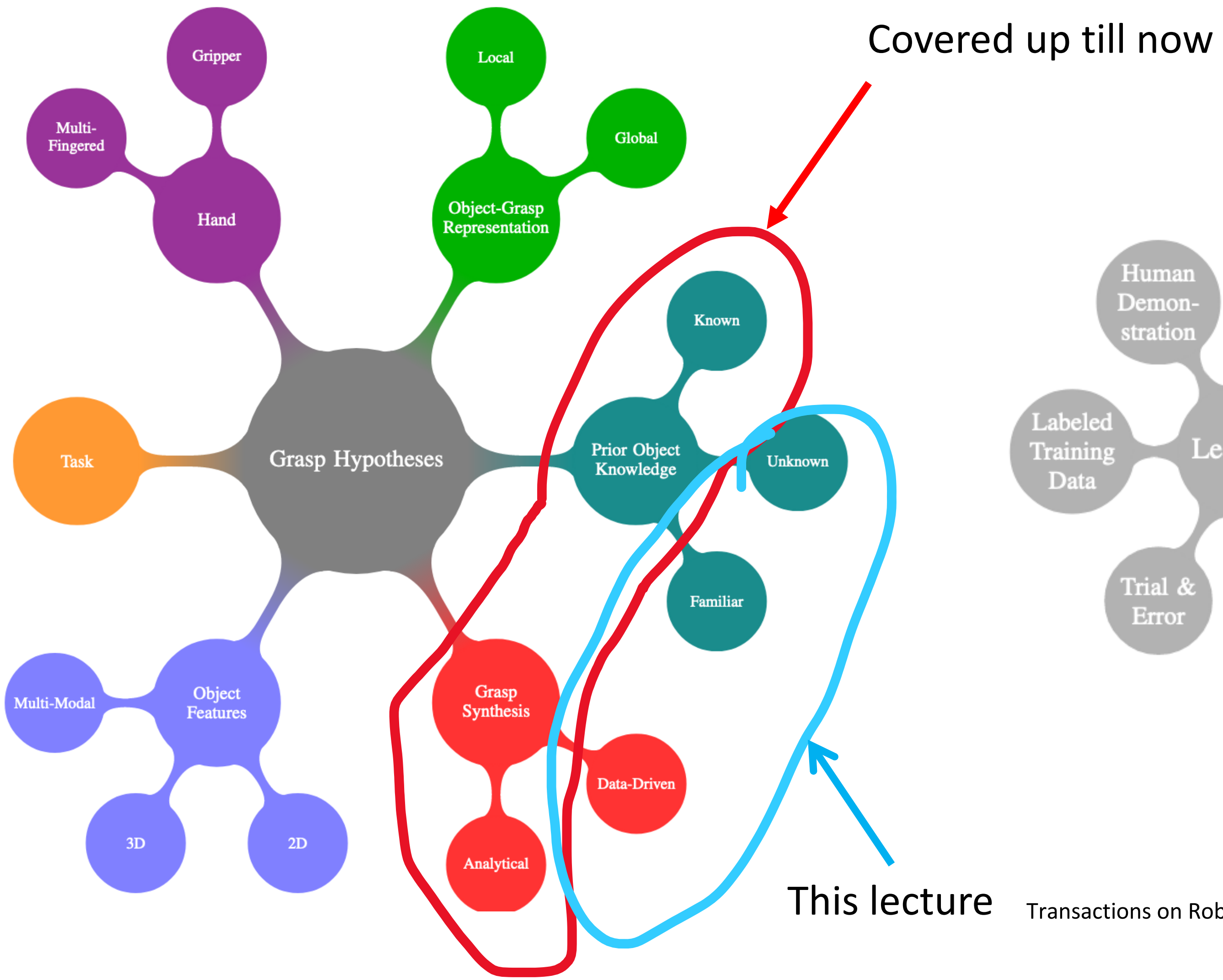


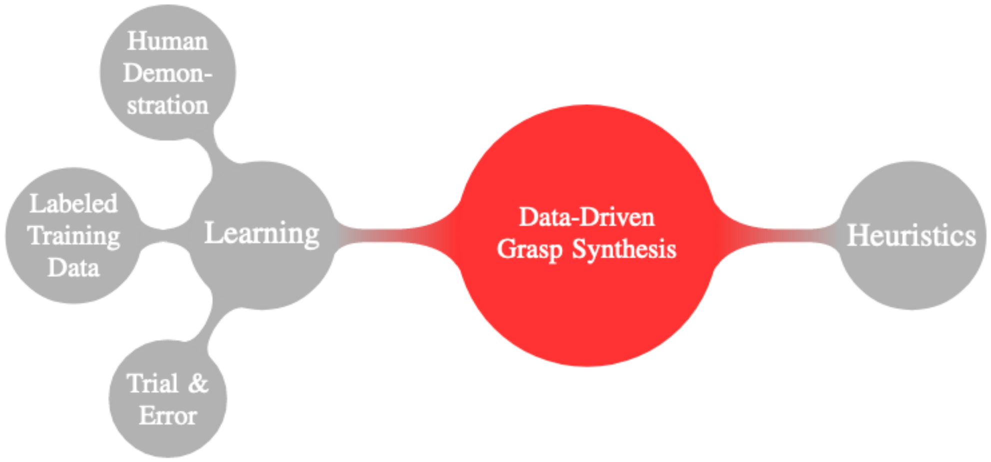
Acquiring a grasp + Closed Loop



Grasp Force Optimization

Data-Driven Approaches to Grasping



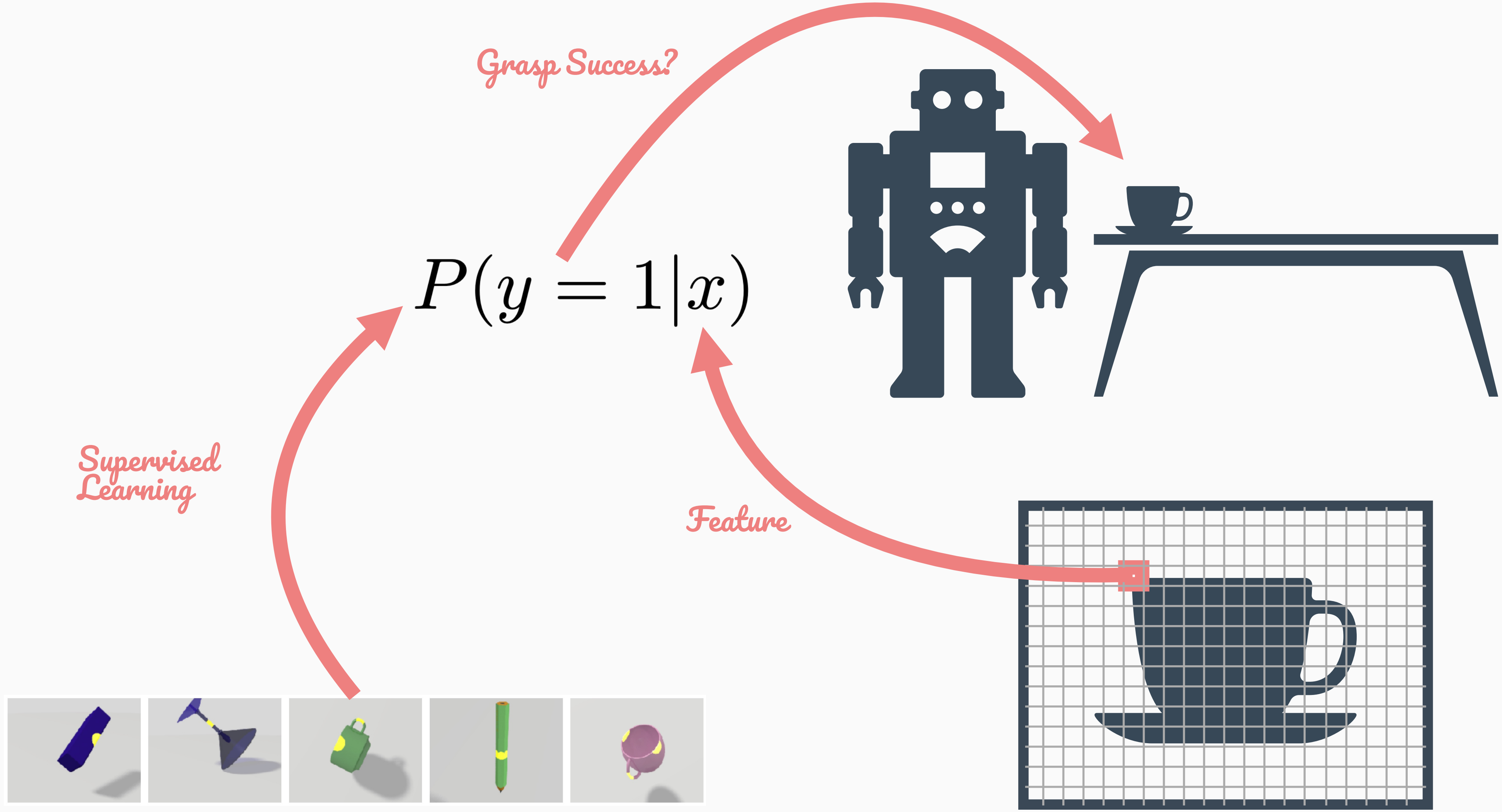


Transactions on Robotics 2014, "Data-Driven Grasp Synthesis – A survey" by Bohg et al.

Detecting 2D Grasping Points

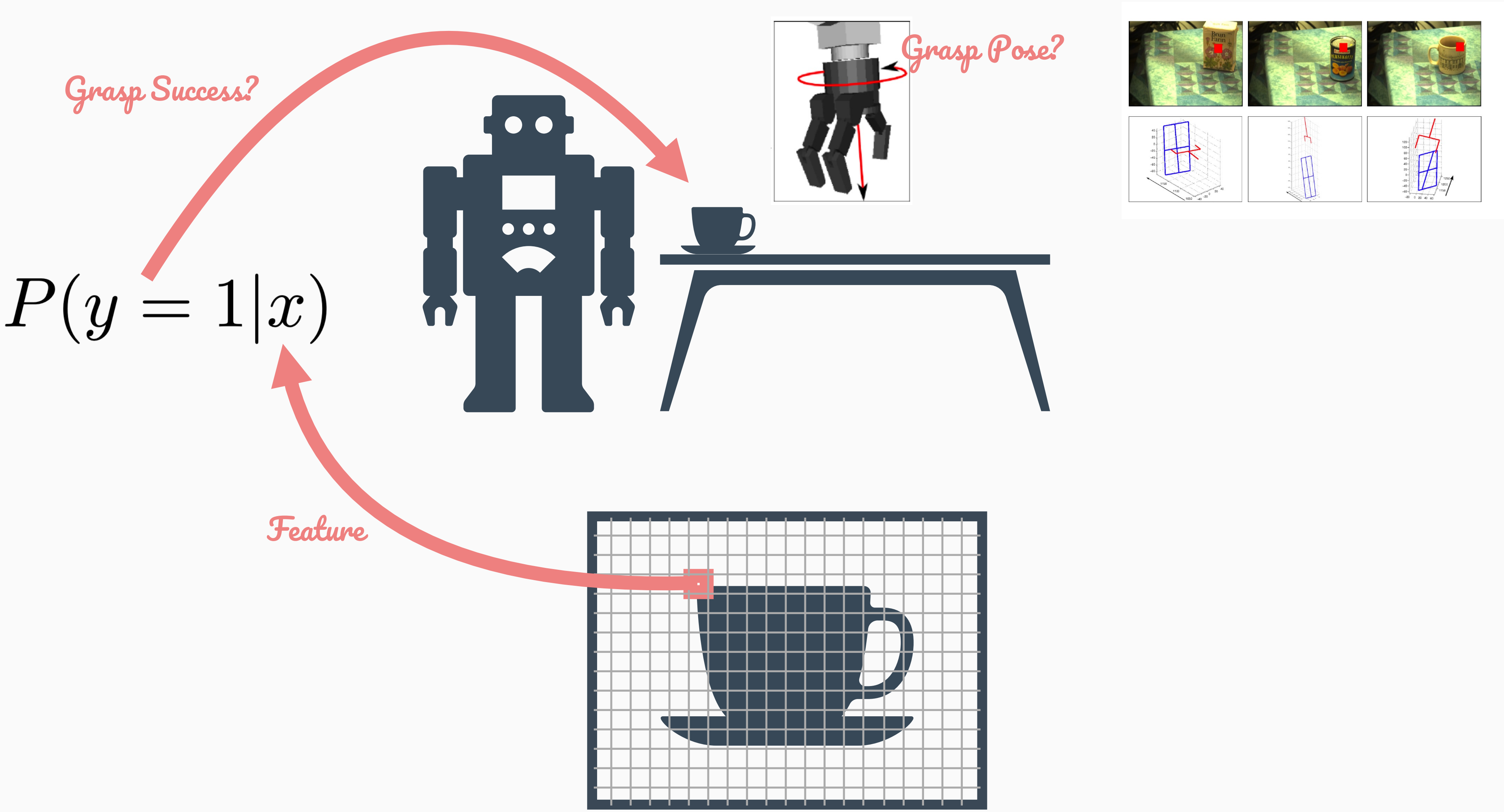
Bohg and Kragic. *Learning Grasping Points with Shape Context.* Robotics and Autonomous Systems. 2010

Grasp Point Detection as a Classification Problem



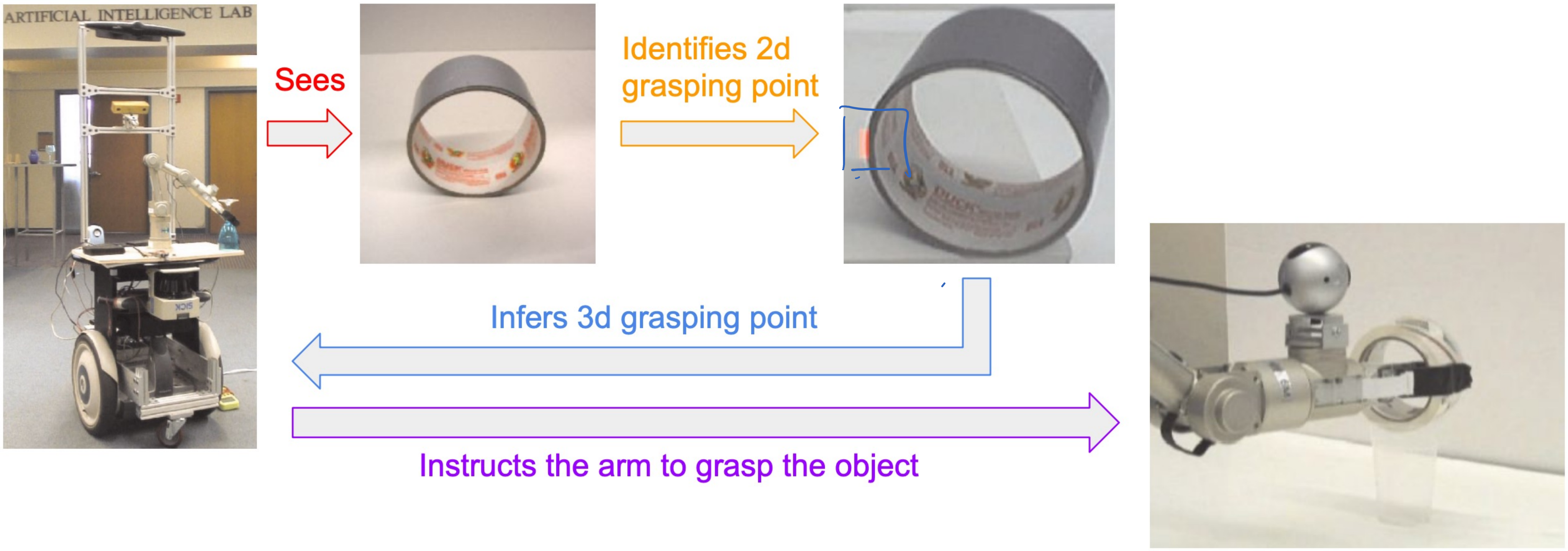
Saxena et al. *Robotic Grasping of Novel Objects.* NeurIPS 2006

From 2D Grasping Points to 6D Grasp Pose

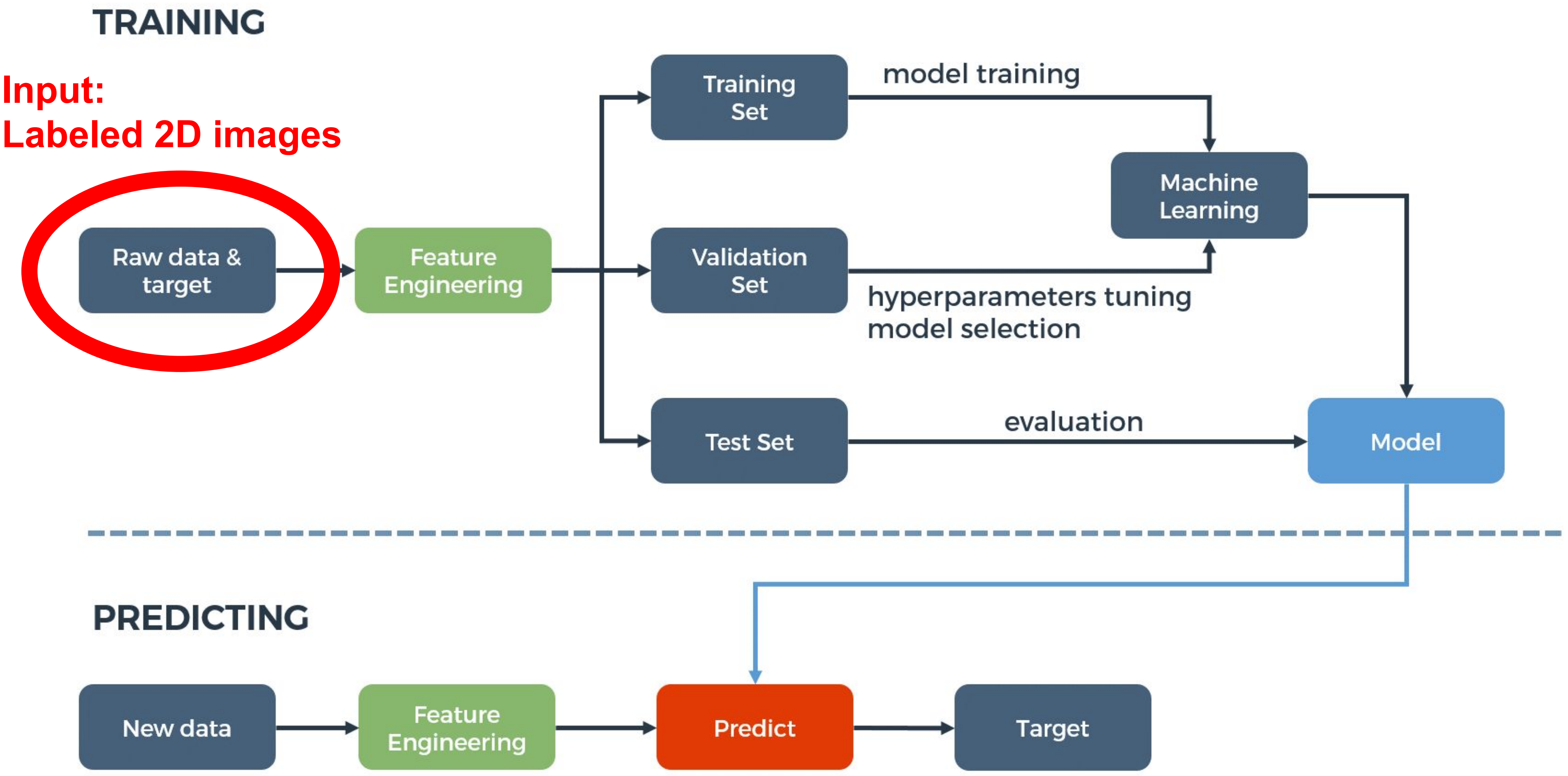


Robotic Grasping of Novel Objects using Vision. Saxena et al. IJRR 2008.

Grasping previously unseen objects using only 2D images *without* 3D meshes

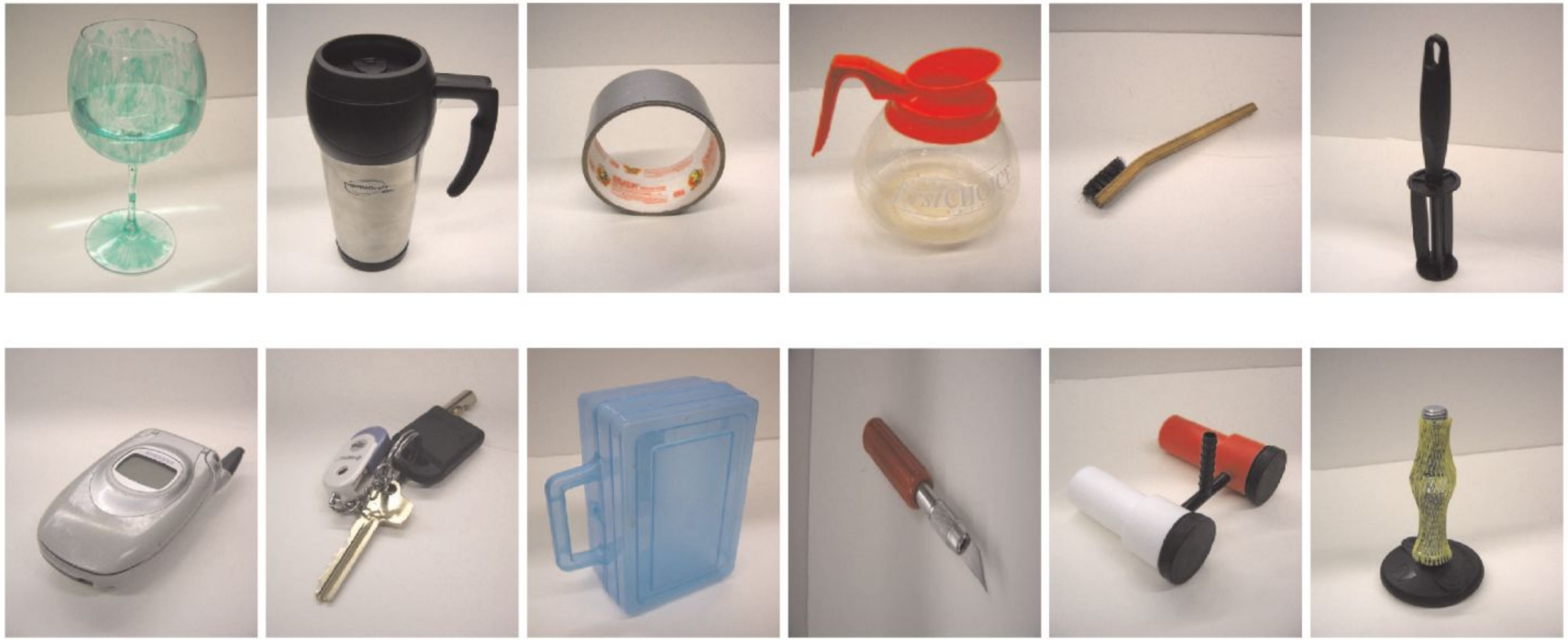


Supervised learning pipeline



Data collection

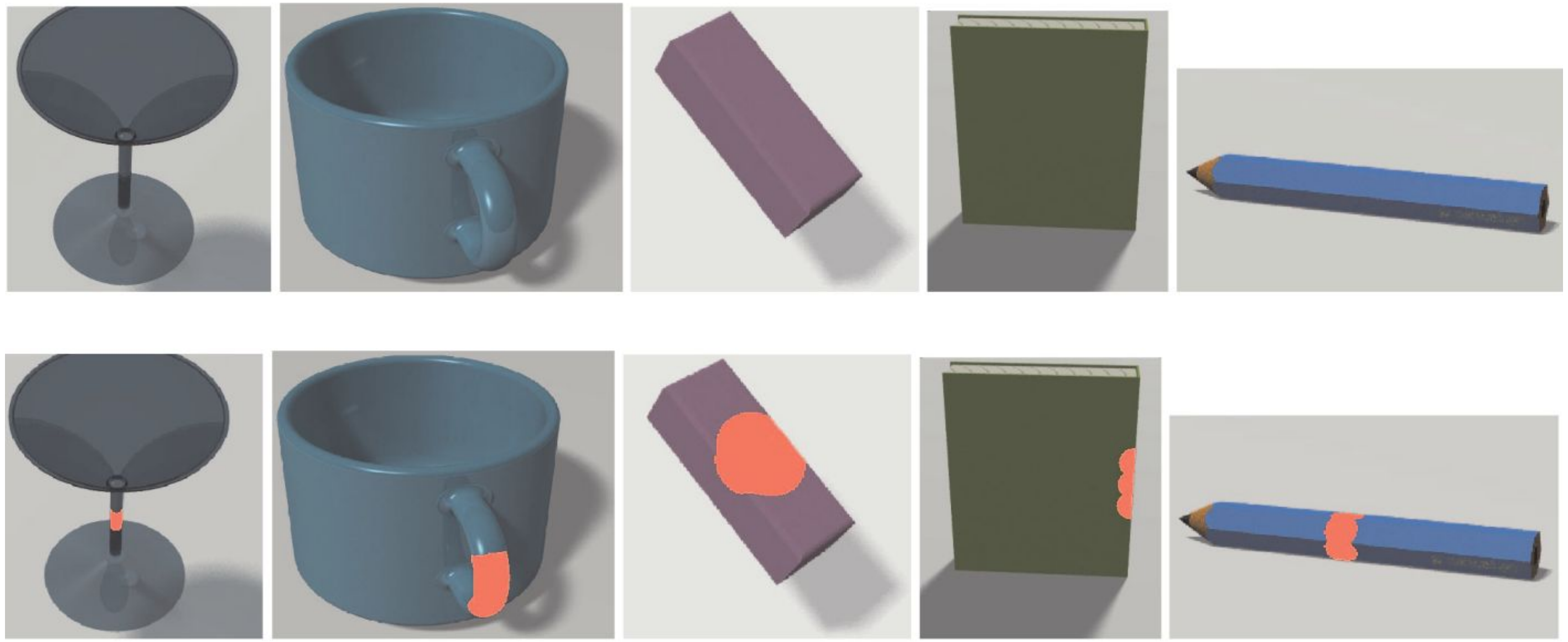
We could collect real images…



...but labeling them is cumbersome / prone to errors.

Data collection

Solution? Use synthetic data!



2500 images
5 object classes

Realistic rendering using ray tracing.
Enables automatic labeling: random lighting, color, orientation, size...

Supervised learning pipeline

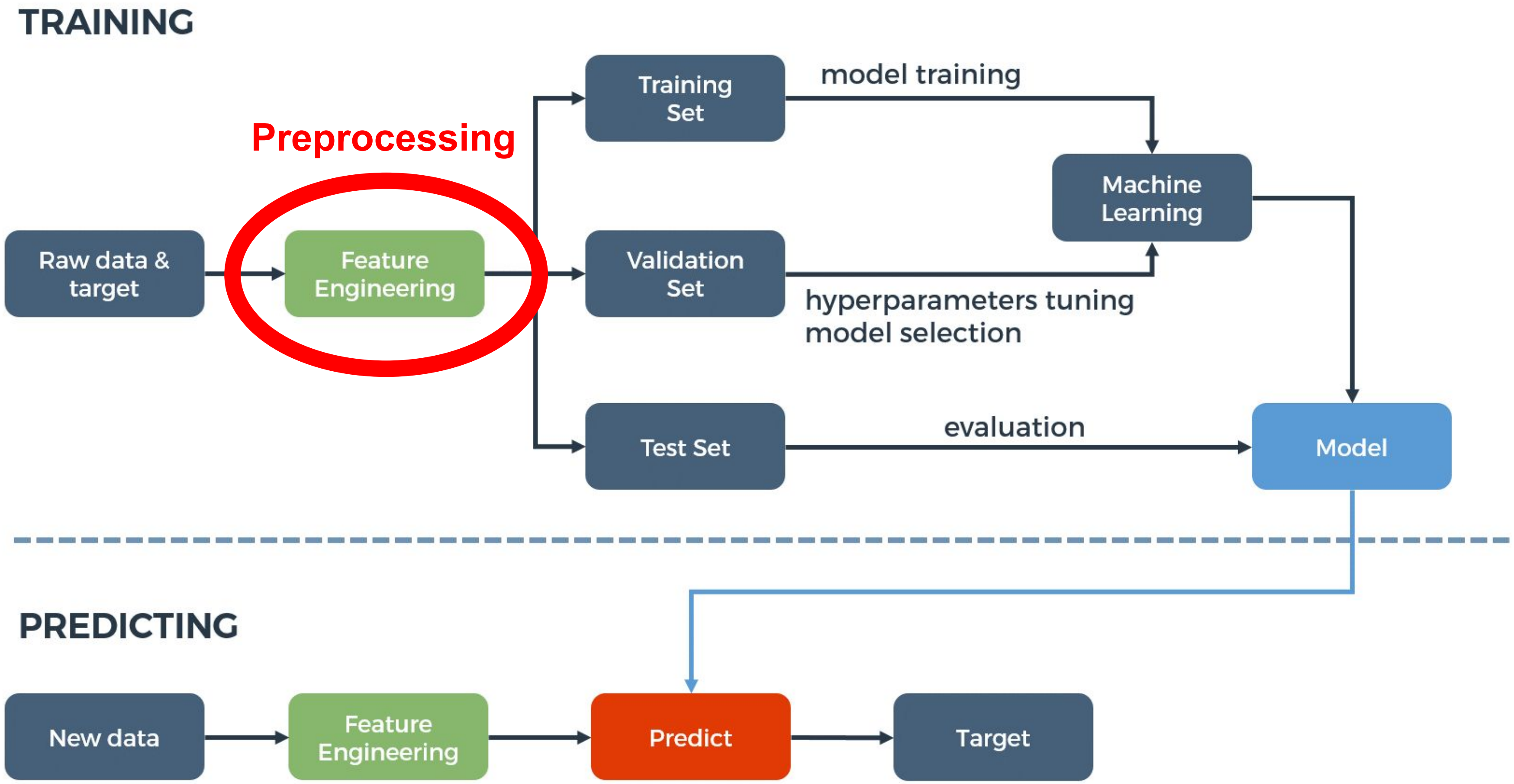
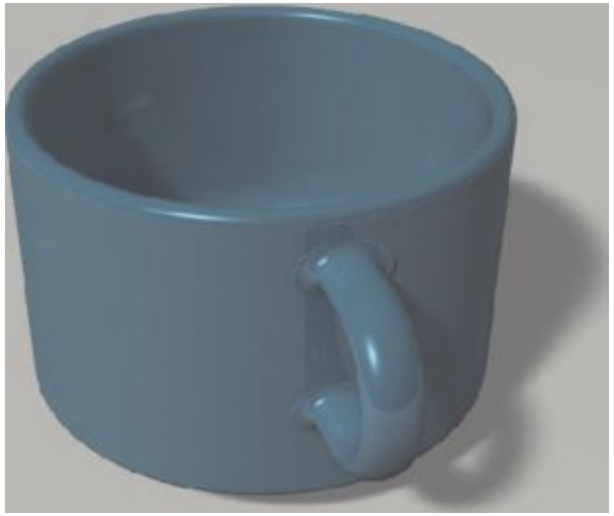


Image preprocessing



RGB -> Y (luma) Cb (chroma) Cr (chroma)

Y: intensity; Cb: B - Y; Cr: R - Y

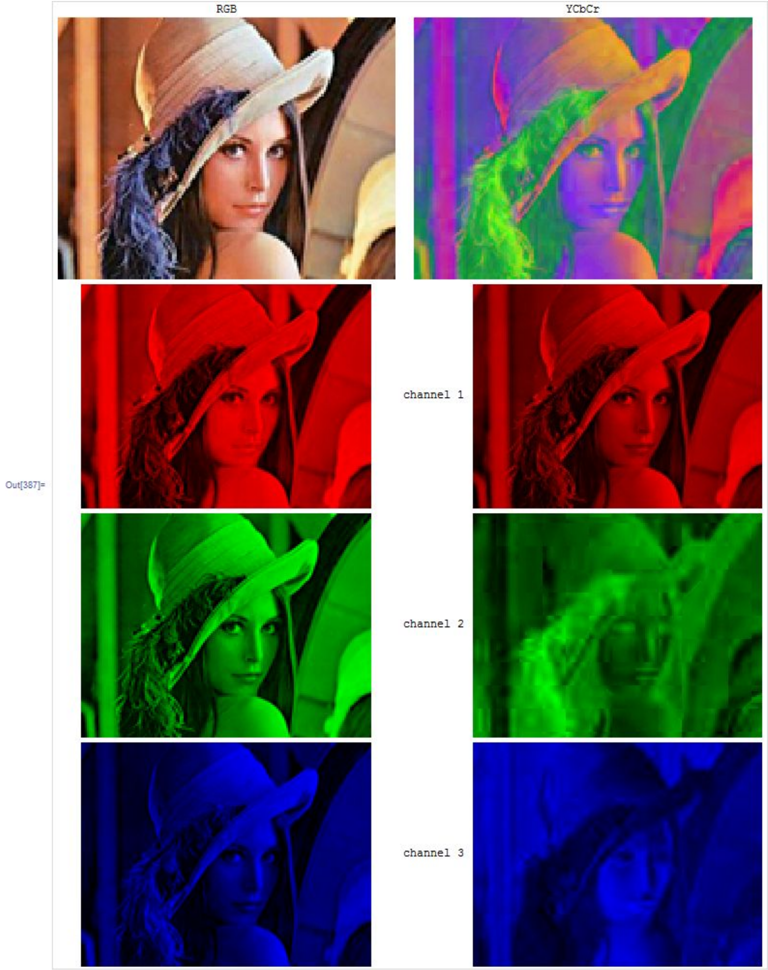
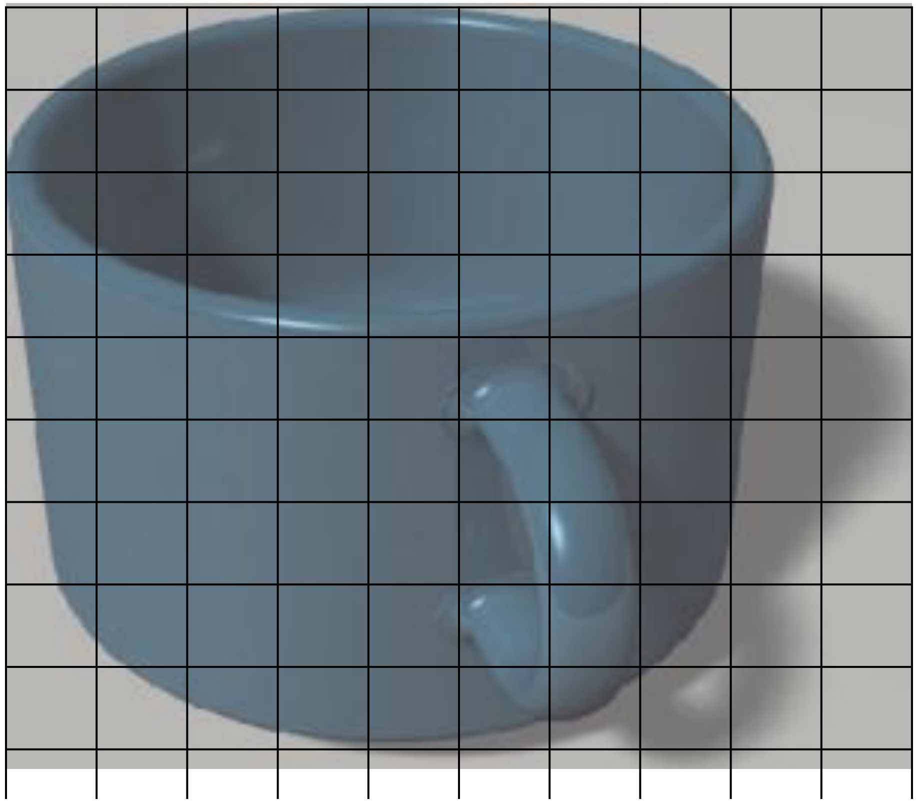


Image preprocessing



Edge filters (Y):

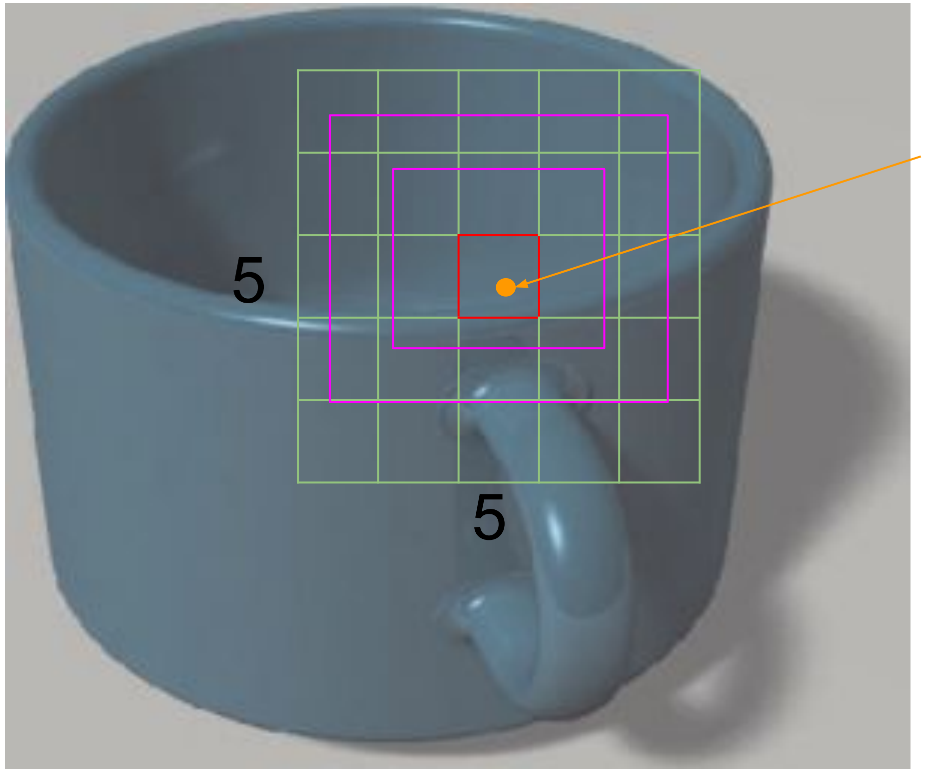
Texture filters (Y):

Average filter (Cb/Cr):

6 (edge) + 9 (texture) + 1 (average) * 2 =

17 features per patch

Image preprocessing



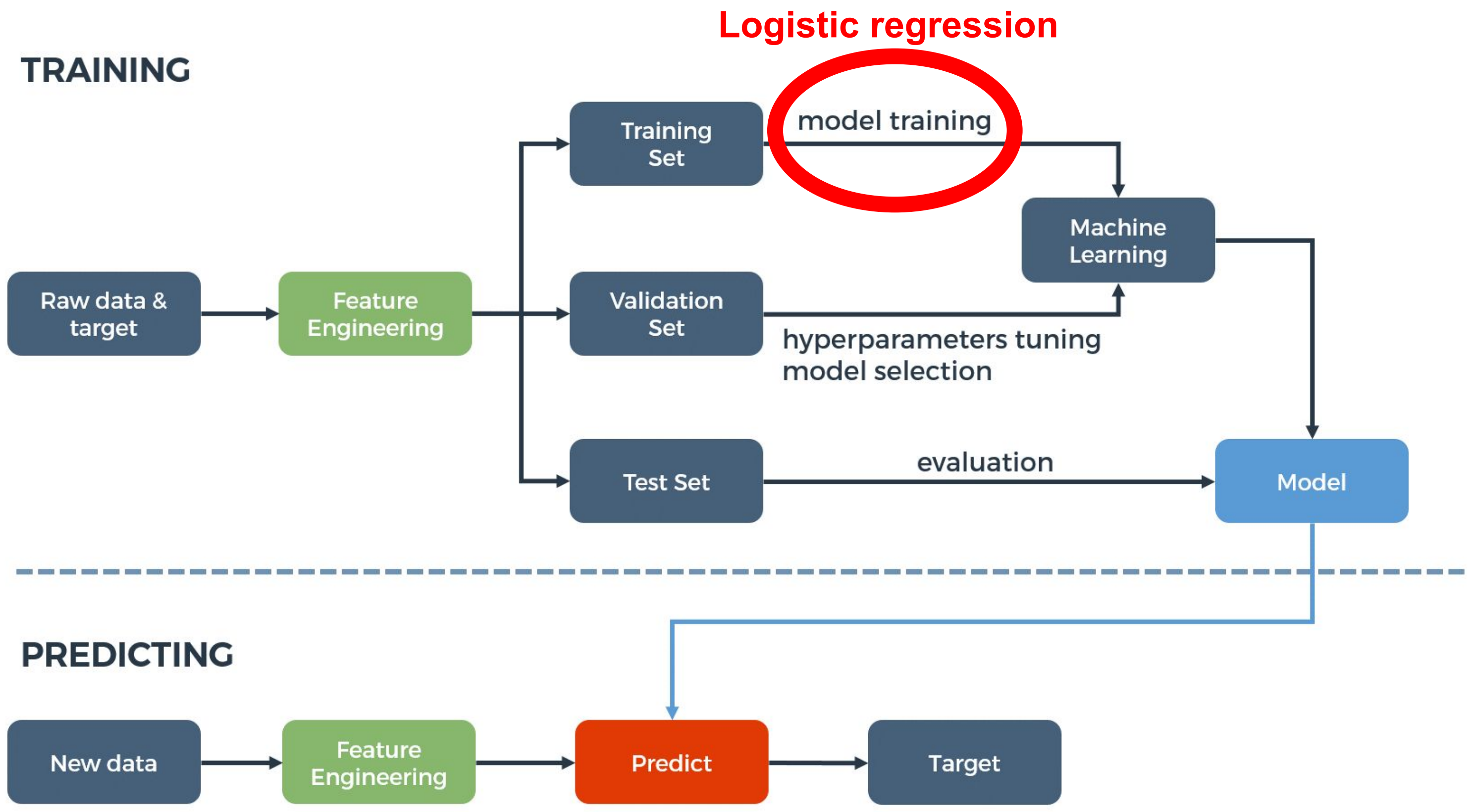
Apply filters on :

- 3 different scales for the patch centered at the pixel of interest
- 1 scale for the 24 surrounding patches in a 5x5 window

17 (# features/patch) * (3 + 24) =

459 features per patch of interest

Supervised learning pipeline



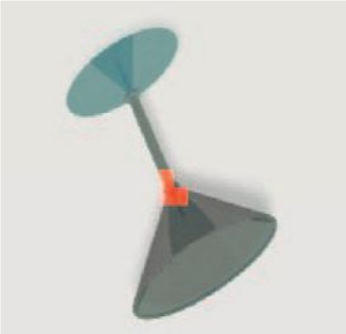
Binary classification task

Is a given pixel *(u,v)* on the image a grasping point (1) or not (0)?

Binary classification task

Prediction time:

$$P(z(u,v) = 1 | C) = \sigma(\hat{\theta}^{\top} x)$$

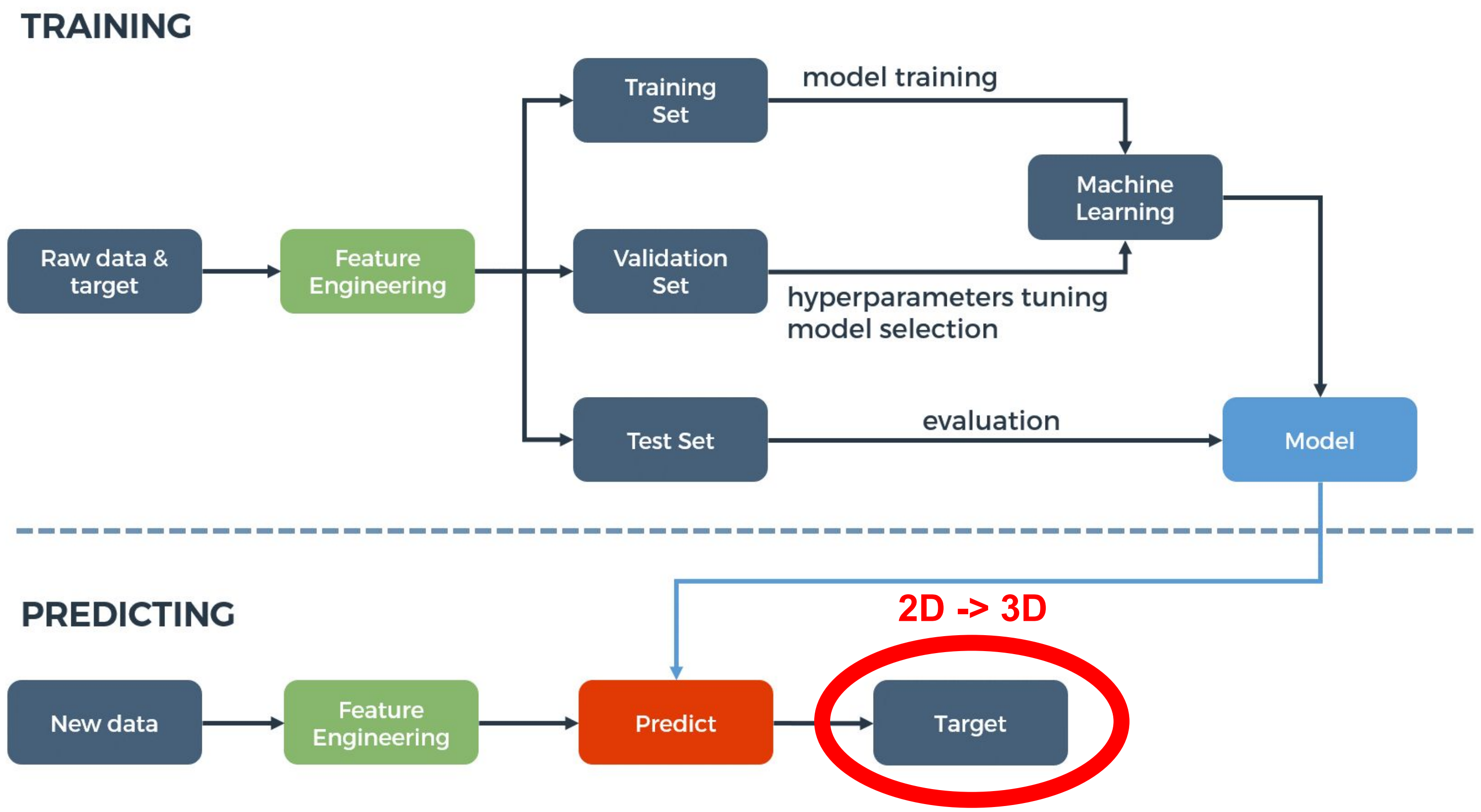


How likely is pixel *(u,v)* on image *C* a grasping point?

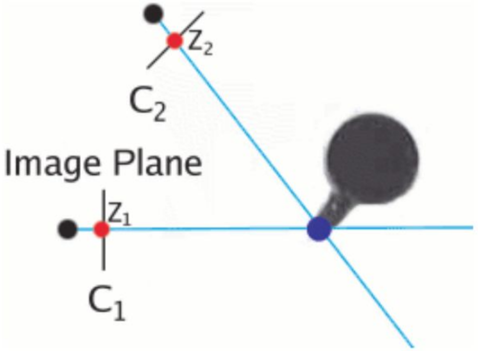
Learned parameter

Features of the patch centered on *(u,v)*

Supervised learning pipeline



2D -> 3D



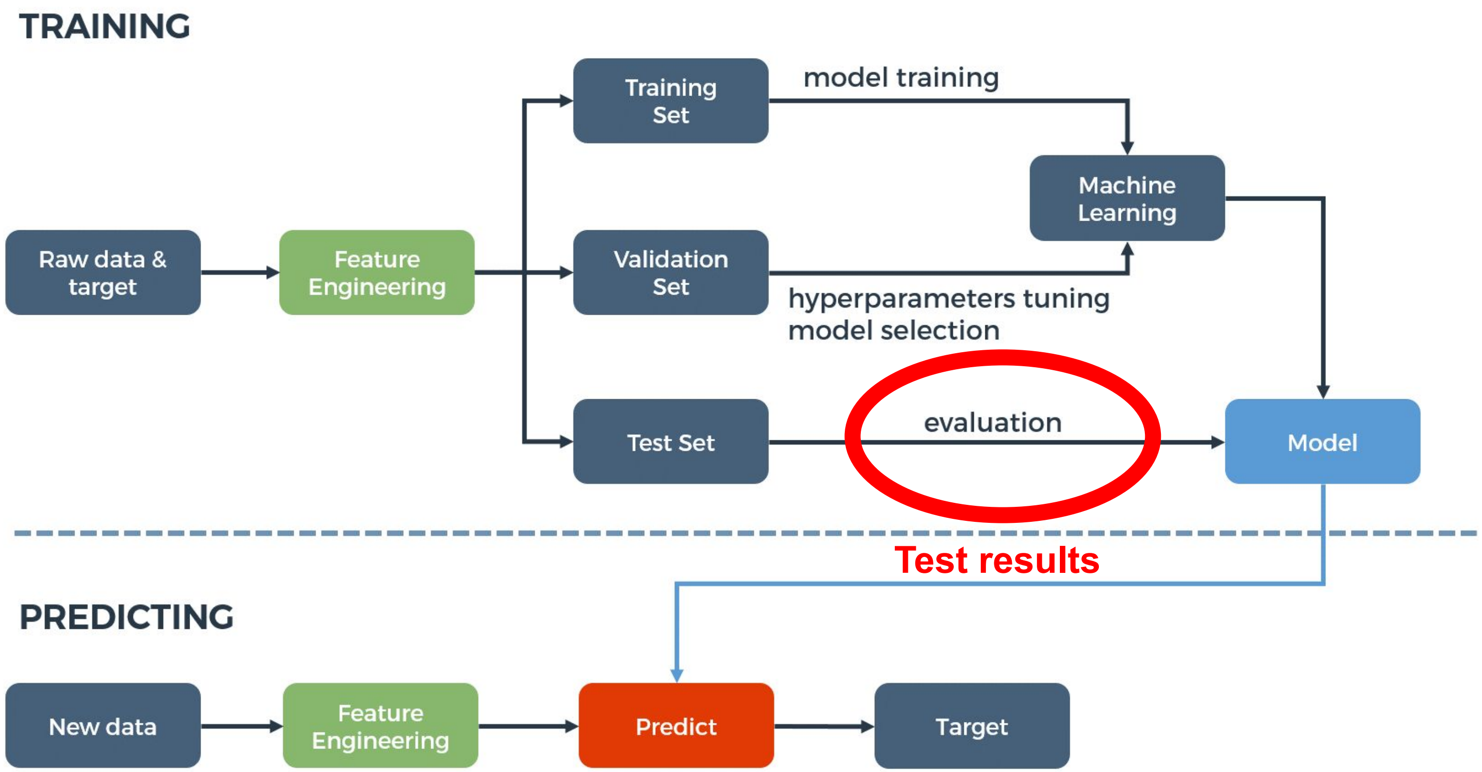
Link 2D to 3D intuitively:

$$z_i(u, v) = 0 \iff y_{r_1}(u, v) = 0 \wedge \ldots \wedge y_{r_k}(u, v) = 0$$

Pixel is not a grasping point

No grid cells along the ray passing through the pixel contain a grasping point

Supervised learning pipeline



Hardware setup

5 dof arm

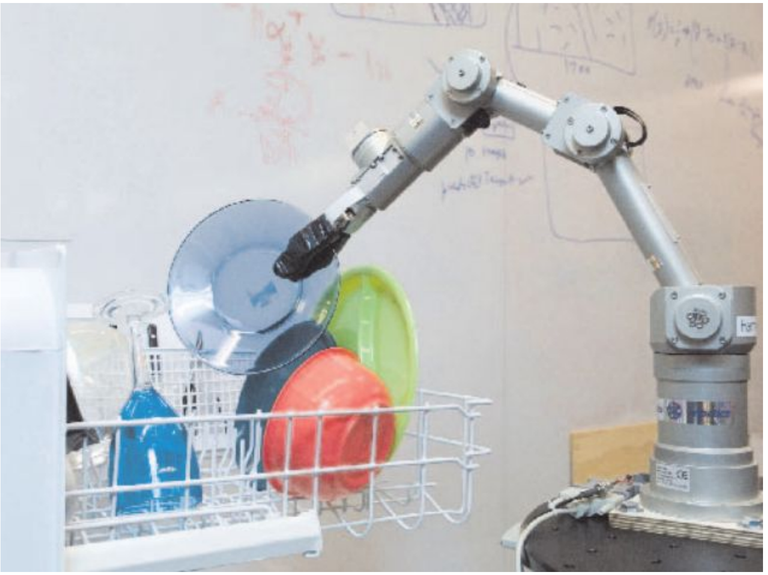
Random object location on uncluttered table top

Evaluation results

1. Synthetic data:
   Classification accuracy on unseen images is 94.2% (2D).
   Accuracy on unseen images after triangulation is higher (3D), mean error 0.84 cm.
2. Real data:
   Mean error after triangulation (3D) 1.84 cm.
   Picked up novel objects 87.8% of the time.

Application task: unloading dishwasher

Added real images + depth measurements



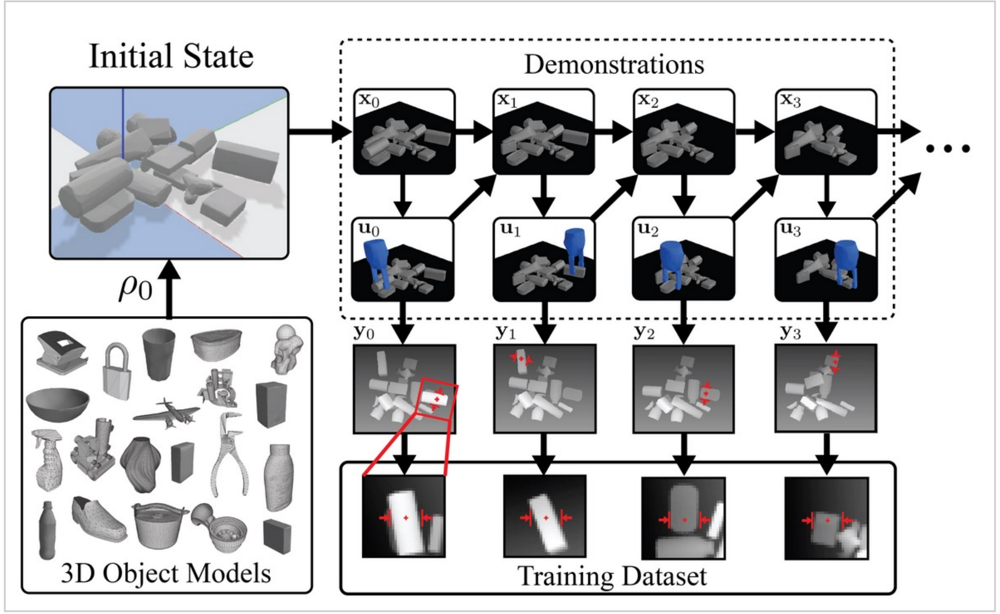
Tested on	Grasp success rate
Plates	100%
Bowls	80%
Mugs	60%
Wine glass	80%
Overall	80%

Conclusion

- Learning-based method
- Only input is 2D images, no 3D mesh model needed
- **Generalizes** to previously unseen objects
- Cool applications!

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

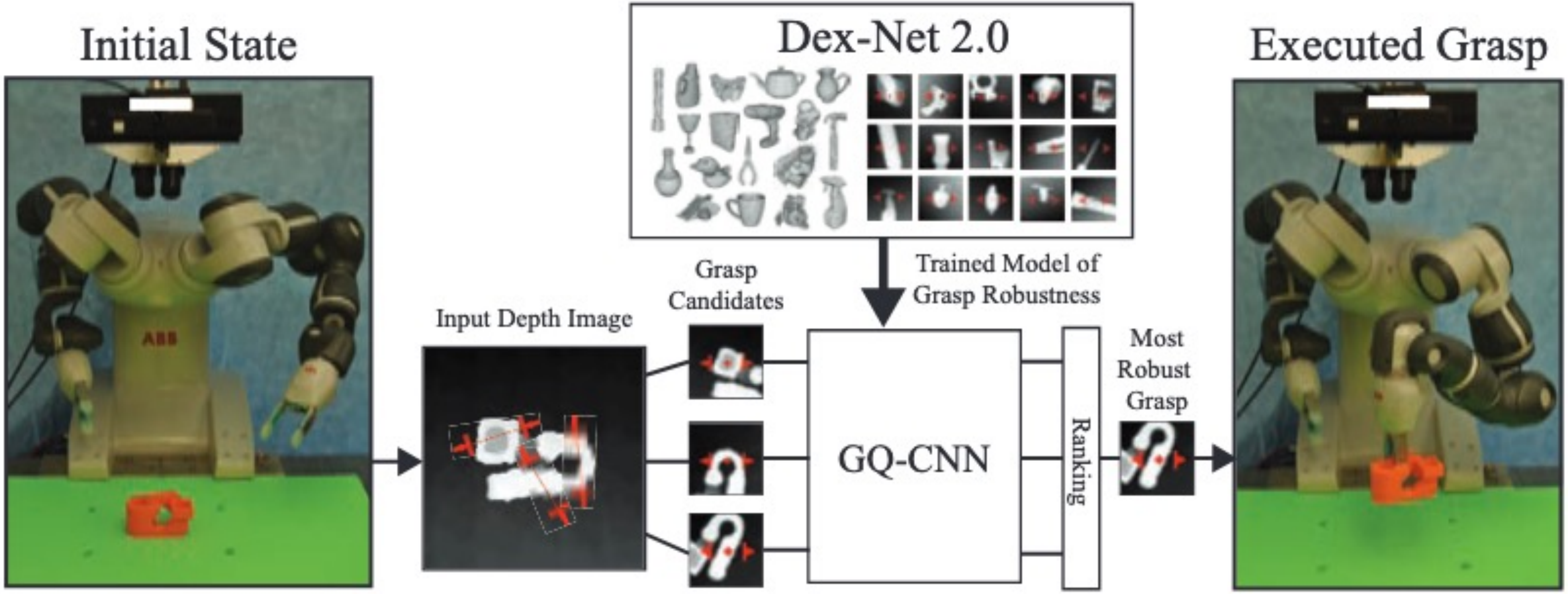
- DexNet 1.0 – 4.0 – Berkeley – AutoLab
- Google Arm Farm



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

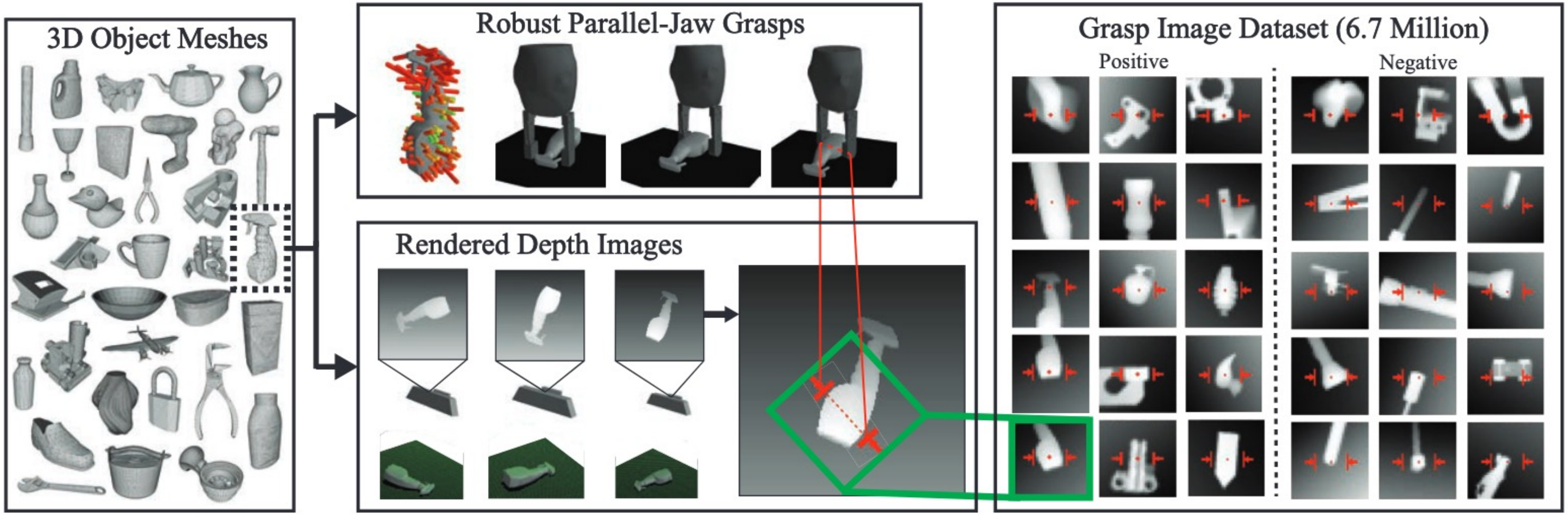
"**Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences**" by Mahler and Goldberg. CORL 2017. https://berkeleyautomation.github.io/dex-net

Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics



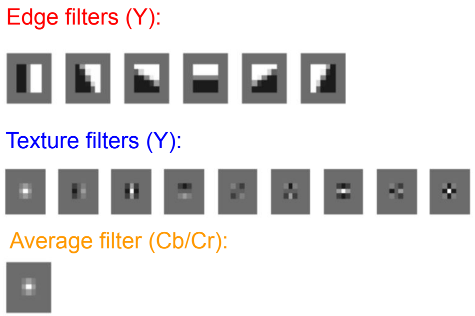
” **Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics**" by Mahler et al.. RSS 2017. https://berkeleyautomation.github.io/dex-net

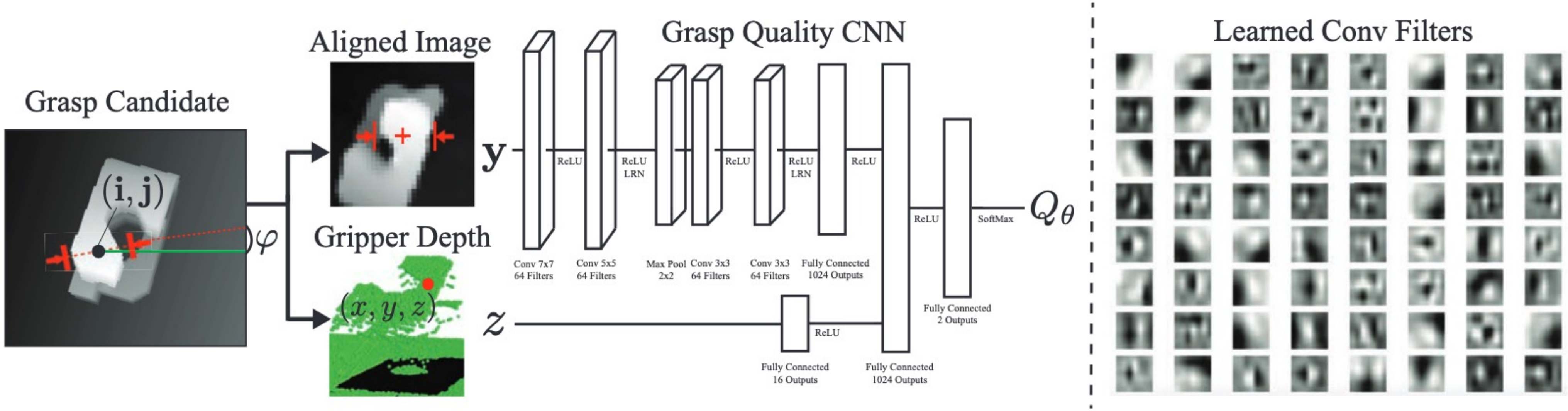
Dataset Generation



” **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

Grasp Classification Network





At test time: $\pi_\theta(y) = argmax_{u \in C}\ Q_\theta(u, y)$ where y = pointcloud, u = grasp parameters

" **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

Video

” **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

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

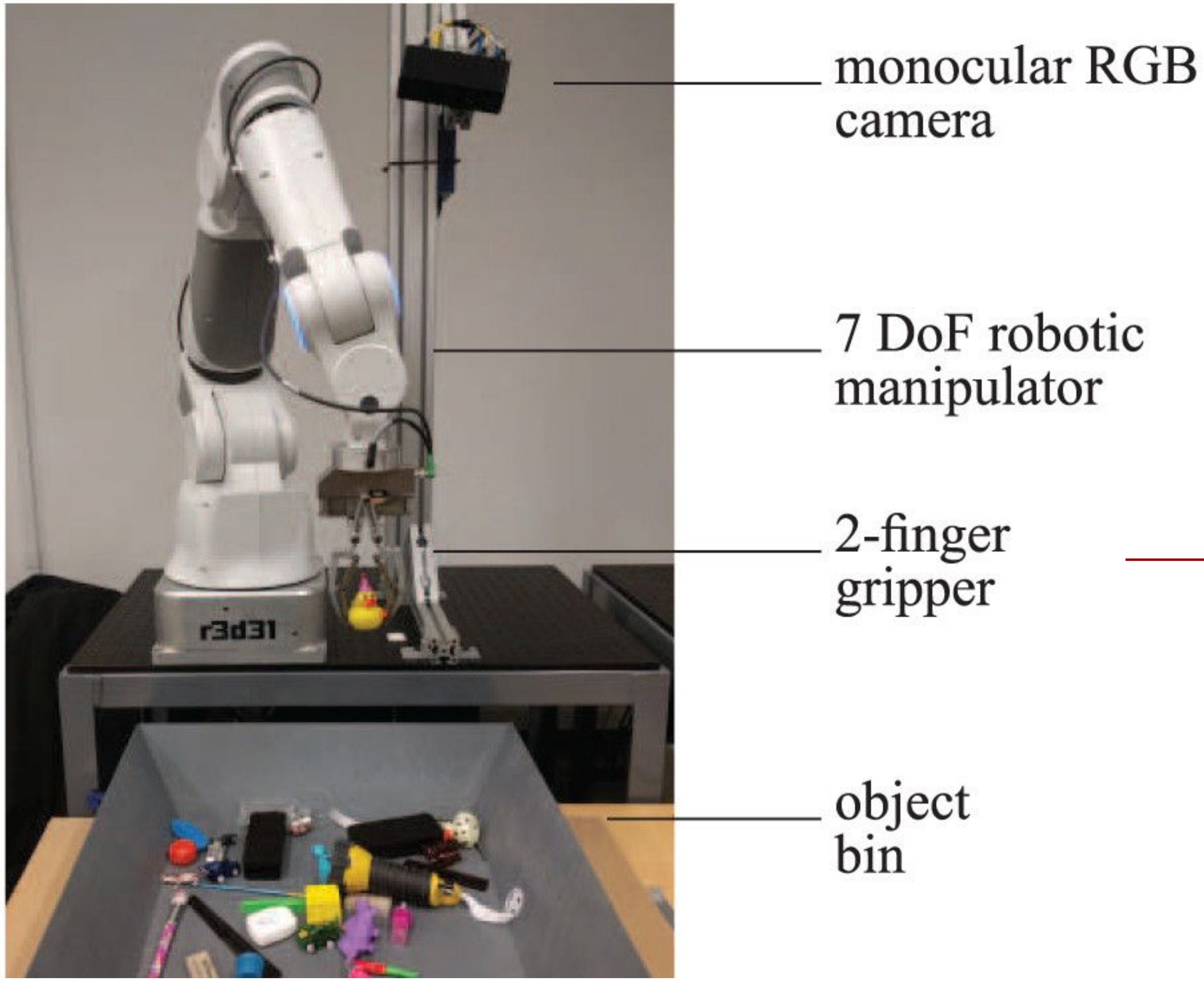
- ~~3D Model of Object~~
  - ~~Depth Sensing~~
  - ~~Wrist Mounted Camera~~
- ~~Specific Representation of Geometry~~
- ~~Contact Model~~
- ~~Simulated Data~~
- ~~Hand-Annotations~~
- ~~Hand-Designed Path Planner~~

RGB Camera

Mounted Over-the-Shoulder

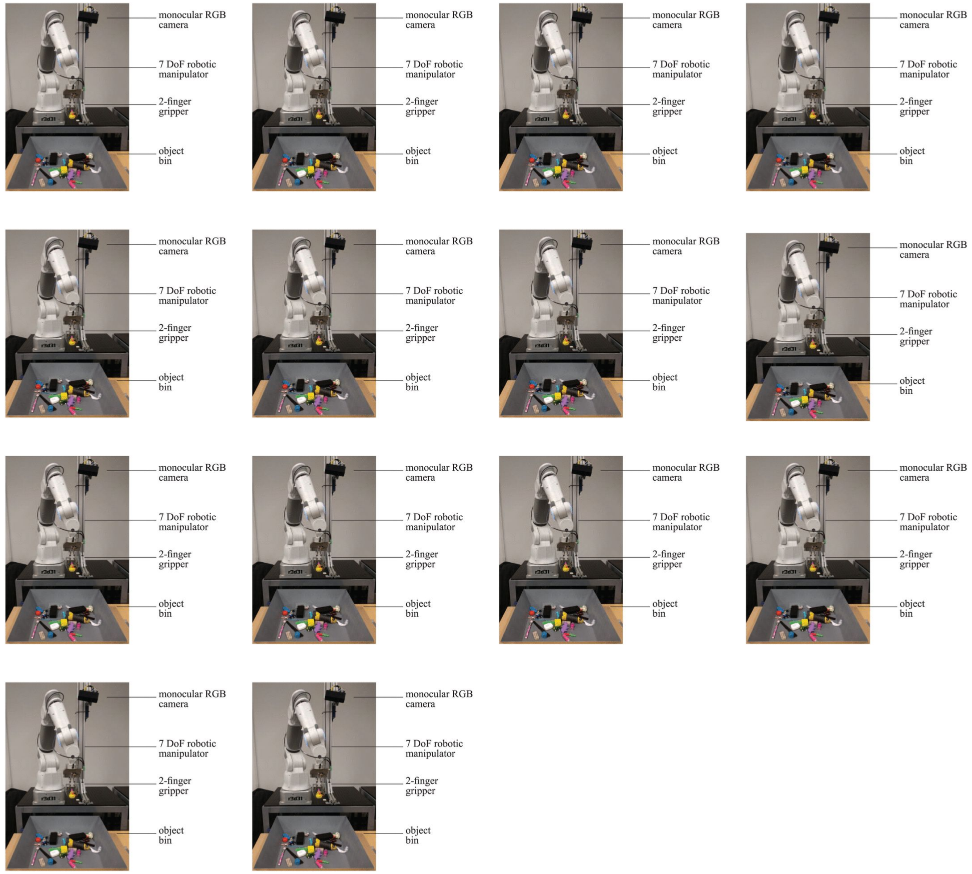
- ~~Camera to Base Calibration~~

So what do we have?



→ Underactuated to conform to object geometry

So what do we have?



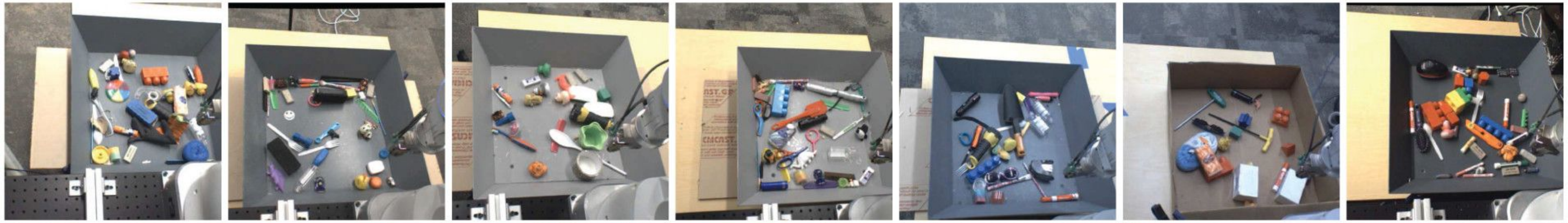
+ Time

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)

$$(p_T - p_t)$$

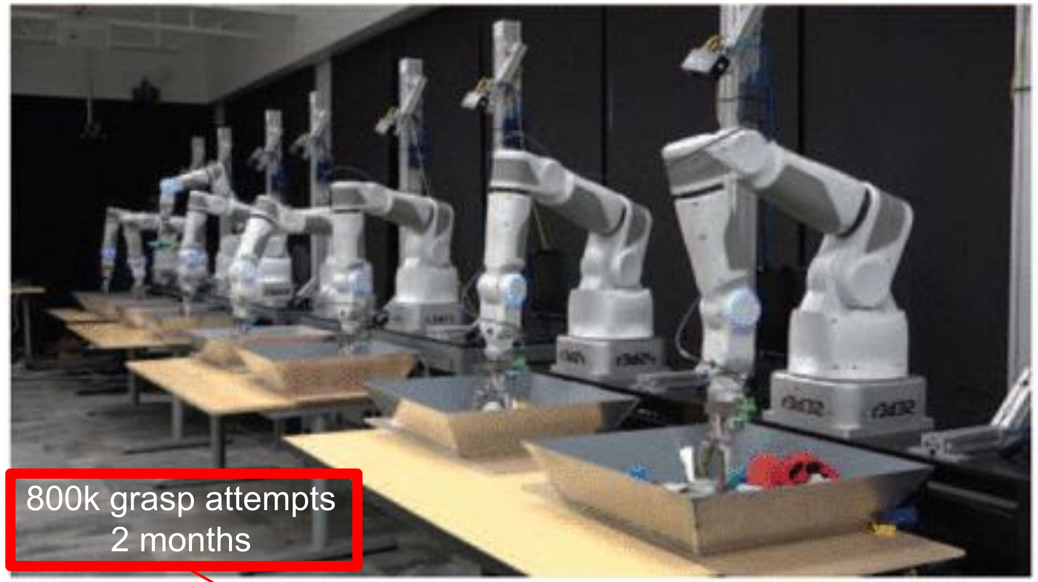
Success

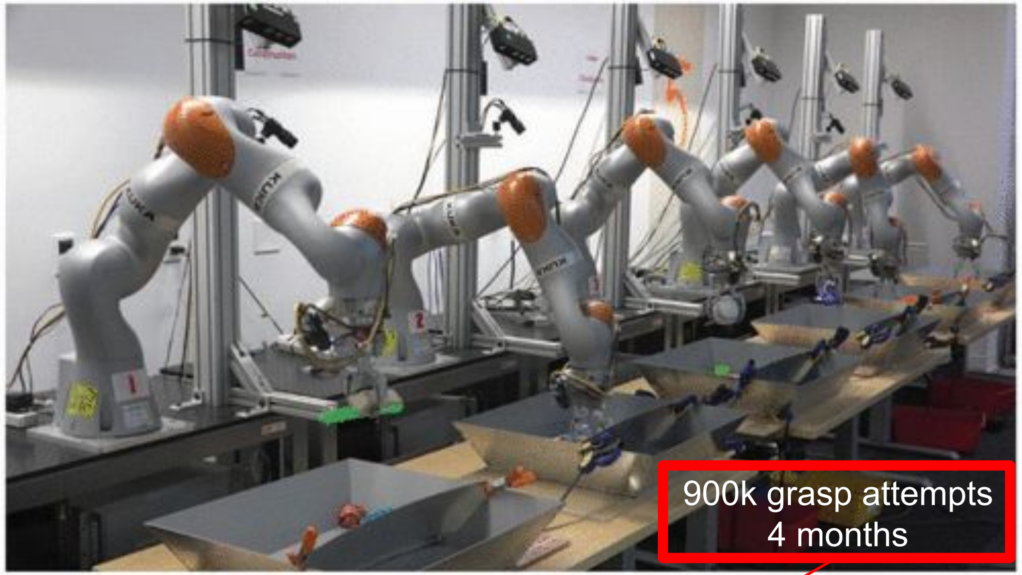
or

Failure

Dataset

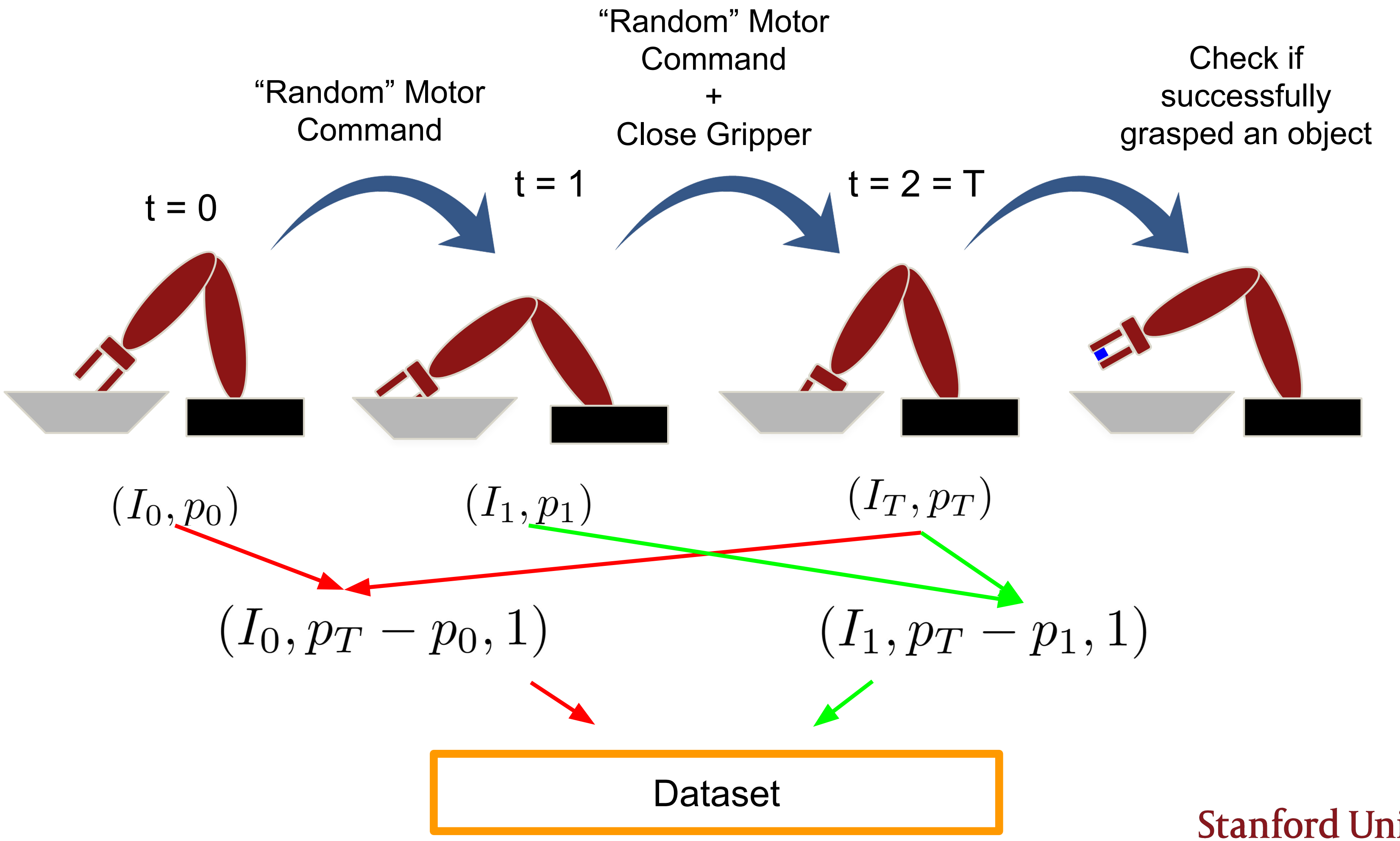
<u>Two Rounds of Self-Supervised Data Collection</u>





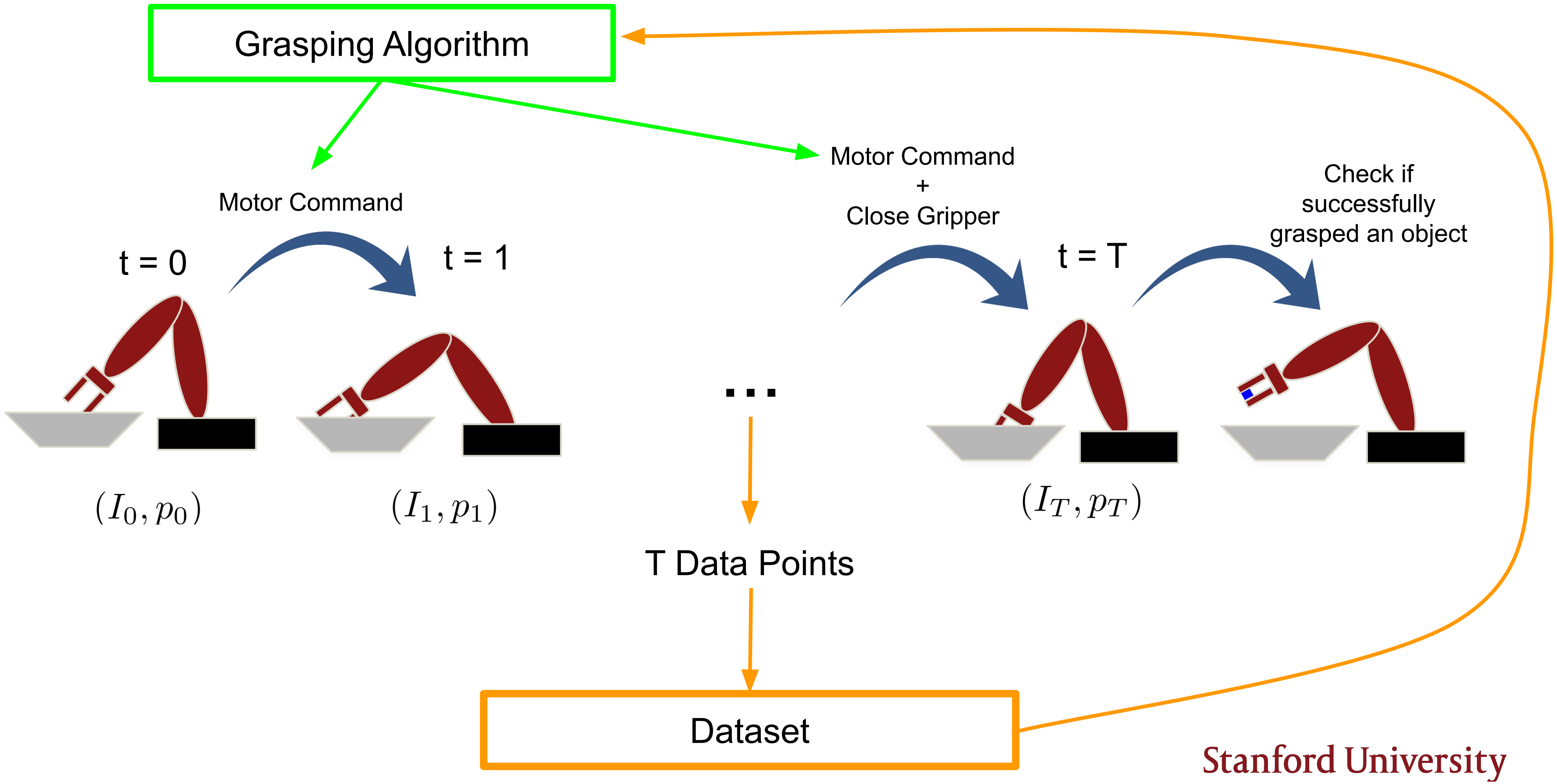
1.7M Grasp Attempts

Self-Supervised Data Collection: Phase 1

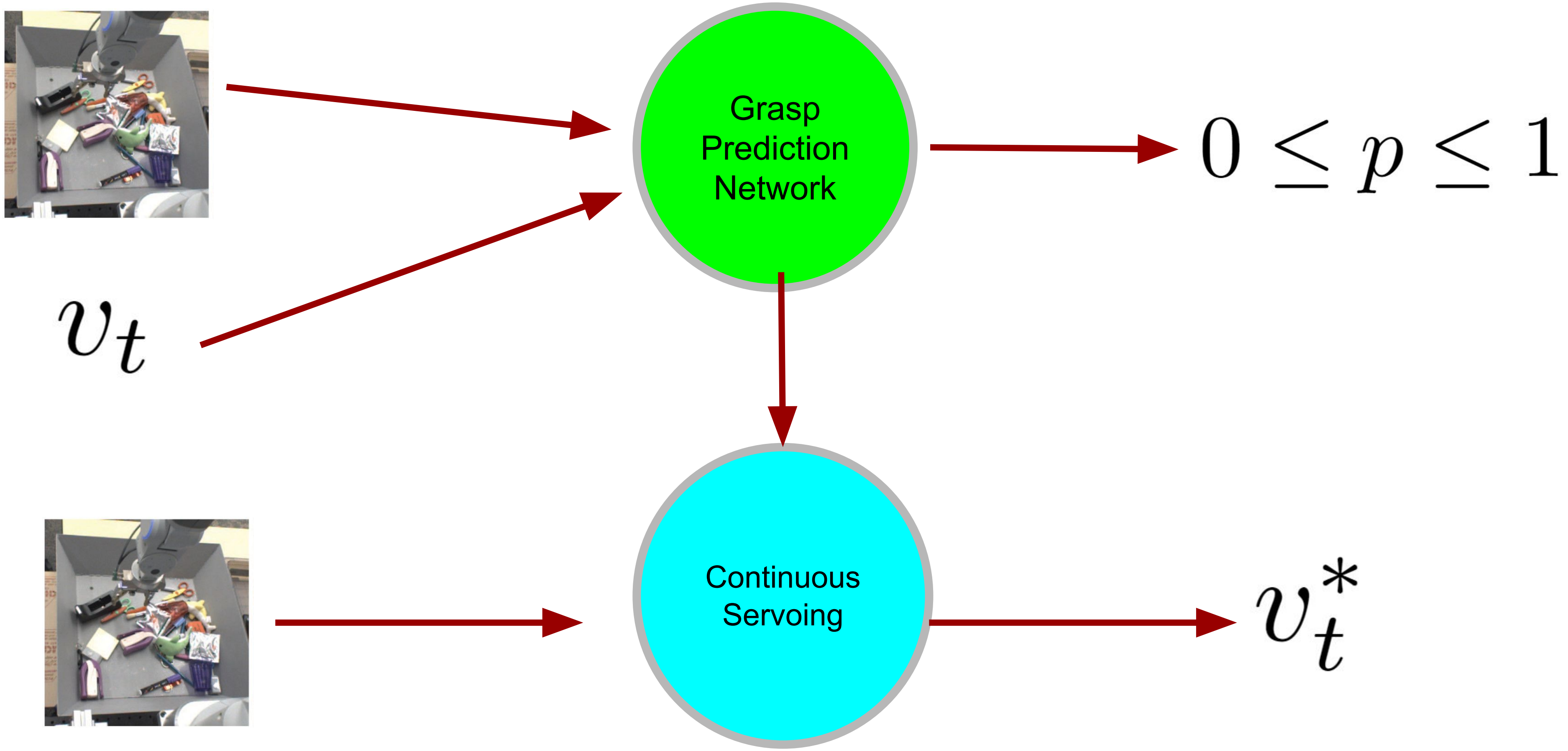


Stanford University

Self-Supervised Data Collection: Phase 2

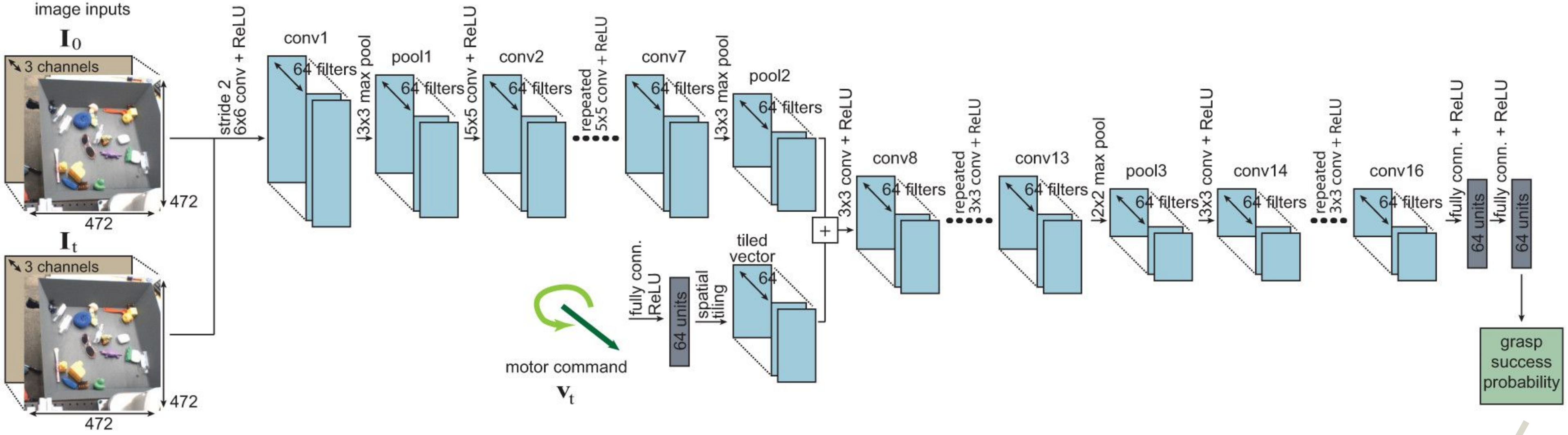


Grasping Algorithm



Grasp Prediction Network

Forward Pass

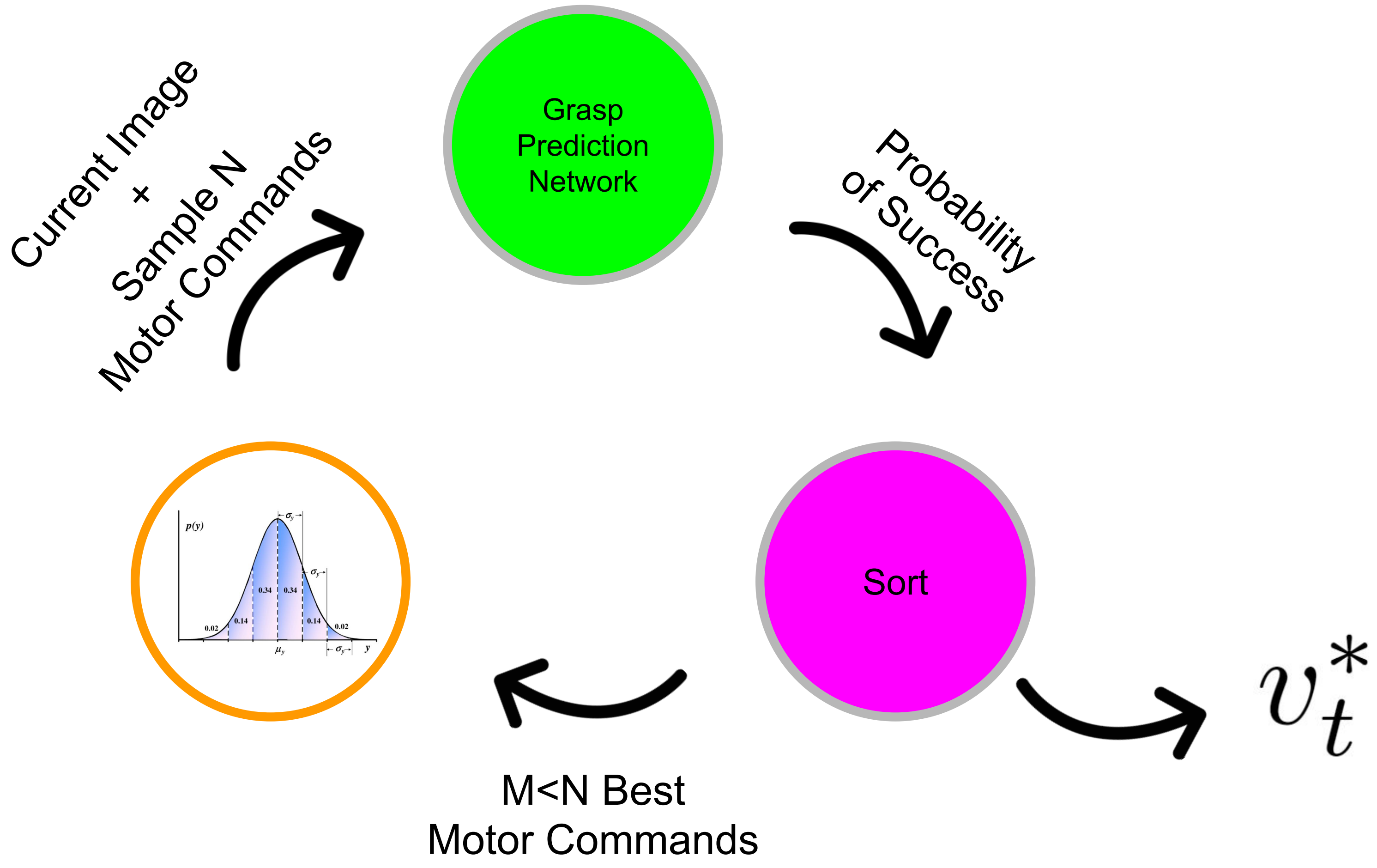


Backward Pass

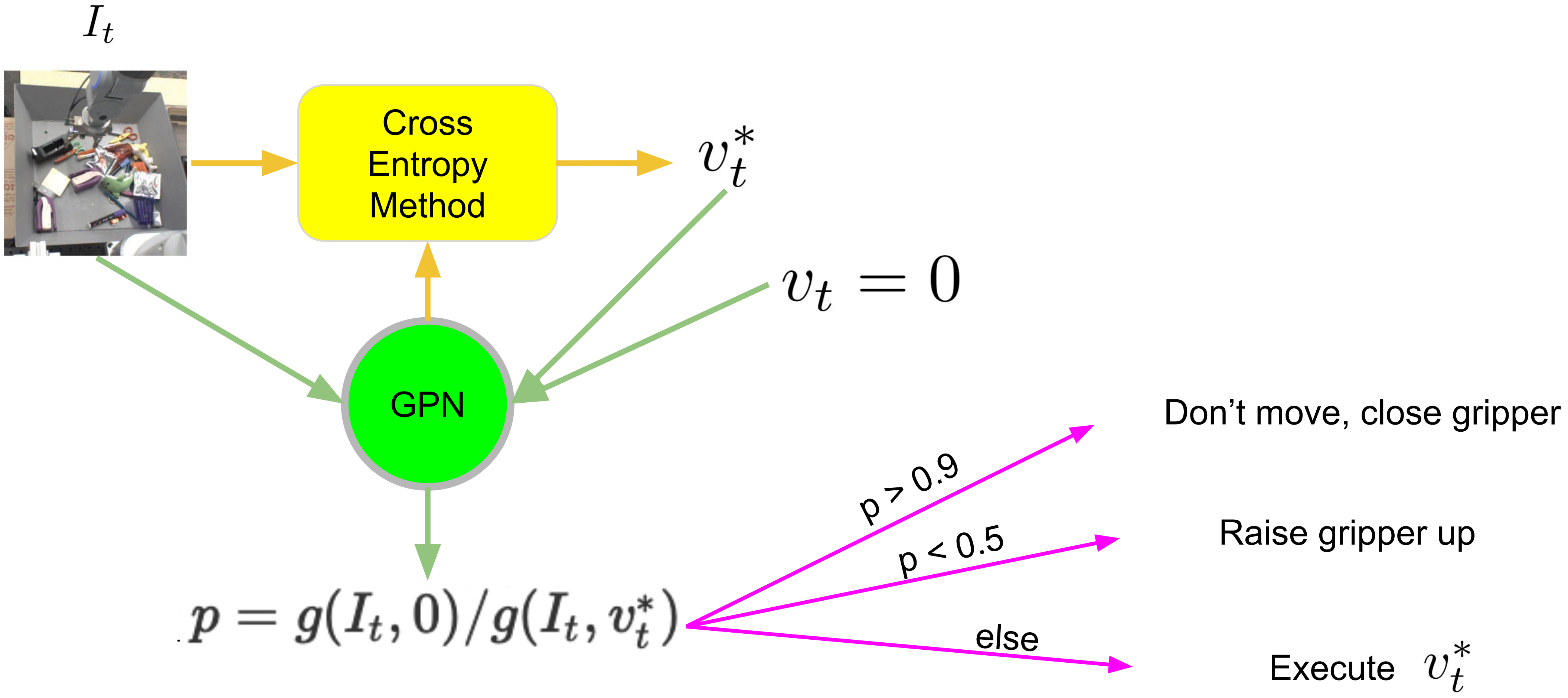
Cross-Entropy Loss

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

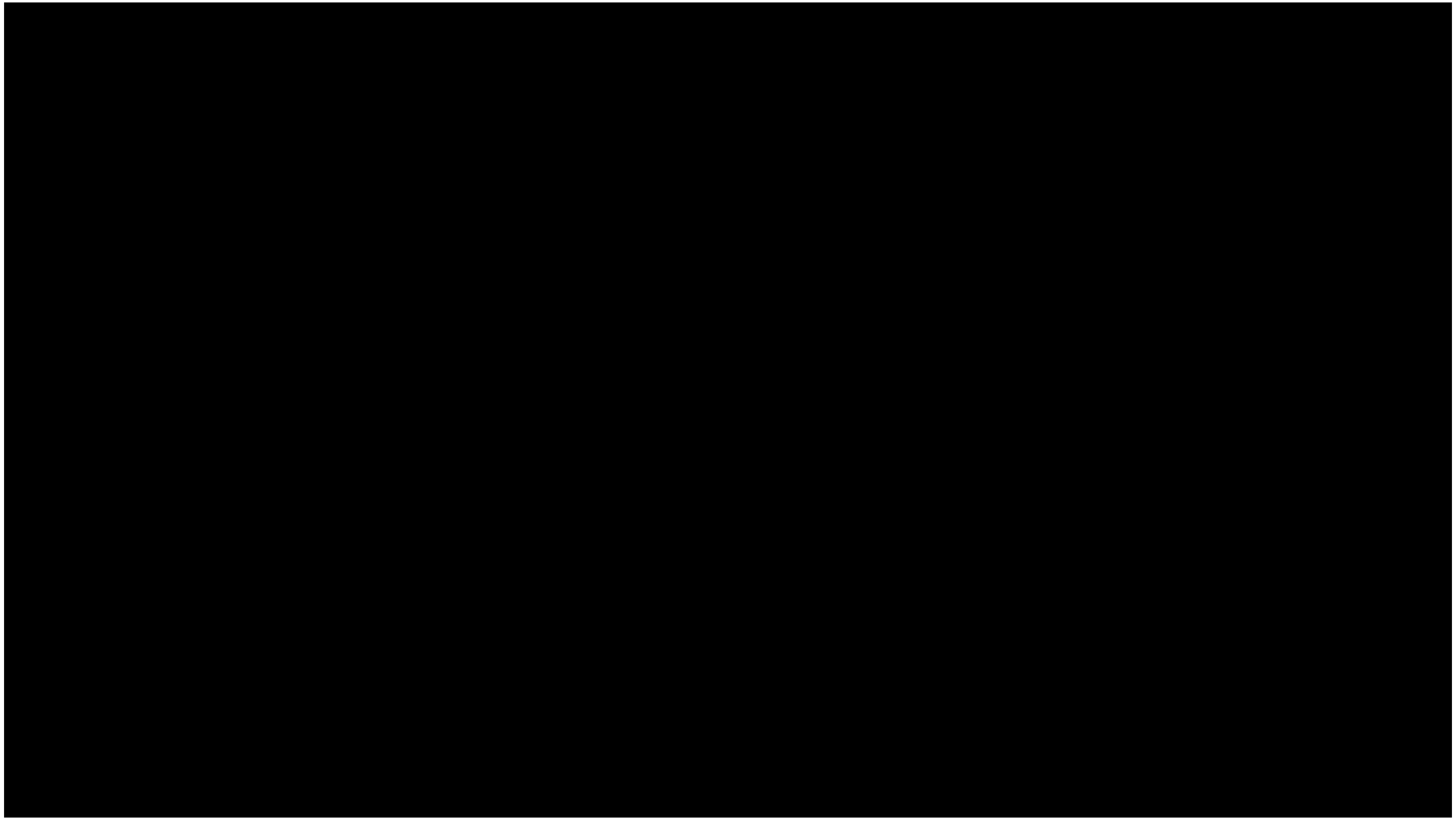
Continuous Servoing: Cross-Entropy Method



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

Unable to react to objects moving

Performs better and requires fewer assumptions

Discussion

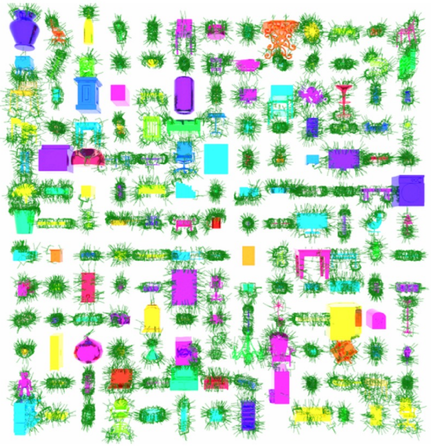
- **End-to-end learning** can achieve good results with **few assumptions**
- It requires **a lot of data** to achieve good performance
  - More tolerable the more **generalizable**
    - Variation in hardware was **small-scale**

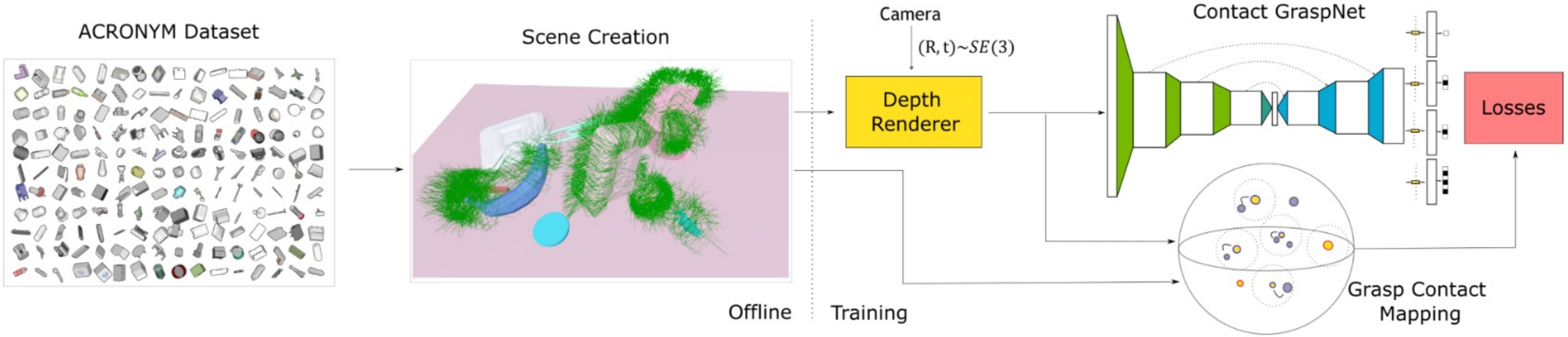
Conclusion: Two Approaches

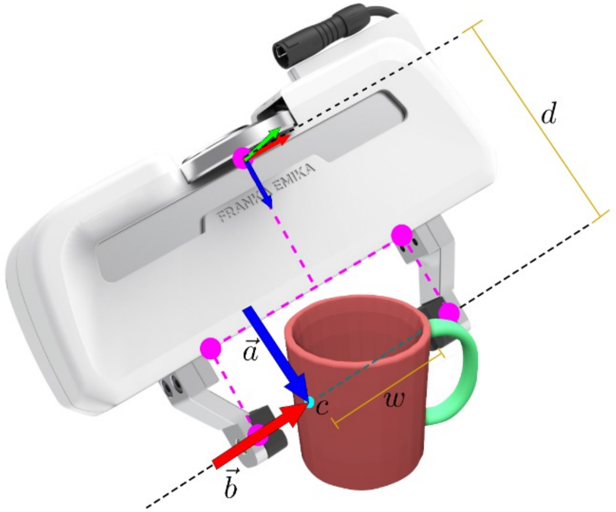
	Dex-Net	Arm Farm
Setup	Single object in simulation	Bin of objects in real world
Number Data Points	13,000 objects, 2.5M grasps	1,100 objects, 1.7M grasps
Data Point	(object, grasp, label = probability of success)	(Image, motor command, label = ground truth success)
Diversity of Objects	Rigid, Opaque	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

Still Missing

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







Suggested Reading

- **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.

” **Dex-Net 2.0: Deep Learning to Pla Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics**" by Mahler et al.. RSS 2017. https://berkeleyautomation.github.io/dex-net

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

Next time

• Interactive Perception