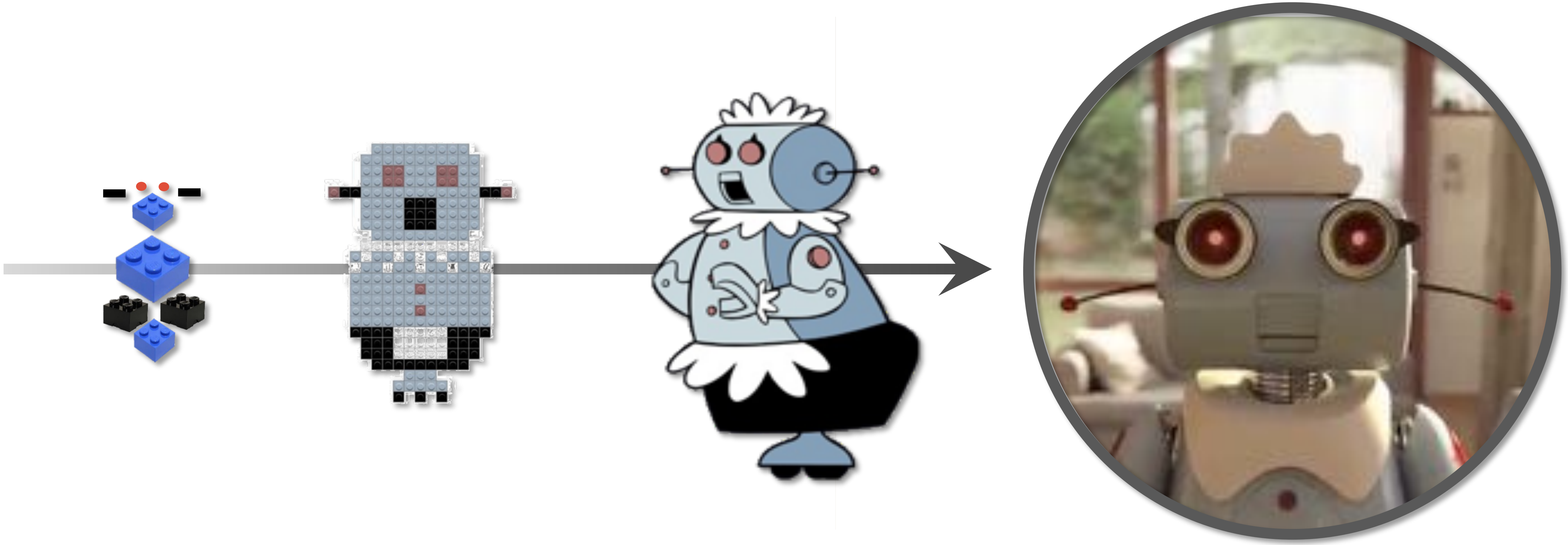
Paving the path to Robot Autonomy with Simulation



Animesh Garg

People AI & Robots
pair

UNIVERSITY OF TORONTO

VECTOR INSTITUTE

NVIDIA

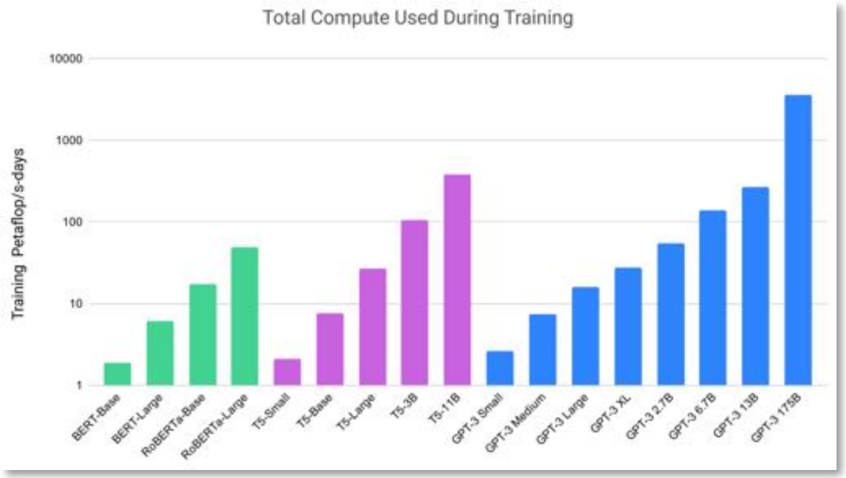
Generalizable Autonomy: Computer Vision & Language

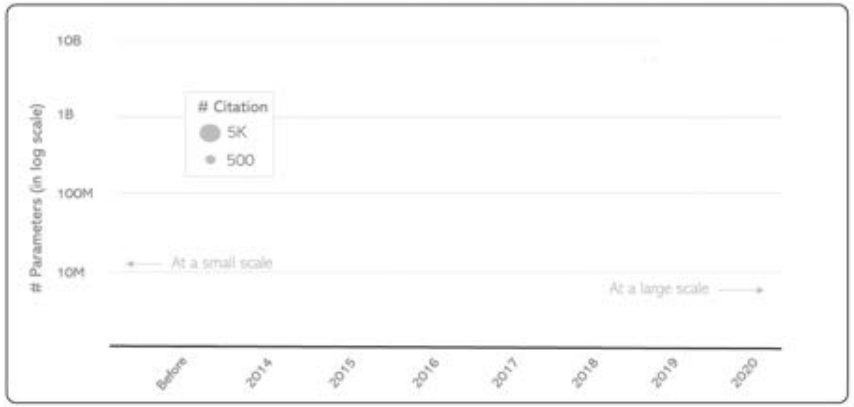
Structured Models + Data + Compute → Performance

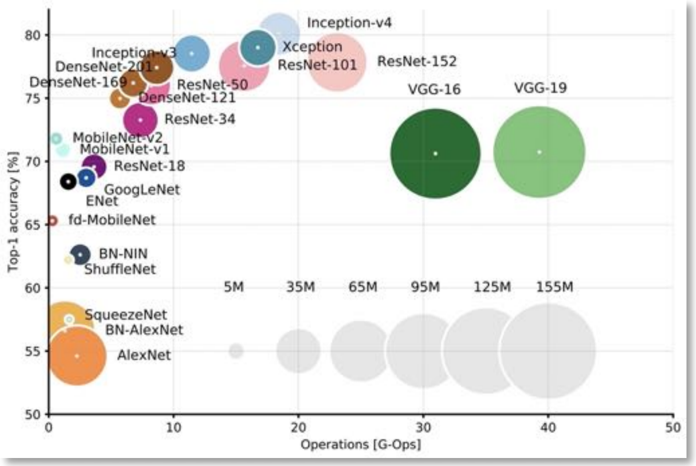
Open Images Dataset

SQuAD
The Stanford Question Answering Dataset

Common Crawl







Model	EM	F1
Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
IE-Net (ensemble) *RICOH_SRCB_DML*	90.939	93.214

Generalizable Autonomy: Computer Vision & Language

Ingredients of Modern Machine Learning & Applications

Large Structured Models

- Over-parameterized
- Structured Biases

IID Data & Datasets

- Concise problem Definition
- IID Data, easier to label

Distributed Deployment

- Large Scale Compute
- Distributed Deployment

Visual Perception

Natural Language

Passive Offline Decisions

Intelligent Robotics

Embodied

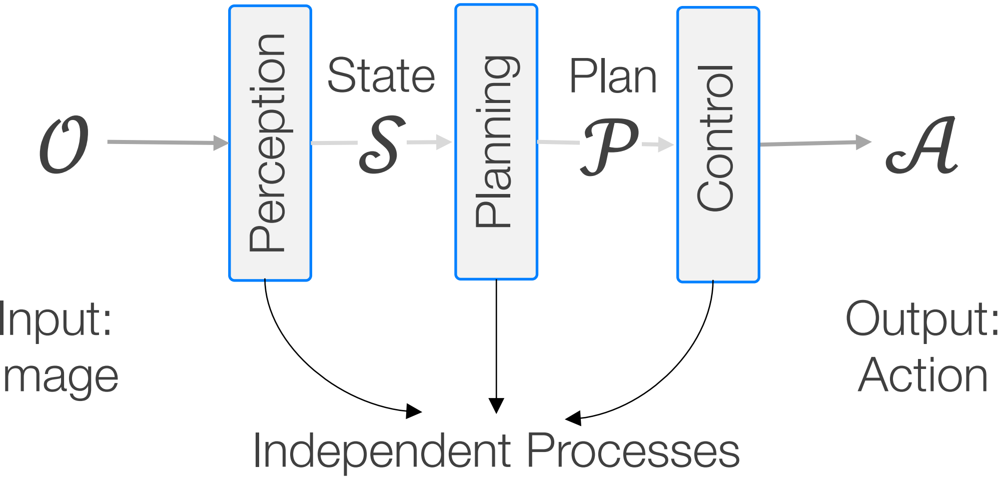
Paving the path to Robot Autonomy with Simulation

Domain Expertise

One problem,
One solution!

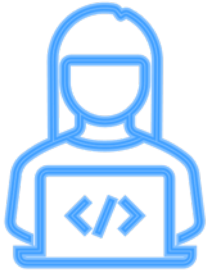
Structured Environments

Data Driven



Generalizable Autonomy: Duality of Discovery & Bias

Domain Expertise



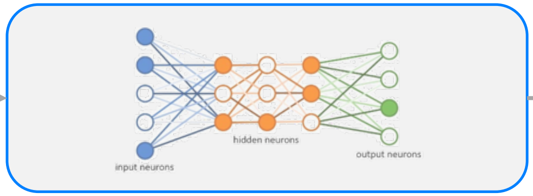
Just add data…

The Unreasonable Effectiveness of Data

Alon Halevy, Peter Norvig, and Fernando Pereira, *Google*

Data Driven

$\mathcal{O}$



$\mathcal{A}$

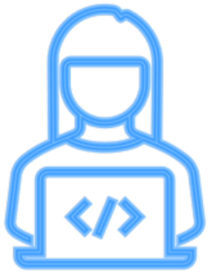
Input: Image

End-to-End Policy π

Output: Action

Generalizable Autonomy: Duality of Discovery & Bias

Domain Expertise



One problem,
One solution!

- ✗ Need for experts
- ✗ Limited applicability
- ✗ Perf vs Flexibility

Just add data…

- ✗ Computational sustainability
- ✗ Data accessibility
- ✗ Out-of-distribution errors

Data Driven

Neither achieves
Generality at Scale

Generalizable Autonomy: Duality of Discovery & Bias

Domain Expertise ⟷ Data Driven

Generalizable Autonomy

Structure + Data

Structure

- Domain knowledge,
- Inductive bias,
- Symmetries,
- Priors
- …

Data

- Online & Offline,
- Simulation & Real,
- Labelled & self-supervised
- Human in the loop
- …

Paving the path to Robot Autonomy with Simulation

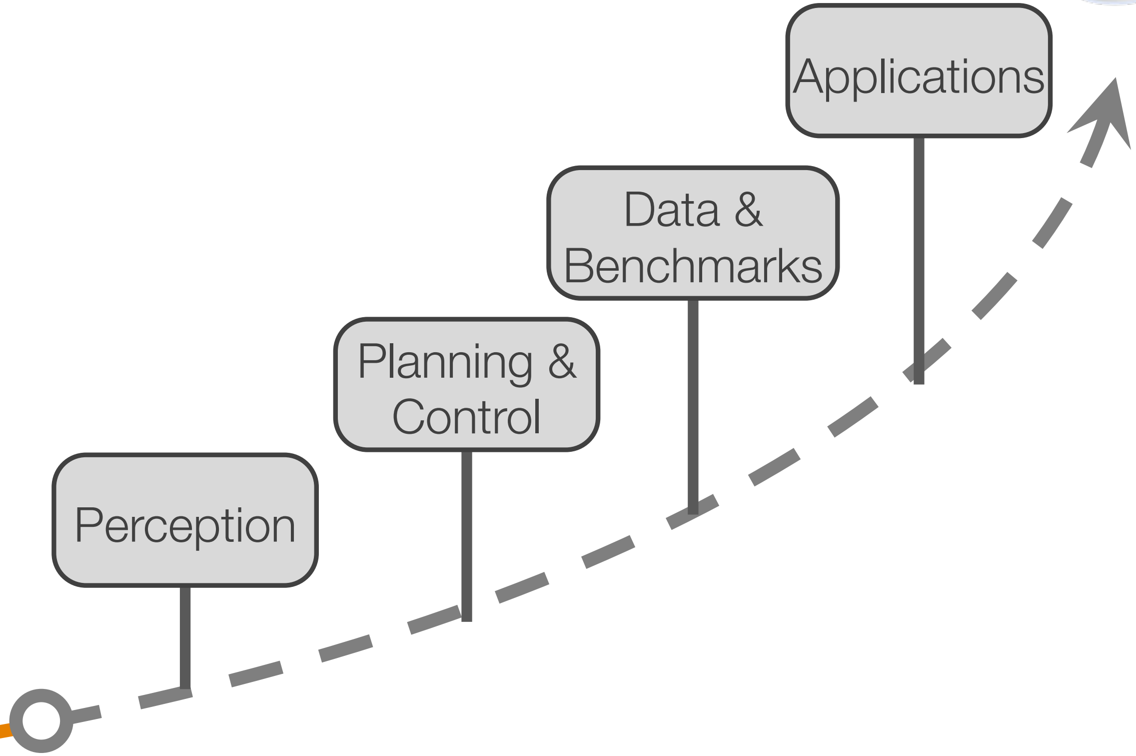

Why it will be years before robot butlers take over your household chores

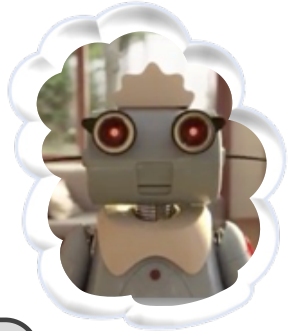
Home robots are good at doing one thing. Experts spell out the challenges preventing them from doing more.

By **Dalvin Brown**

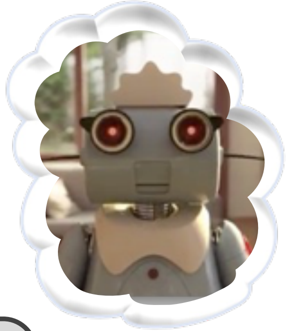
March 23, 2021 at 3:00 a.m. PDT

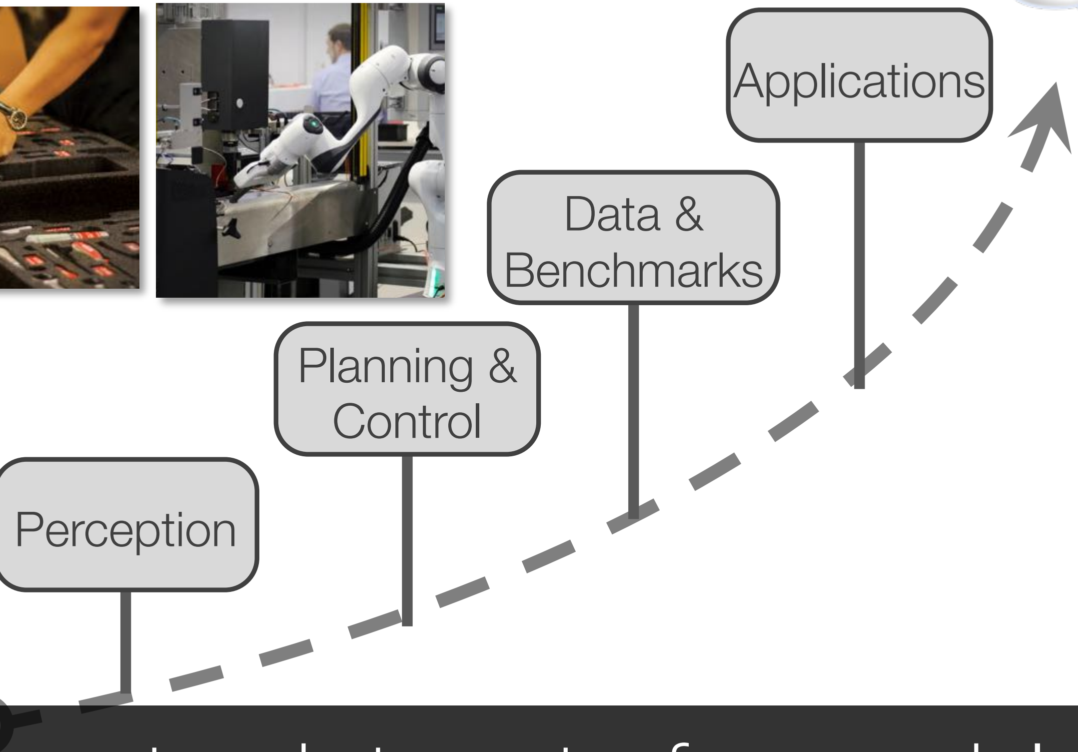
Listen to article 6 min





Paving the path to Robot Autonomy with Simulation

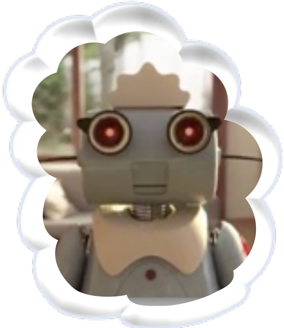




Too many problems to create datasets for each!

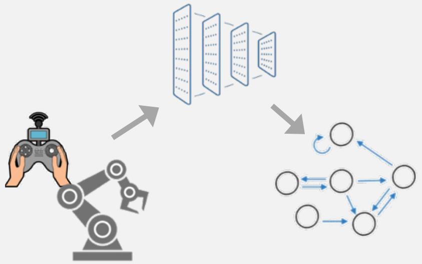
Paving the path to Robot Autonomy with Simulation

Vision: Simulation is Data Factory for Robotics

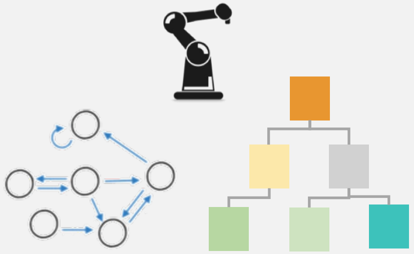


Simulation Systems

Interactive Estimation

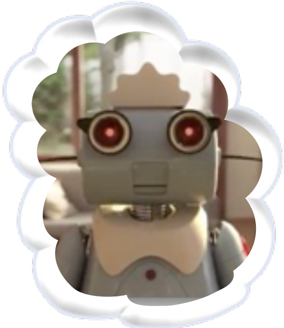


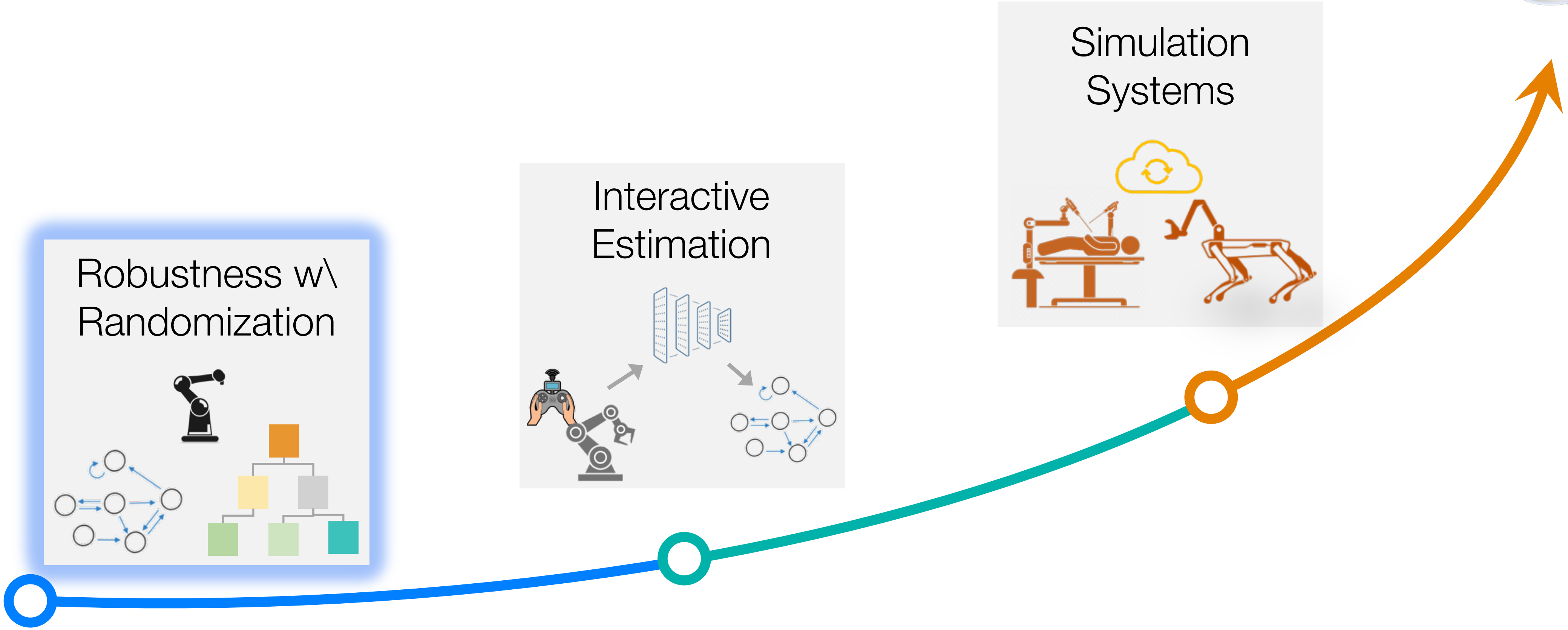
Robustness w\ Randomization



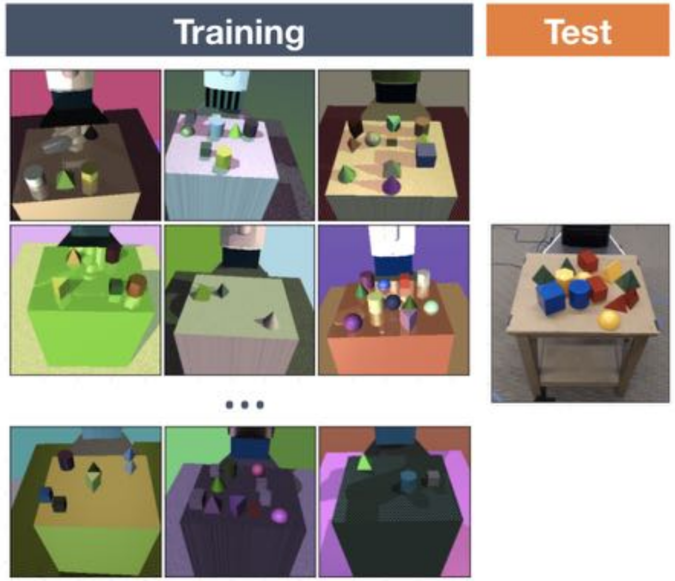
Paving the path to Robot Autonomy with Simulation

Vision: Simulation is Data Factory for Robotics

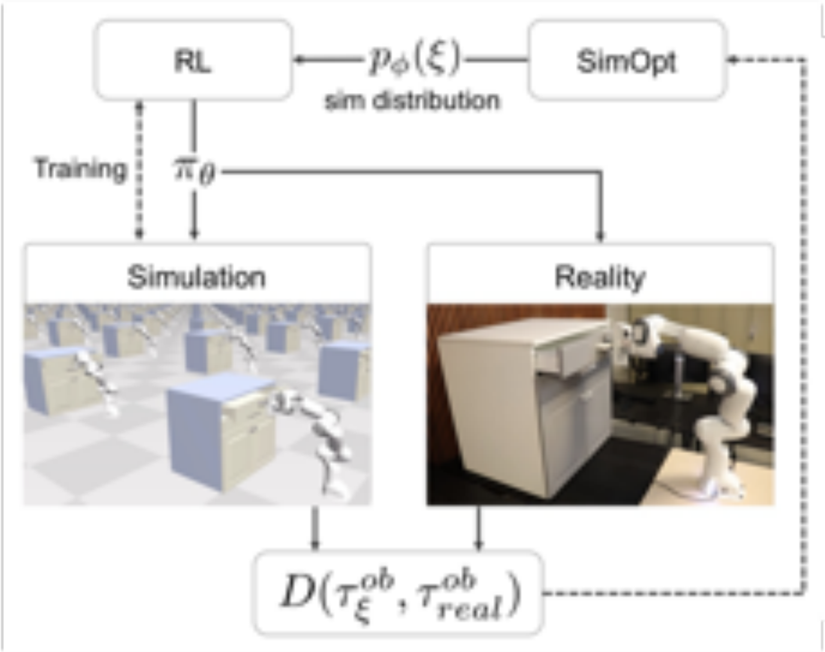




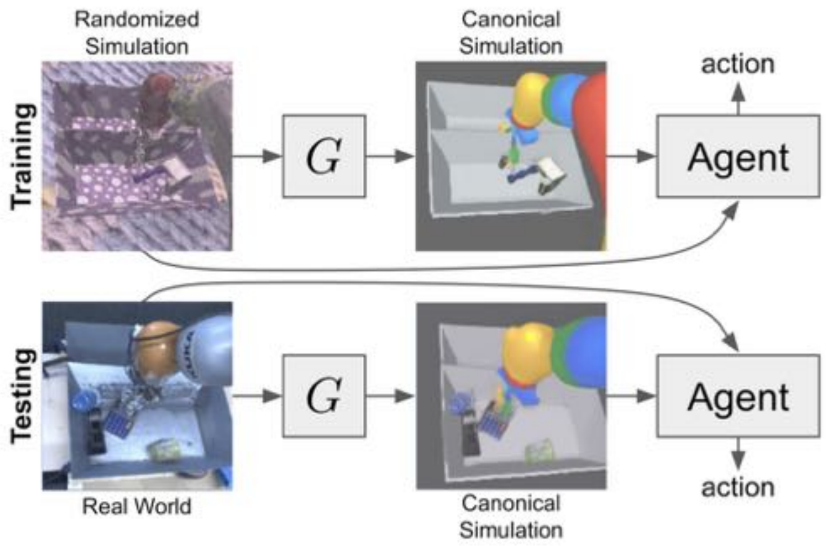
Domain Randomization



Tobin et al 2017



Chebotar et al 2019



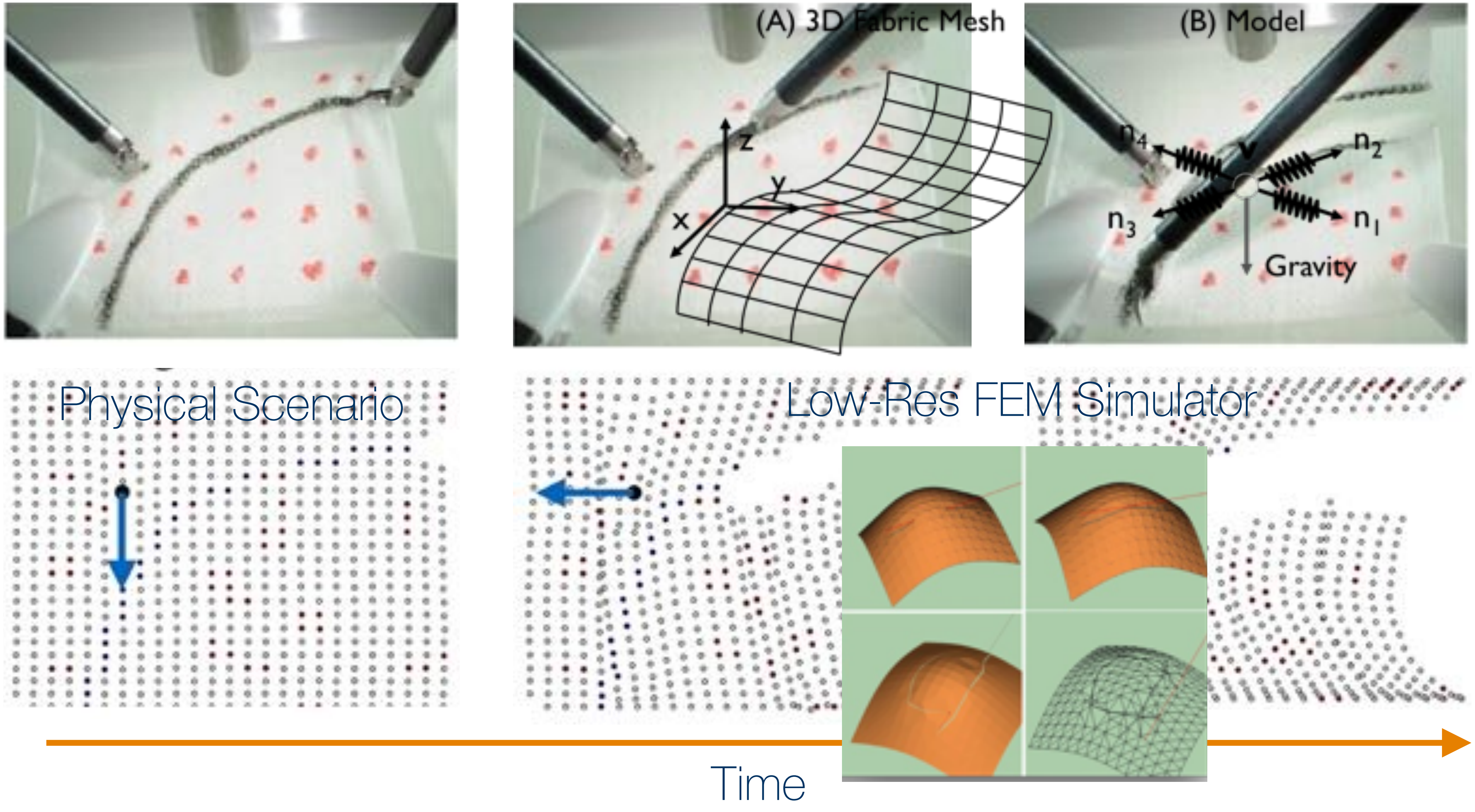
James et al 2019

Uniform Domain Randomization

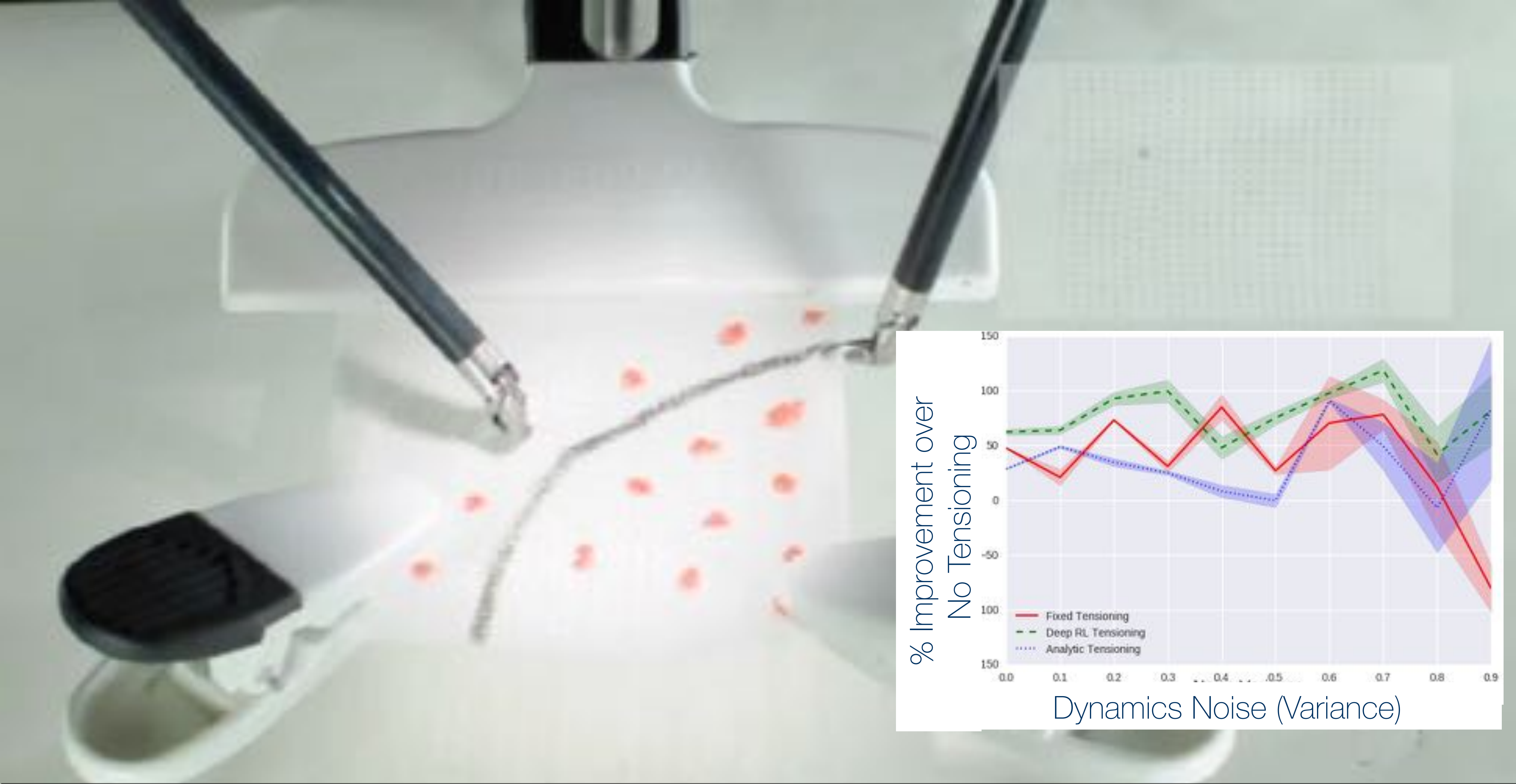
Online System ID & Adaptation

Handling Visual Observations

Learning Efficiently: Simulators



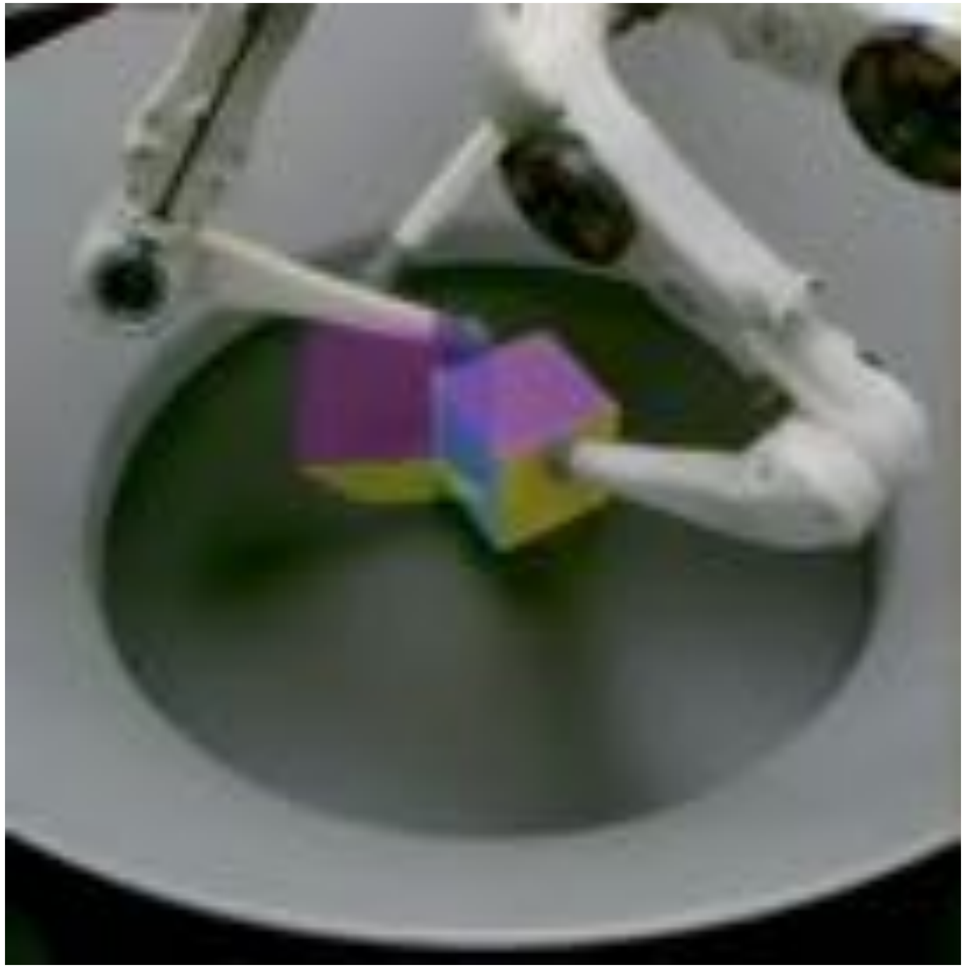
[Thanenjeyan et al. ICRA'17]

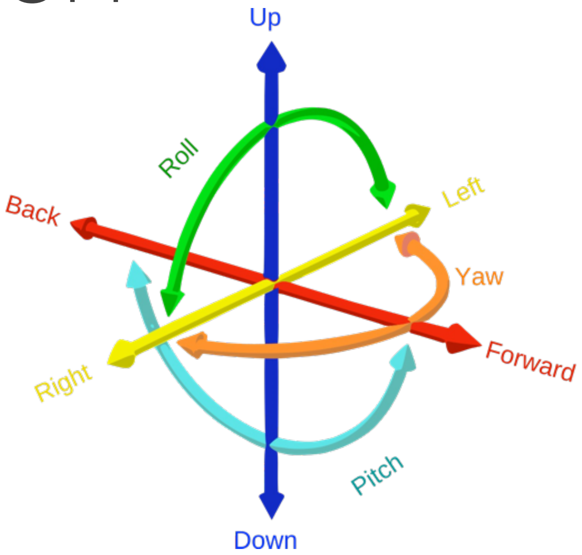


Autonomous Cutting

[ICRA'17]

Multi-finger In-Hand Manipulation





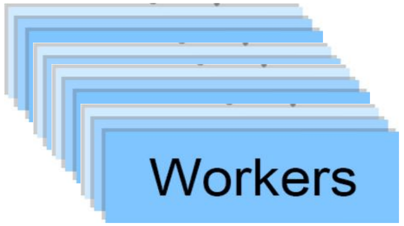
Real Robot Challenge: Trifinger platforms

Task: repose in 6-DoF
(position + orientation)

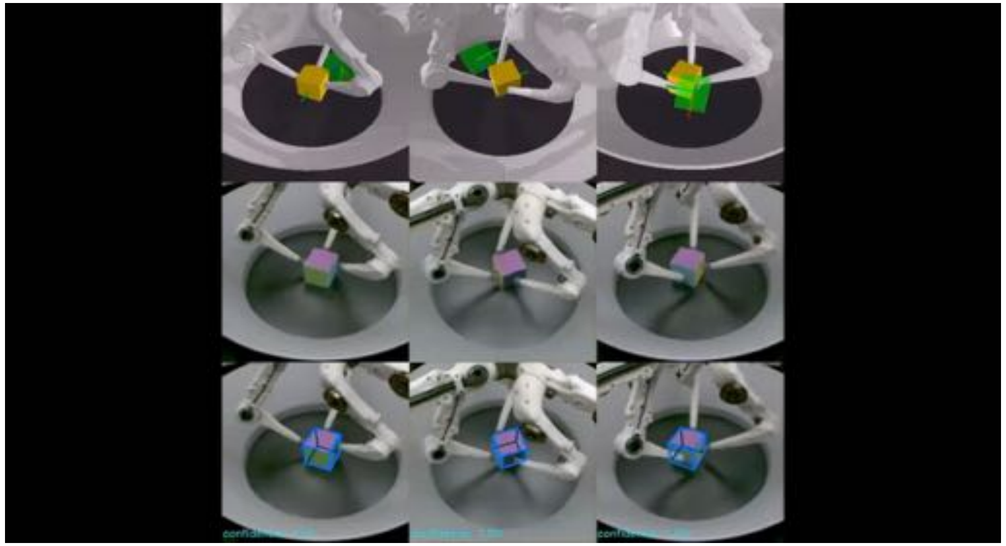
Development done remotely in simulation using Isaac Gym, no physical robot access

Allshire et al arxiv 2021

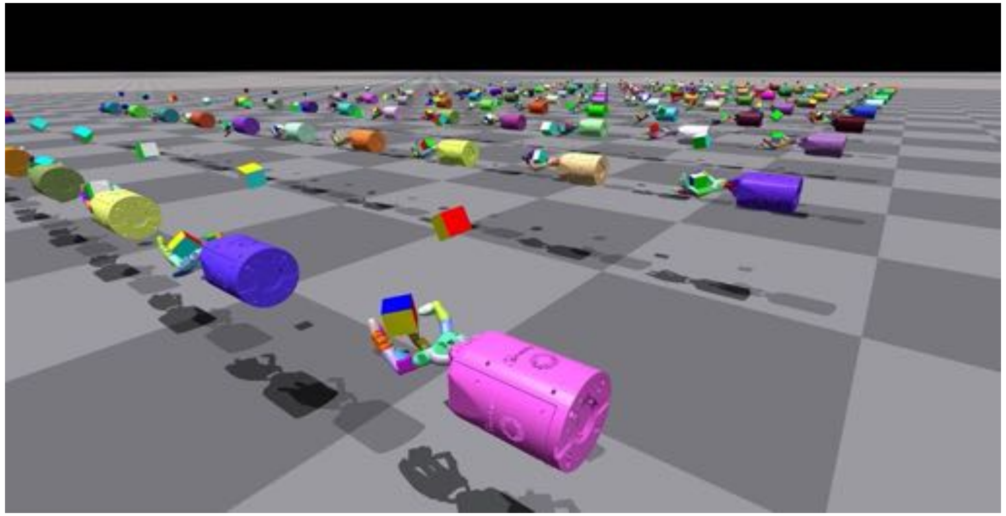
Multi-finger In-Hand Manipulation



Dextrous Manipulation via Simulation
OpenAI, 2018



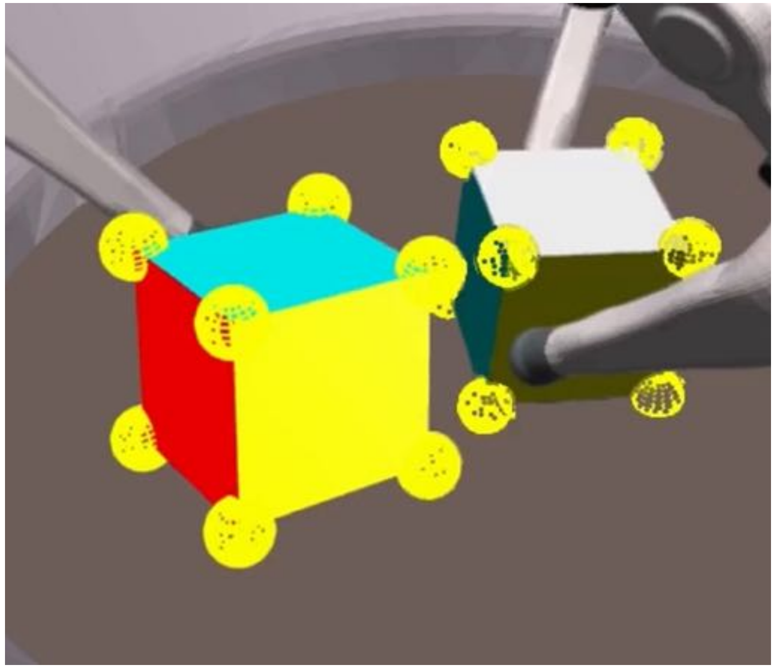
Real Robot Challenge
Structured Policies

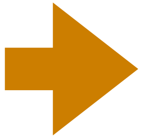


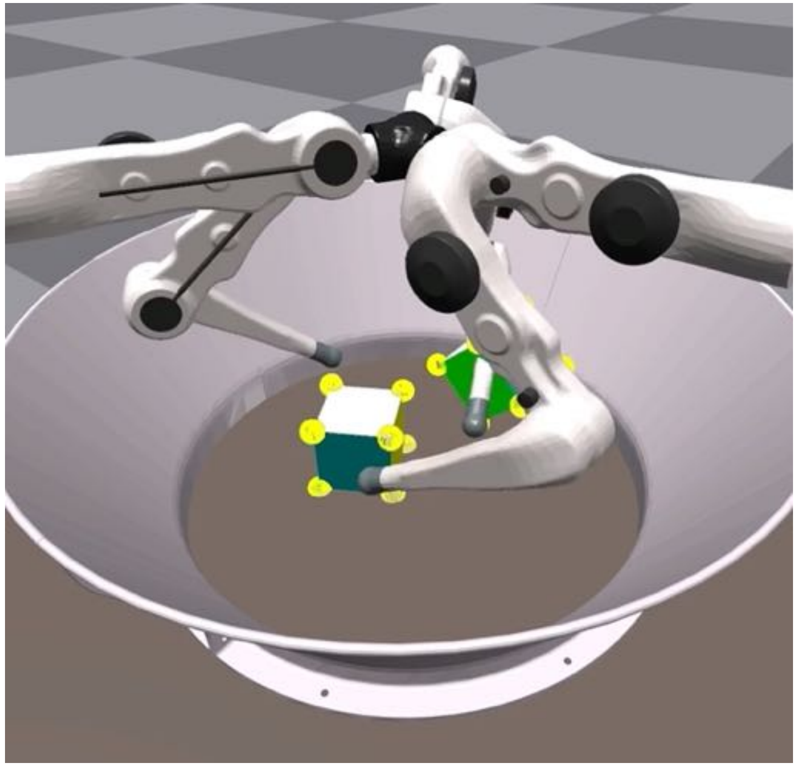
GPU-Simulated Manipulation
Isaac Gym

Multi-finger In-Hand Manipulation

- Traditional reward & observation performed poorly
- A better representation than position + quaternion in {observation, reward}?
- Allow for 6-DoF reposing



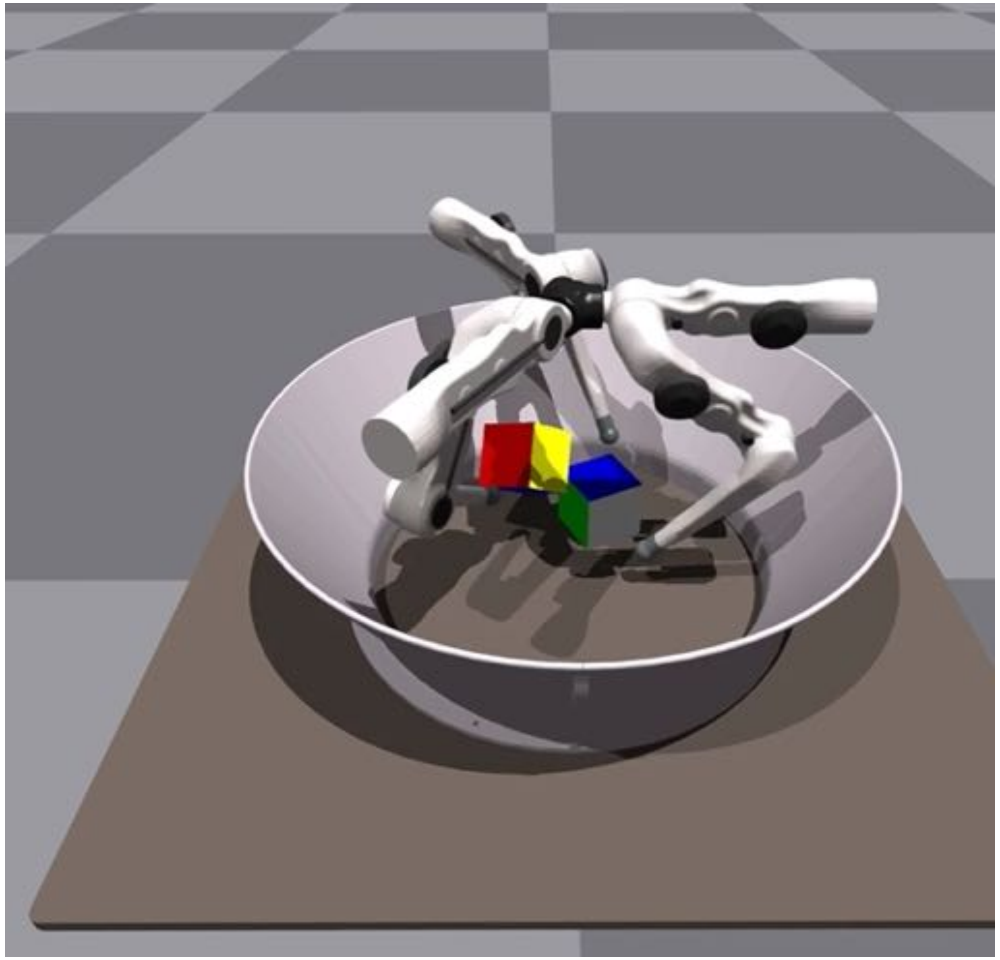


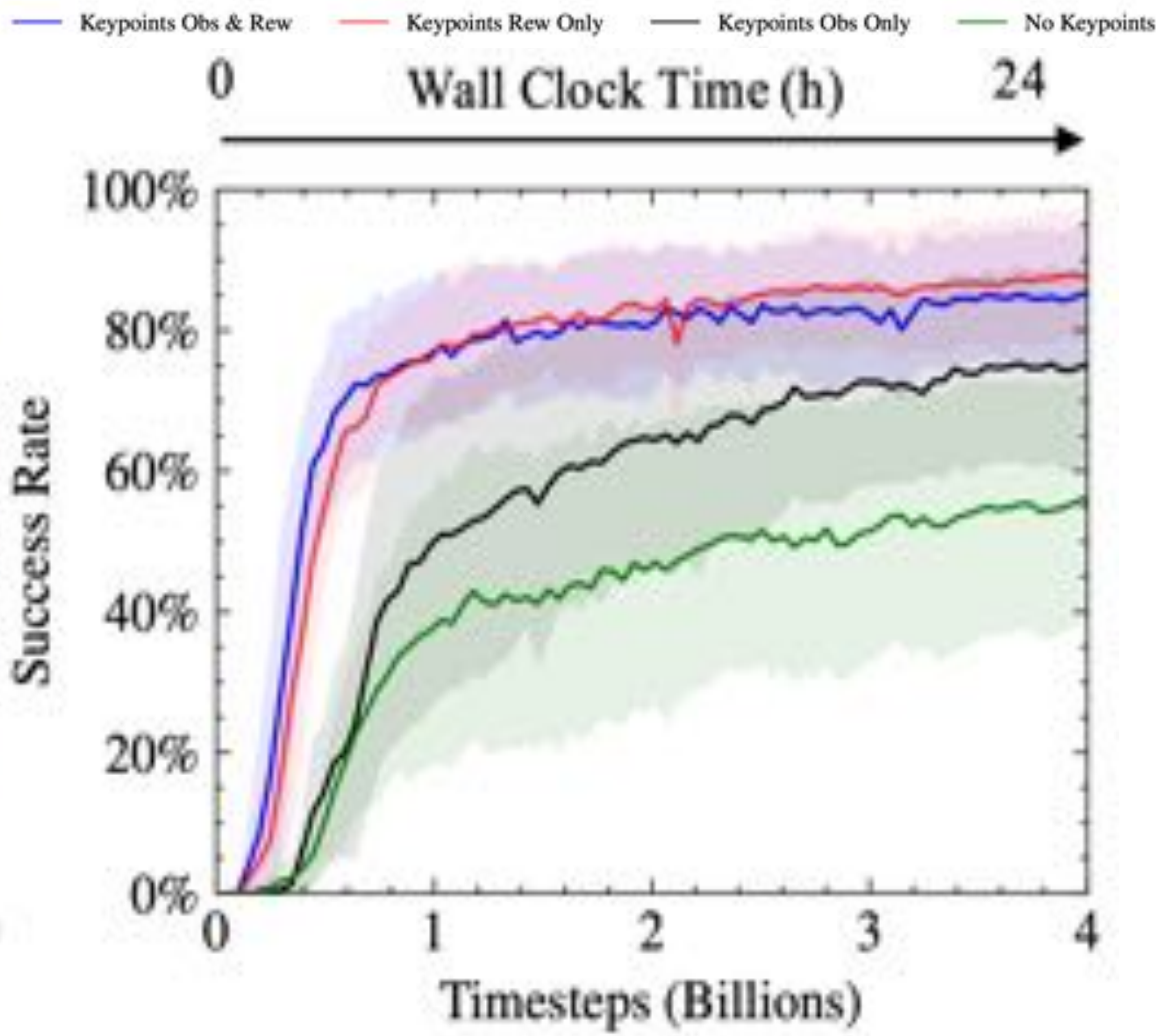


Allshire et al arxiv 2021

Multi-finger In-Hand Manipulation

- Able to train in <24h on 1 GPU





Allshire et al arxiv 2021

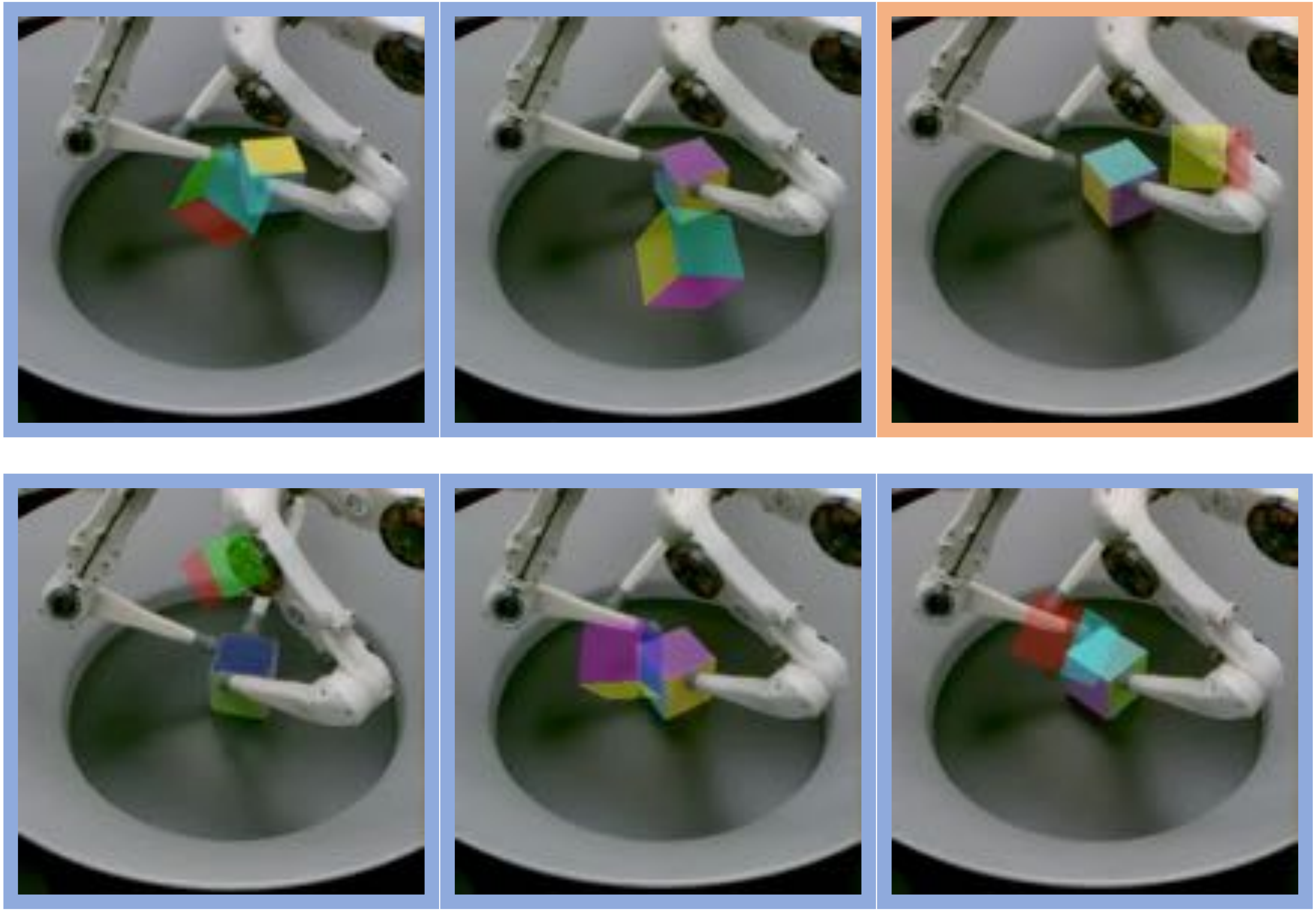
Sim2Real
Results

Robotics as a Service

No physical robot access

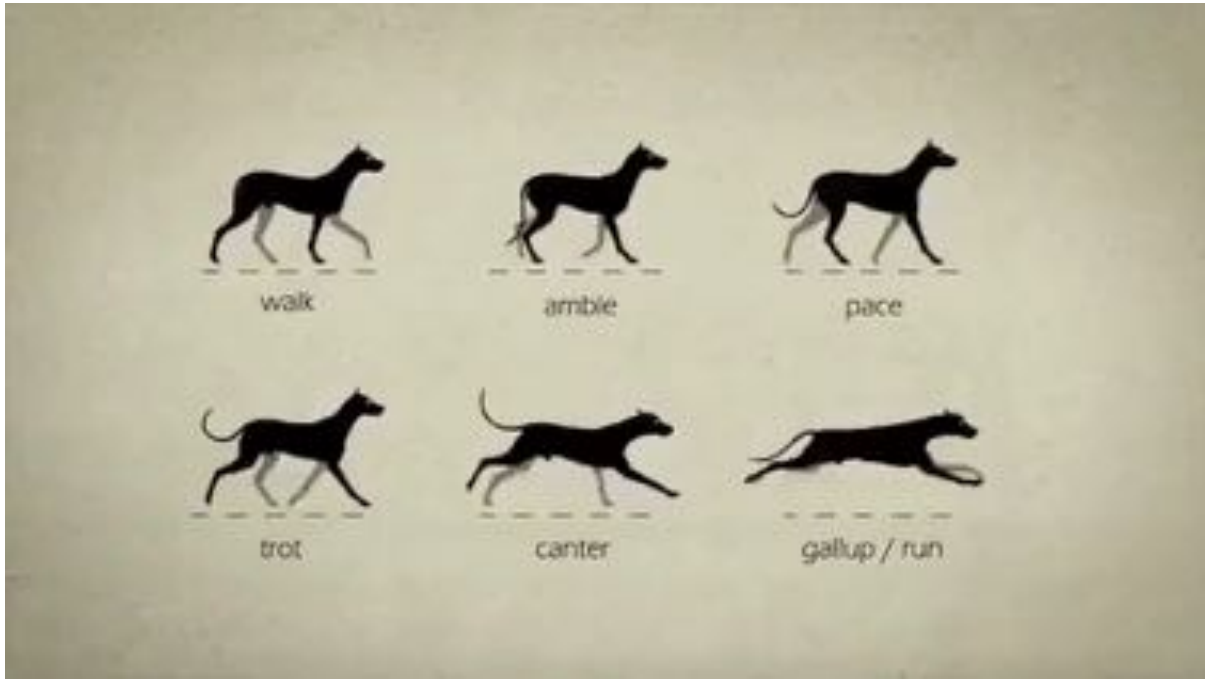
83% Success

(Real Time Videos)



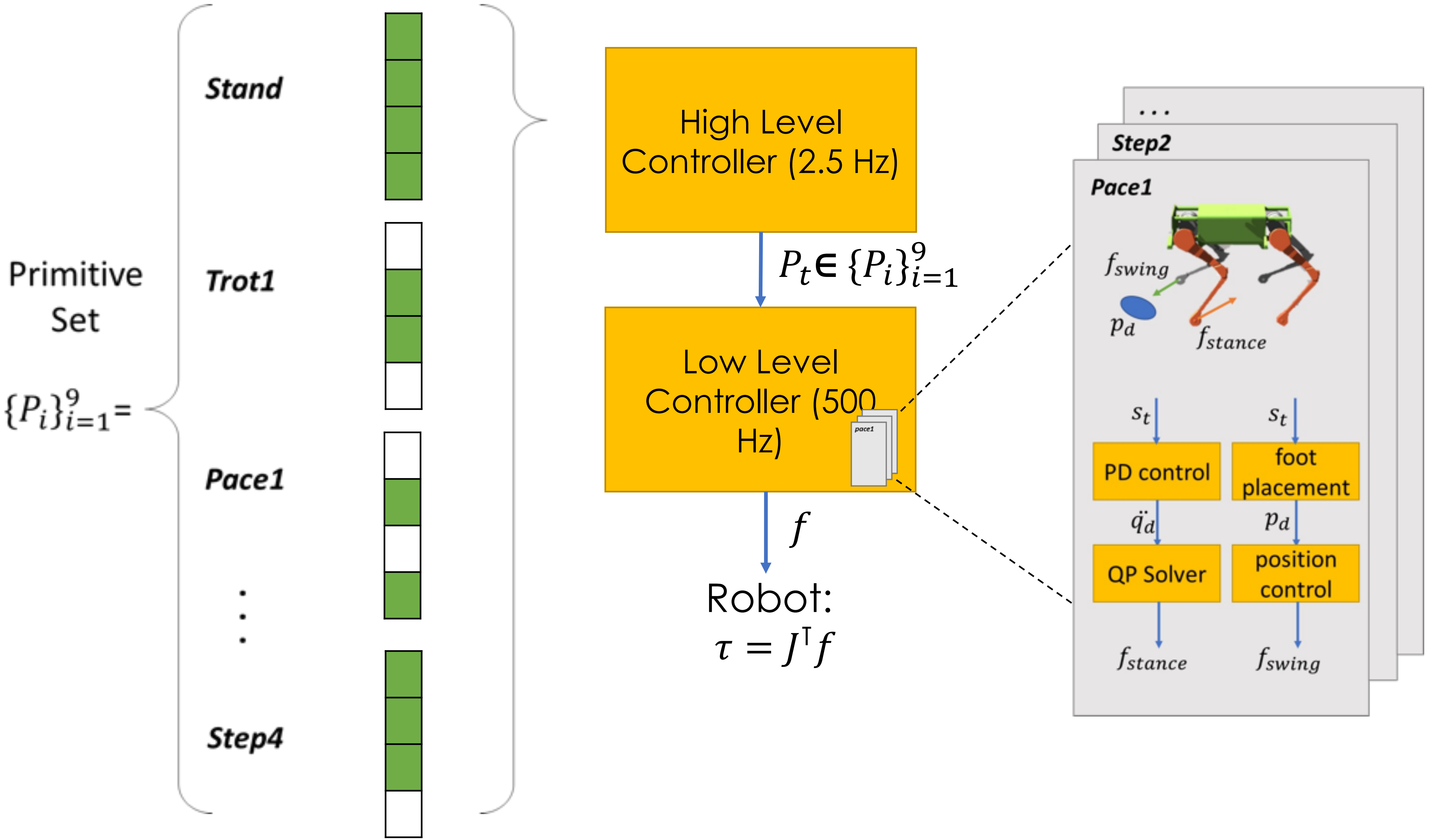
Sim2Real: Learning to Walk

Locomotion: Situation Specific Gaits



common quadrupedal gaits

custom gait



Primitive Set
$\{P_i\}_{i=1}^{9}=$
Stand
Trot1
Pace1
Step4
High Level Controller (2.5 Hz)
$P_t \in \{P_i\}_{i=1}^{9}$
Low Level Controller (500 Hz)
pace1
f
Robot:
$\tau = J^{\intercal} f$
...
Step2
Pace1
f_{swing}
p_d
f_{stance}
s_t
s_t
PD control
foot placement
$\ddot{q}_d$
p_d
QP Solver
position control
f_{stance}
f_{swing}

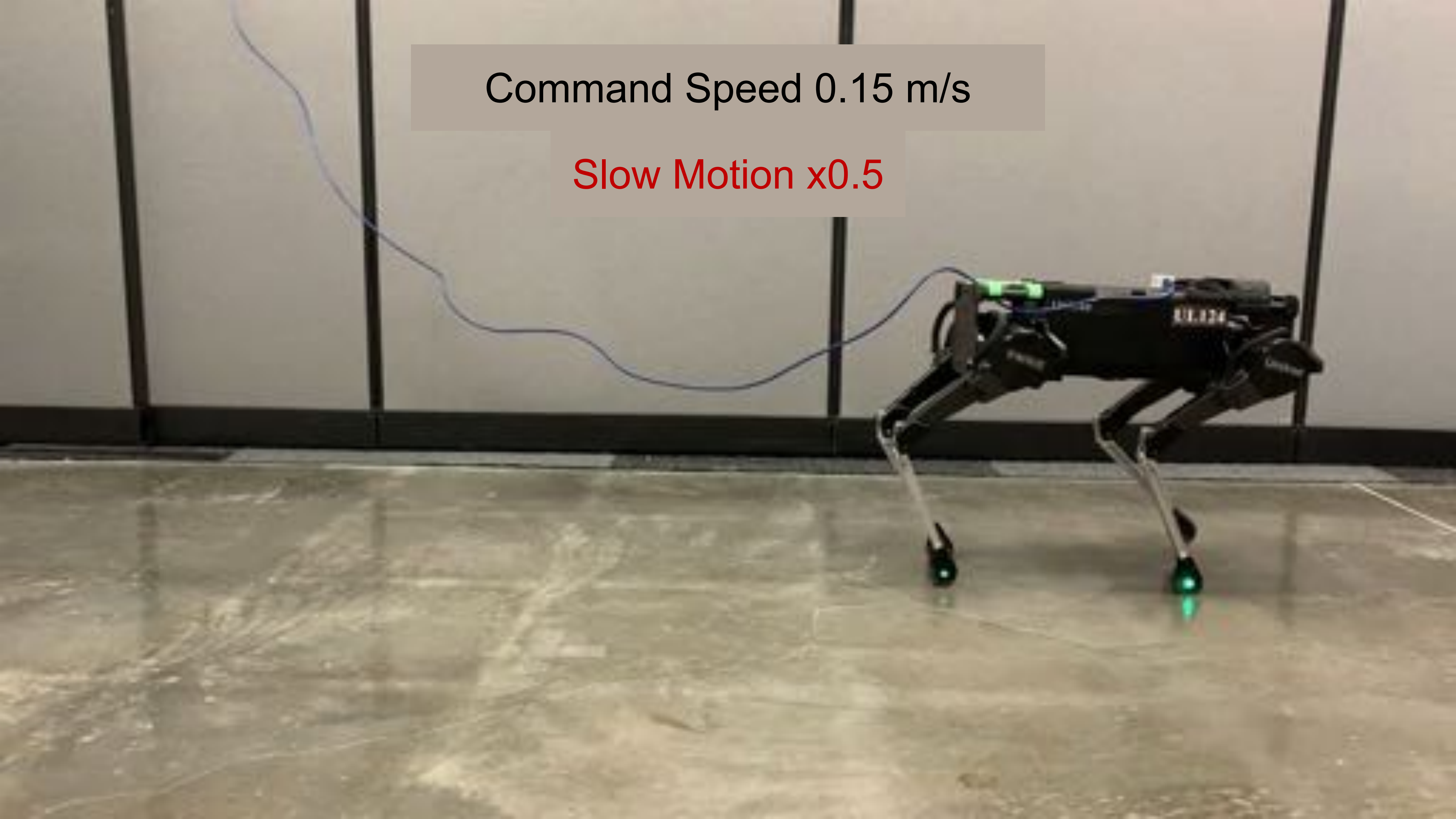
Training Setup

Variations:
- Treadmill belts
- Treadmill speed
- Robot orientation

Rewards:
- Stay balance
- Stay in place
- Minimize energy

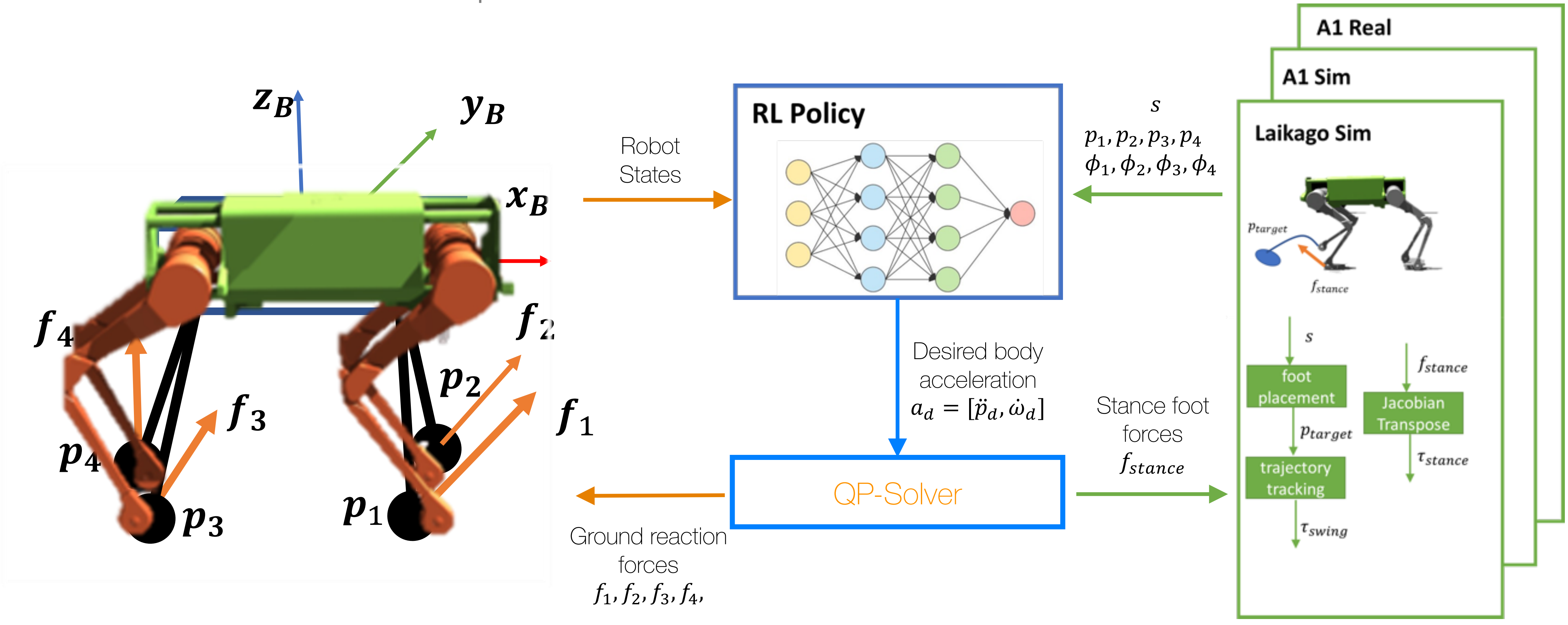
Treadmill Speed 0 m/s

Treadmill Speed 0.15 m/s


Command Speed 0.15 m/s
Slow Motion x0.5

Representations RL: Task Spaces

Full model to Simplified Model



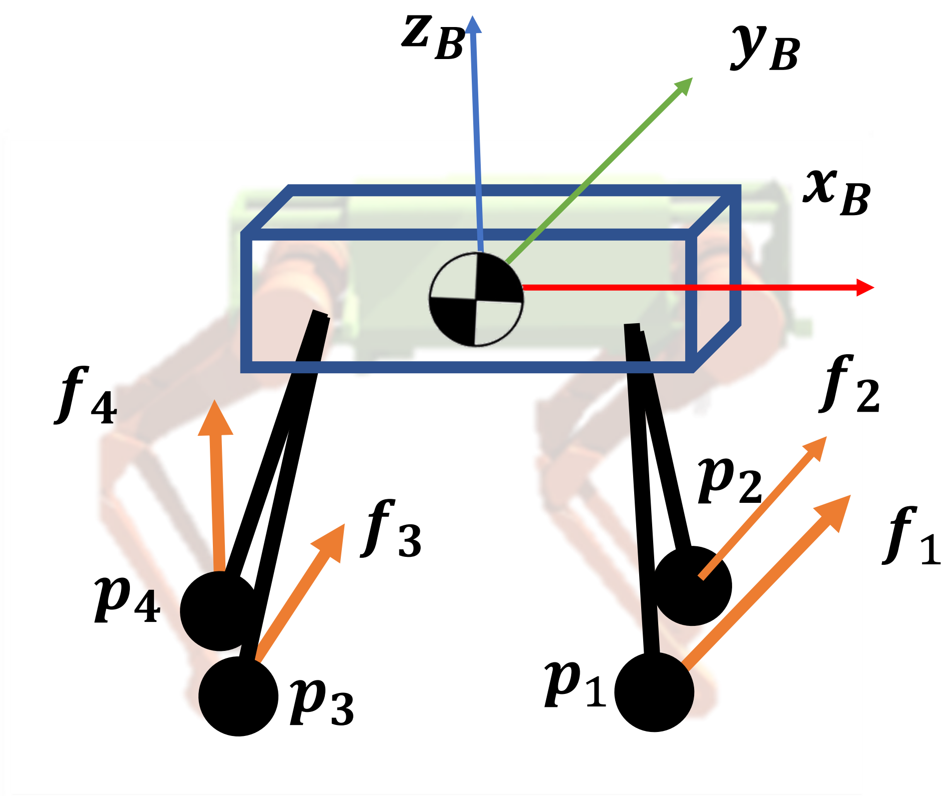
Centroidal Task Space

Full Robot Model

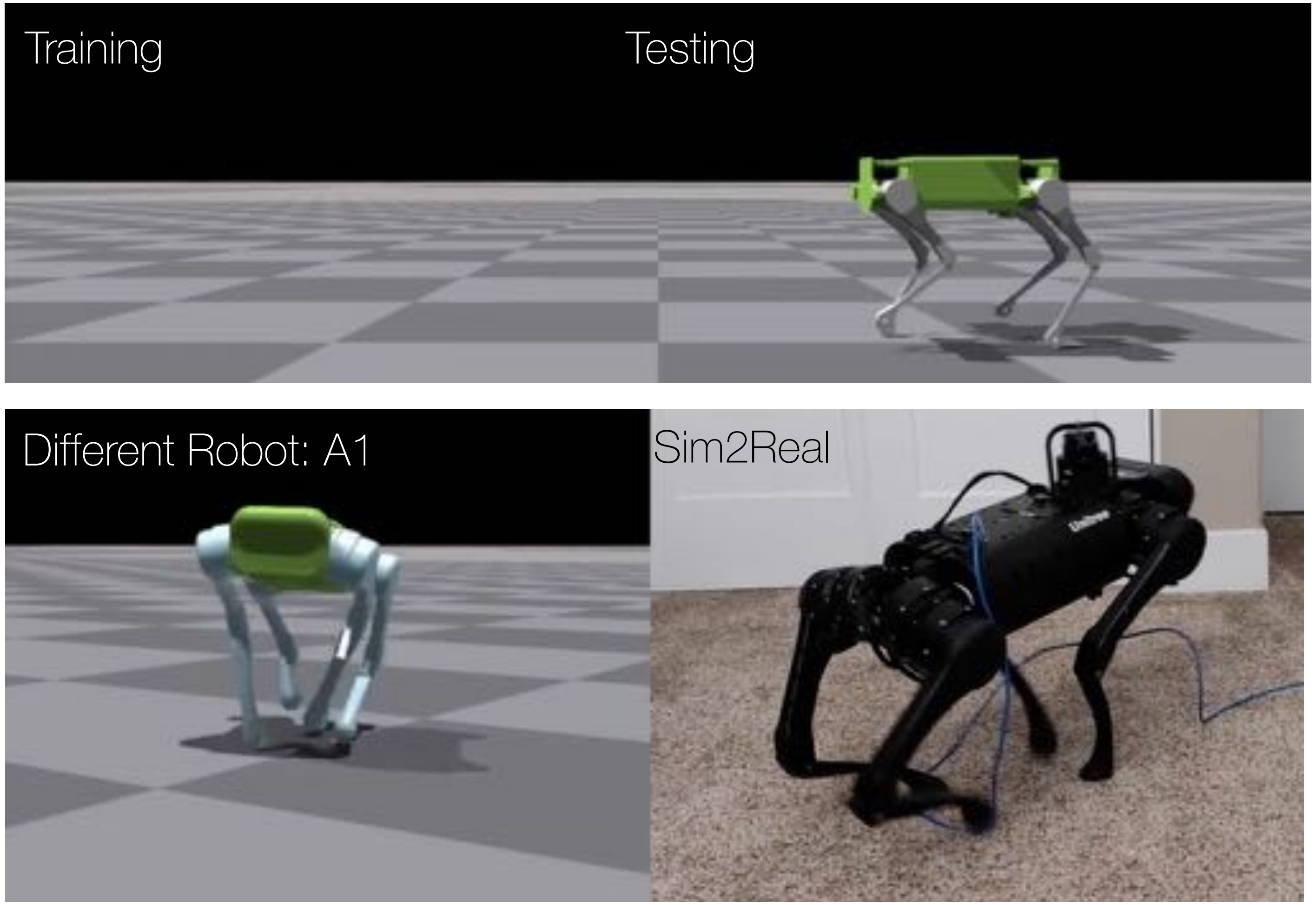
GLiDE: Generalizable Quadrupedal Locomotion, under review 2021

Representations RL: Task Spaces

Full model to Simplified Model



Centroidal Task Space

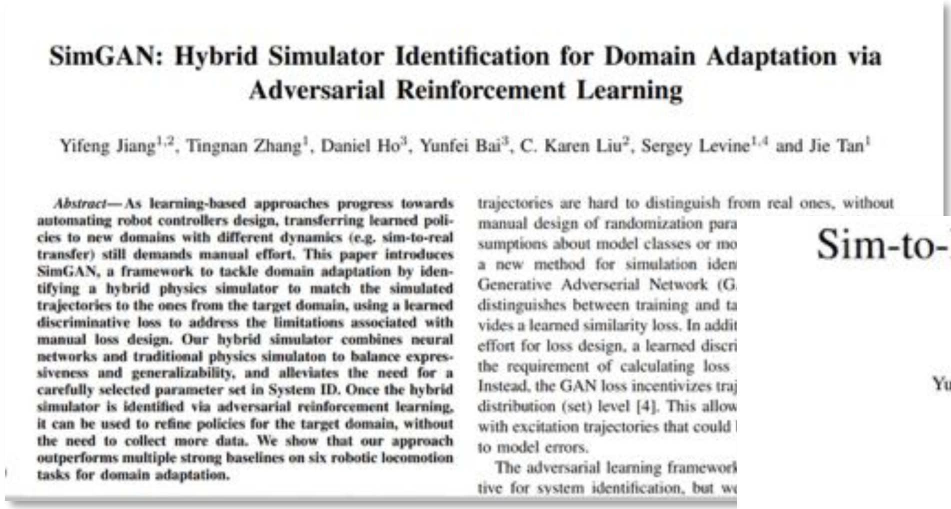


GLiDE: Generalizable Quadrupedal Locomotion, under review 2021

Sim-to-Real: Is Randomization all we need?

Myth 1: Sim-to-Real is Hard

Myth 2: Randomization is Necessary

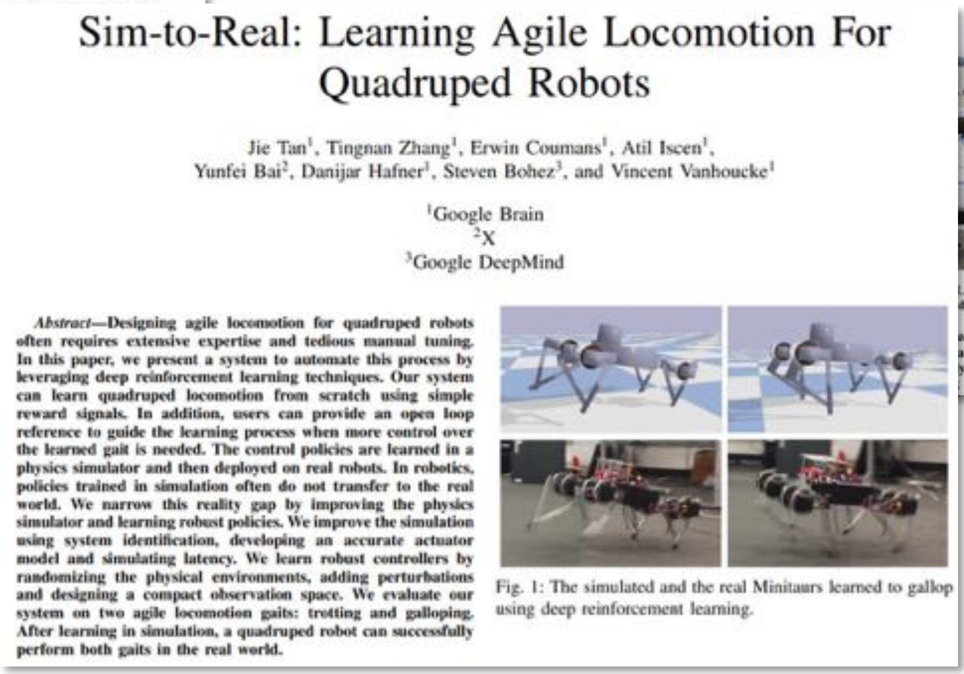

SimGAN: Hybrid Simulator Identification for Domain Adaptation via Adversarial Reinforcement Learning

Yifeng Jiang[1,2], Tingnan Zhang[1], Daniel Ho[3], Yunfei Bai[3], C. Karen Liu[2], Sergey Levine[1,4] and Jie Tan[1]

***Abstract*—As learning-based approaches progress towards automating robot controllers design, transferring learned policies to new domains with different dynamics (e.g. sim-to-real transfer) still demands manual effort. This paper introduces SimGAN, a framework to tackle domain adaptation by identifying a hybrid physics simulator to match the simulated trajectories to the ones from the target domain, using a learned discriminative loss to address the limitations associated with manual loss design. Our hybrid simulator combines neural networks and traditional physics simulaton to balance expressiveness and generalizability, and alleviates the need for a carefully selected parameter set in System ID. Once the hybrid simulator is identified via adversarial reinforcement learning, it can be used to refine policies for the target domain, without the need to collect more data. We show that our approach outperforms multiple strong baselines on six robotic locomotion tasks for domain adaptation.**

trajectories are hard to distinguish from real ones, without manual design of randomization para sumptions about model classes or mo a new method for simulation iden Generative Adverserial Network (G distinguishes between training and ta vides a learned similarity loss. In addit effort for loss design, a learned discri the requirement of calculating loss Instead, the GAN loss incentivizes traj distribution (set) level [4]. This allow with excitation trajectories that could to model errors.

The adversarial learning framework tive for system identification, but we

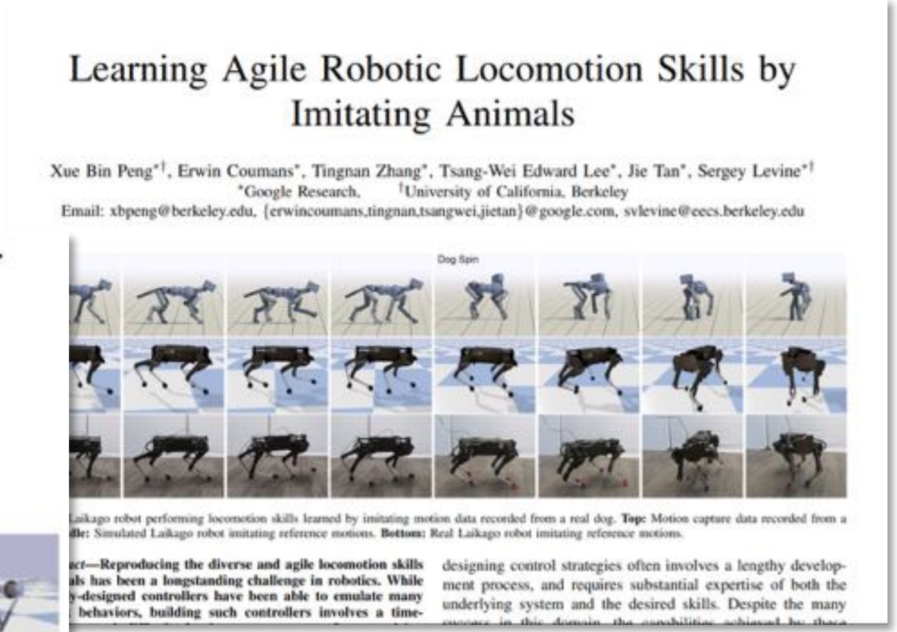

Sim-to-Real: Learning Agile Locomotion For Quadruped Robots

Jie Tan[1], Tingnan Zhang[1], Erwin Coumans[1], Atil Iscen[1], Yunfei Bai[2], Danijar Hafner[1], Steven Bohez[3], and Vincent Vanhoucke[1]

[1]Google Brain
[2]X
[3]Google DeepMind

***Abstract*—Designing agile locomotion for quadruped robots often requires extensive expertise and tedious manual tuning. In this paper, we present a system to automate this process by leveraging deep reinforcement learning techniques. Our system can learn quadruped locomotion from scratch using simple reward signals. In addition, users can provide an open loop reference to guide the learning process when more control over the learned gait is needed. The control policies are learned in a physics simulator and then deployed on real robots. In robotics, policies trained in simulation often do not transfer to the real world. We narrow this reality gap by improving the physics simulator and learning robust policies. We improve the simulation using system identification, developing an accurate actuator model and simulating latency. We learn robust controllers by randomizing the physical environments, adding perturbations and designing a compact observation space. We evaluate our system on two agile locomotion gaits: trotting and galloping. After learning in simulation, a quadruped robot can successfully perform both gaits in the real world.**

Fig. 1: The simulated and the real Minitaurs learned to gallop using deep reinforcement learning.


Learning Agile Robotic Locomotion Skills by Imitating Animals

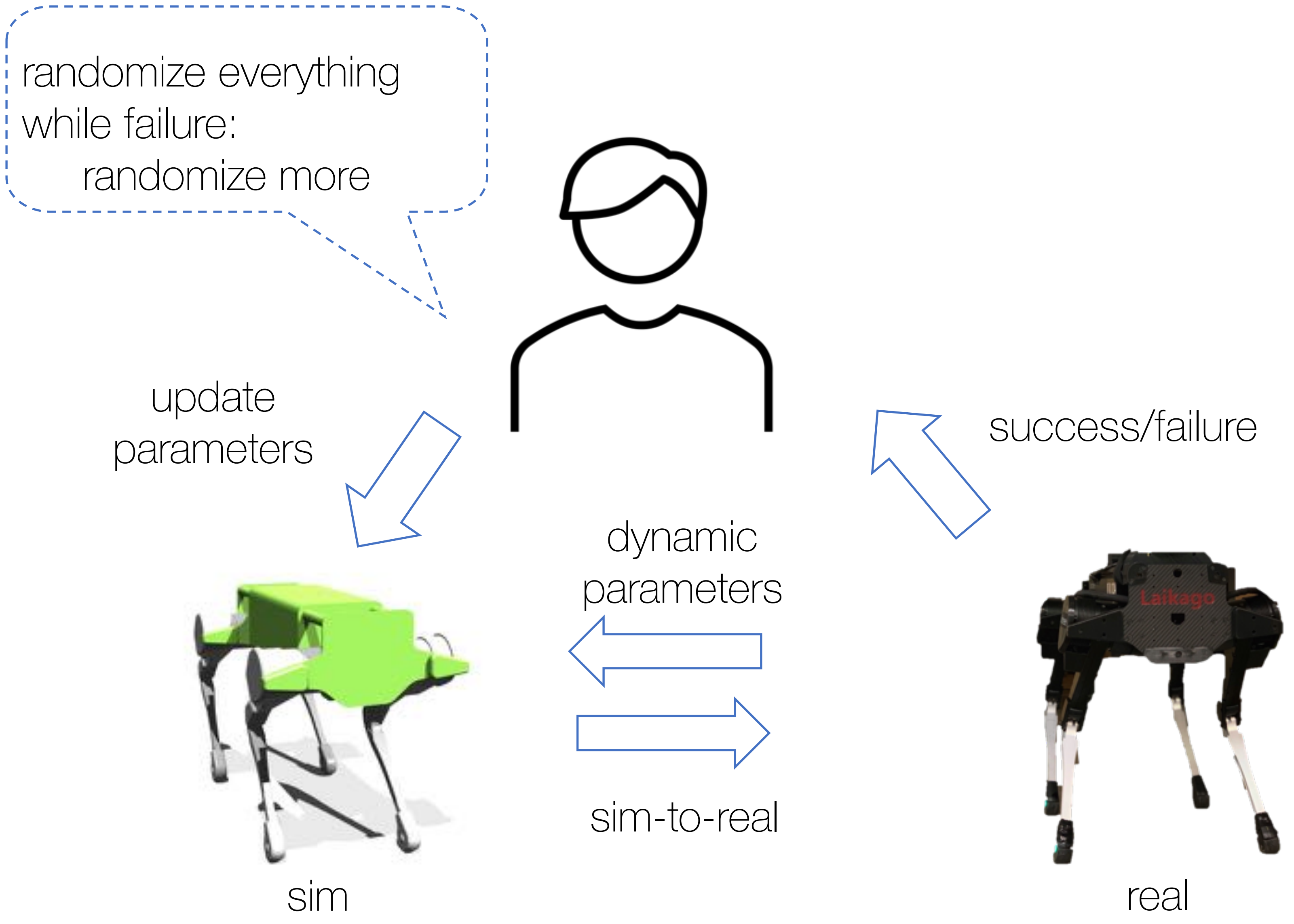
Xue Bin Peng*†, Erwin Coumans*, Tingnan Zhang*, Tsang-Wei Edward Lee*, Jie Tan*, Sergey Levine*†
*Google Research, †University of California, Berkeley
Email: xbpeng@berkeley.edu, {erwincoumans,tingnan,tsangwei,jietan}@google.com, svlevine@eecs.berkeley.edu

Laikago robot performing locomotion skills learned by imitating motion data recorded from a real dog. **Top:** Motion capture data recorded from a **dle:** Simulated Laikago robot imitating reference motions. **Bottom:** Real Laikago robot imitating reference motions.

act—Reproducing the diverse and agile locomotion skills als has been a longstanding challenge in robotics. While y-designed controllers have been able to emulate many behaviors, building such controllers involves a time-

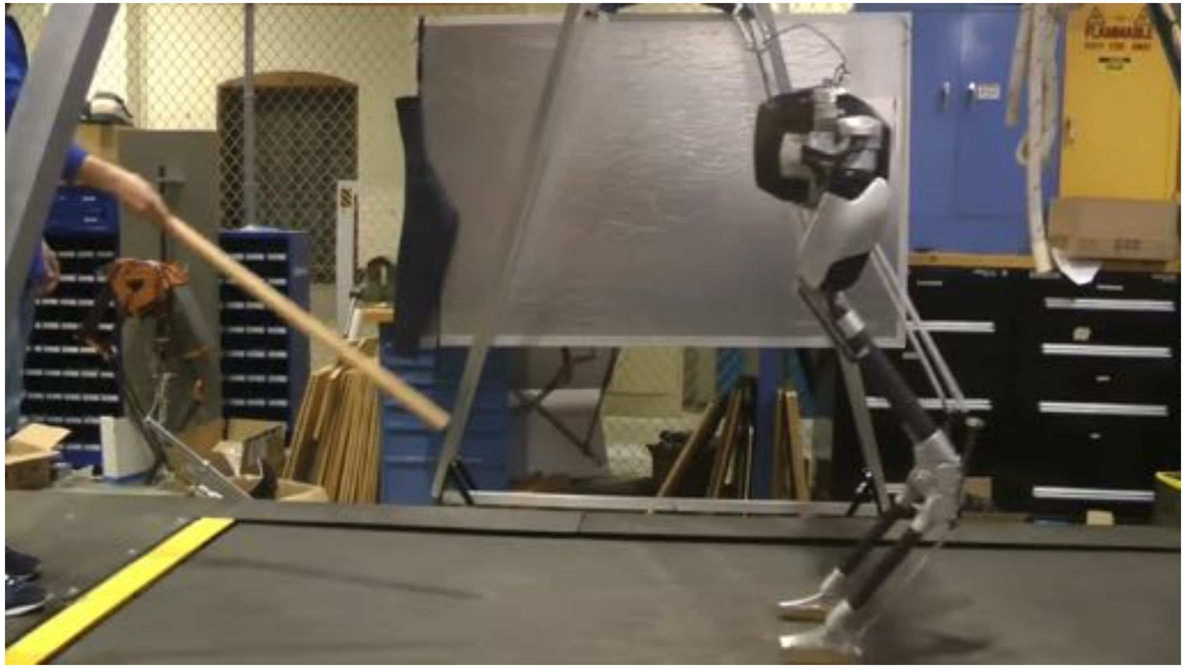
designing control strategies often involves a lengthy development process, and requires substantial expertise of both the underlying system and the desired skills. Despite the many

Sim-to-Real


randomize everything
while failure:
randomize more
update
parameters
success/failure
dynamic
parameters
sim-to-real
sim
real
Laikago

Sim-to-Real: Without Randomization

Dynamics Randomization: Necessary?



Learning Locomotion Skills for Cassie: Iterative Design and Sim-to-Real
CoRL 2020

Dynamics Randomization Revisited: A Case Study for Quadrupedal Locomotion
ICRA 2021

Sim-to-Real: With Randomization

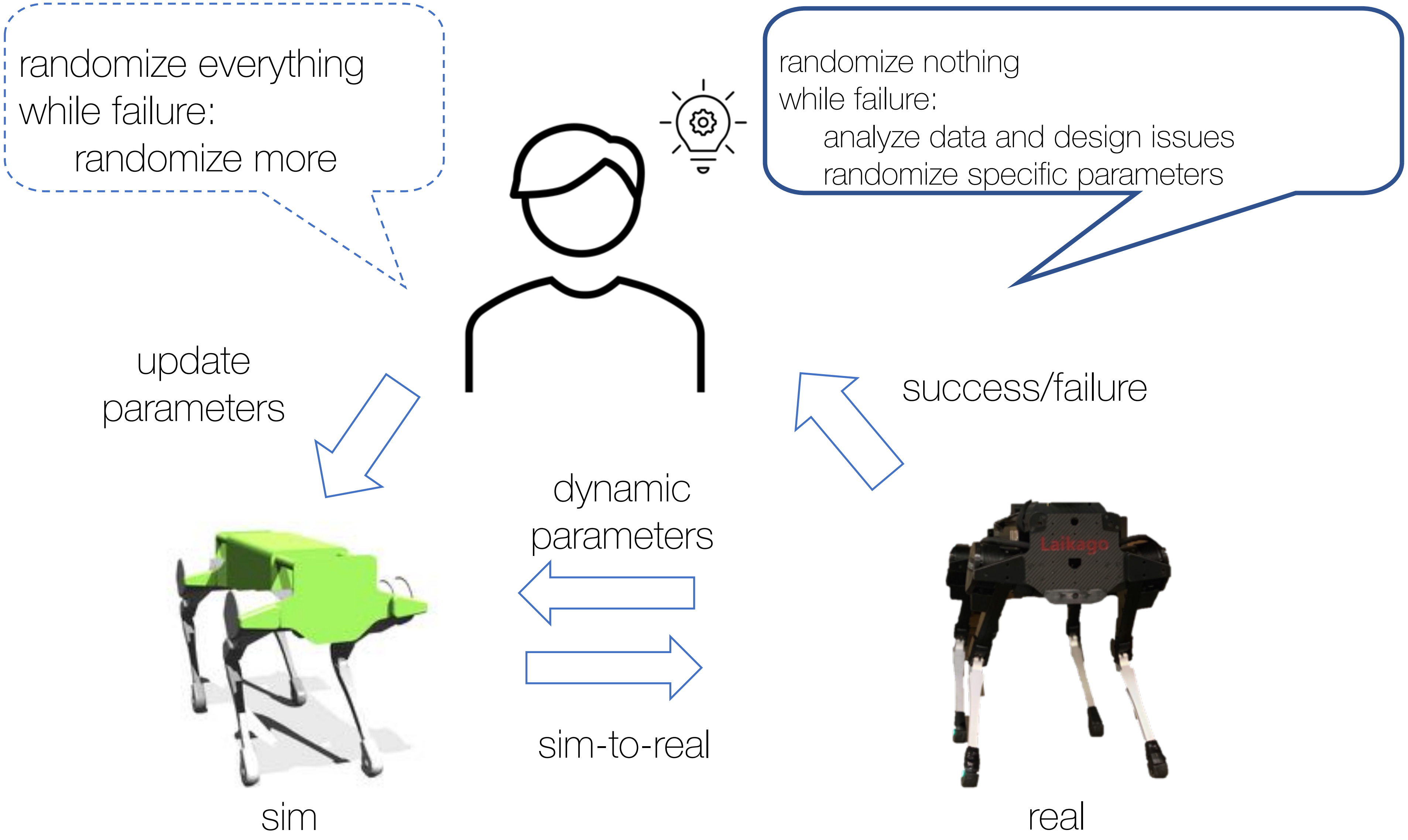
Dynamics Randomization: Sufficient?

Design Choices Matter

No Velocity Feedback

High Joint Gains

Sim-to-Real

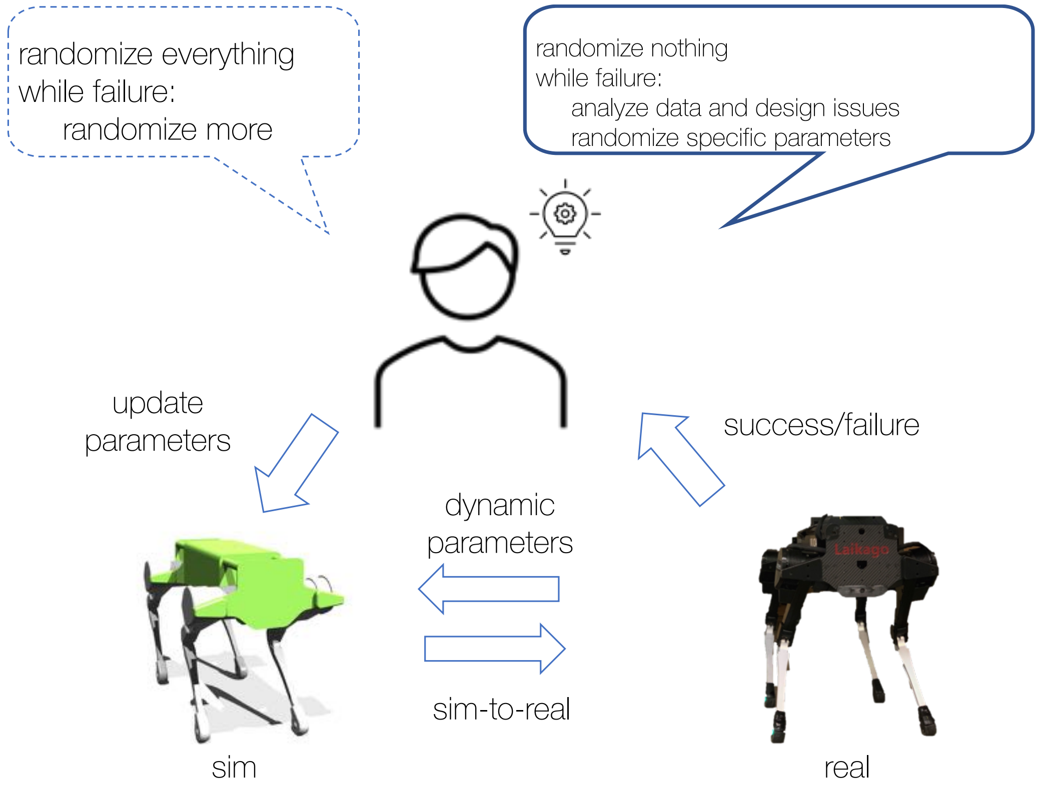

randomize everything
while failure:
randomize more
randomize nothing
while failure:
analyze data and design issues
randomize specific parameters
update parameters
success/failure
dynamic parameters
sim-to-real
Laikago
sim
real

Sim-to-Real

Dynamics Randomization: Neither Necessary nor Sufficient?

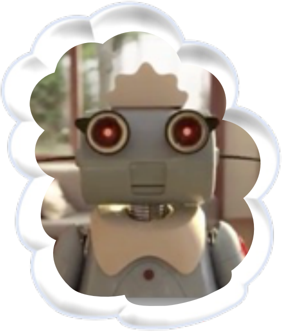
Dynamics Randomization can be avoided given right design choices.

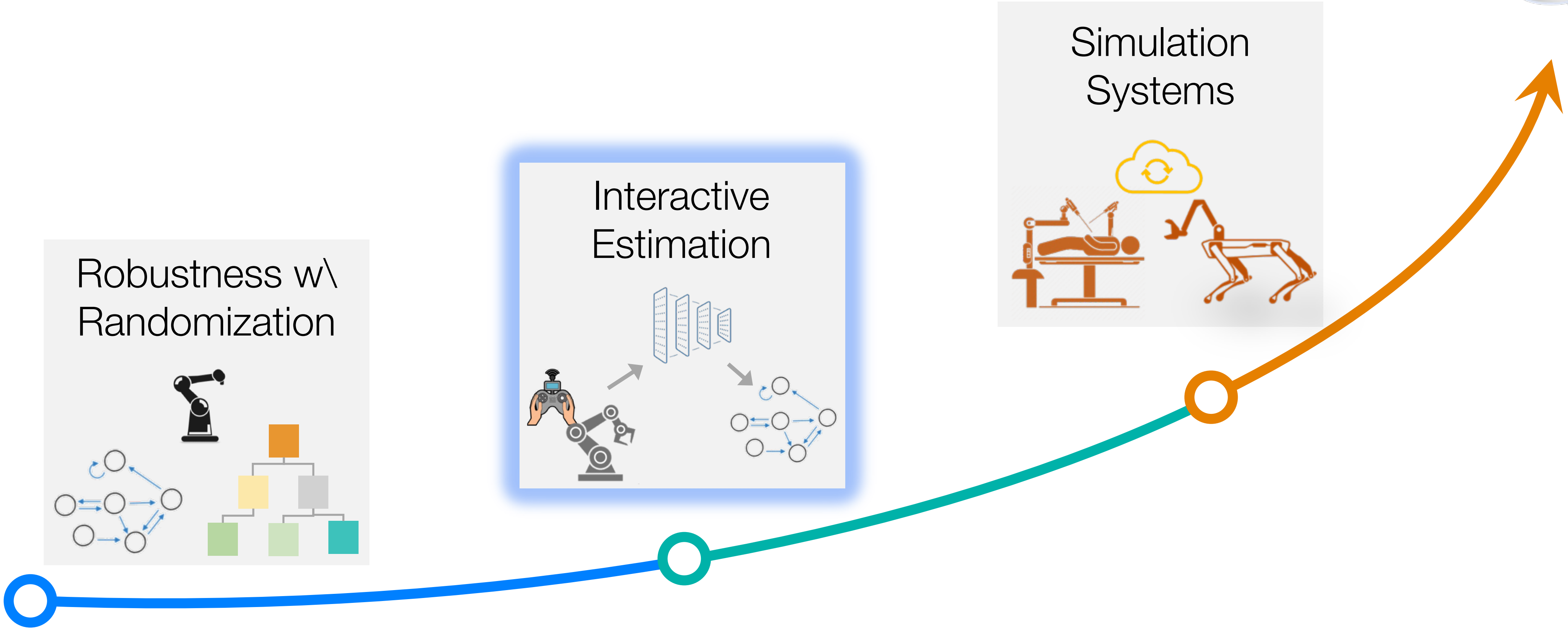
Should only be used based on domain understanding



Paving the path to Robot Autonomy with Simulation

Vision: Simulation is Data Factory for Robotics



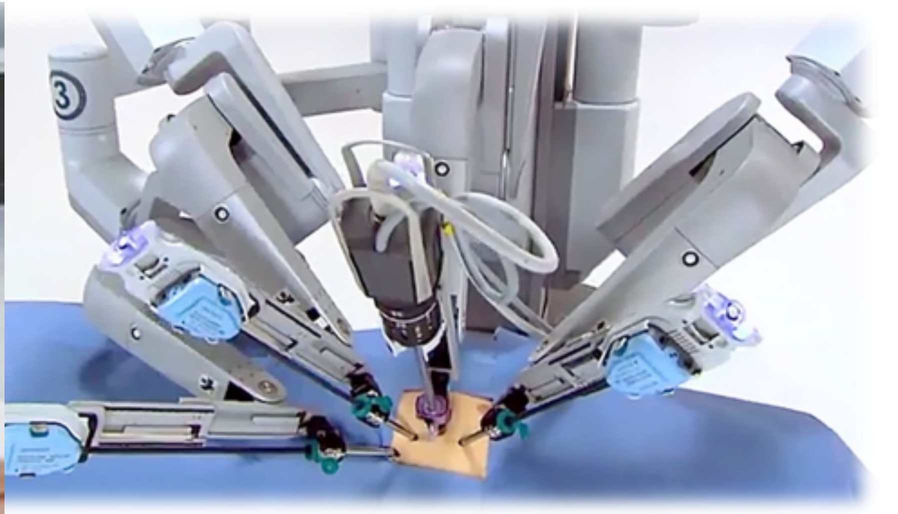


Why simulate cutting?

- Applications in food-processing, robotic surgery, household robotics
- Design of cutting machines
- Optimal motion of the cutting tool for a particular material
- Safe trajectory generation through accurate force predictions

Moley Robotics

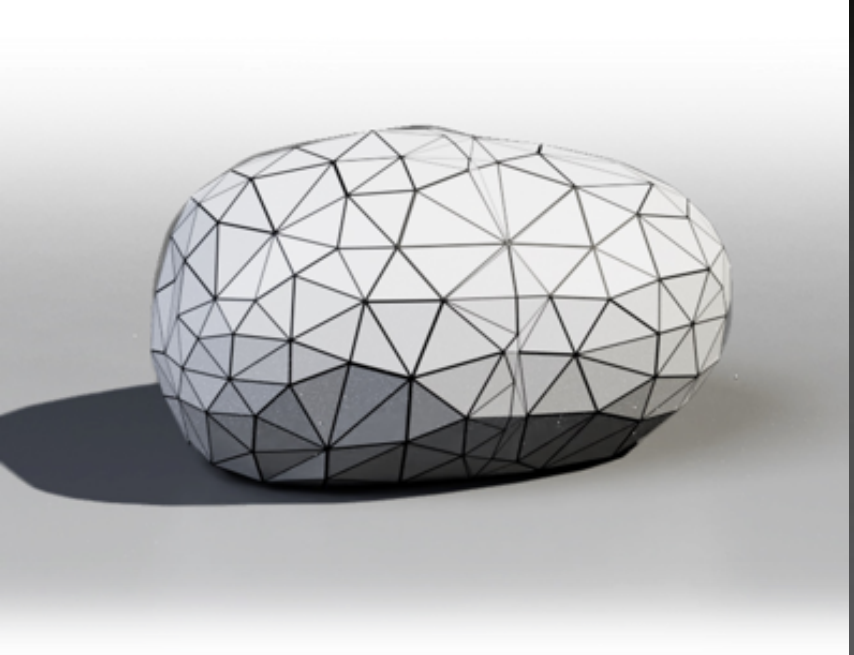
foodmanufacture.co.uk



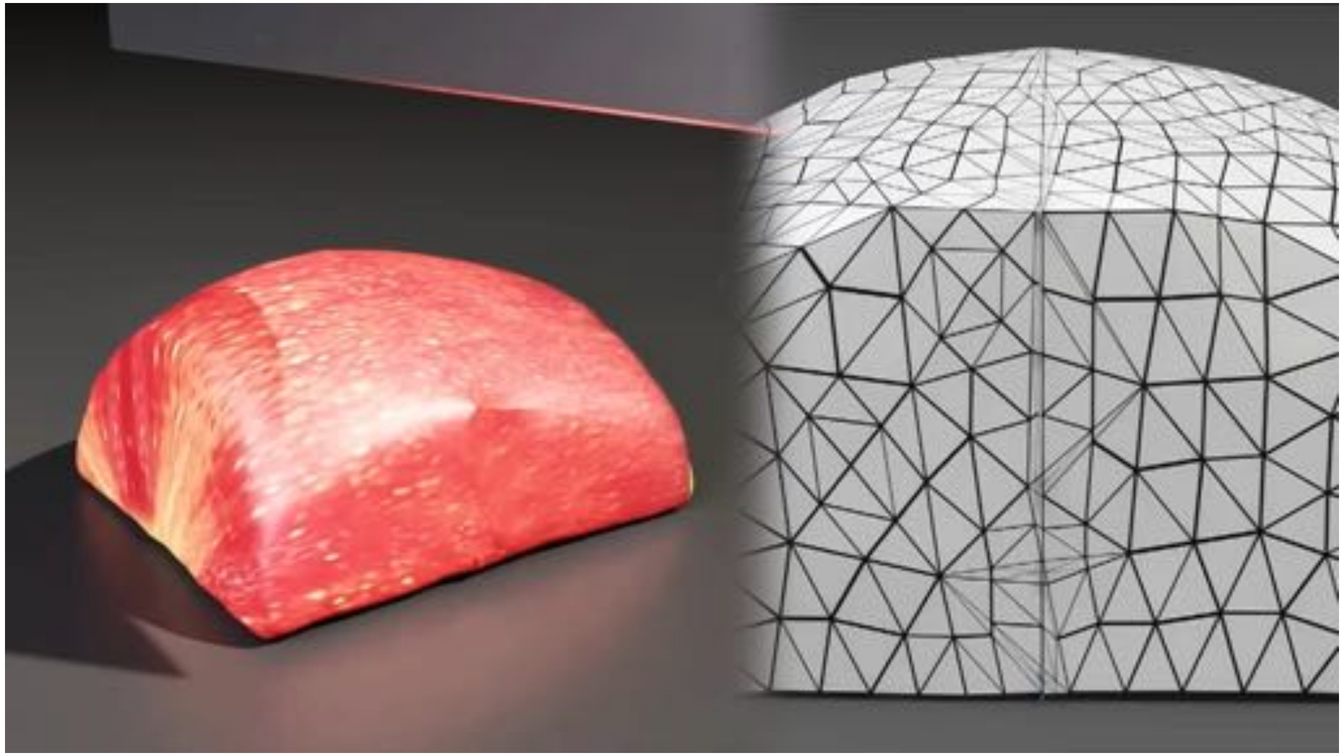
robohub.com

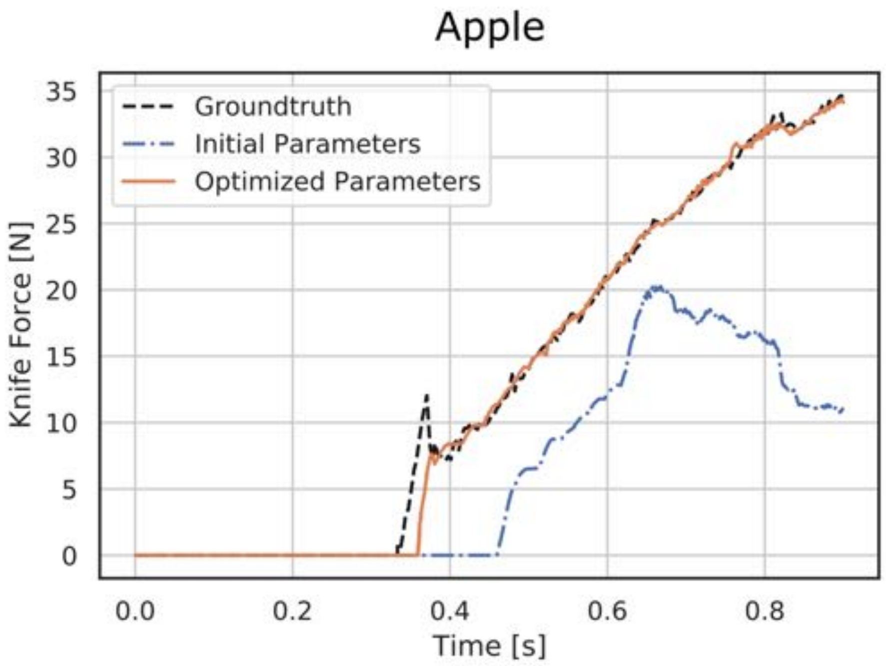
Approach

- Simulate deformable objects through Finite Element Method
- Continuous model for crack propagation, damage mechanics
- Detailed model for contact mechanics achieves realistic prediction of knife forces



Weakening of Cutting Springs

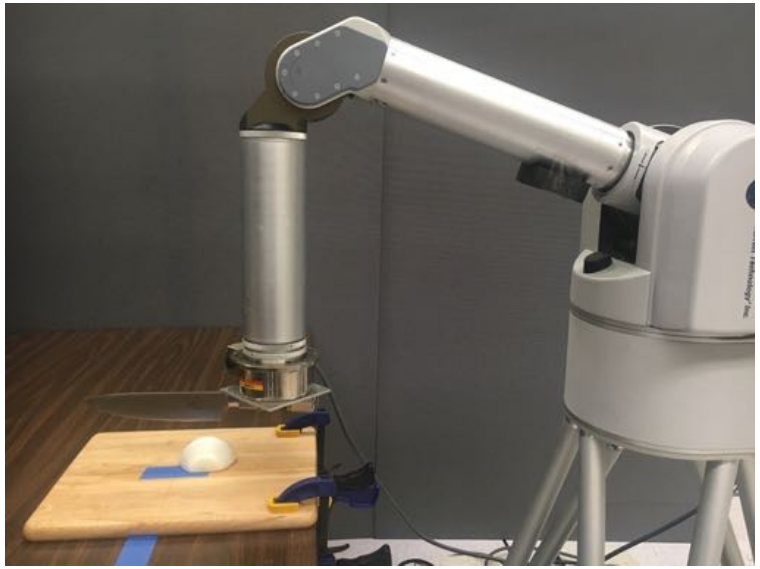


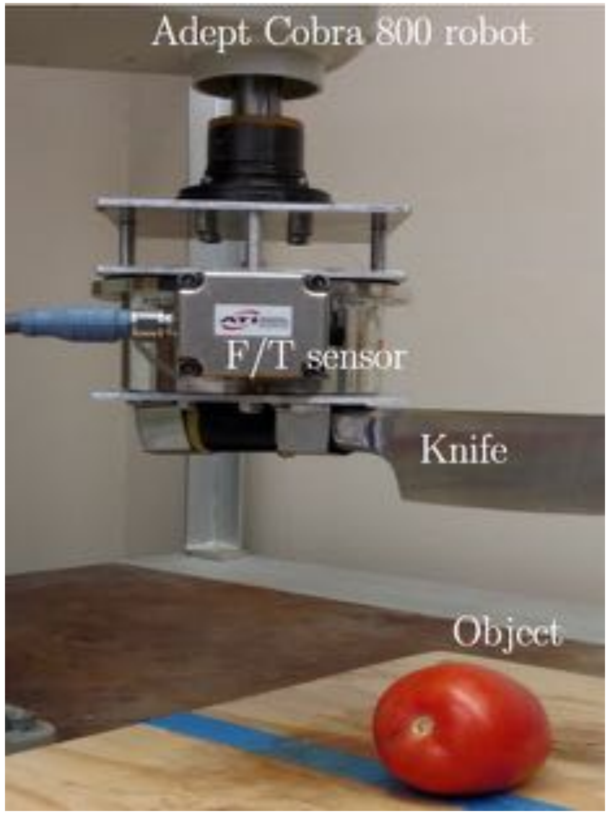


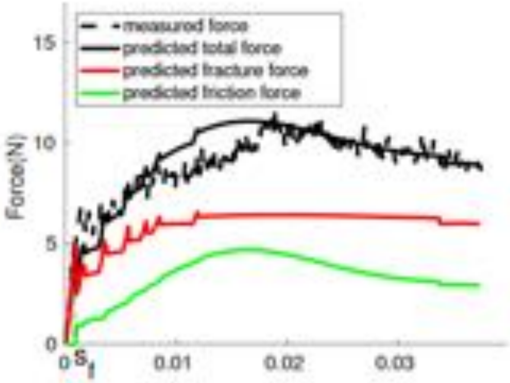
Progressive weakening of cutting springs:

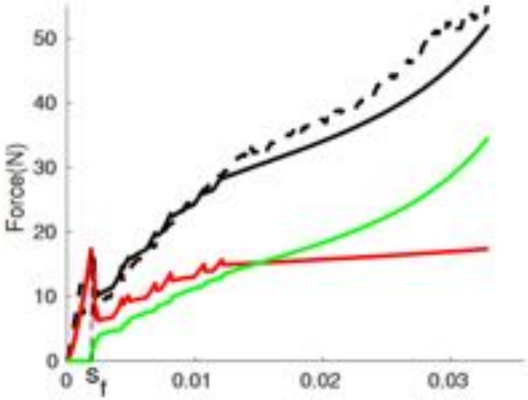
$$k'_e = k_e - \gamma \|f_{knife}\|$$

Real-robot Force Measurements

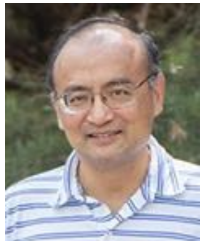






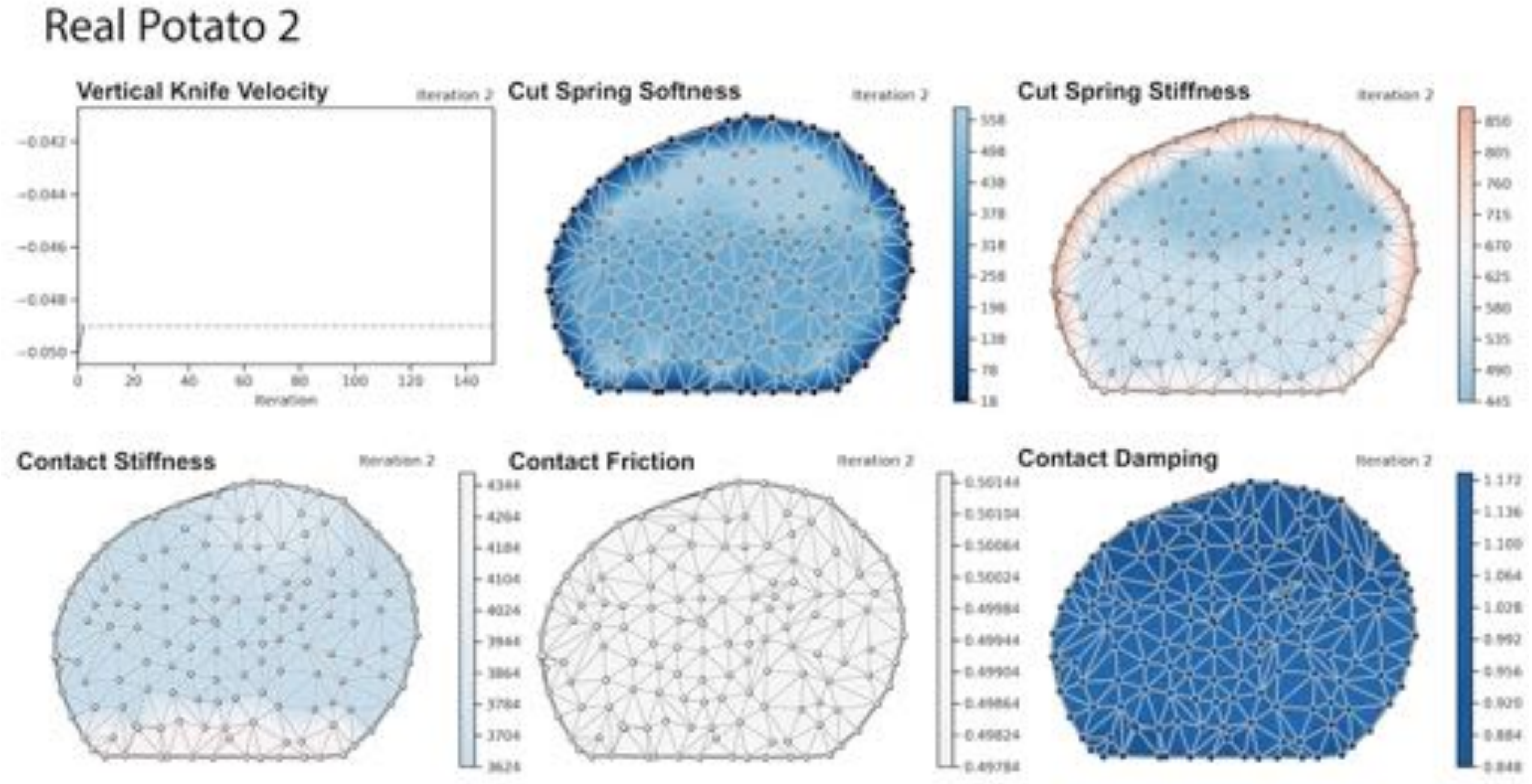


Prajjwal Jamdagni



Yan-Bin Jia

Inference of Simulation Parameters



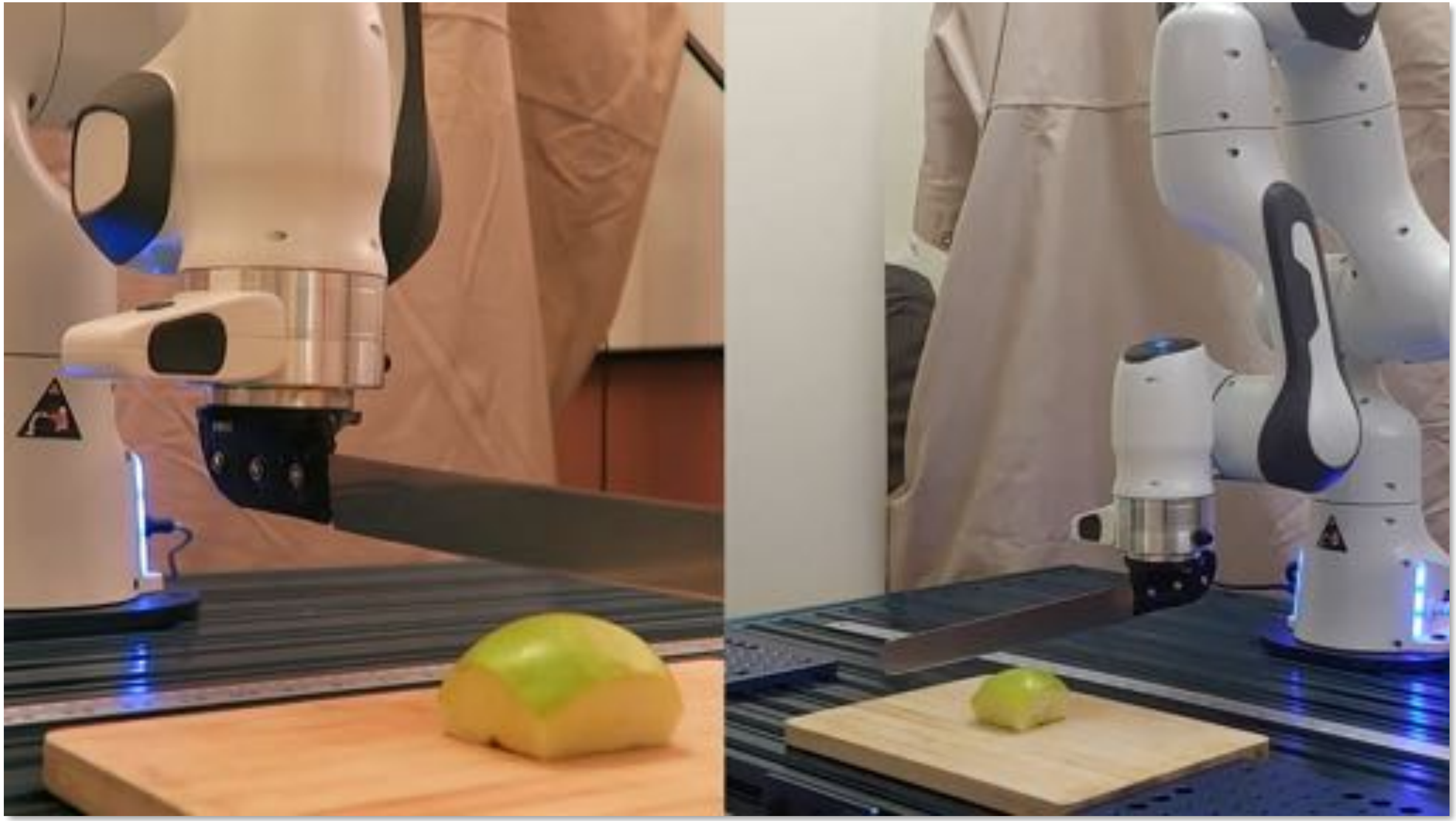
Trajectory Optimization

$$\text{minimize} \qquad \mathcal{L} = \frac{1}{T}\int f(t, \mathbf{a}, \mathbf{b}, \mathbf{c}) + \dot{y}_{\text{knife}}(t)\, dt$$

$$\text{s.t.} \quad z_{\text{knife}}(t) \leq \frac{1}{2} l_{\text{knife}}$$

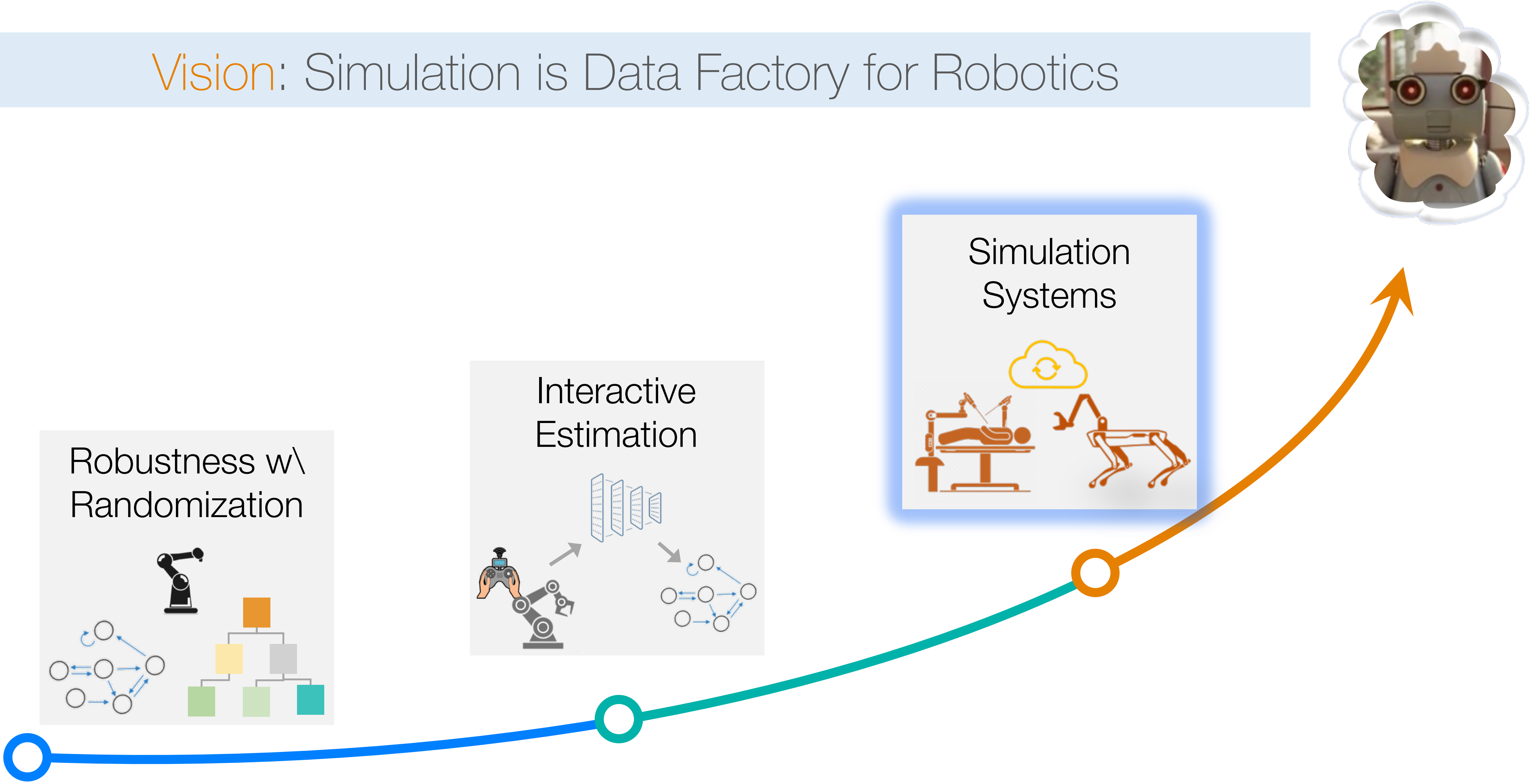
Real Robot Transfer

Model-predictive cutting on the real robot



Paving the path to Robot Autonomy with Simulation

Vision: Simulation is Data Factory for Robotics



Isaac Sim: Ease of Use

Application: Mobile Manipulation

Human environments are full of objects designed "*for us* and *by us*"

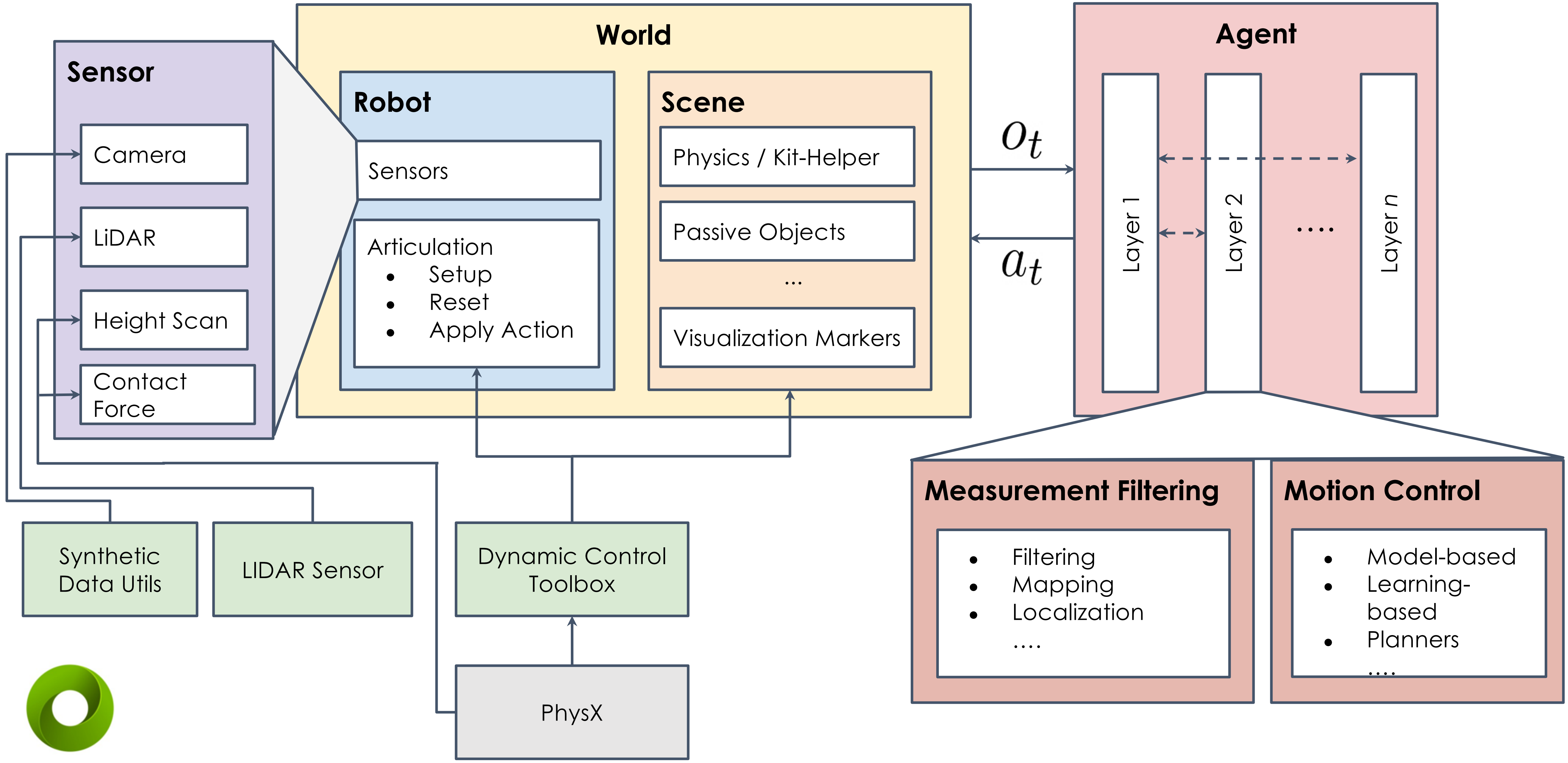
Isaac Sim: Ease of Use

Application: Mobile Manipulation

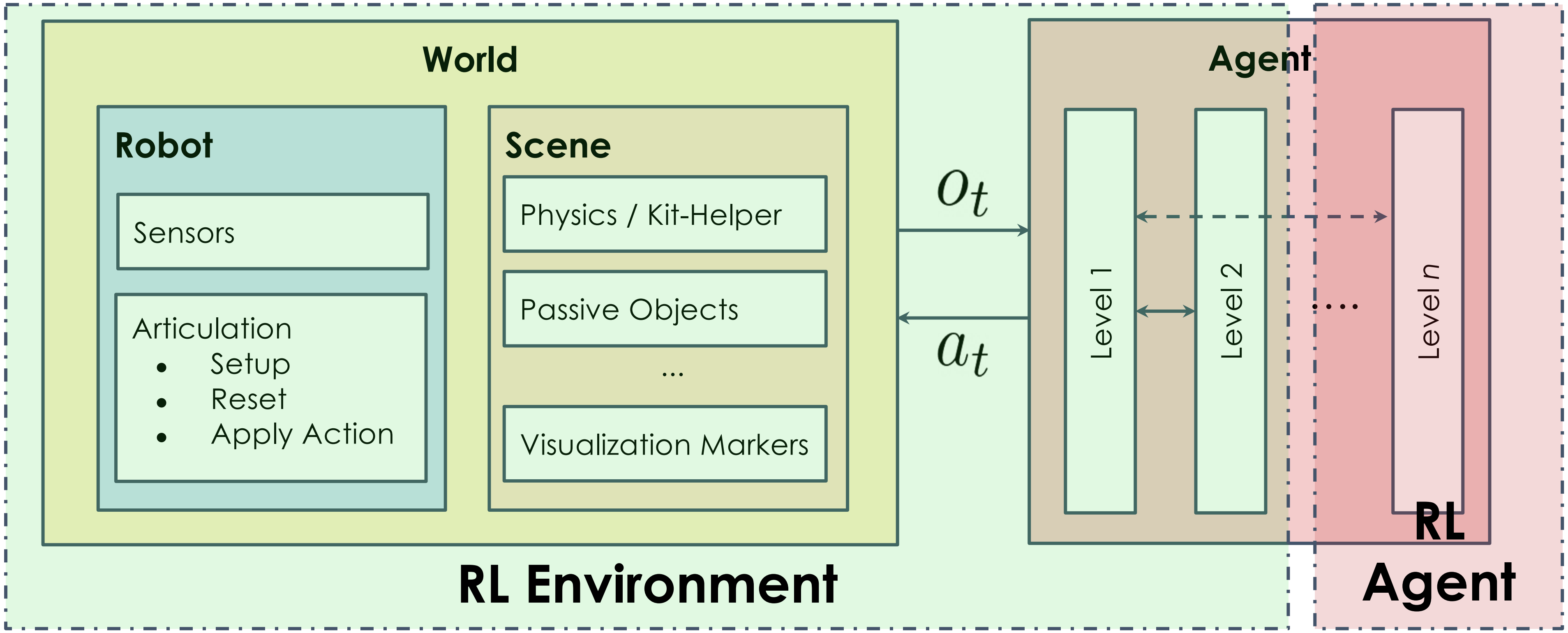
Design mobile manipulation system for articulated object interaction in human environments like kitchens

- Generalize to various kitchen layouts
- Handle intra-category variations
- Possess real-time capabilities to handle dynamic variations

Isaac Sim: Ease of Use

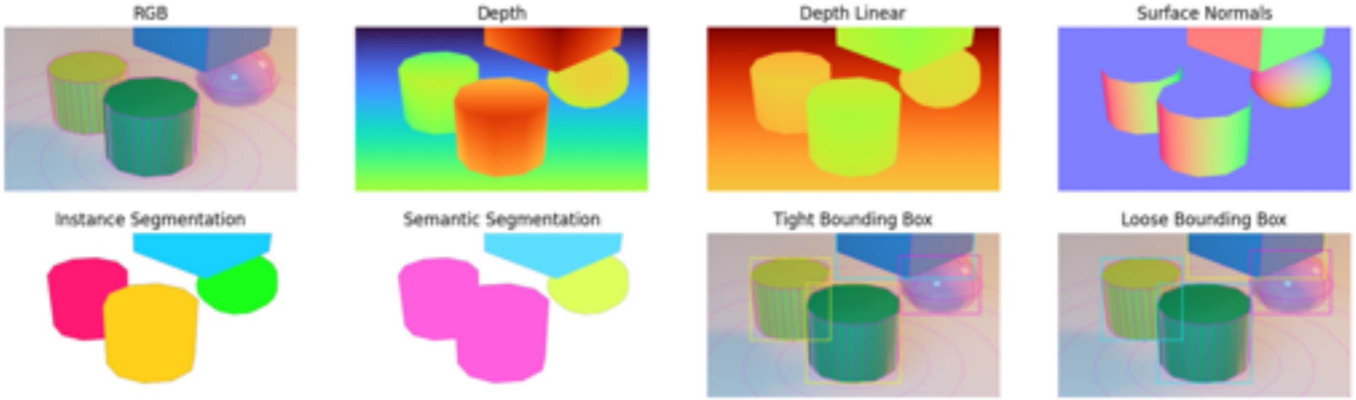


Isaac Sim: Ease of Use for RL

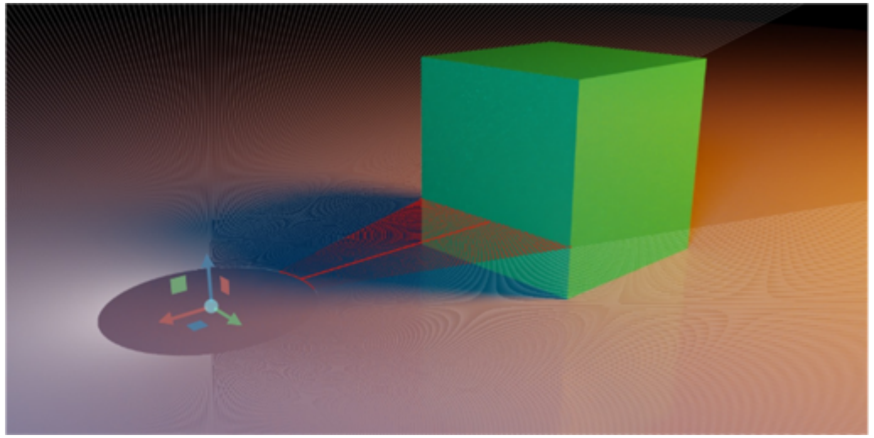


- Each layer interfaces with the next layer via "observations-actions"
- Interfaces are modular enough to ensure the "world" acts the same in simulation and real-world

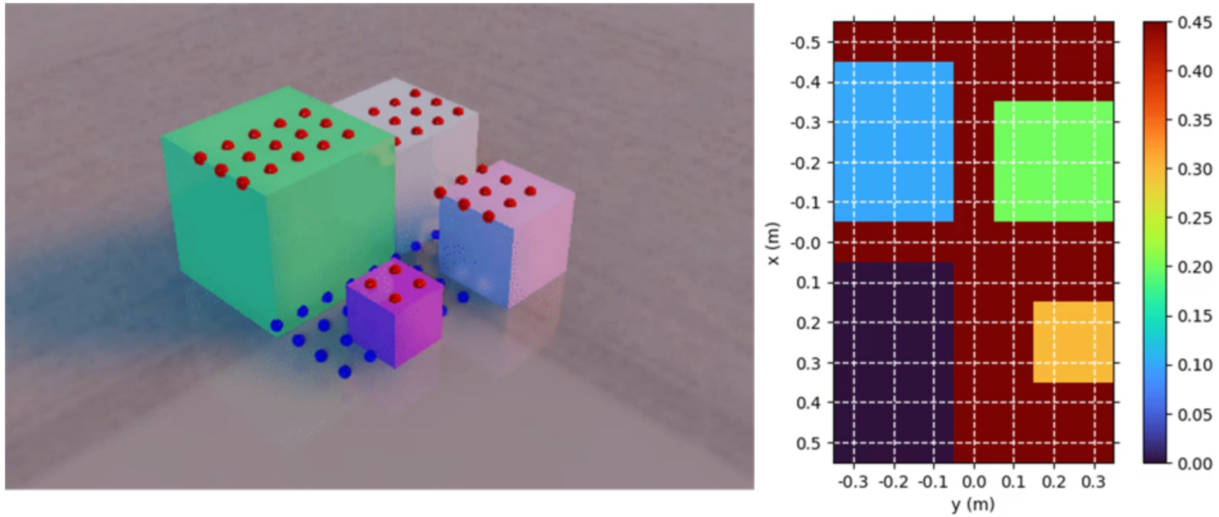
Isaac Sim: Ease of Use for RL



Multiple cameras



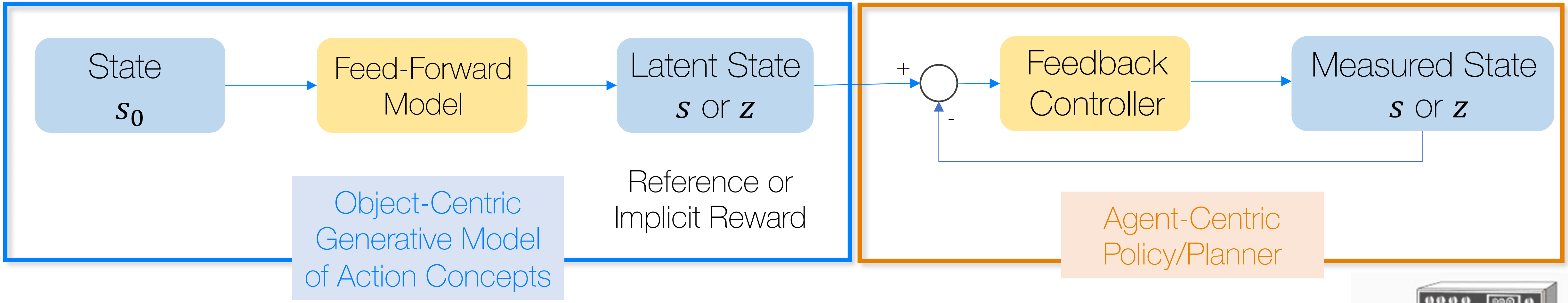
LiDAR



Height Scanner using PhysX raycasting

Contact Sensor

Structure in Compositional Planning



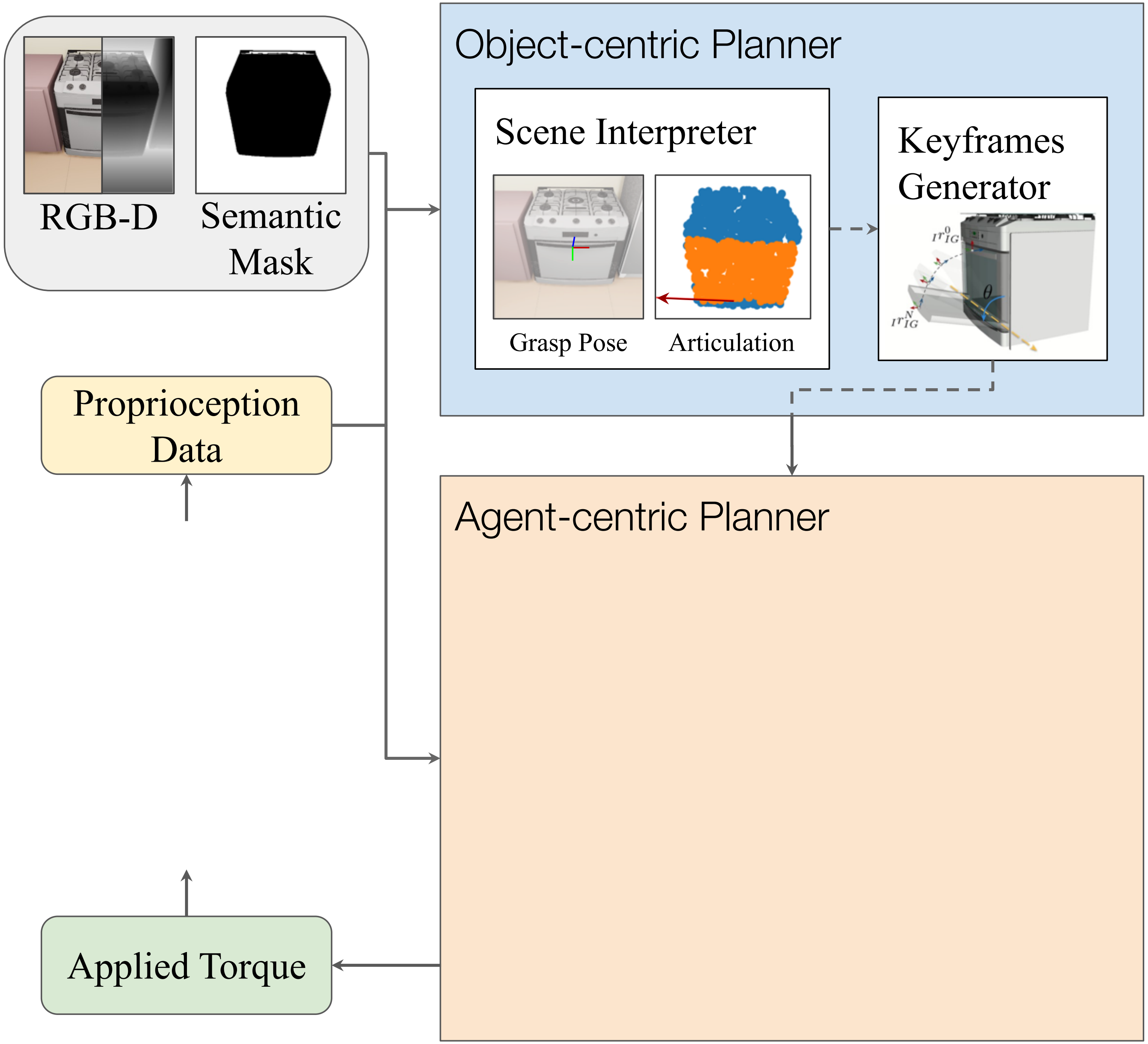
Input

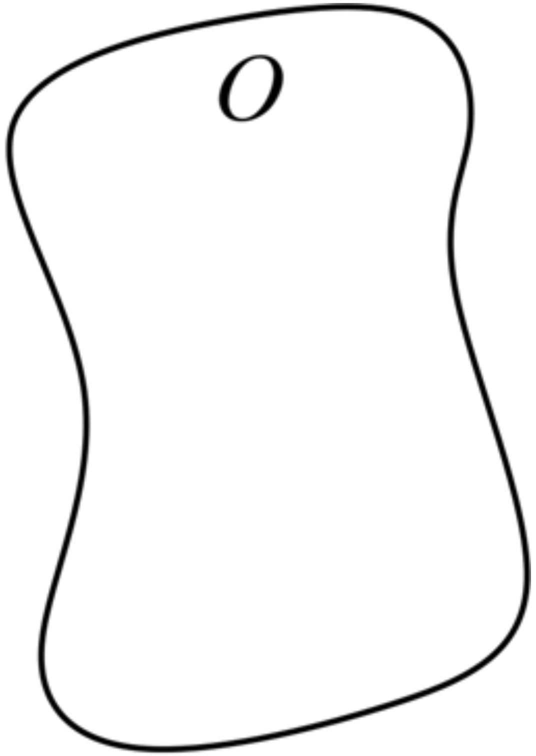
→ "Take" "Jug"
→ "Open" "Fridge"
→ "Put" "Jug" in "Fridge"

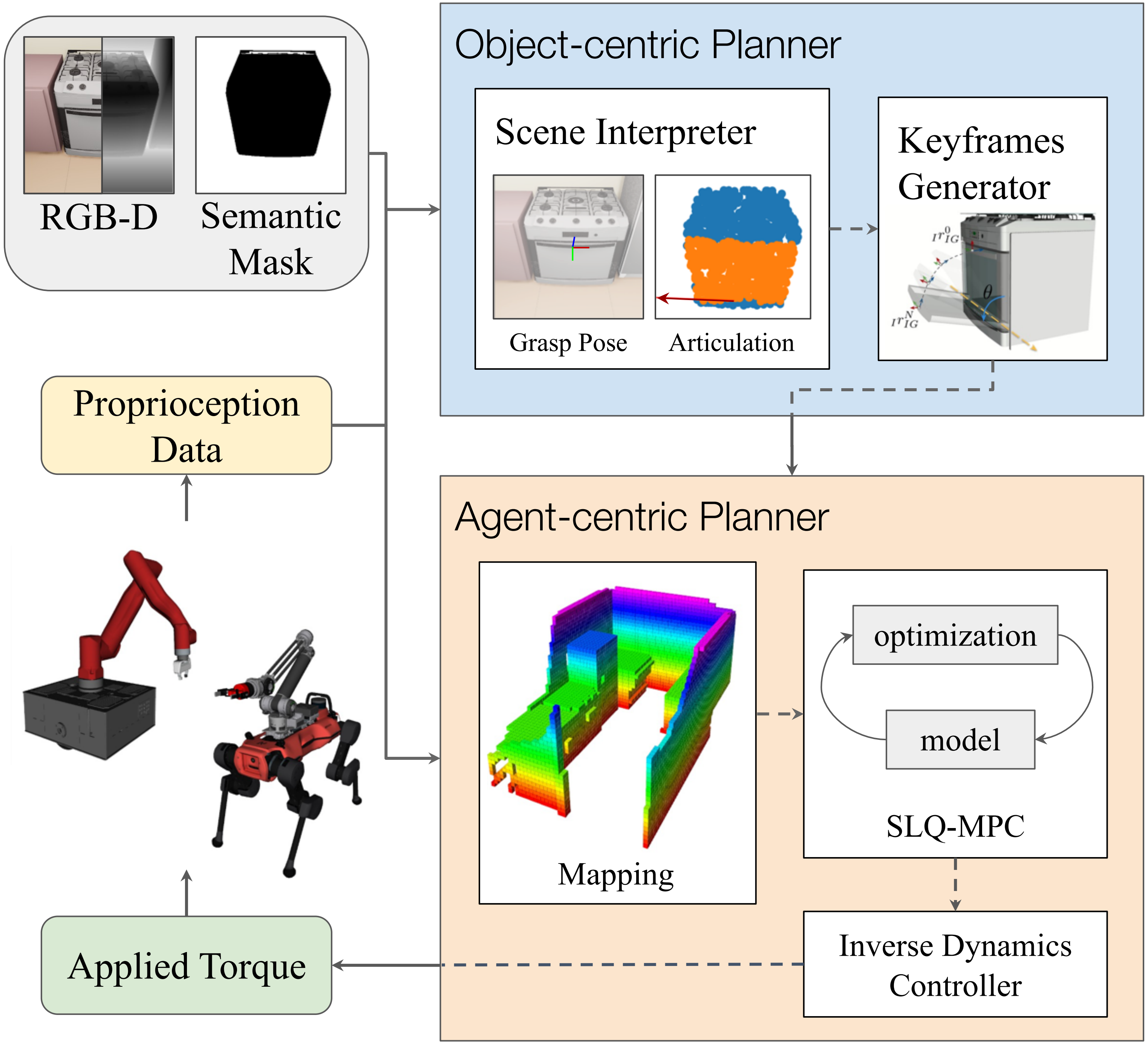
Goal Generation

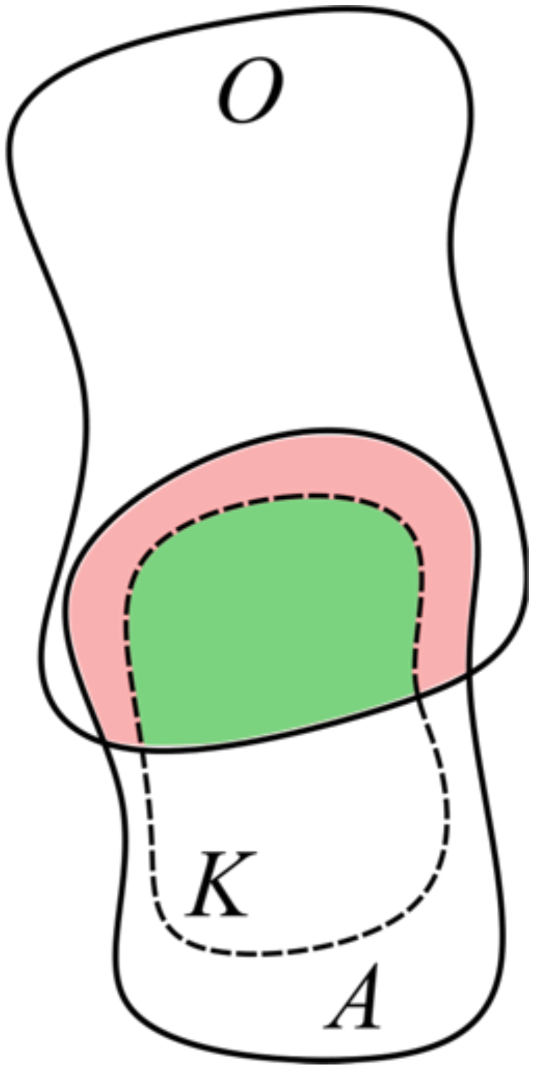
Goal-conditioned Reactive controller

Solvable online for different agents

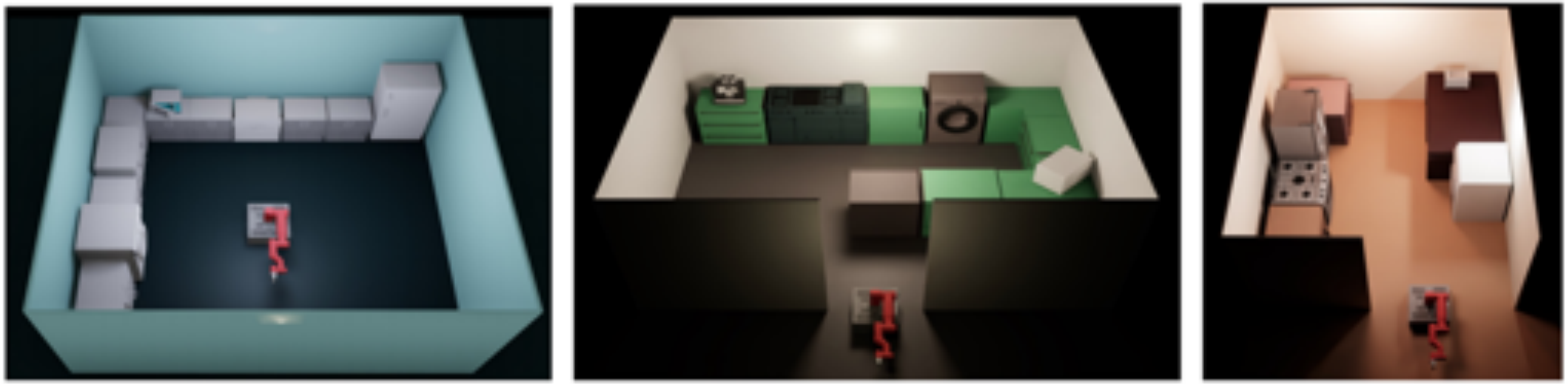

RGB-D
Semantic Mask
Object-centric Planner
Scene Interpreter
Grasp Pose
Articulation
Keyframes Generator
Proprioception Data
Agent-centric Planner
Applied Torque


O


RGB-D
Semantic Mask
Object-centric Planner
Scene Interpreter
Grasp Pose
Articulation
Keyframes Generator
Proprioception Data
Agent-centric Planner
Mapping
optimization
model
SLQ-MPC
Inverse Dynamics Controller
Applied Torque


O
K
A

Structure in Compositional Planning: Setup



Different kitchen layouts designed on NVIDIA Isaac Sim using PartNet-Mobility dataset

(a) Drawers (b) Ovens (c) Washing Machines

Static Scene: novel instances of known articulated object category

IROS 2022 (under review)


x1.5


Hardware: Legged-base
IROS 2022 (under review)

Paving the path to Robot Autonomy with Simulation

Vision: Simulation is Data Factory for Robotics