Lecture 12
Manipulation Learning

AncoraSIR.com

Manipulation Learning Problems

AncoraSIR.com

Five Categories of Manipulation Learning

A framework to be covered in the next half of the class

- **Learning to define the state space**
  - The robot must discover the state features and degrees of freedom attached to each object in its environment
- **Learning a transition model of the environment**
  - The robot must learn a model of how its actions affect the task state, and the resulting background cost, for use in planning
- **Learning motor skills**
  - The robot attempts to learn a motor control policy that directly achieves some goal, typically via reinforcement learning
- **Learn to characterize that motor skill**
  - Given the motor skills, the robot learns a description of the circumstances under which it can be successfully executed, and a model of the resulting state change
- **Learning compositional and hierarchical structure**
  - Aims to learn hierarchical knowledge that enables the robot to become more effective at solving new tasks in the family

AncoraSIR.com

Learning to Define the State Space

The robot must discover the state features and degrees of freedom attached to each object in its environment

- Object Representations
  - Types of object representation
    - object pose, object shape, material properties, interactions or relative properties
  - Object representation hierarchy
    - Point-level, Part-level, Object-level representations
- Passive and Interactive Perception
- Learning About Objects and Their Properties
  - Discovering objects
  - Discovering degrees of freedom
  - Estimating object properties
- Feature Learning and Selection
  - Unsupervised vs. supervised approaches

Learning a Transition Model of the Environment

The robot must learn a model of how its actions affect the task state, and the resulting background cost, for use in planning

- Representing and Learning Transition Models
  - Continuous Models
  - Discrete Models
  - Hybrid Models
- Stochasticity and Uncertainty in Transition Models
  - Stochasticity
  - Model Uncertainty
- Self-supervision and Exploration for Learning Transitions
- Transferring and Reusing Transition Models

AncoraSIR.com

Learning Motor Skills

The robot attempts to learn a motor control policy that directly achieves some goal, typically via reinforcement learning

- The Spectrum of Policy Structure
  - Nonparametric Policies
  - Generic Fixed-size Parametric Policies
  - Restricted Parametric Policies
  - Goal-based Policies
- Reinforcement Learning
  - Model-Based RL vs Model-Free RL
  - Value Function Methods vs Policy Search Methods
  - On-Policy vs Off-Policy Learning
  - Exploration Strategies
- Imitation Learning
  - Behavioral Cloning
  - Reward Inference
  - Learning from Observation
  - Corrective Interactions
- Skill Transfer
  - Direct Skill Re-use
  - Parameterized Skills
  - Metalearning
  - Domain Adaptation
  - Sequential Transfer and Curriculum Learning
- Safety and Performance Guarantees
  - Performance Metrics
  - Classes of Guarantees and Bounding Methods

AncoraSIR.com

Learn to Characterize that Motor Skill

The robot learns a description of the circumstances under which it can be successfully executed, and a model of the resulting state change

- Pre- and Postconditions as Propositions and Predicates
  - Classifier Representation
  - Distribution Representation
  - Modularity and Transfer
- Learning Pre- and Postcondition Groundings
- Skill Monitoring and Outcome Detection
  - Learning Goal and Error Classifiers
  - Detecting Deviations from Nominal Sensory Values
  - Verifying Predicates
- Predicates and Skill Synthesis
  - Representing and Synthesizing Skill Parameters
  - Preconditions and Affordances

Learning Compositional and Hierarchical Structure

Aims to learn hierarchical knowledge that enables the robot to become more effective at solving new tasks in the family

- The Form of a Motor Skill
- Segmenting Trajectories into Component Skills
  - Segmentation Based on Skill Similarity
  - Segmentation Based on Specific Events
- Discovering Skills While Solving Tasks
- Learning Decision-Making Abstractions
  - Learning Abstract Policy Representations
  - Learning Abstract State Spaces

Learning a Transition Model

仿生设计与学习实验室
Bionic Design & Learning Lab

AncoraSIR.com

Learning a Transition Model of the Environment

The robot must learn a model of how its actions affect the task state, and the resulting background cost, for use in planning

Transition Models of the Environment

Continuous, Discrete and Hybrid Models

Stochasticity ← **Probabilistic Models** → **Uncertainty**

Learning a New Model

Self-supervision and Exploration

Reuse an Existing Model

Transferring and Reusing

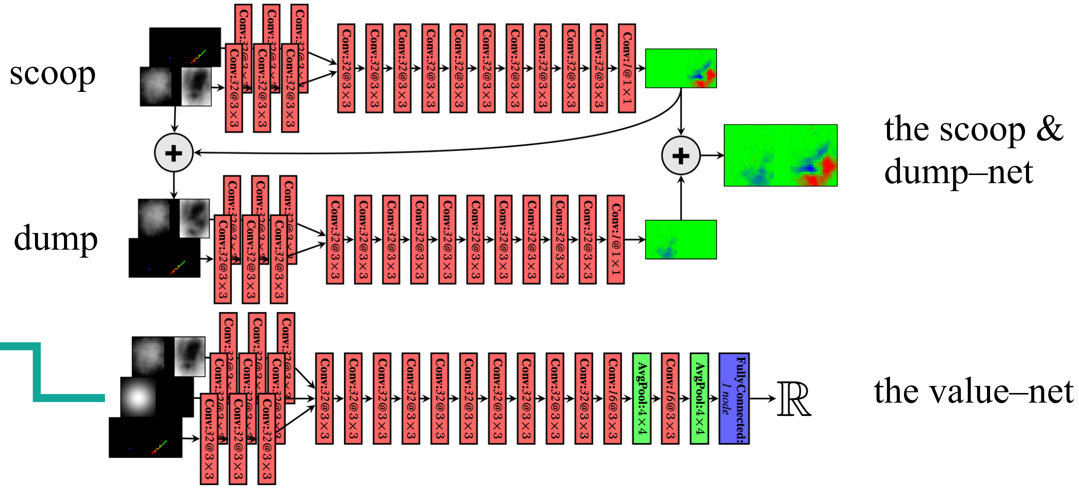
AncoraSIR.com

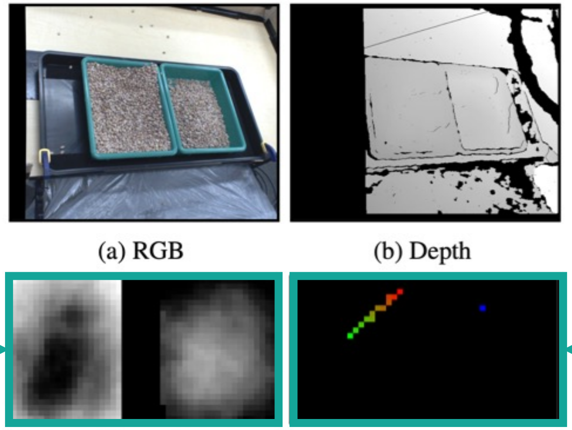
Continuous Transition Models

*Regression methods can be used to learn **low-level** transition models for predicting the next state as a set of continuous values, even when the set of actions is discrete*

Scoop & Dump Task $l(h_0, a_0, \ldots, a_T, h_g) = \|\mathcal{F}(h_0, a_0, \ldots, a_T) - h_g\|_1$

h_0: a given ***initial state*** of the environment
h_g: a given ***goal state*** of the environment
$a_{0,\ldots,T}$: a series of ***robot actions***
$\|Norm\|_1$: L1 norm to be minimized as distance to the goal state
$\mathcal{F}(h_0, a_{0,\ldots,T}) = \boxed{h_{T+1}}$ applies actions sequentiall to reach h_{T+1}



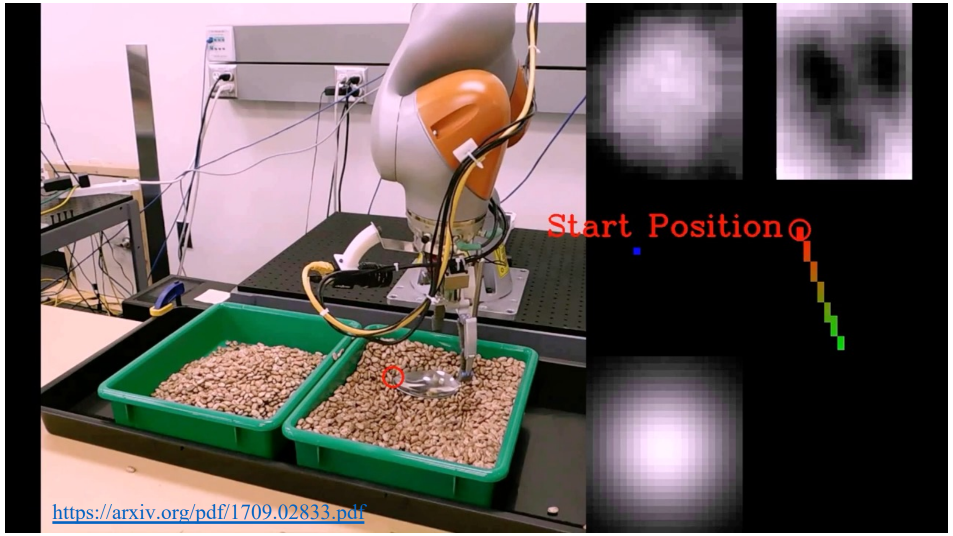


(a) RGB (b) Depth

(c) Height-Map (d) Action Map

Robot action (a 9D vector)

- *Scoop action*
  - the start location (2D)
  - the start angle (1D)
  - the end location (2D)
  - the end angle (1D)
  - the roll angle (1D)
- *Dump action*
  - the dump location (2D)

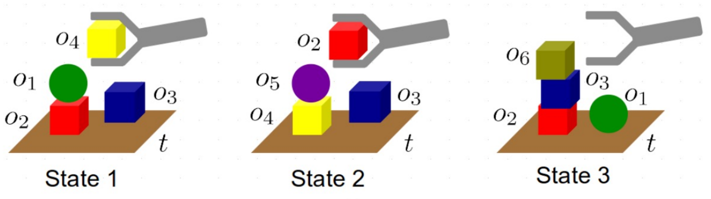


https://arxiv.org/pdf/1709.02833.pdf

AncoraSIR.com

Discrete Transition Models

*Learn transitions for tasks with discrete state and action spaces, typically capturing **high-level** tasks*



(a)

State	Enumerated	Factored	Relational
1	s_1	$on\_o_1\_o_2$, $on\_o_2\_t$, $on\_o_3\_t$, $inhand\_o_4$	$on(o_1,o_2)$, $on(o_2,t)$, $on(o_3,t)$, $inhand(o_4)$, $ball(o_1)$, $cube(o_2)$, $cube(o_3)$, $cube(o_4)$, $table(t)$
2	s_2	$on\_o_3\_t$, $on\_o_4\_t$, $on\_o_5\_o_4$, $inhand\_o_2$	$on(o_3,t)$, $on(o_4,t)$, $on(o_5,o_4)$, $inhand(o_2)$, $cube(o_2)$, $cube(o_3)$, $cube(o_4)$, $ball(o_5)$, $table(t)$
3	s_3	$on\_o_1\_t$, $on\_o_2\_t$, $on\_o_3\_o_2$, $on\_o_6\_o_3$	$on(o_1,t)$, $on(o_2,t)$, $on(o_3,o_2)$, $on(o_6,o_3)$, $cube(o_2)$, $cube(o_3)$, $cube(o_6)$, $ball(o_1)$, $table(t)$

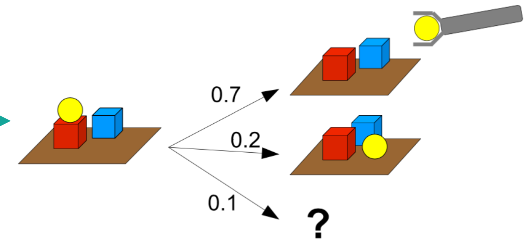
(b)

$$\mathcal{E} = \{$$

$$\begin{aligned}
grab(d):&\quad cube(a), cube(b), ball(c), ball(d), table(t), on(a,t), on(b,t), on(c,a), \underline{on(d,b)}\dots\\
\rightarrow&\quad cube(a), cube(b), ball(c), ball(d), table(t), on(a,t), on(b,t), on(c,a), \underline{inhand(d)}\dots\\
puton(t):&\quad cube(a), cube(b), ball(c), ball(d), table(t), on(a,t), on(b,t), on(c,a), \underline{inhand(d)}\dots\\
\rightarrow&\quad cube(a), cube(b), ball(c), ball(d), table(t), on(a,t), on(b,t), on(c,a), \underline{on(d,t)}\dots\\
grab(c):&\quad cube(a), cube(b), ball(c), ball(d), table(t), on(a,t), on(b,t), \underline{on(c,a)}, on(d,t)\dots\\
\rightarrow&\quad cube(a), cube(b), ball(c), ball(d), table(t), on(a,t), on(b,t), \underline{on(d,t)}, \underline{inhand(c)}\dots\\
\vdots&\quad \vdots
\end{aligned}$$

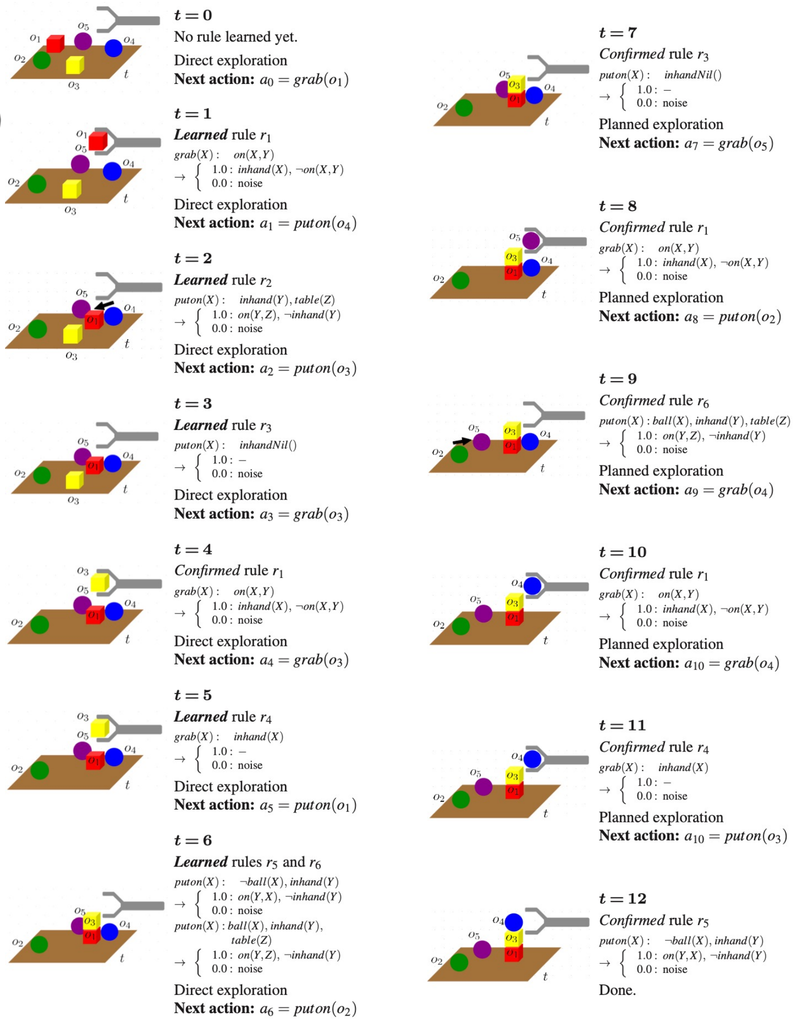
$$\}$$

$$P(s'\,|\,s,a) = P(s'\,|\,s,r) = \sum_{i=1}^{m_r} p_{r,i} P(s'|\Omega_{r,i},s) + p_{r,0} P(s'\,|\,\Omega_{r,0},s)$$



$$grab(X): \quad on(X,Y),\ ball(X),\ cube(Y),\ table(Z) \rightarrow \begin{cases} 0.7 & : \quad inhand(X),\ \neg on(X,Y) \\ 0.2 & : \quad on(X,Z),\ \neg on(X,Y) \\ 0.1 & : \quad noise \end{cases}$$

$$a_r(X) : \phi_r(X) \rightarrow \begin{cases} p_{r,1} & : \quad \Omega_{r,1}(X) \\ & \vdots \\ p_{r,m_r} & : \quad \Omega_{r,m_r}(X) \\ p_{r,0} & : \quad \Omega_{r,0} \end{cases}$$

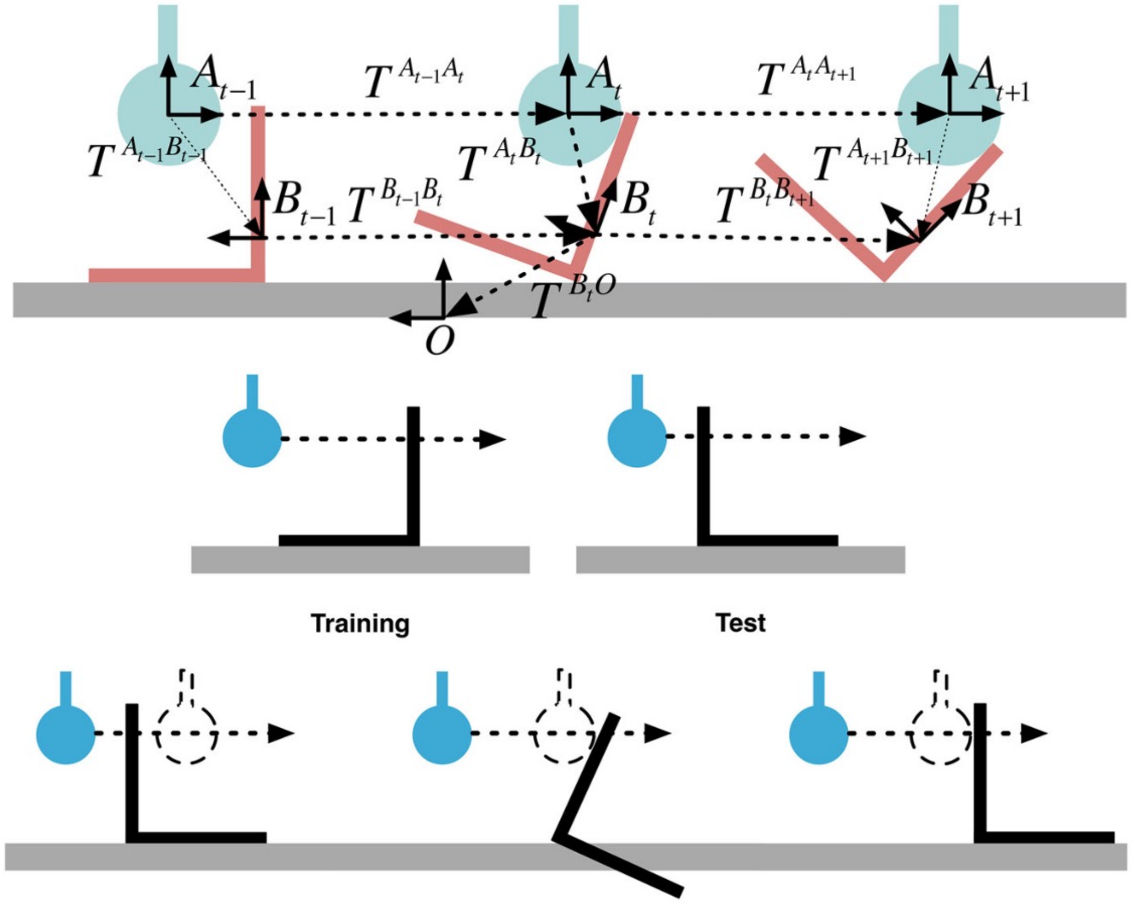


AncoraSIR.com

https://ipvs.informatik.uni-stuttgart.de/mlr/papers/12-lang-JMLR.pdf

Stochasticity

A stochastic process exhibits a randomness in its state transitions



https://sci-hub.tw/10.1007/s10514-016-9571-3

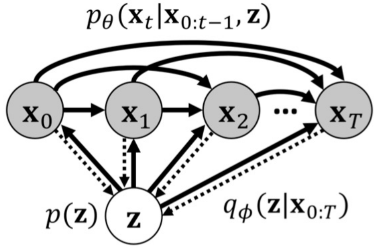
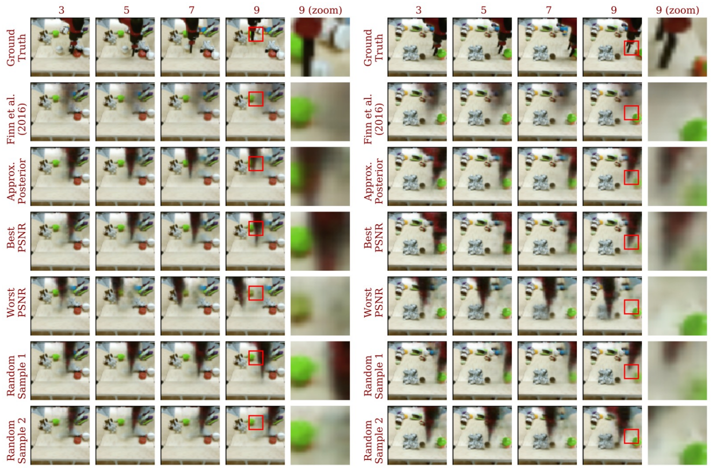


Figure 2: Probabilistic graphical model of stochastic variational video prediction, assuming time-invariant latent. The generative model predicts the next frame conditioned on the previous frames and latent variables (solid lines), while the variational inference model approximates the posterior given all the frames (dotted lines).



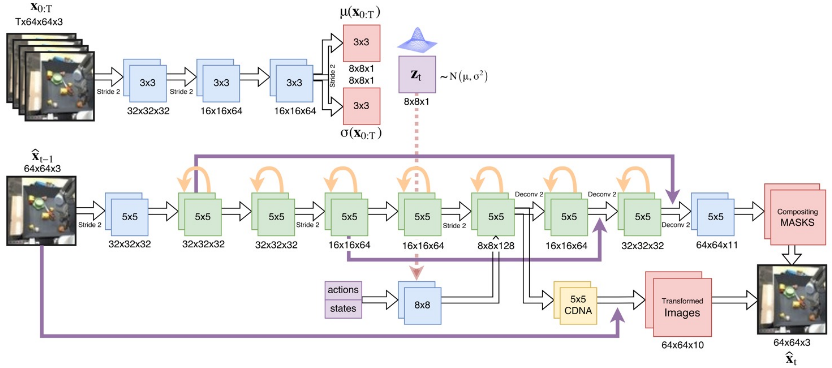


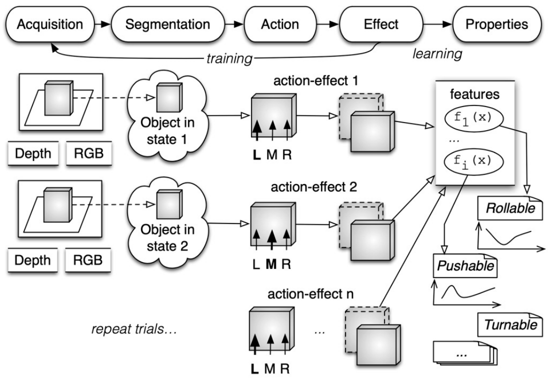
Figure 3: Architecture of SV2P. At training time, the inference network (top) estimates the posterior $q_\phi(\mathbf{z}|\mathbf{x}_{0:T}) = \mathcal{N}(\mu(\mathbf{x}_{0:T}), \sigma(\mathbf{x}_{0:T}))$. The latent value $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}_{0:T})$ is passed to the generative network along with the (optional) action. The generative network (from Finn et al. (2016)) predicts the next frame given the previous frames, latent values, and actions. At test time, $\mathbf{z}$ is sampled from the assumed prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

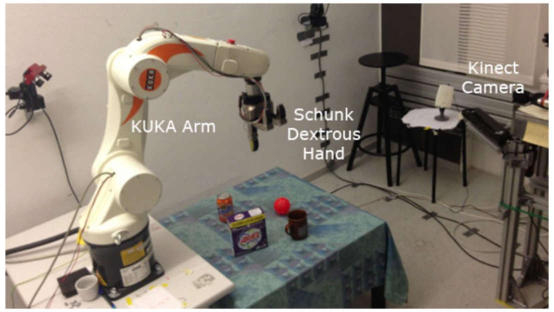
https://arxiv.org/pdf/1710.11252.pdf

AncoraSIR.com

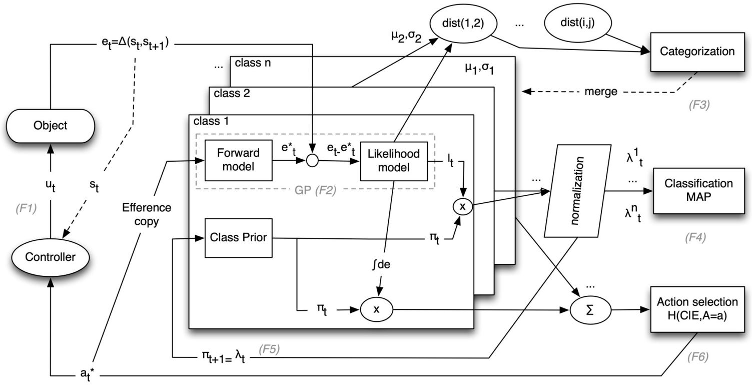
Model Uncertainty

The outcome of an action may be uncertain due to the robot's limited knowledge of the process

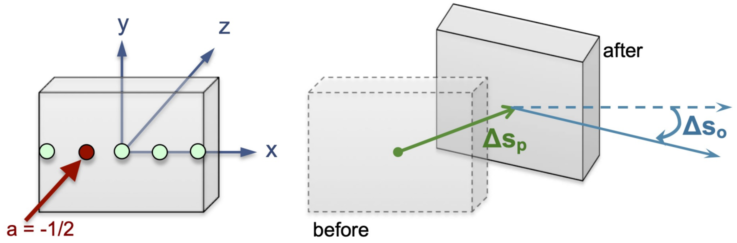


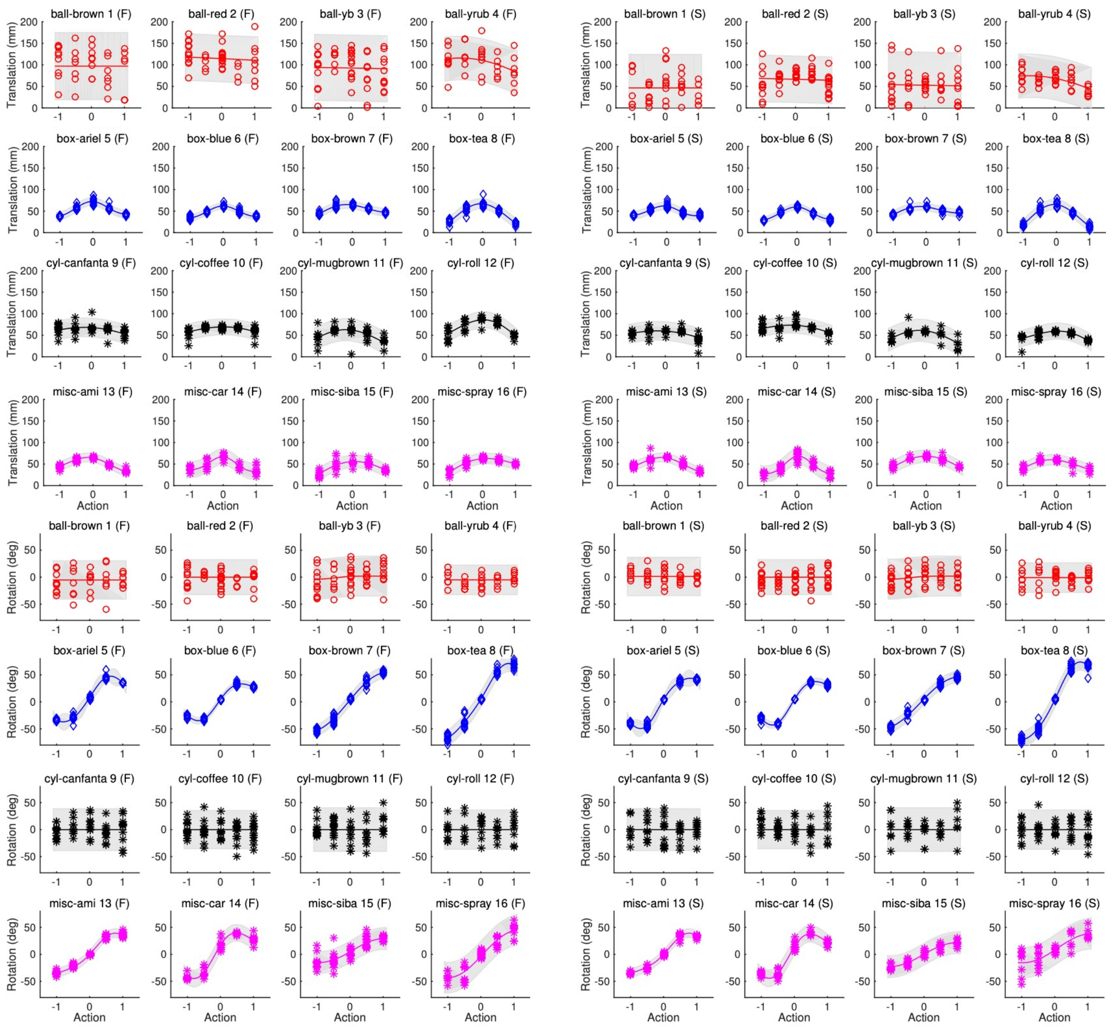


A framework of sensorimotor learning, adopting a modular approach for interactive classification and functional categorization of objects.



A model for pushing

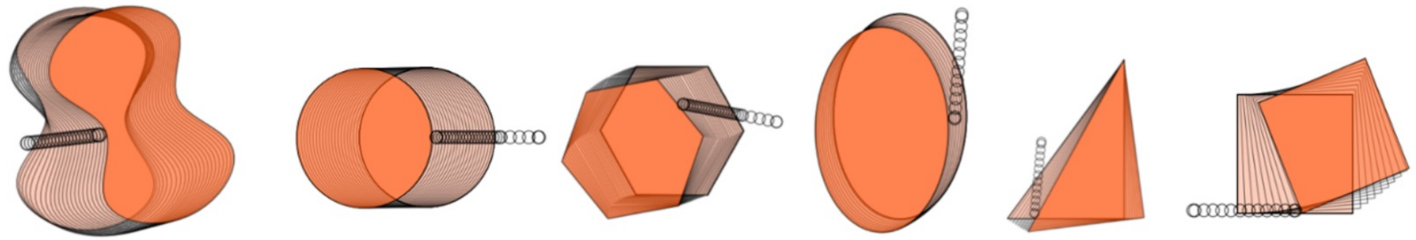




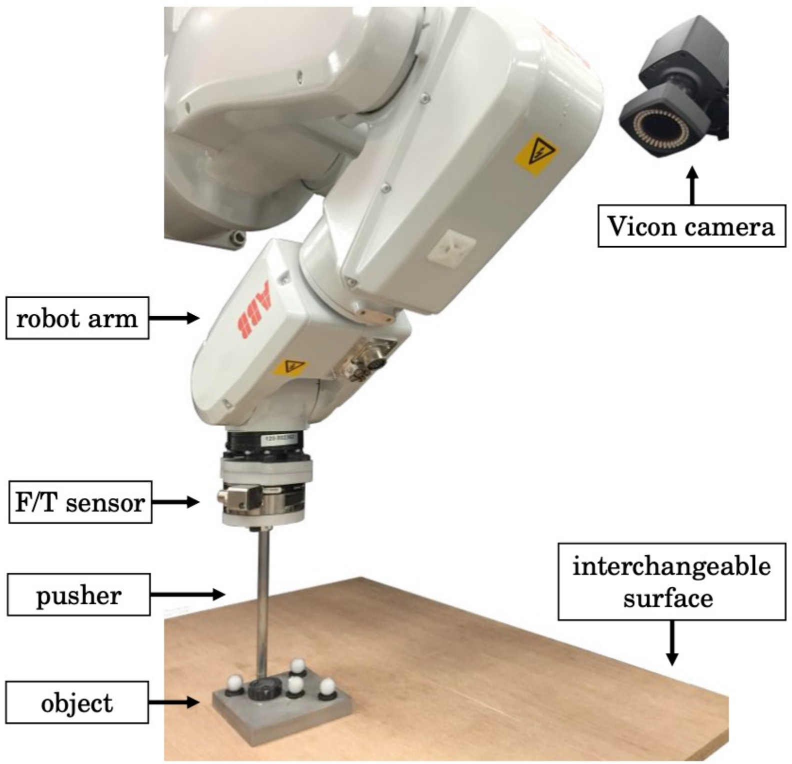
http://kth.diva-portal.org/smash/get/diva2:922036/FULLTEXT02.pdf

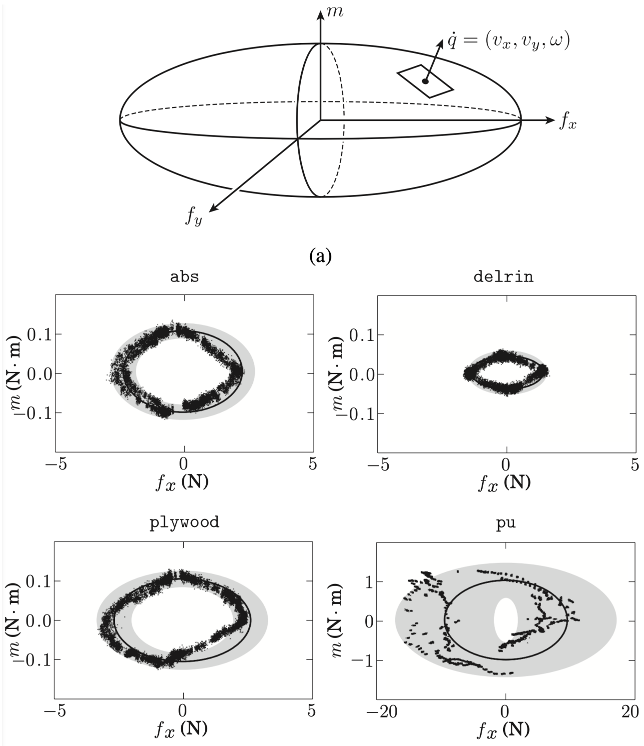
Self-supervision & Exploration

Transition models are usually learned in a self-supervised manner and the robot may adopt different exploration strategies for acquiring samples



Shape	 `rect1`, `rect2`, `rect3`, `hex`, `ellip1`, `ellip2`, `ellip3` , `butter` , `tri1`, `tri2`, `tri3`
Surface	`abs`, `derlin`, `polywood`, `pu`
Speed (mm/s)	10, 20, 50, 75, 100, 150, 200, 300, 400, 500
Acceleration (ms^{-2})	0, 0.1, 0.2, 0.5, 0.75, 1, 1.5, 2, 2.5
Initial contact	33 points for `tri1-3` and `hex`, 40 for `ellip1-3` and `butter`, and 44 for `rect1-3`
Initial push direction	0°, 20°, 40°, 60°, 80°, -20°, -40°, -60°, -80°

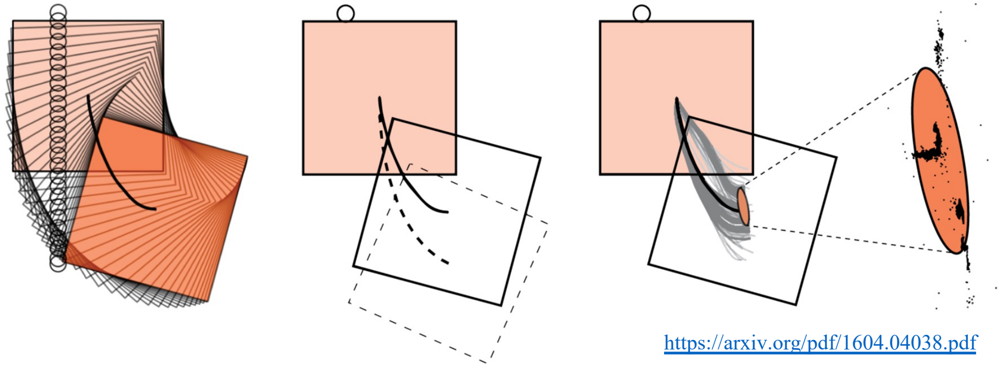




(a)

(b)

Object	Mass (g)	Dimension (mm)	Moment of inertia (g·m^2)
`rect1`	837	w:90, h:90	1.13
`rect2`	1045	w:90, h:112.5	1.81
`rect3`	1251	w:90, h:135	2.74
`hex`	983	circumradius: 60.5	1.50
`ellip1`	894	w:105, h:105	1.23
`ellip2`	1110	w:105, h:130.9	1.95
`ellip3`	1334	w:105, h:157	2.97
`butter`	1197	w1:95.3, w2:54.7, h: 156	2.95
`tri1`	803	leg1: 125.9, leg2: 125.9	1.41
`tri2`	983	leg1: 125.9, leg2: 151.0	2.11
`tri3`	1133	leg1: 125.6, leg2: 176.5	2.96

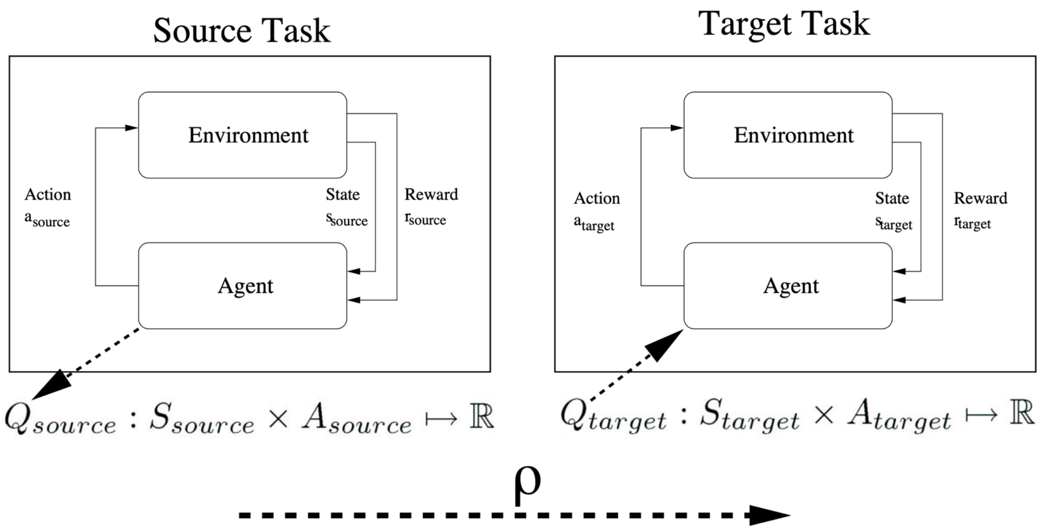


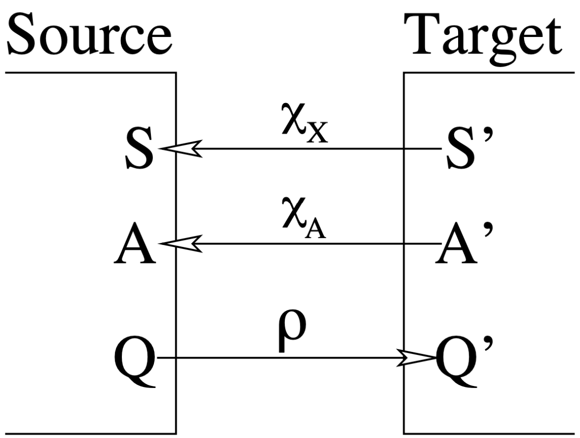
https://arxiv.org/pdf/1604.04038.pdf

(a) (b) (c)

Transferring and Reusing Transition Models

Transition models are not inherently linked to a specific task and can therefore often be transferred and reused between different manipulation tasks and even task families.



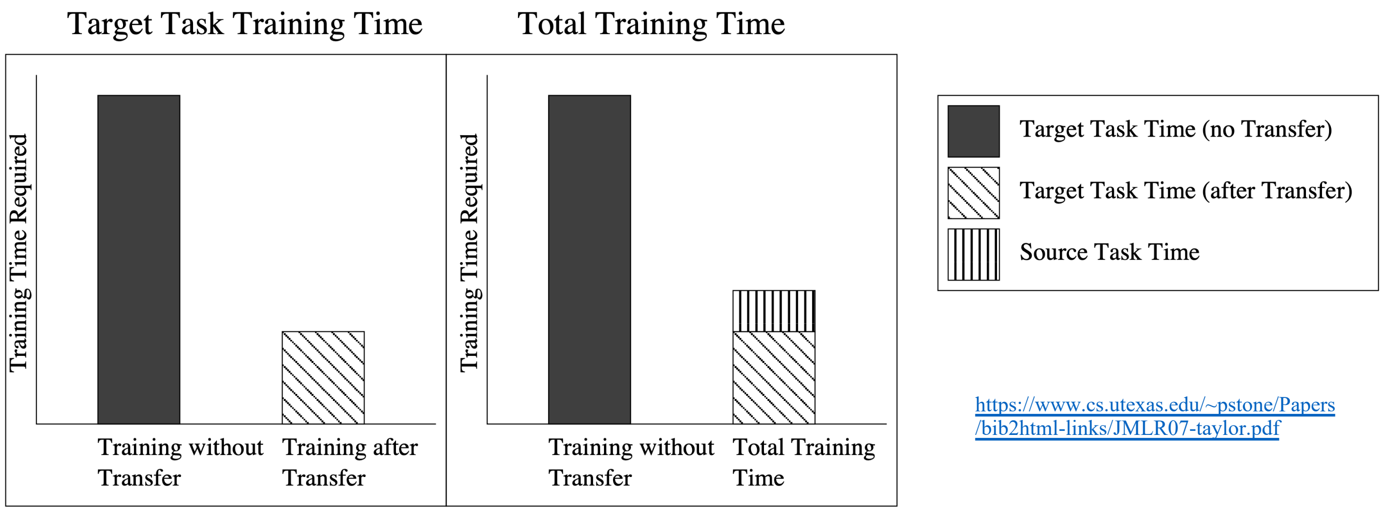


$$\chi_X(x_{i,target}) = x_{j,source}.$$

$$\chi_A(a_{i,target}) = a_{j,source}.$$

Evaluation of Transfer

1. **Asymptotic Performance**: Measure the performance after convergence in the target task.
2. **Initial Performance**: Measure the initial performance in the target task.
3. **Total Reward**: Measure the total accumulated reward during training in the target task.
4. **Area Ratio**: Measure the area between the transfer and non-transfer learning curves.
5. **Time-to-Threshold**: Measure the time needed to reach a performance threshold in the target task.



https://www.cs.utexas.edu/~pstone/Papers/bib2html-links/JMLR07-taylor.pdf

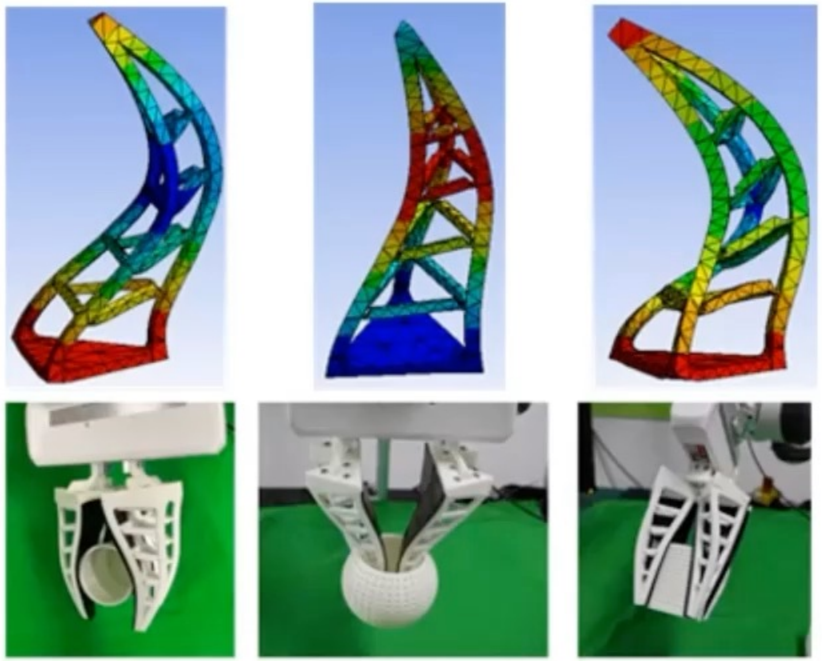
Learning to Touch

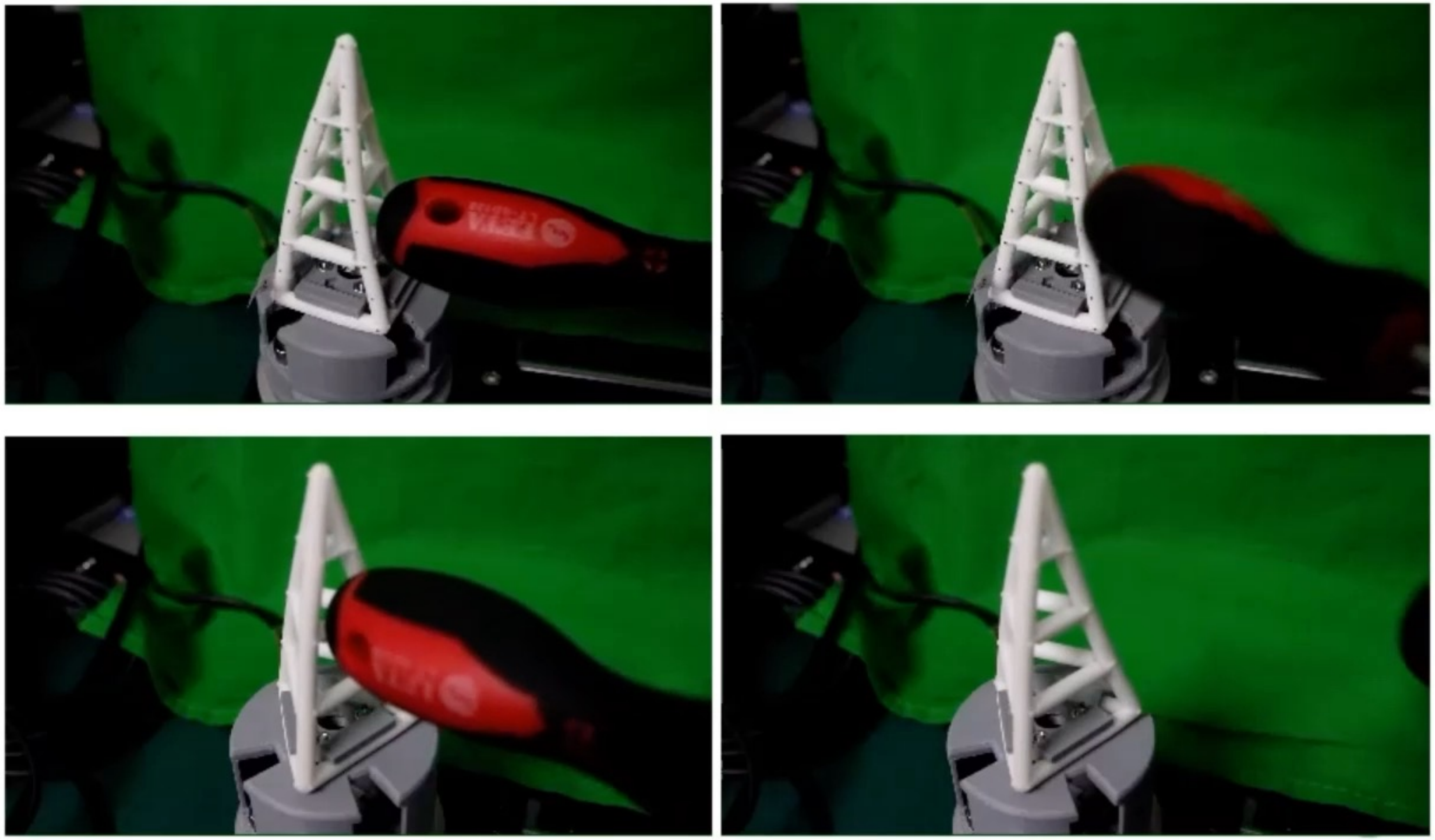
Example set 1 with the DeepClaw Toolkit

AncoraSIR.com

Omni-directional Adaptation

***Solid2Soft Technology** that transforms almost any structure with omni-directional adaptation*



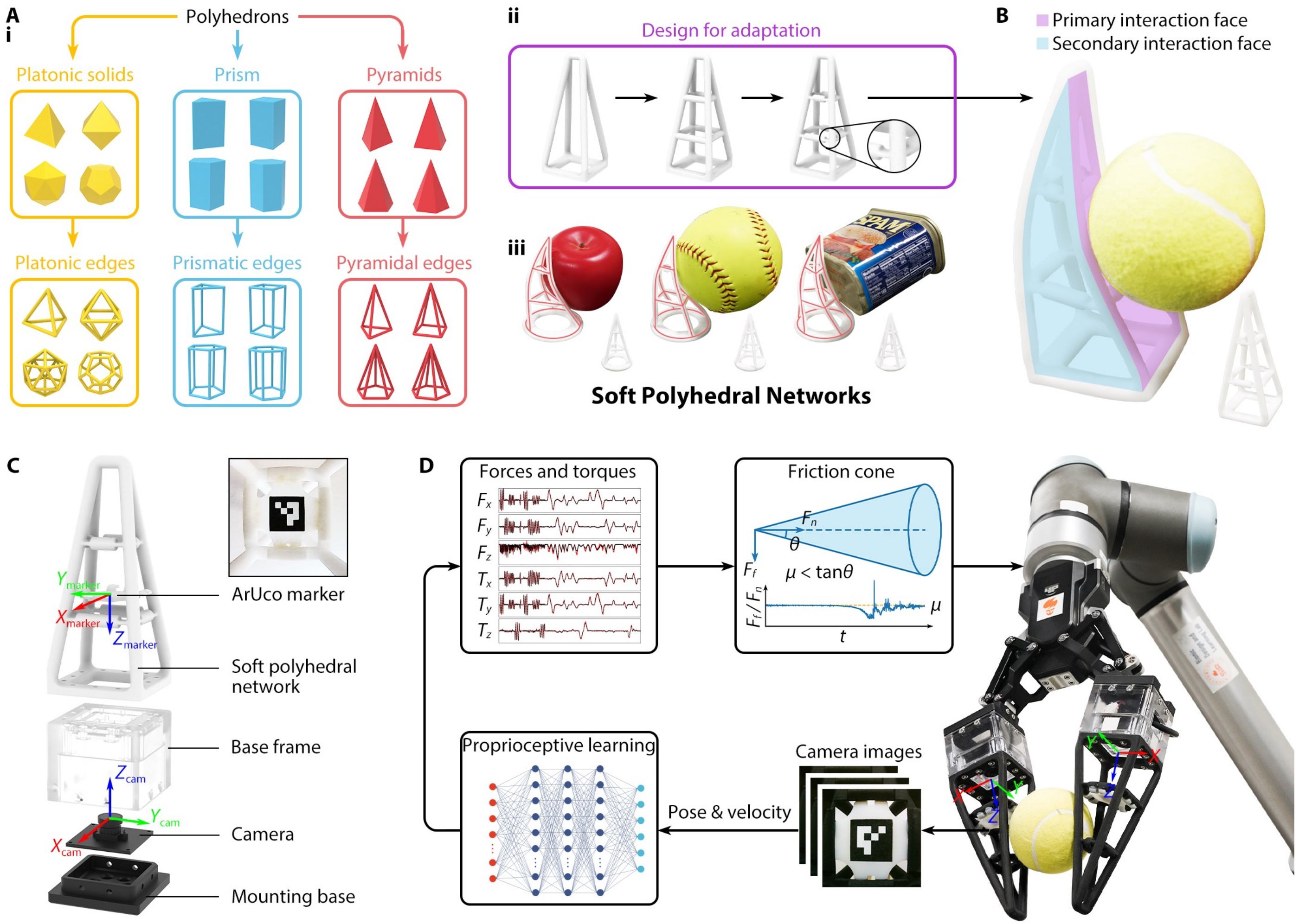


- **Ultra-low-cost** & **soft** 3D structure that capable of omni-directional adaptation with any shape and texture.
- **Simple integration** with any picking system for computationally efficient physical interaction

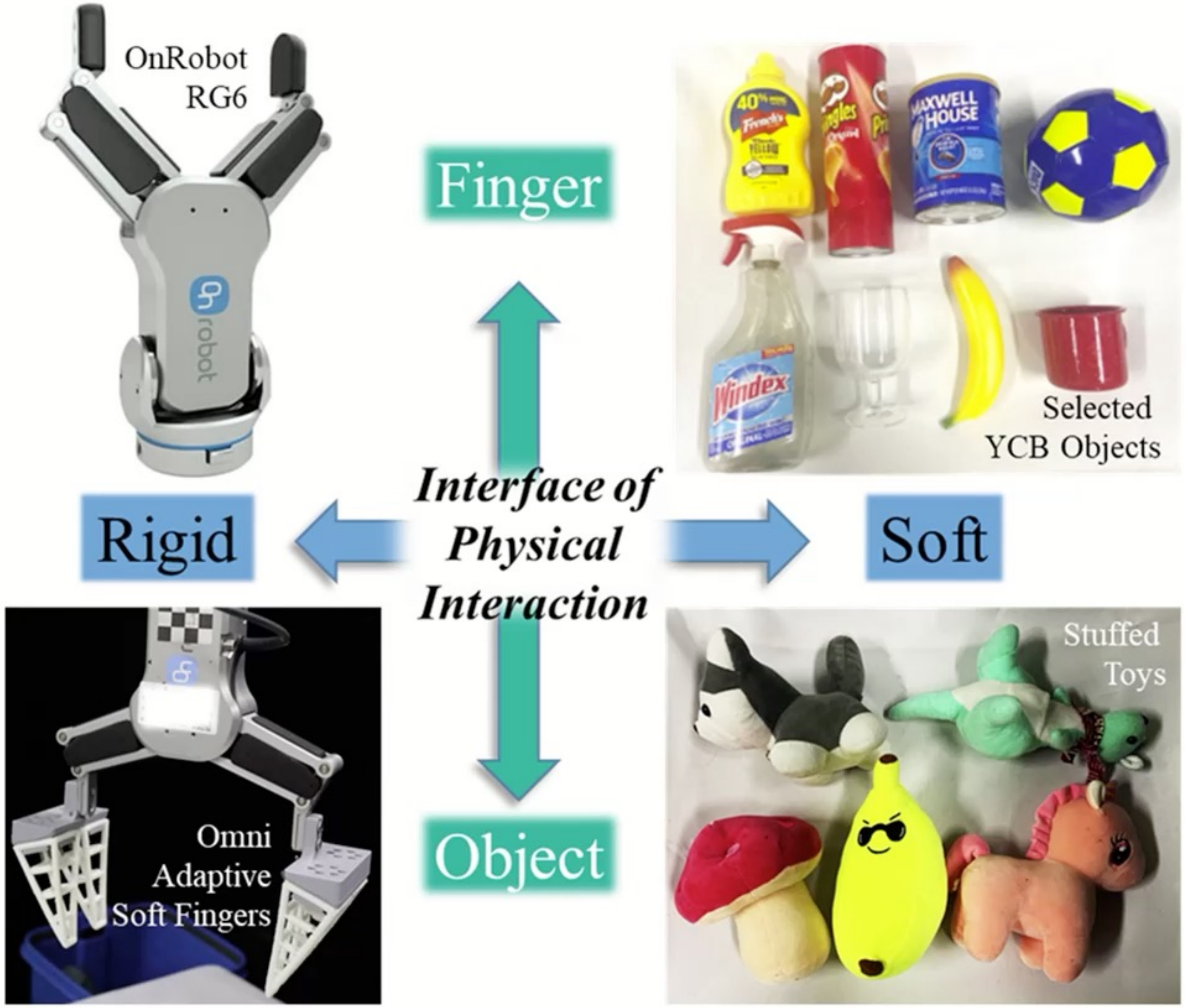
AncoraSIR.com

A General Framework for Vision-based Touch Learning

The DeepClaw toolkit you've been using for the course project

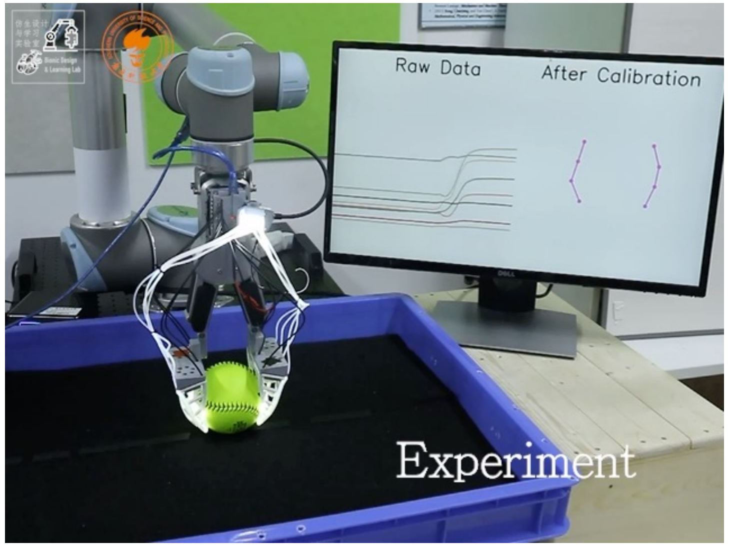


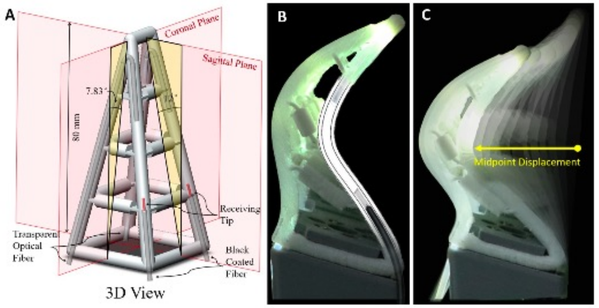
Can We Generalize Learning through Physical Interaction?

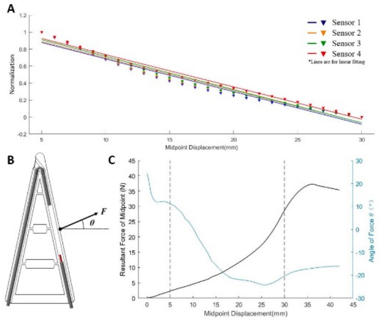


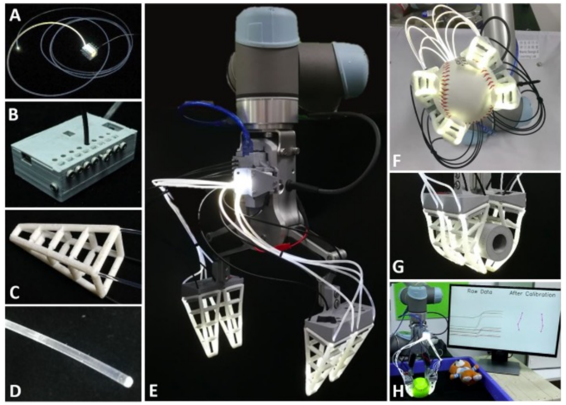
AncoraSIR.com

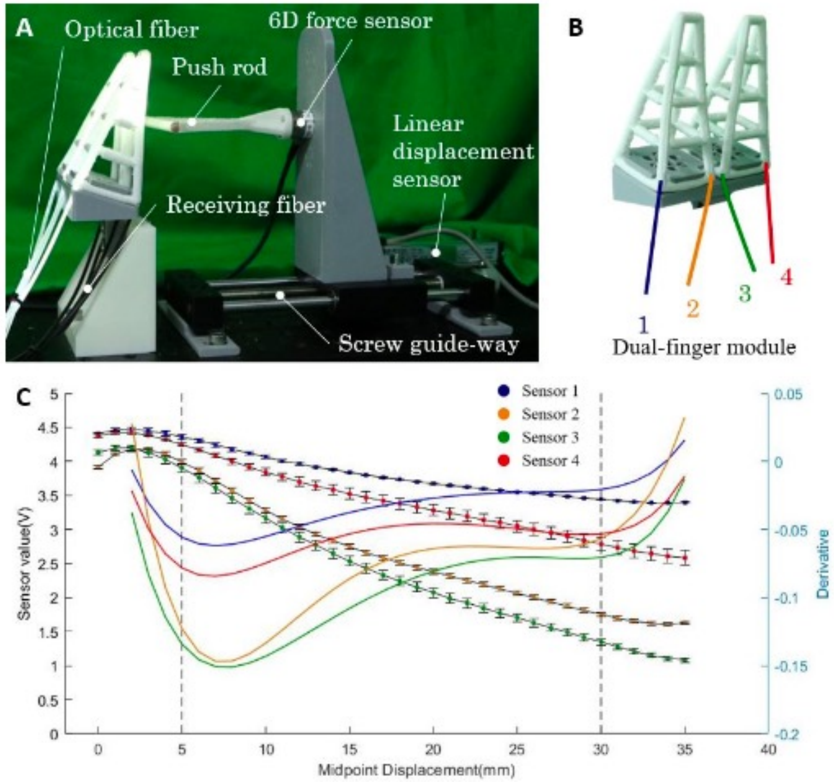
Touch Sensory Integration with Optical Fibers

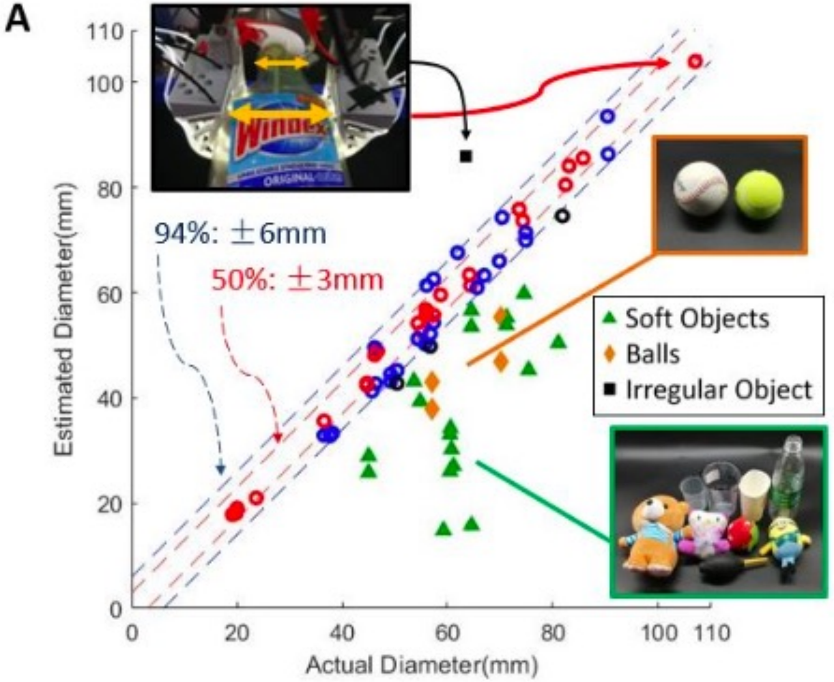




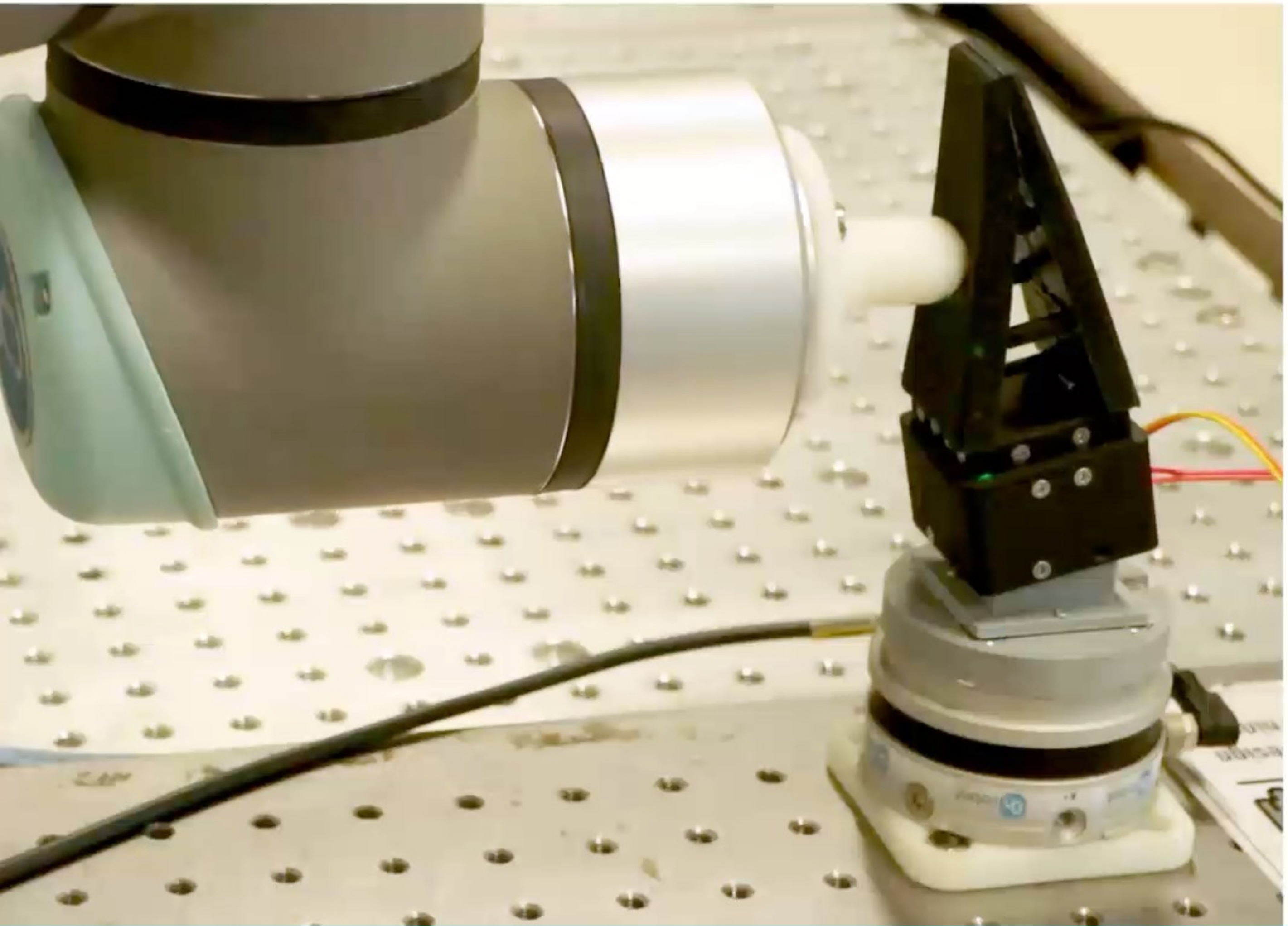


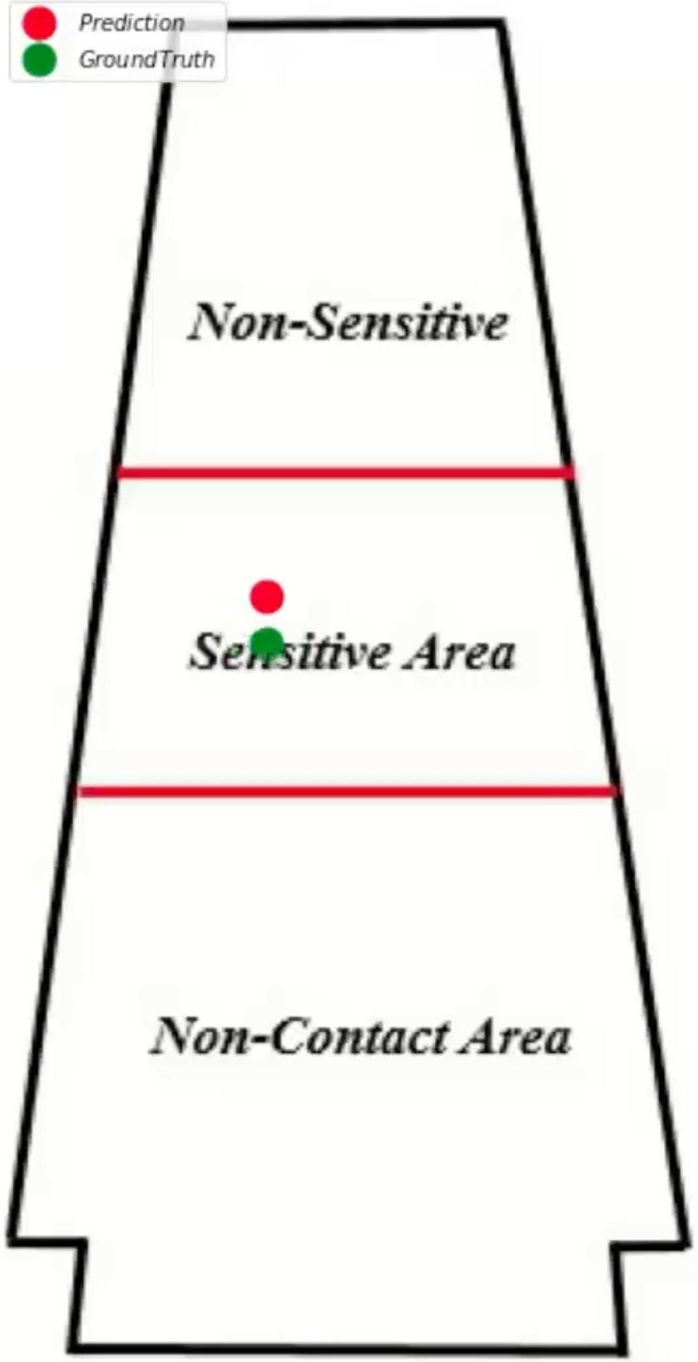




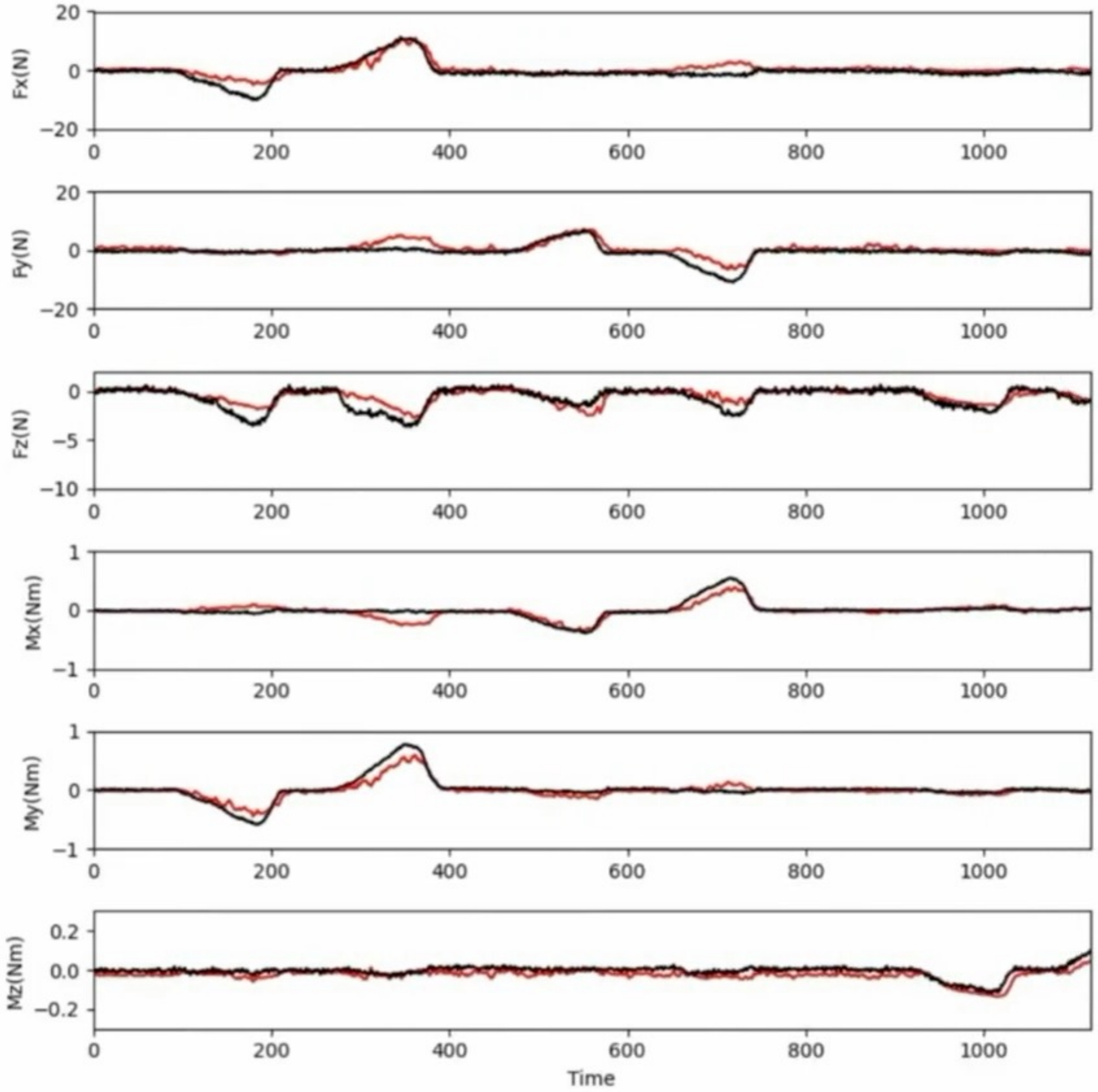


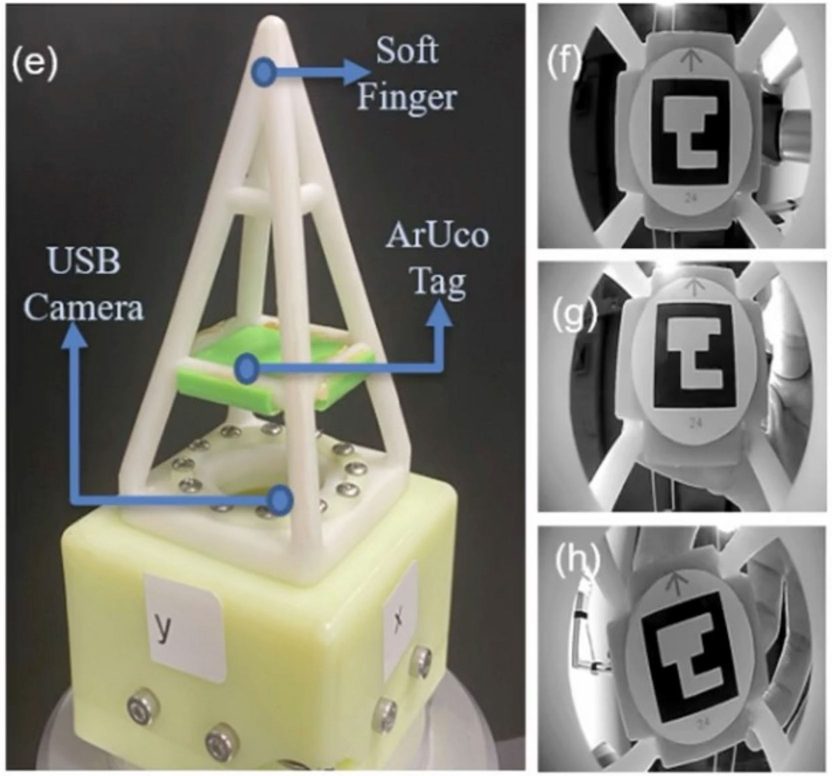
Real-time Prediction of Interactive Position





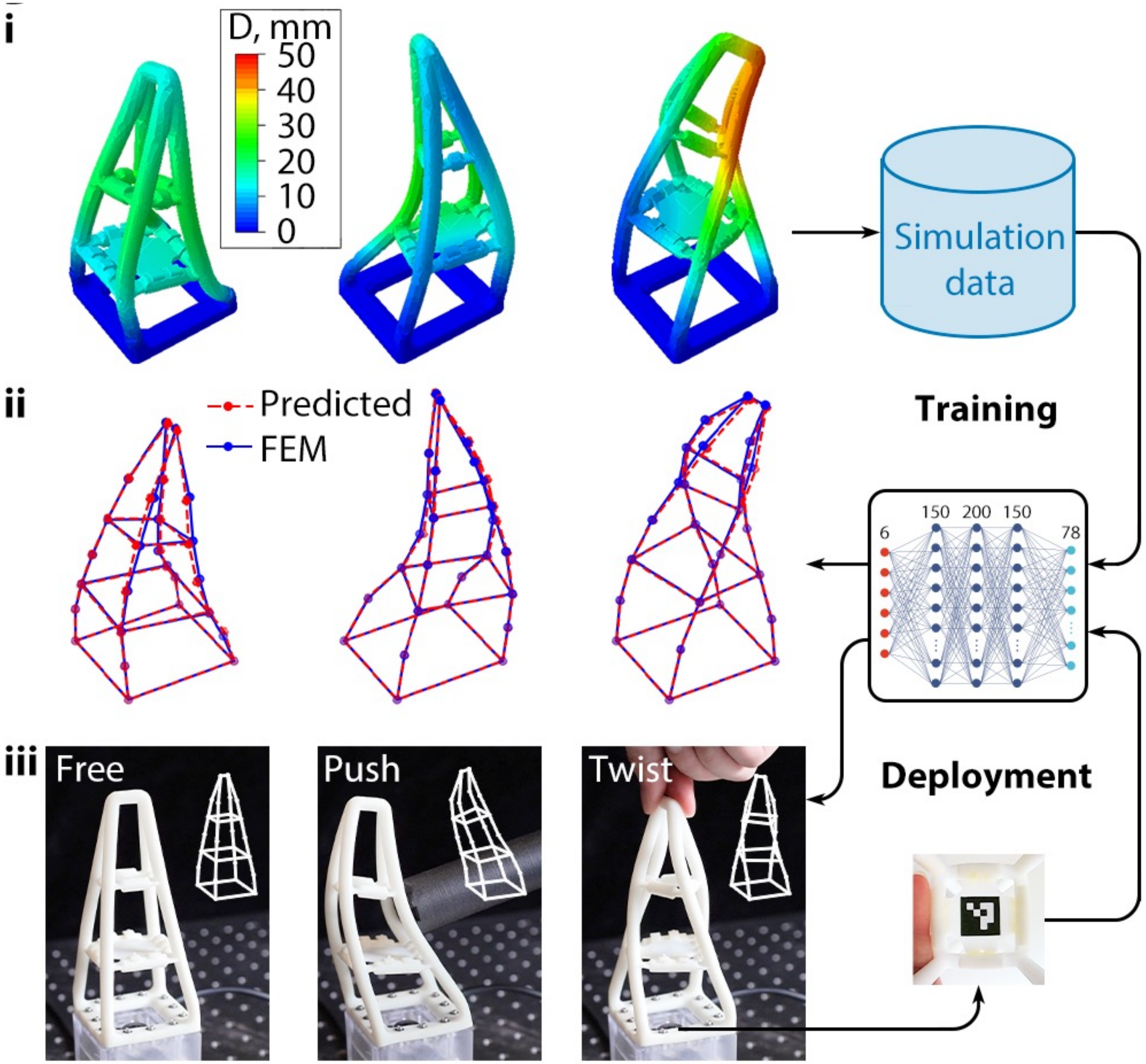
Learning 6D Force/Torque with the 6D Finger Design





We further designed a simple 3-layer network that can estimate all 6D F/T of the finger in realtime by visually tracking the finger's 6D pose inside.

Sim2Real Learning with FEM



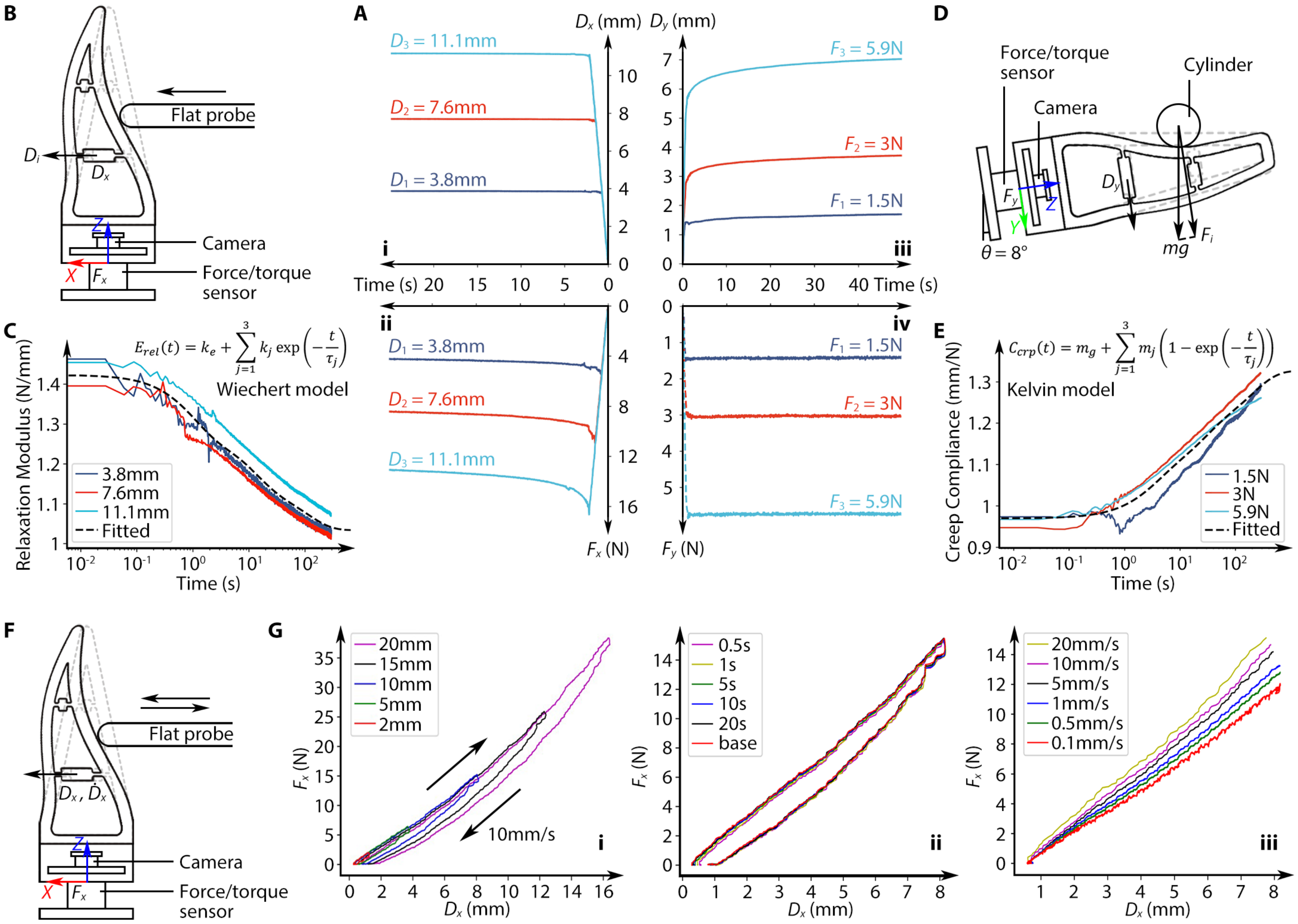
- After (i) collecting FEM simulation data of the soft network under external compressions at various angles and magnitudes,
- (ii) we train a Sim2Real multi-layer perceptron (MLP) to reproduce spatial movement of 26 key points on the soft network.
- (iii) When deployed to the actual soft network, the MLP predictions show a good alignment with observations in scenarios of free standing, pushing, and twisting.

Proprioceptive Learning with Soft Polyhedral Networks

S1. Sim2Real Proprioception for Adaptive Kinesthesia

SUSTech

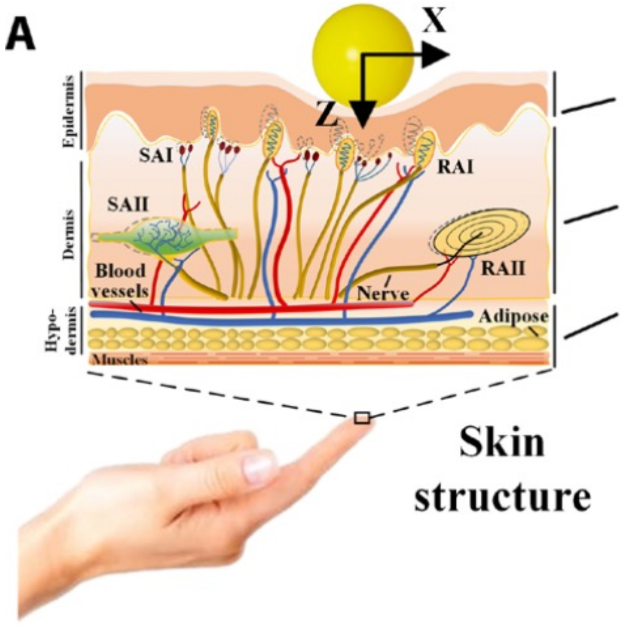
Learning Friction with Material's Viscoelasticity



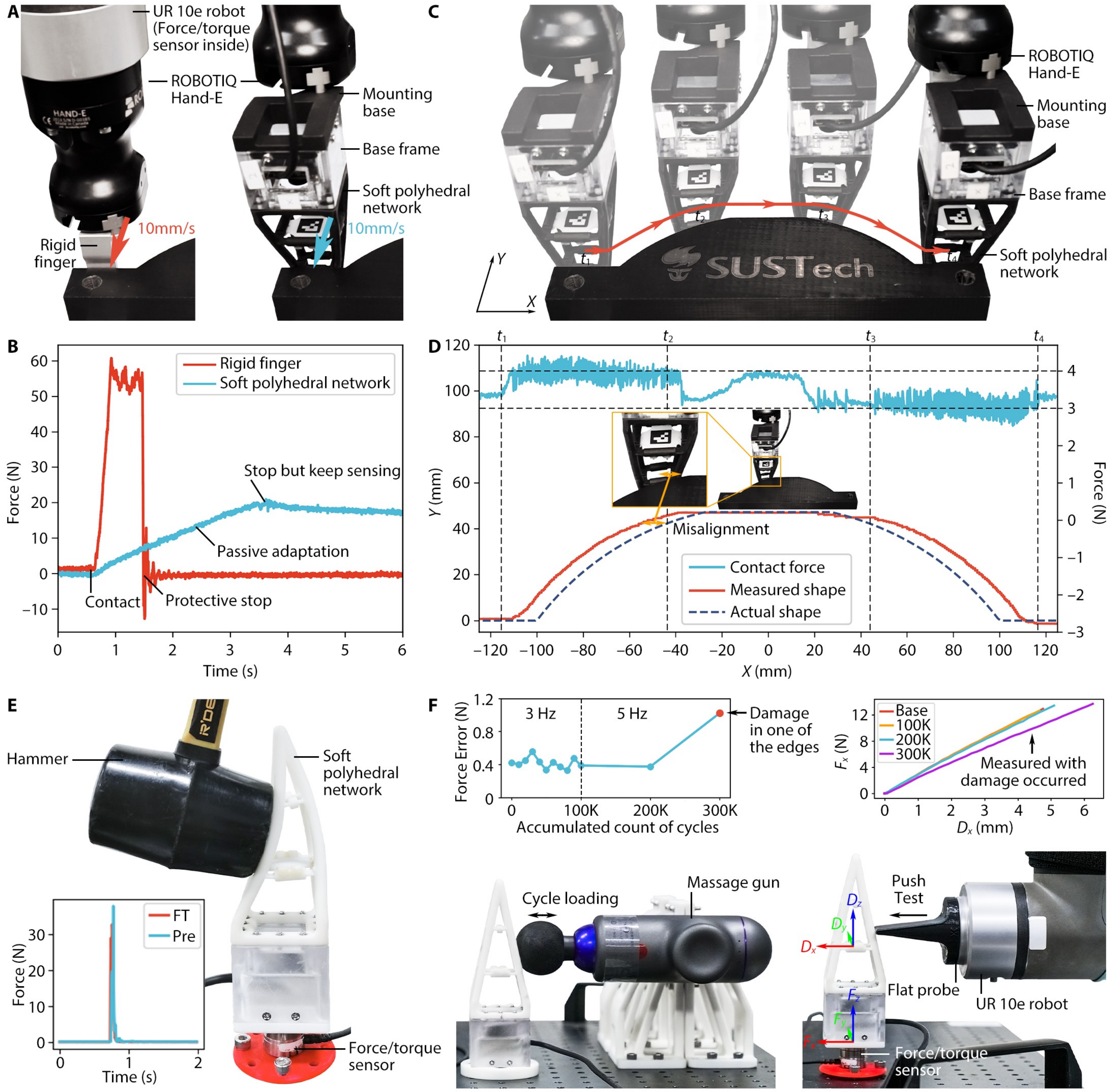
Proprioceptive Learning with Soft Polyhedral Networks

S2. Viscoelastic Sensitive Grasping for Friction Estimation

SUSTech



SUSTech
Southern University of Science and Technology



Tactile Reconstruction

Proprioceptive Learning with Soft Polyhedral Networks

S4. Impact Absorption and Tactile Reconstruction using Visual Force Learning

Proprioceptive Learning with Soft Polyhedral Networks

S5. High-speed Fatigue Testing with a Massage Gun

Proprioceptive Learning with Soft Polyhedral Networks

S3. Competitive Grasping for an Orange

SUSTech

仿生设计与学习实验室 Bionic Design & Learning Lab

Proprioceptive Learning with Soft Polyhedral Networks

S6. Reprogrammable Teleoperation for Peg-in-hole

Learning Skill Policies

AncoraSIR.com

The Final Learning Goal for a Robot

To acquire a behavior, or skill controller, that will perform a desired manipulation task

Skill Controller

Task (*Goal*)

		Specific	A Class
Experience (*Examples*)	From the world (or a model of it)	**Reinforcement Learning**	**Skill Transfer**
	From demonstration trajectory	**Imitation Learning**	

Uncertainty (Safety)

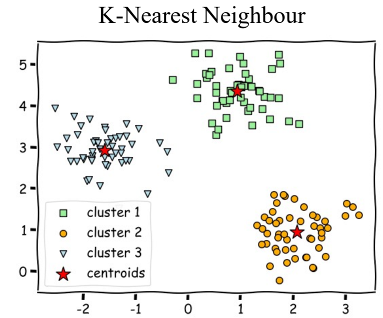
Performance Guarantees (Metrics)

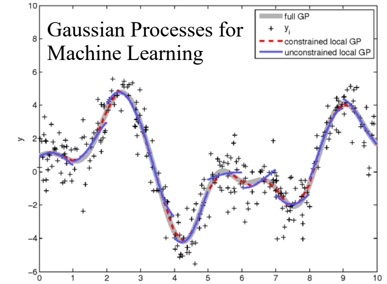
Working?

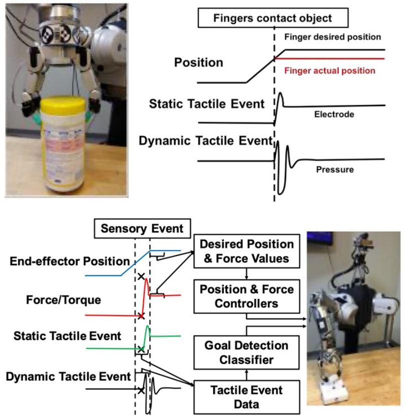
AncoraSIR.com

The Spectrum of Policy Structure

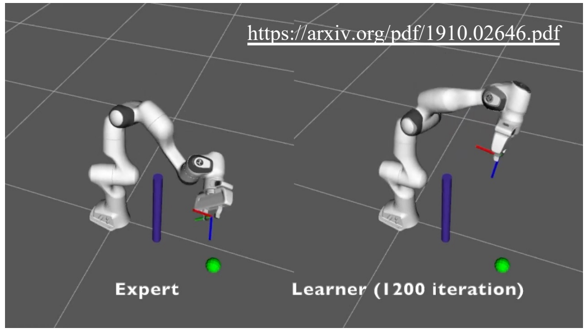
The choice of policy representation is a critical design decision for any robot learning algorithm

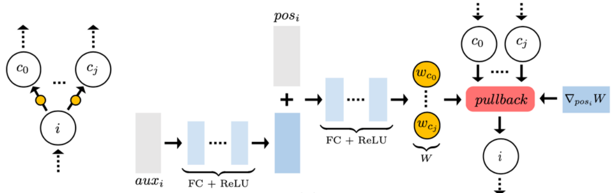






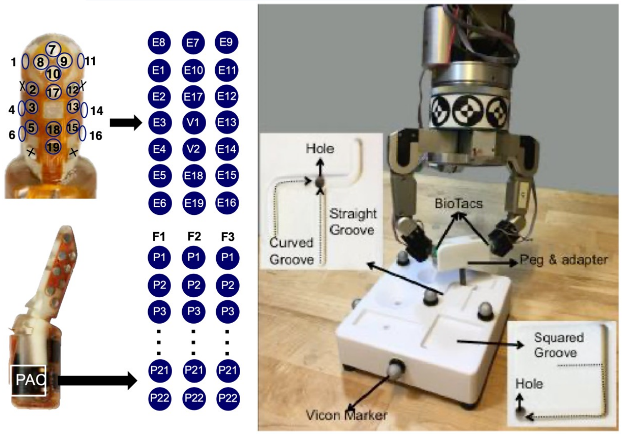
https://arxiv.org/pdf/1910.02646.pdf





(a)

https://harryzhesu.github.io/pdf/su2016learning.pdf



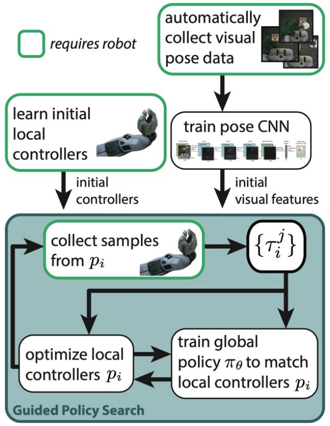
Non-Parametric Policies

Generic Fixed-size Parametric Policies

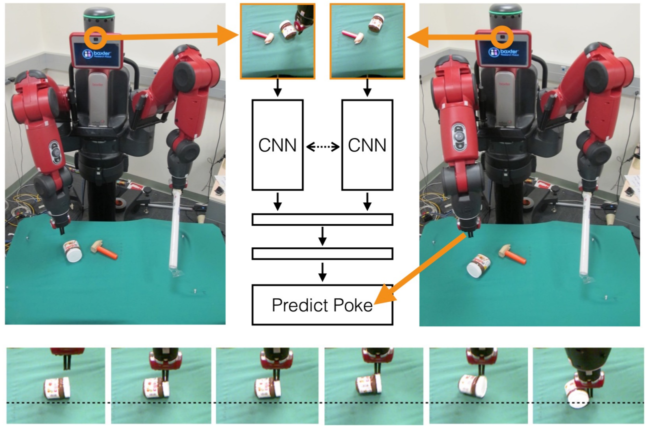
Restricted Parametric Policies

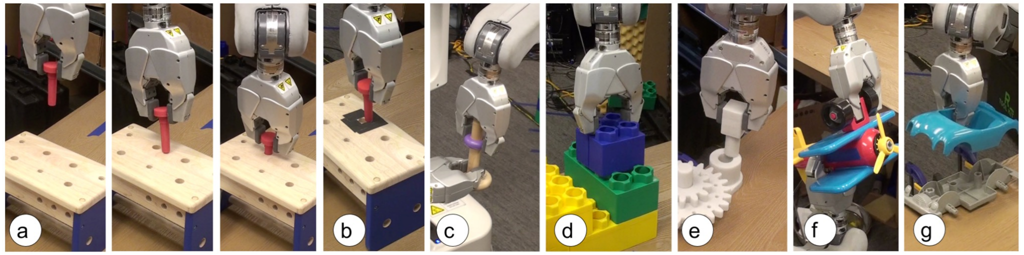
Goal-based Policies

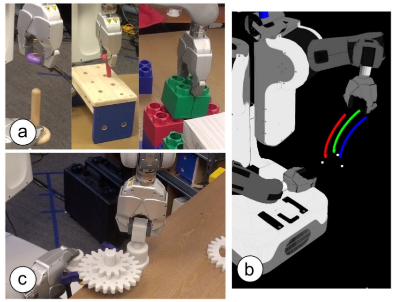
https://arxiv.org/pdf/1504.00702.pdf

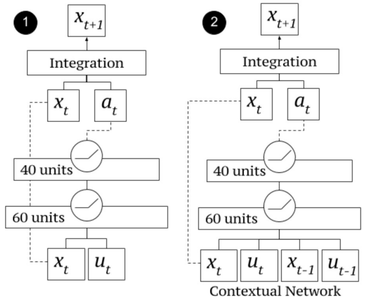


https://arxiv.org/pdf/1606.07419.pdf





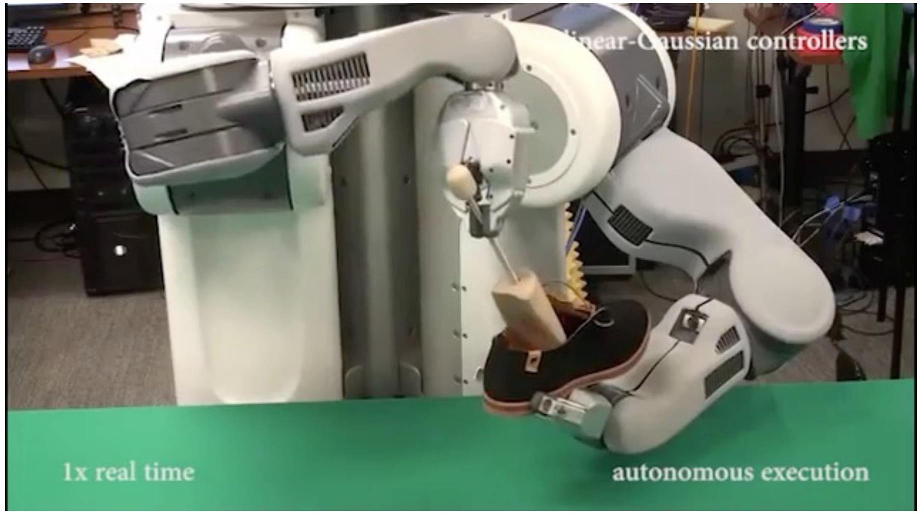


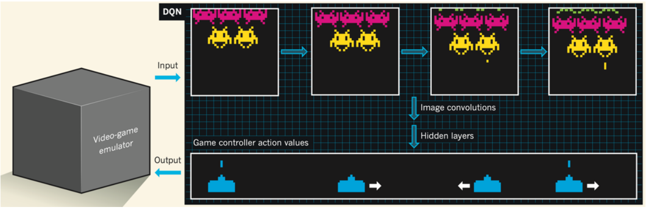


https://arxiv.org/pdf/1509.06841.pdf

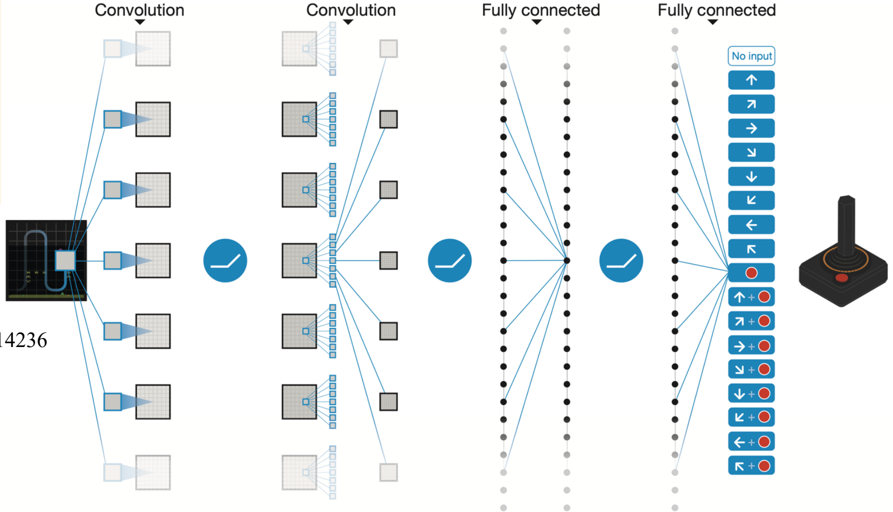
Reinforcement Learning

For any given policy representation, RL can be used to learn policy parameters for skill controllers.





https://www.nature.com/articles/518486a.pdf



doi:10.1038/nature14236

Algorithm 1 Guided policy search with unknown dynamics

1: **for** iteration $k = 1$ to K **do**
2: Generate samples $\{\tau_i^j\}$ from each linear Gaussian controller $p_i(\tau)$ by running it on the robot
3: Minimize $\sum_{i,t} \lambda_{i,t} D_{\mathrm{KL}}(p_i(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|p_i(\mathbf{x}_t,\mathbf{u}_t))$ with respect to θ using samples $\{\tau_i^j\}$
4: Update $p_i(\mathbf{u}_t|\mathbf{x}_t)$ using the LQG-like method
5: Increment each of the dual variables $\lambda_{i,t}$ by $\alpha D_{\mathrm{KL}}(p_i(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|p_i(\mathbf{x}_t,\mathbf{u}_t))$
6: **end for**
7: **return** optimized policy parameters θ

Value Function or Policy Search

Model-Based or Model-Free

On-Policy vs Off-Policy

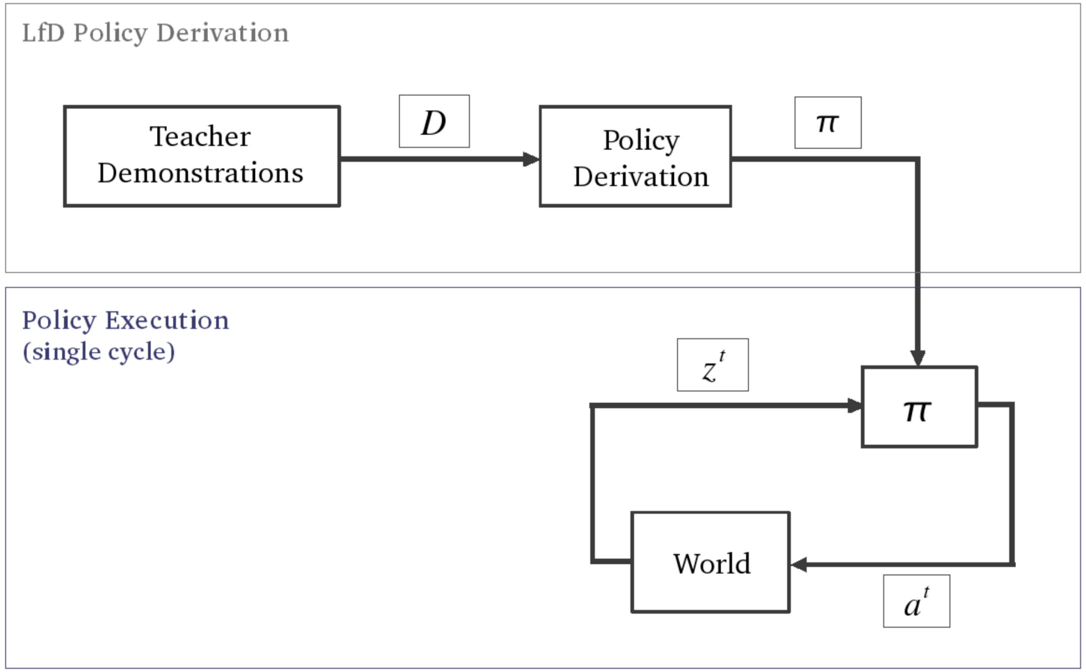
Reinforcement Learning

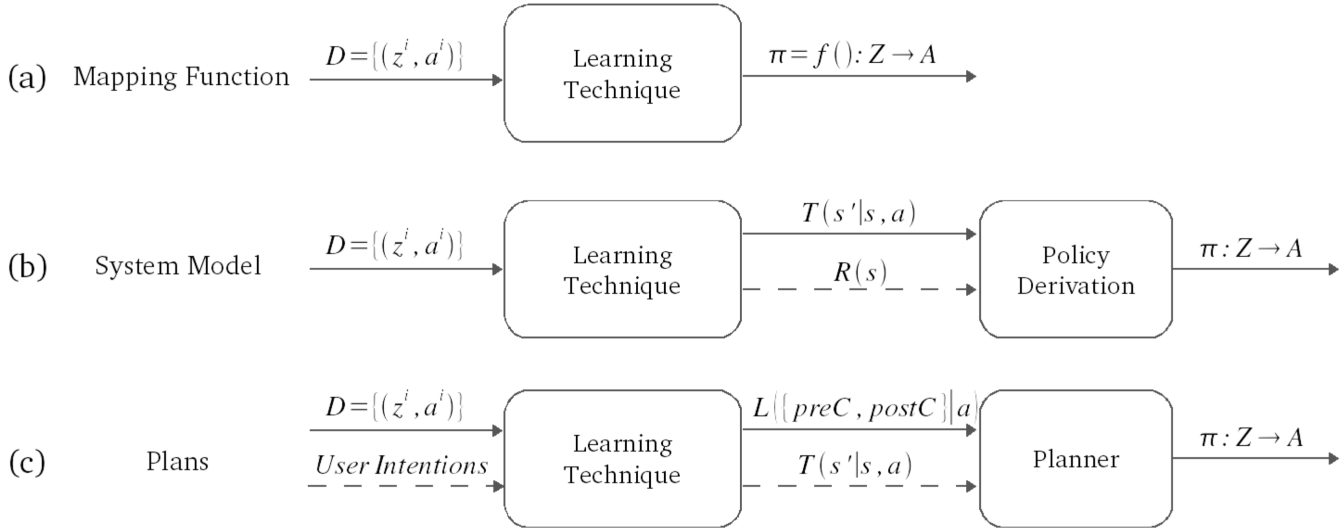
AncoraSIR.com

Imitation Learning

The user simply shows the robot what to do instead of writing code to describe the desired behavior

- Leverage the existing task expertise of (potentially non-expert) humans to
  - Bypass time-consuming exploration that would be required in an RL setting,
  - Communicate user preferences for how a task ought to be done, and
  - Describe concepts that may be difficult to specify formally or programmatically.





Skill Transfer

Skills learned in one task are often transferred to other tasks via a variety of mechanisms

- Direct Skill Re-use
  - Directly re-use a learned skill in a new task related to the one it was learned on
- Parameterized Skills
  - In certain task families, only some aspects of the task context change, while all other task semantics remain the same or are irrelevant
- Metalearning
  - “learn to learn”—in other words, learn something about a distribution of tasks that allows for more efficient learning on any particular task from that distribution in the future.
- Domain Adaptation
  - some task families retain all of their high-level semantics across instances, differing only in lower-level details
- Sequential Transfer and Curriculum Learning
  - It is sometimes advantageous to view multiple instances transfer as a sequential learning problem

Safety and Performance Guarantees

How well will the policy perform across the distribution of situations that it will face?

- Future applications will require behaviors that are safe and correct with high confidence
  - Robots that operate alongside humans in homes and workplaces must not cause injuries, destroy property, or damage themselves;
  - Safety-critical tasks such as surgery and nuclear waste disposal must be completed with a high degree of reliability;
  - Robots that work with populations that rely on them, such as the disabled or elderly, must be dependable
- Performance Metrics
  - Cumulative reward under some reward function
    - Whether the *expected return* of a policy is being bounded,
    - Whether a *risk-aware function* of return is used
    - When a policy must obey performance bounds
- Classes of Guarantees and Bounding Methods
  - Safe learning is even more difficult, with limited real-world data collection abilities
  - A research gap to provide strong performance guarantees in low-data and poor-model robotics settings

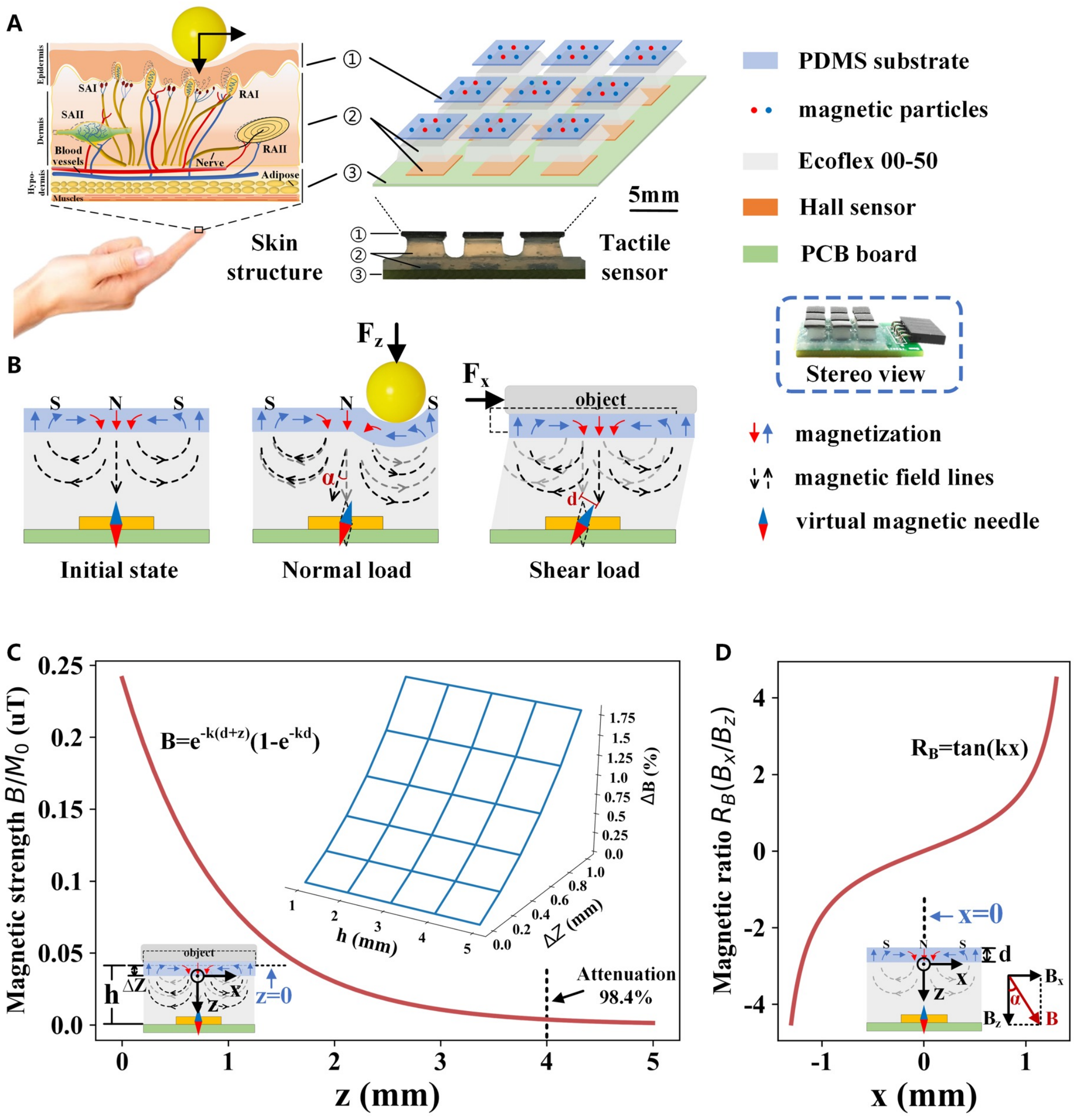
AncoraSIR.com

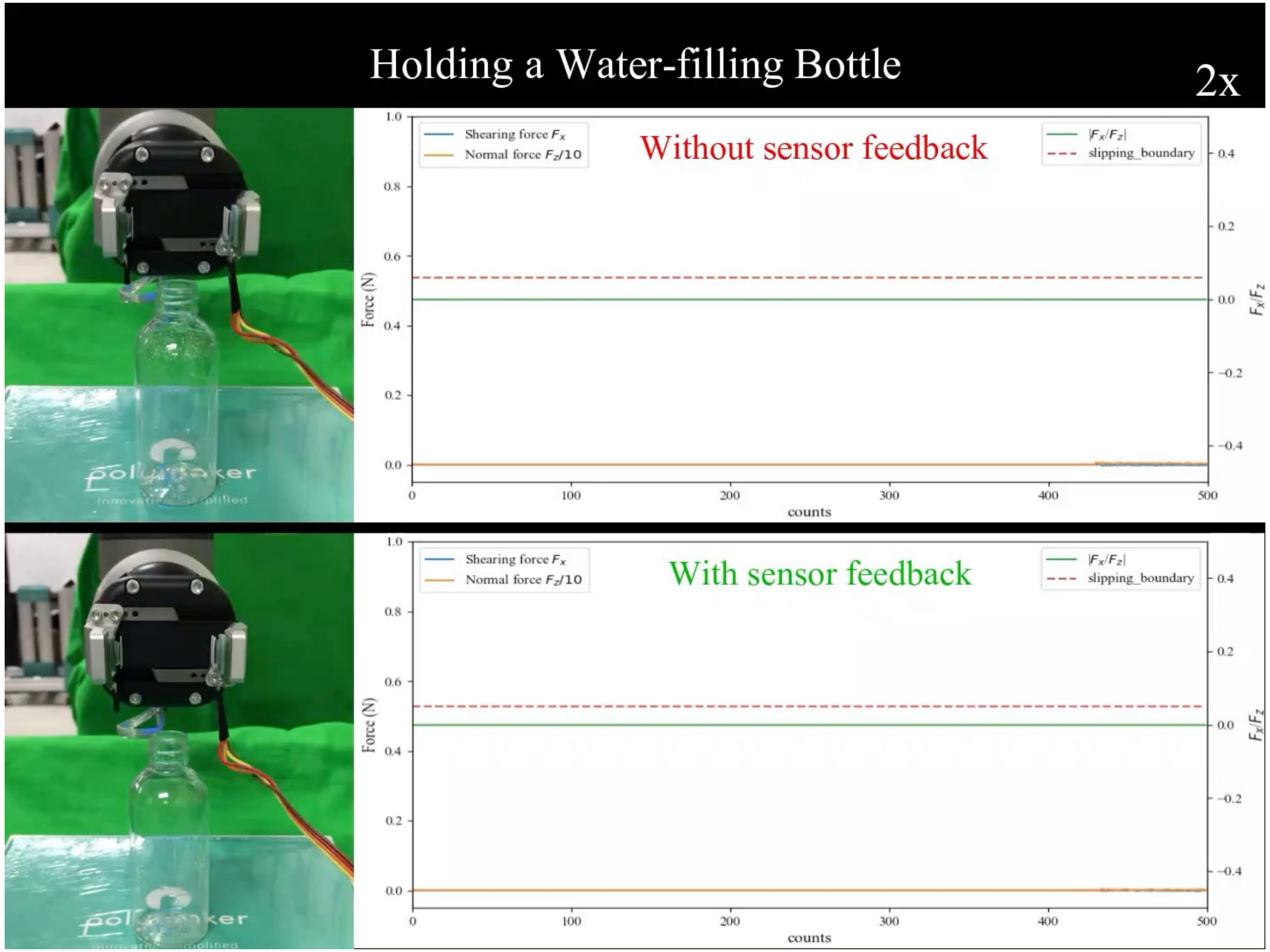
Learning to Touch

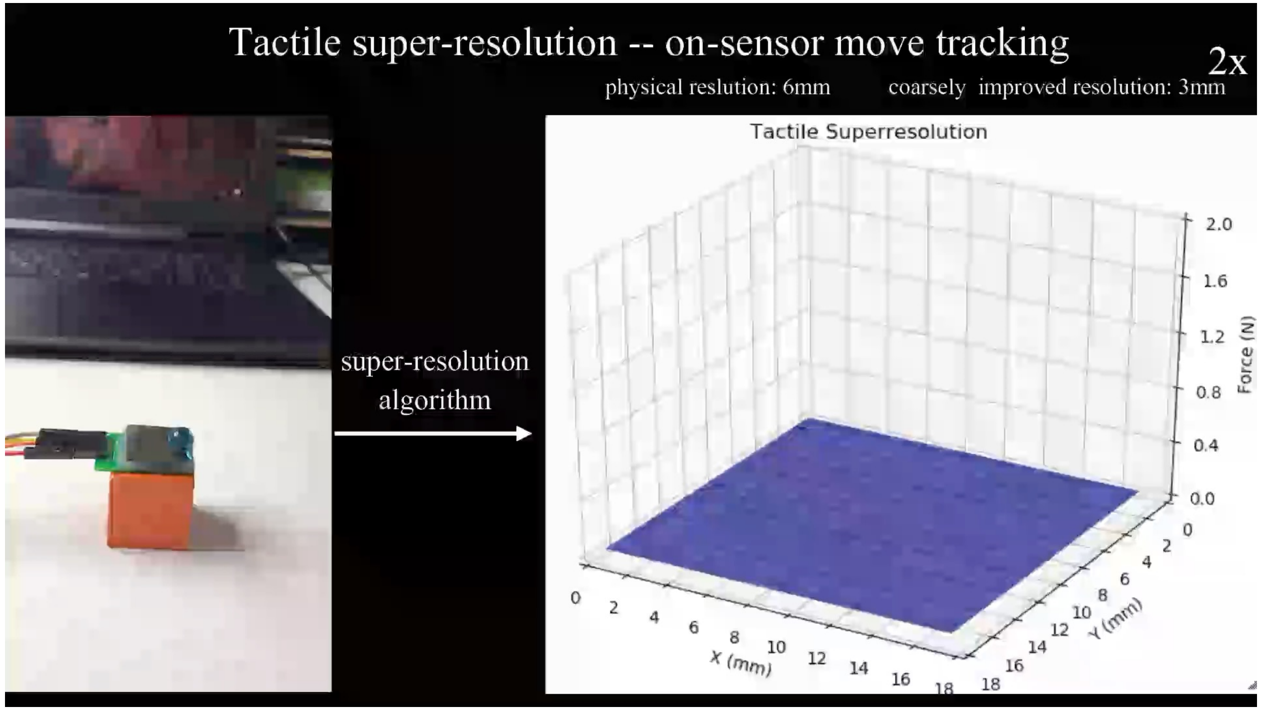
Example set 2 with the DeepClaw Toolkit

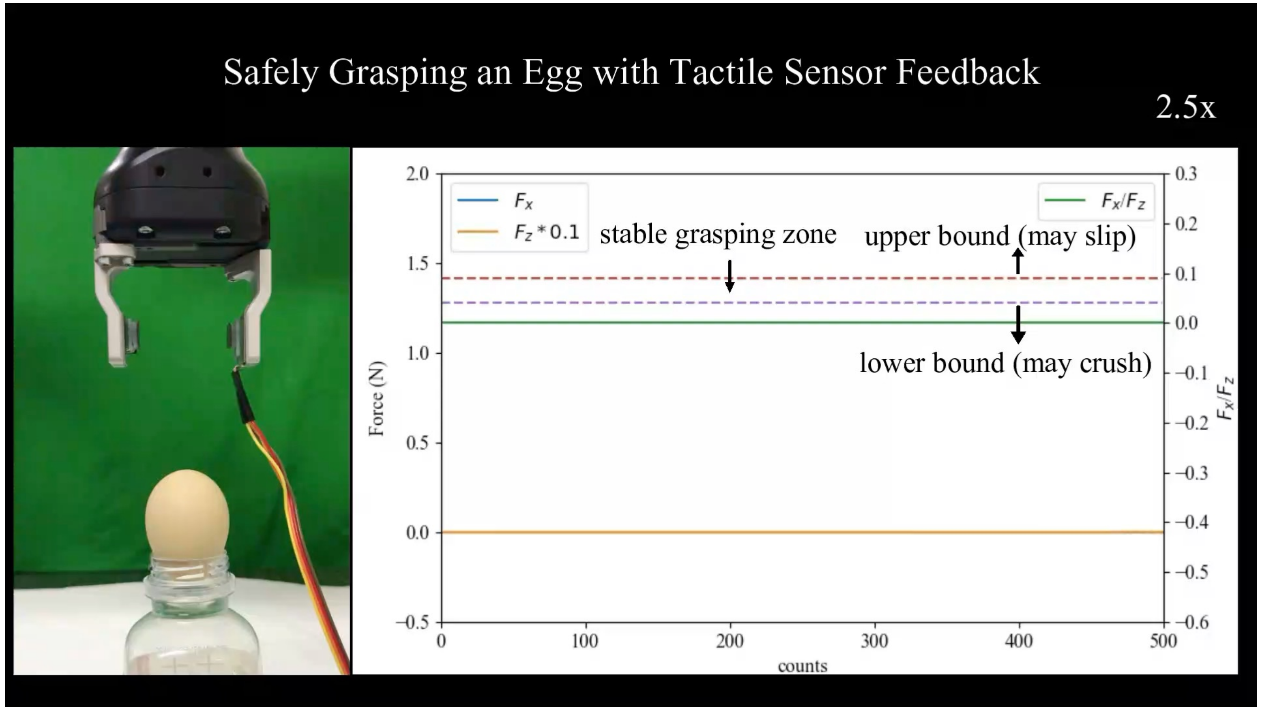
AncoraSIR.com

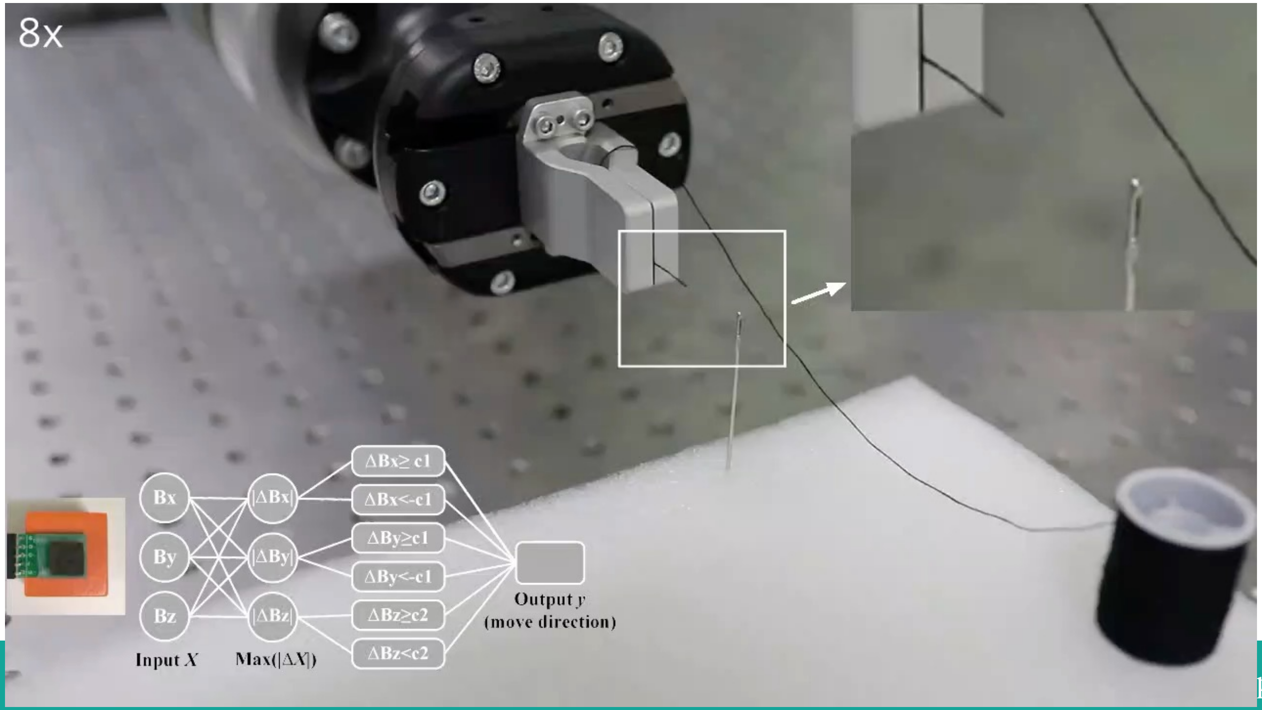
Magnetized Tactile Skin with Super-resolution

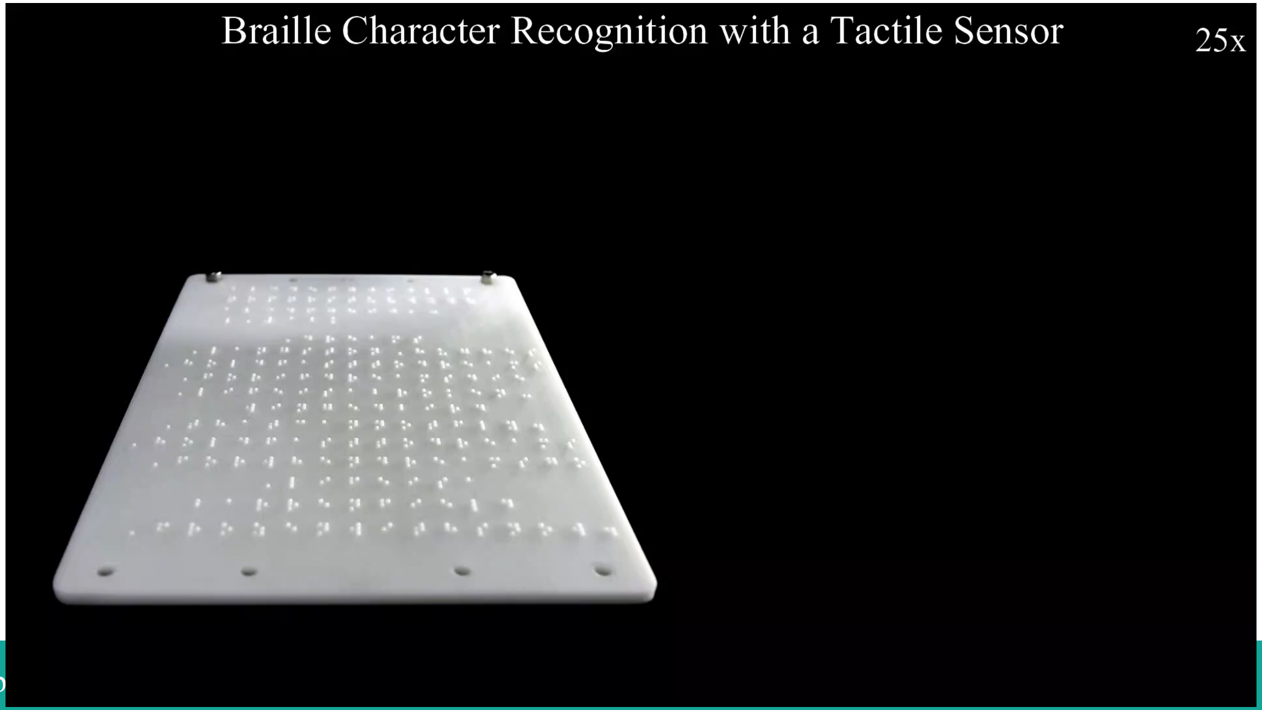




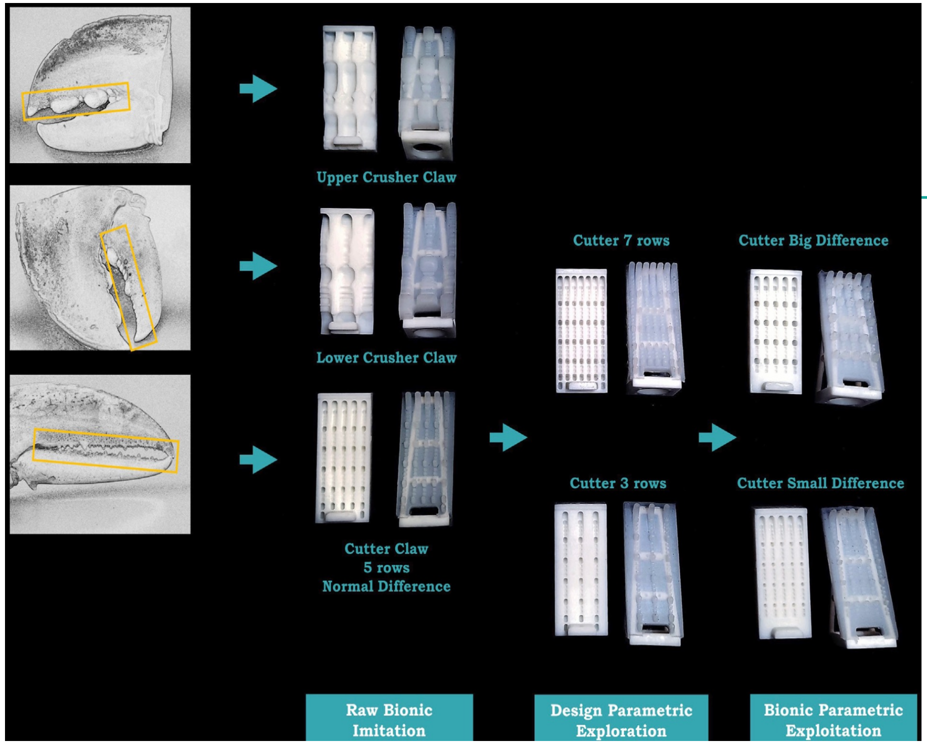

Tactile super-resolution -- on-sensor move tracking
2x
physical reslution: 6mm
coarsely improved resolution: 3mm
super-resolution
algorithm
Tactile Superresolution
Force (N)
X (mm)
Y (mm)

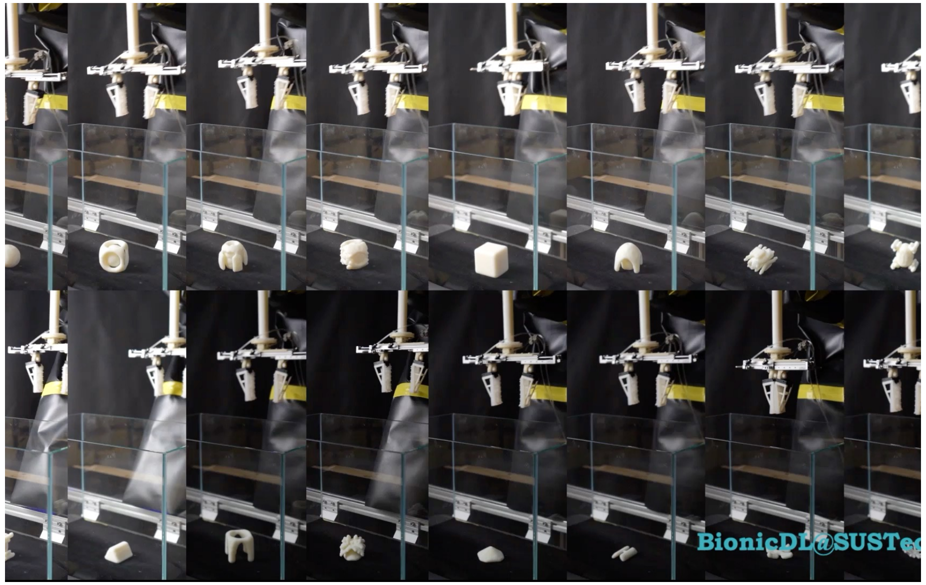

Safely Grasping an Egg with Tactile Sensor Feedback
2.5x
F_x
$F_z * 0.1$
F_x/F_z
stable grasping zone
upper bound (may slip)
lower bound (may crush)
Force (N)
counts

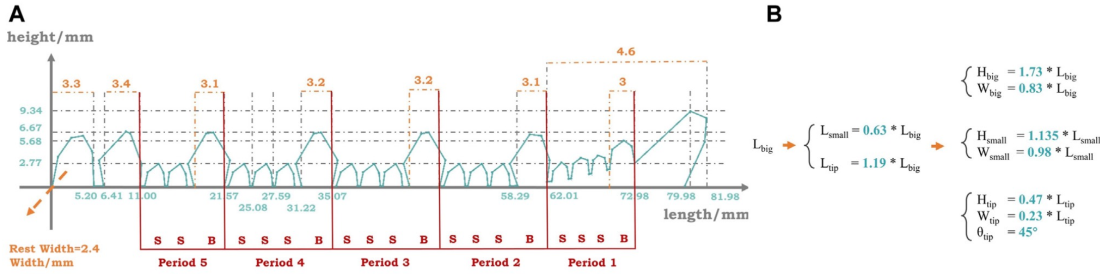

8x
Bx
By
Bz
|ΔBx|
|ΔBy|
|ΔBz|
ΔBx≥ c1
ΔBx<-c1
ΔBy≥c1
ΔBy<-c1
ΔBz≥c2
ΔBz<c2
Output y
(move direction)
Input X
Max(|ΔX|)

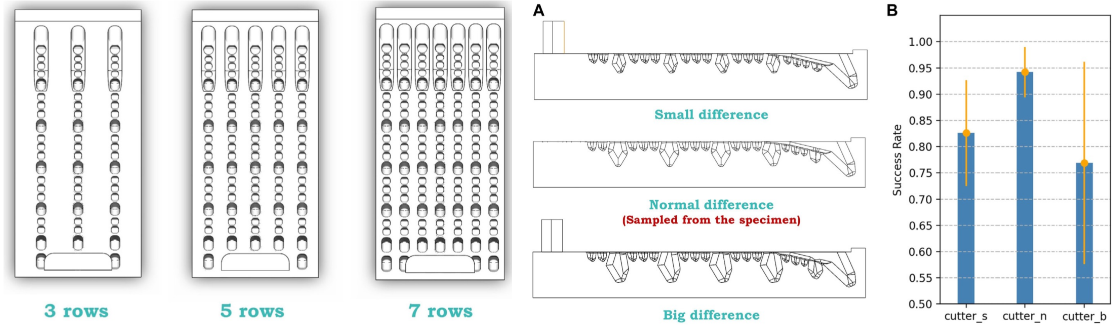

Braille Character Recognition with a Tactile Sensor
25x

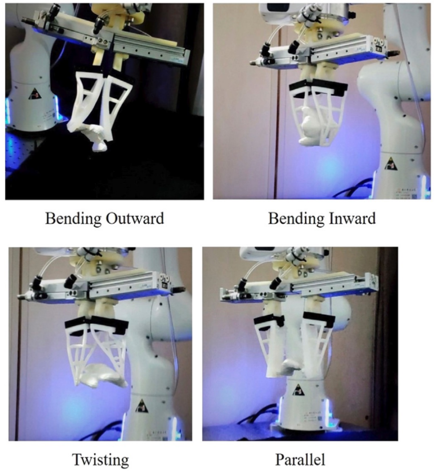
Bio-inspired Design for Grasp Learning Underwater

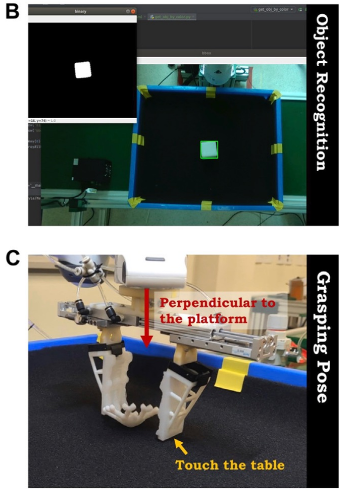






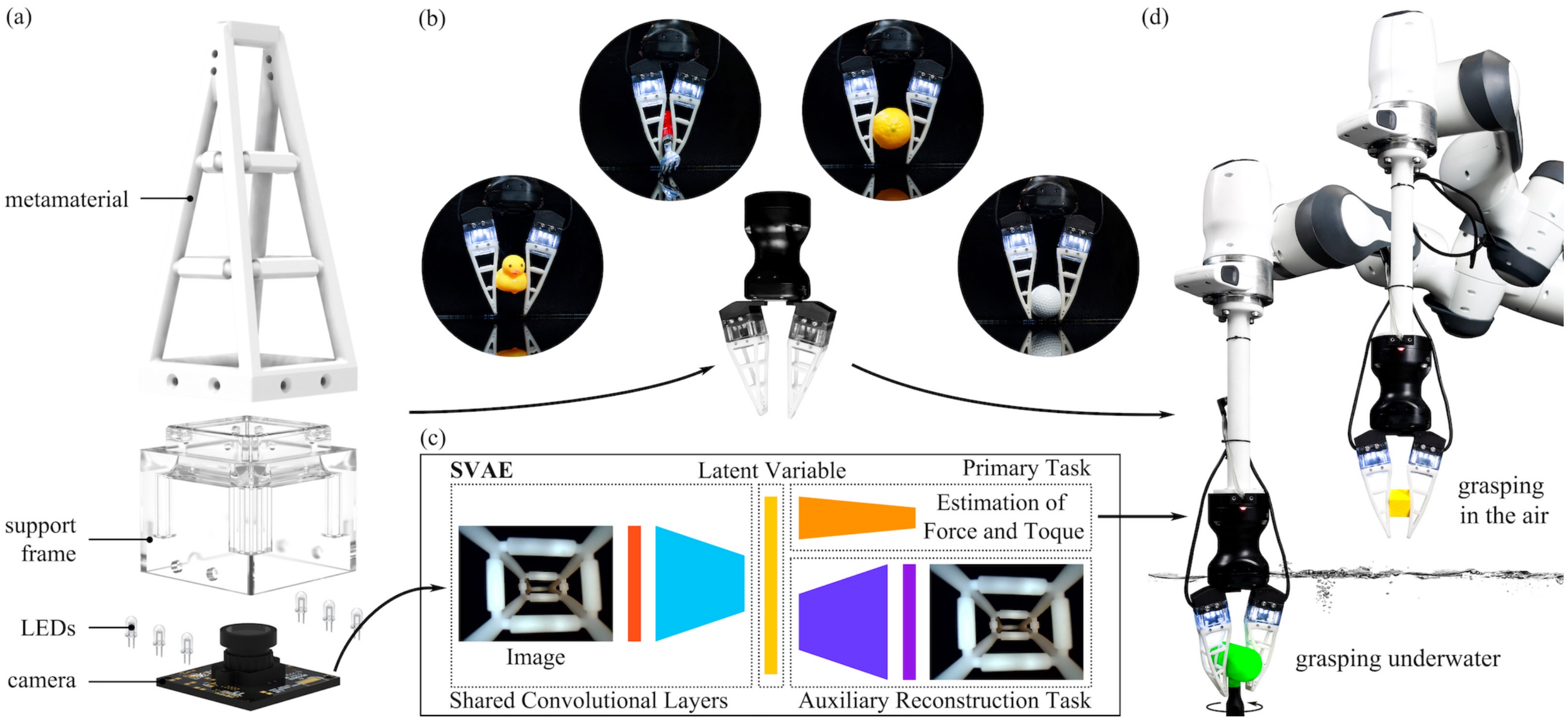






Learning to Touch Underwater

Using Generative Models

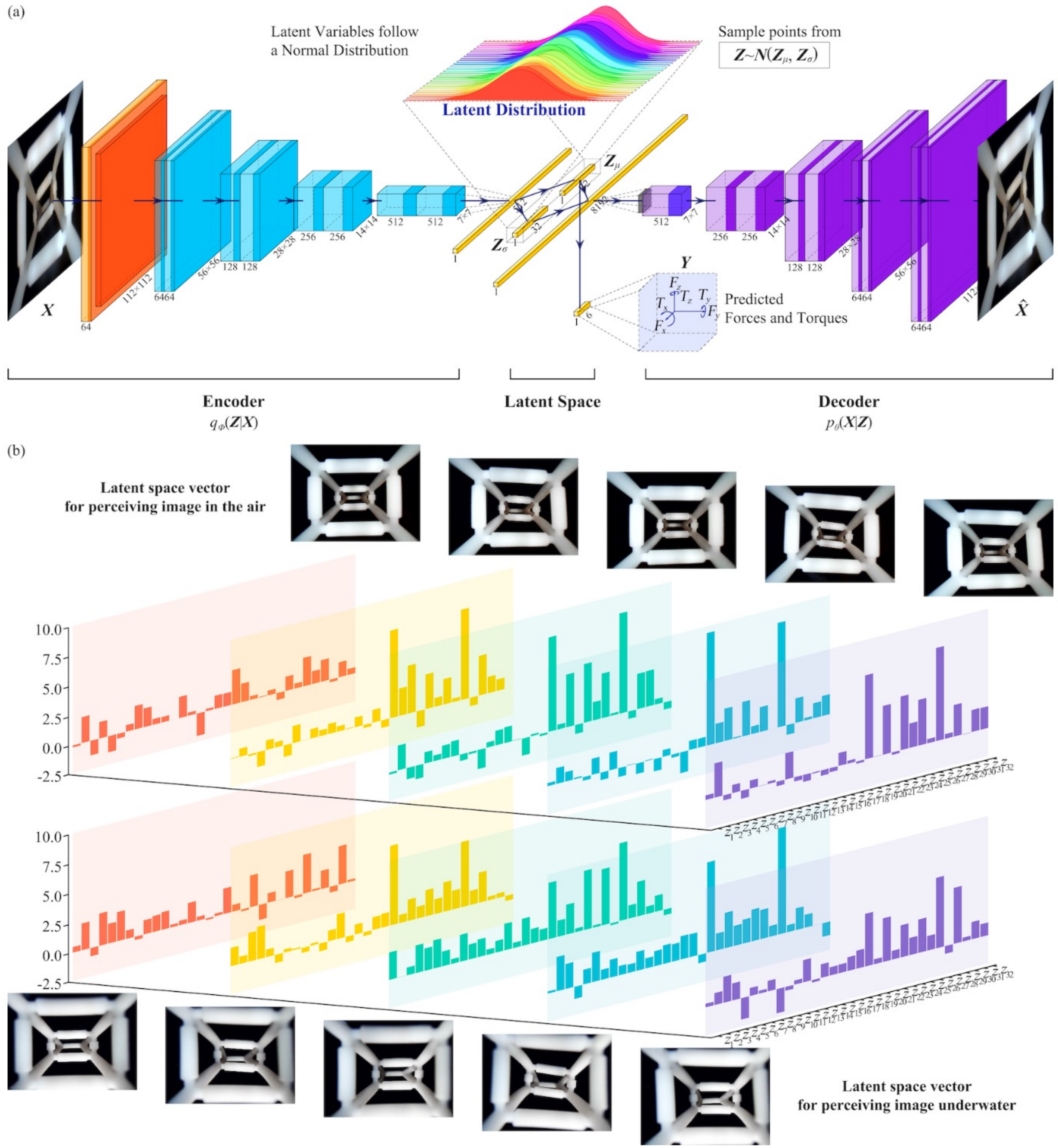


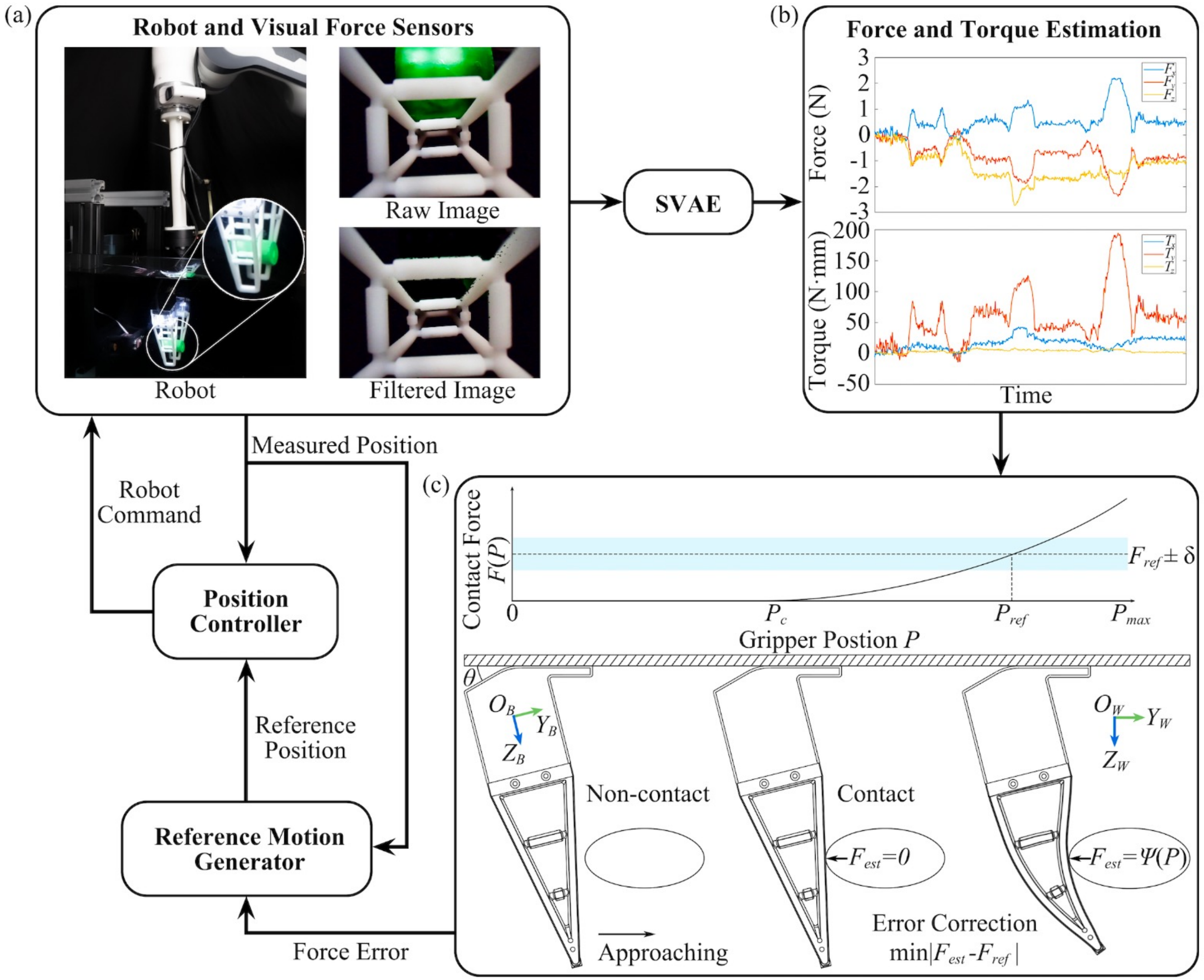
AncoraSIR.com

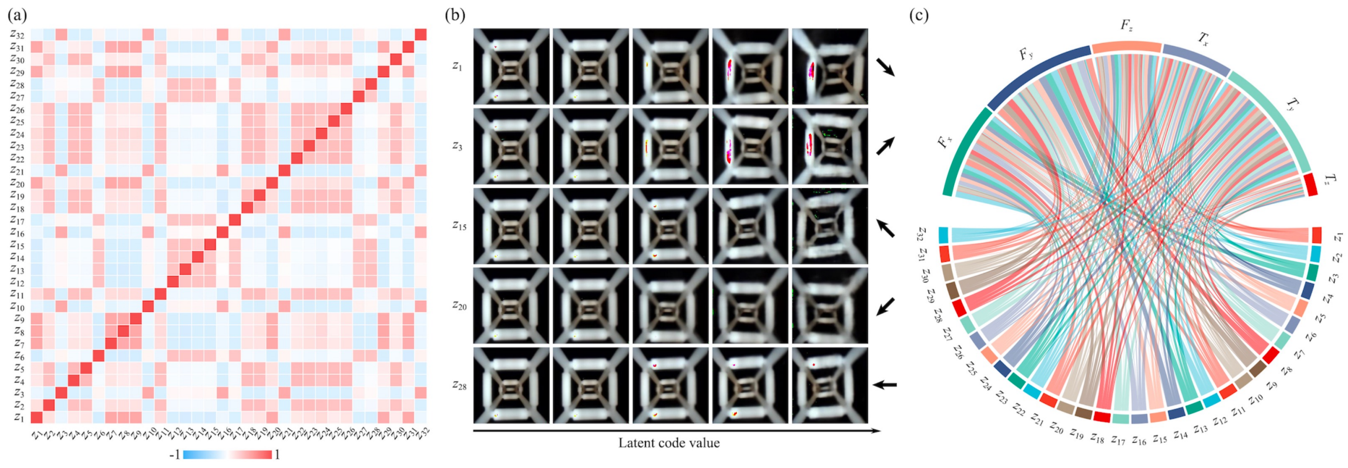
Southern University
of Science and Technology

Variational Autoencoder

Supervised Learning

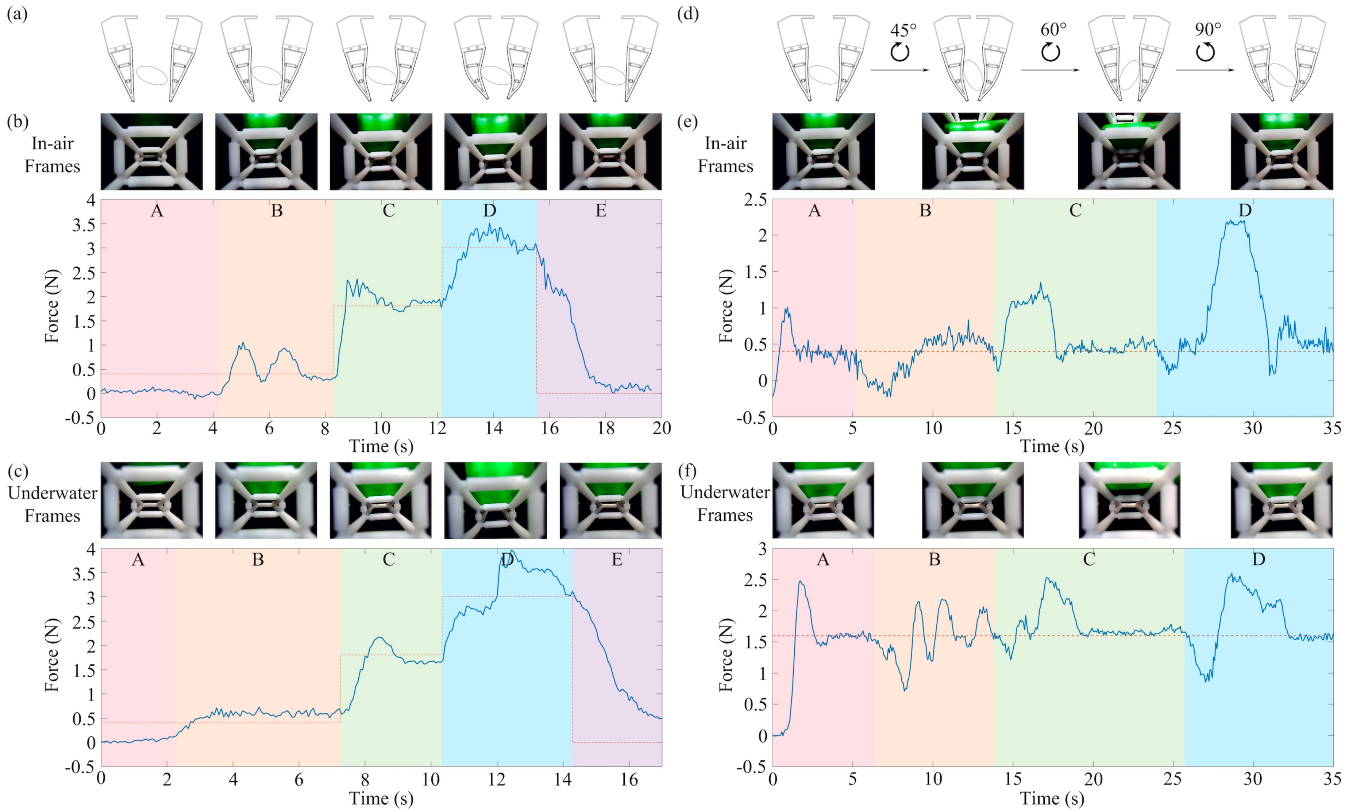


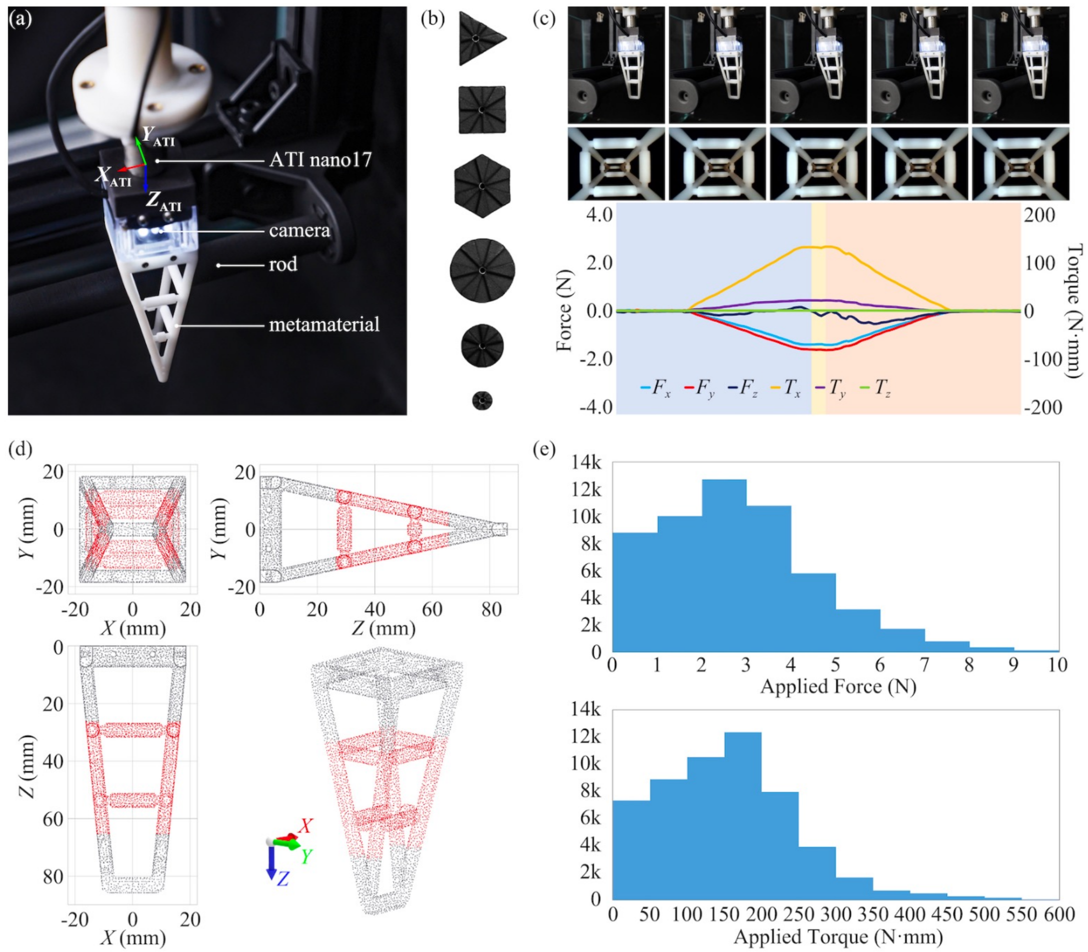




Latent Variables

For learning-based explanability





Thank you~

songcy@sustech.edu.cn