Vikash Kumar

DeepRL Course Guest Lecture

Applied

Deep Reinforcement Learning

Vikash Kumar

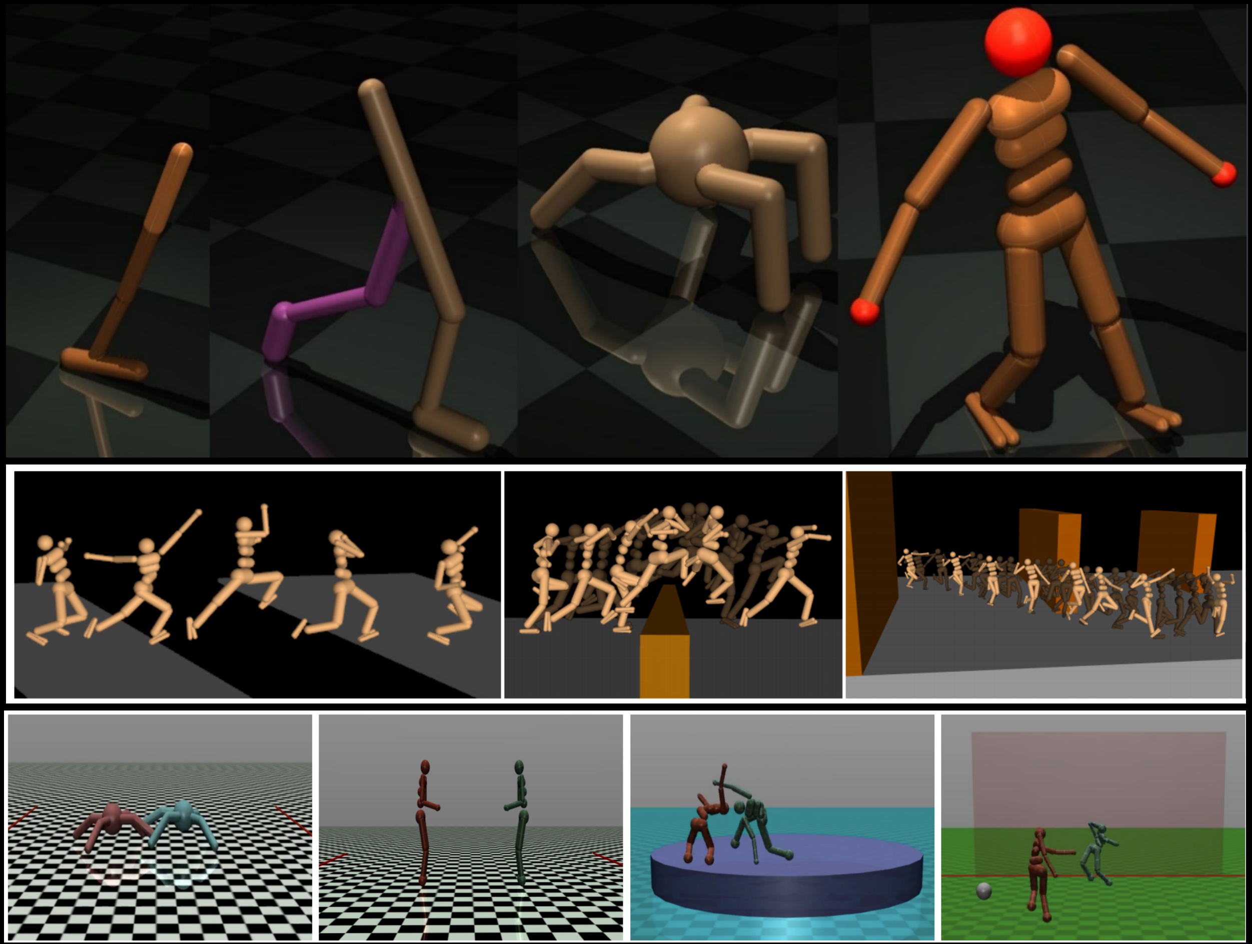
University of Washington Computer Science & Engineering

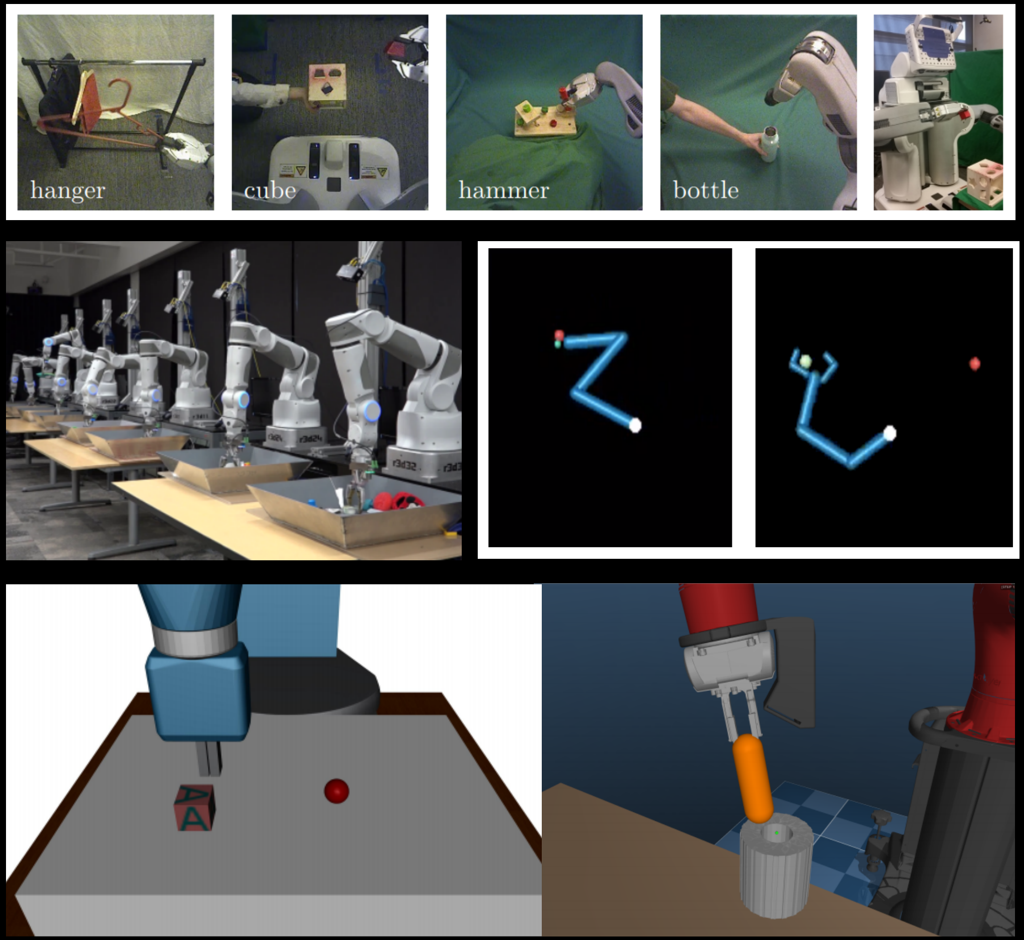
OpenAI

Berkeley University of California

Google Brain

Deep RL for continuous control





Question

1) What's new?
2) Gap/assumptions between simulations and reality?
3) Can the wall clock time required for skill acquisition on <u>physical hardware</u> be reduced to practical time scales?

Robot selection & acquisition → **Infrastructure & setup** → Software layer → **Algorithm & experiment design** → Skill acquisition →

Life long learning

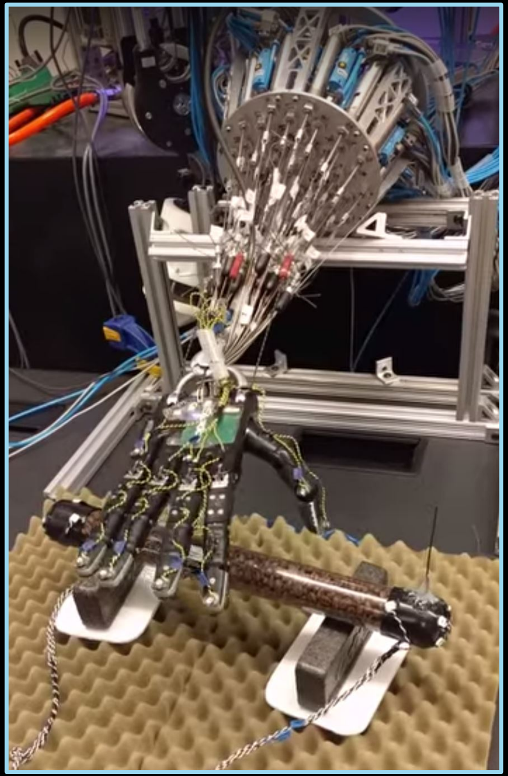
Question

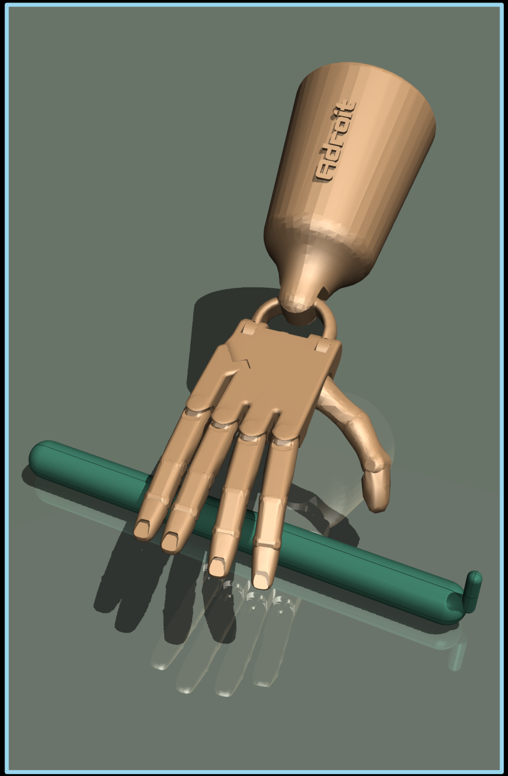
1) What's new?
2) Gap/assumptions between simulations and reality?
3) Can the wall clock time required for skill acquisition on physical hardware be reduced to practical time scales?

Robot selection & acquisition → **Infrastructure & setup** → Software layer → **Algorithm & experiment design** → Skill acquisition

Life long learning

Hardware: ADROIT MANIPULATION PLATFORM

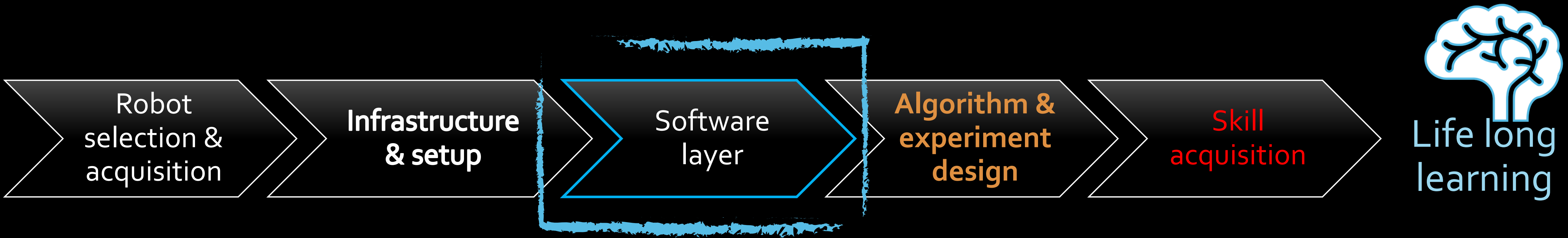




- 24 DoF hand
- Low friction & stiction
- Sensing
  - Joint angle (500hz)
  - Finger tip touch (500hz)
  - Tendon length (9khz)
  - Tendon tension (9khz)

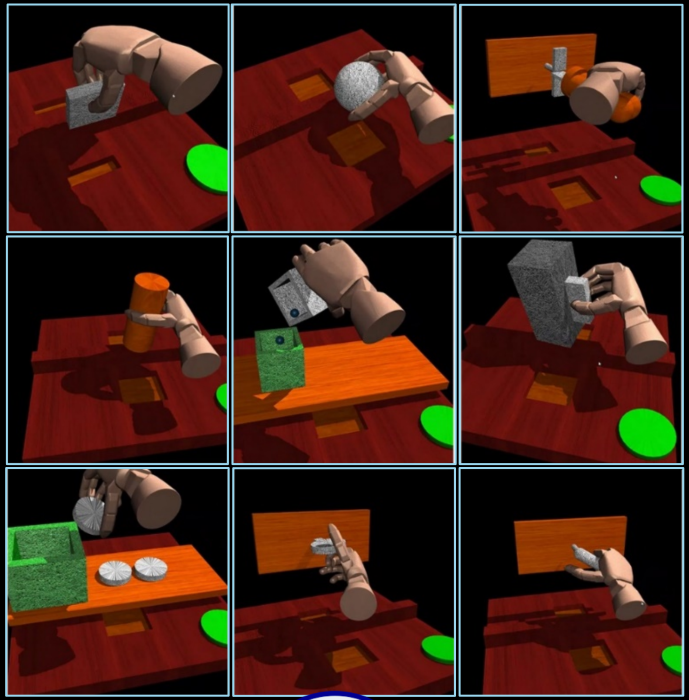
Software

- Fast and efficient simulators (mujoco + mujoco_py)
- Standard Algorithmic APIs (Baselines, RLlab, NPG)
- Fast and easy switch between software and hardware
- Physically realistic demonstrations (mujoco-vr)

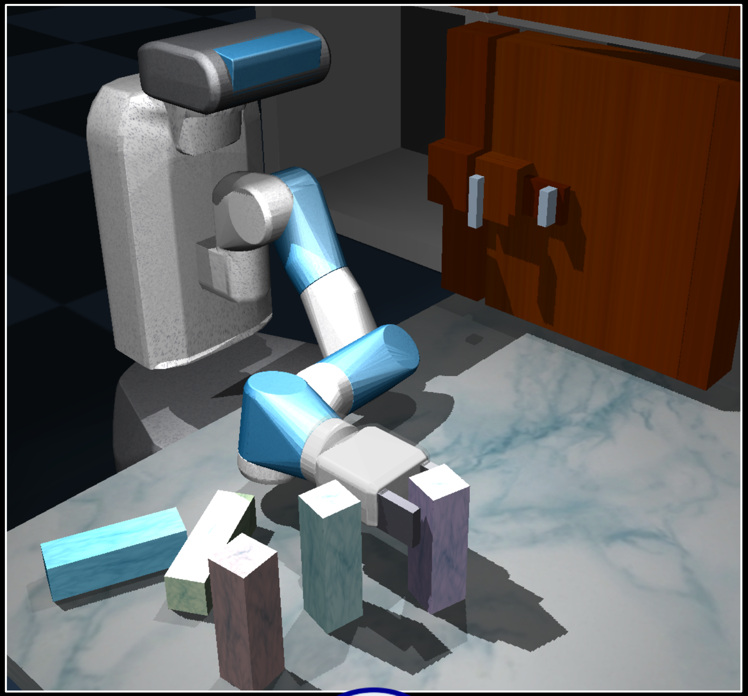


Software: Physically Realistic Demonstrations

MuJoCo HAPTIX

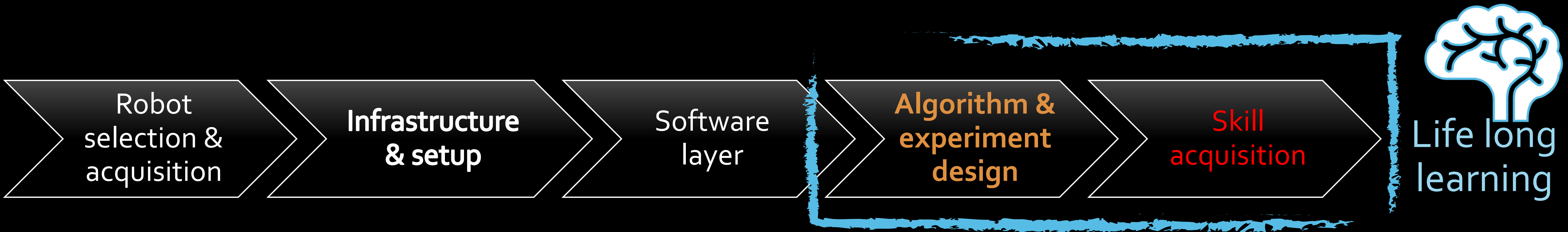


MuJoCo-VR



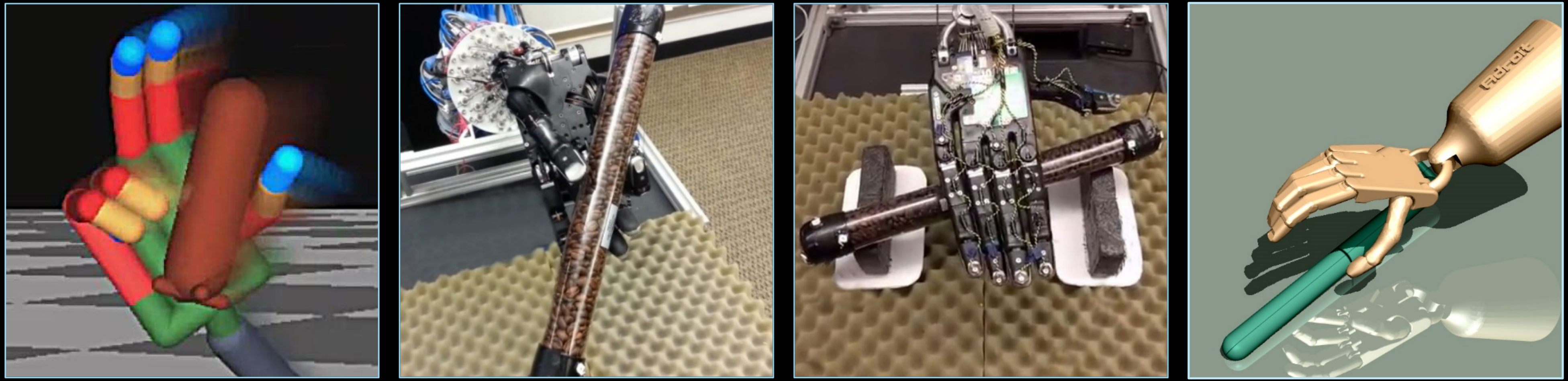
Question

1) What's new?
2) Gap/assumptions separating simulations & reality?
3) Can the wall clock time required for skill acquisition on physical hardware be reduced to practical time scales?

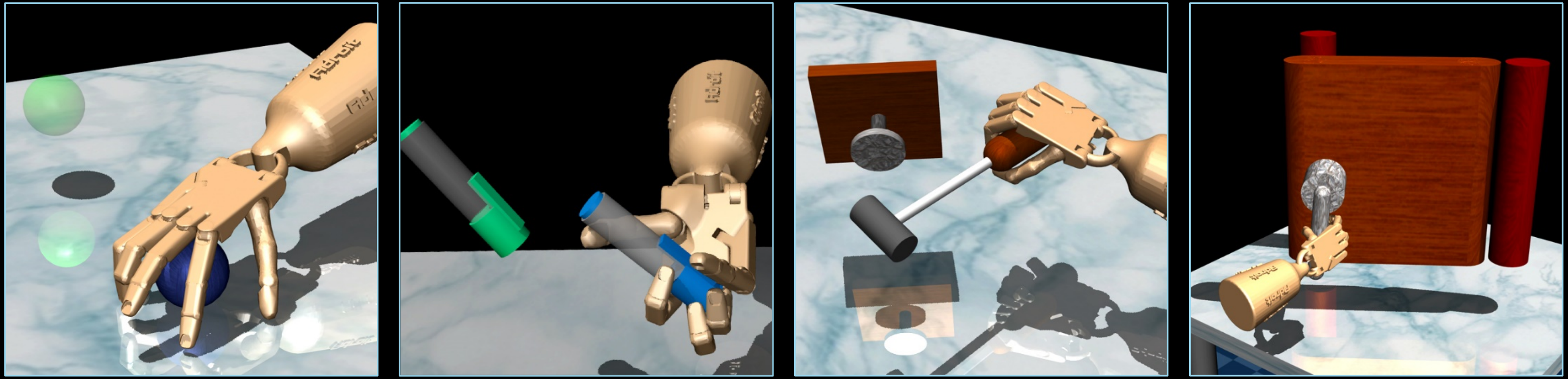


Algorithmic Paradigms

Model Based

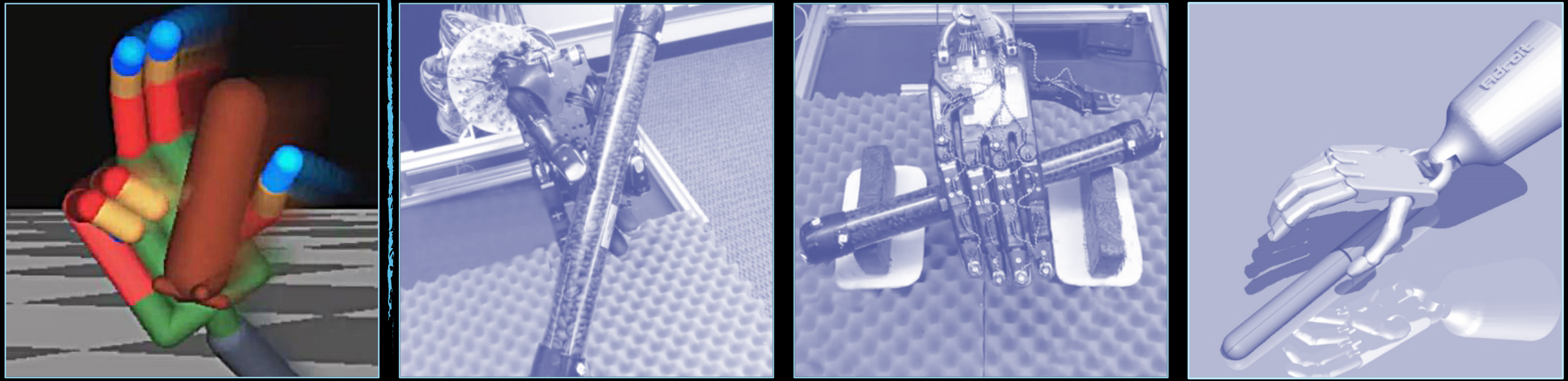


Model Free



Algorithmic Paradigms: Known Global Model

Model
Based



Model
Free

Approaches

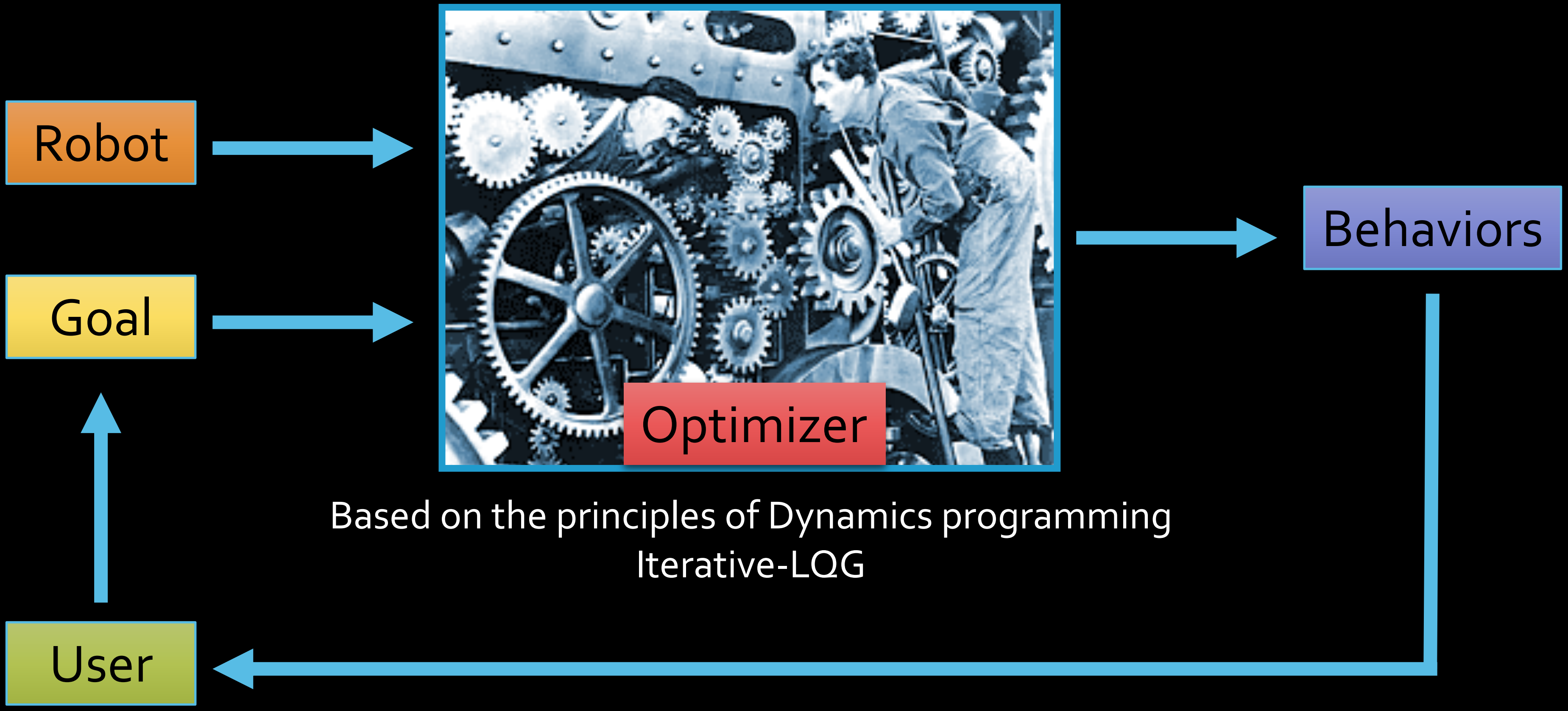
- Traditional approach: Plans movements
  - Manual scripting
  - Inverse kinematics

- Modern approaches: Plans behaviors
  - Optimal control
    - Developed to control slow evolving chemical plants
    - Gradient is my signal
    - High level goal directed reasoning
  - Reinforcement Learning
    - Noise is my signal
    - Sparse goals
    - Computational budgets

Optimal Control: Trajectory Optimization

Global Robot Dynamics Model

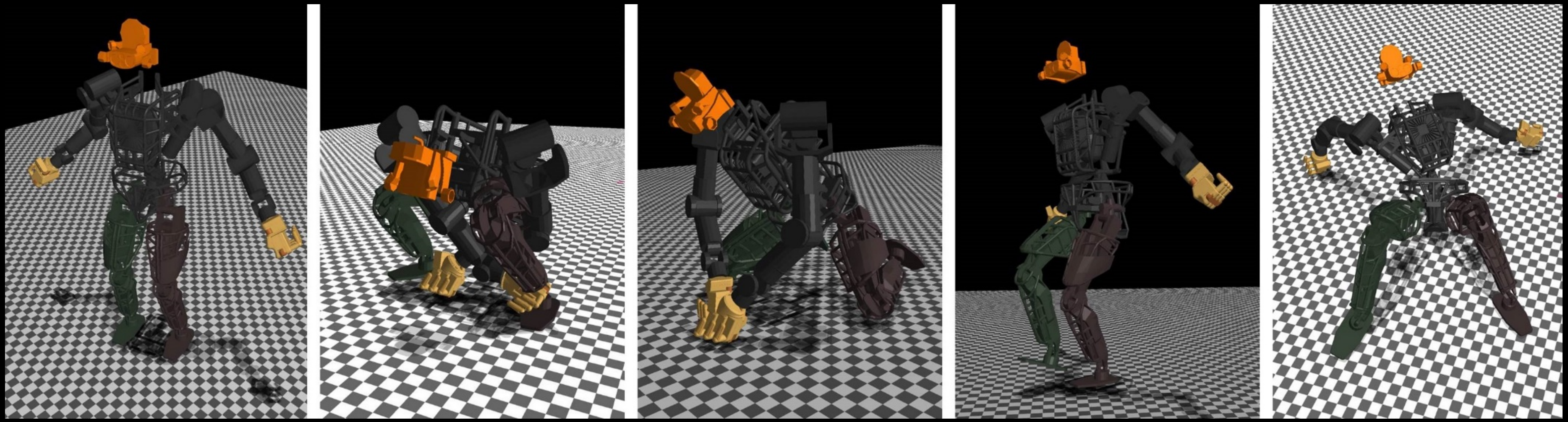
Simple terms with Intuitive meaning



Based on the principles of Dynamics programming
Iterative-LQG

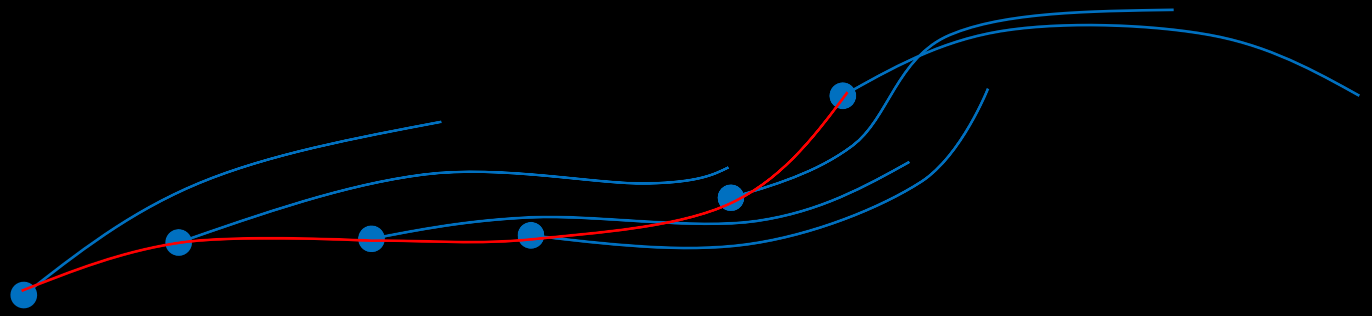
DARPA Robotics challenge

- Fused behaviors
  - Dynamic full body stabilization
  - Head/ hand reach target
  - Head/ hand look



Team :: Tom Erez, Kendall Lowrey, Yuval Tassa, Vikash Kumar, Svet Kolev, Emo Todorov

Optimal Control: Model Predictive Control

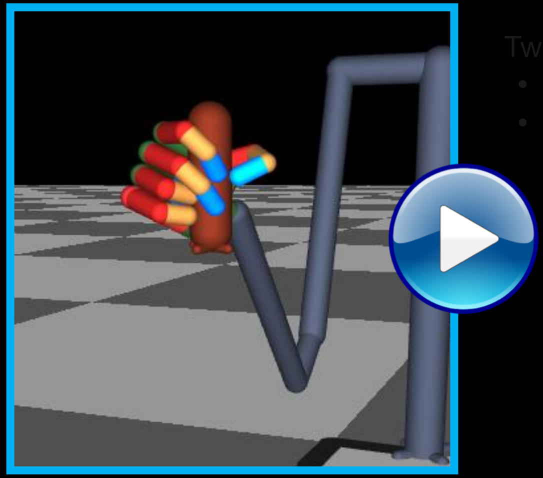


1. At time step *t*, solve a trajectory optimization problem for the desired behavior
2. Execute initial part of the solution
3. Re-evaluate and update your plan

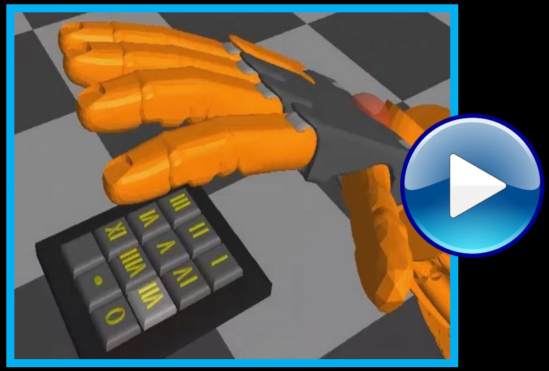
There is always a plan, the plan is re-optimized all the time,
only the initial portion is ever executed.

Optimal Control: Model Predictive Control

- Reusable machinery
- Real time behavior generation
- Behaviors encoded as simple cost terms
  - ➢ Manipulation
    - Distance from goal configuration
    - Regularization (controls and velocities)
    - Distance between hand and object (<10^3)
  - ➢ Typing
    - Desired key press
    - Distance between key-finger tip
    - Autocorrect
    - Regularization (control and velocities)



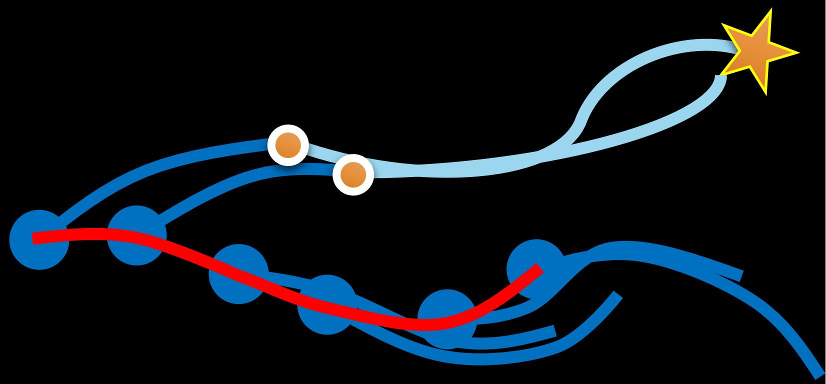
Two subtle sequences
- Partial grip
- Back fumble



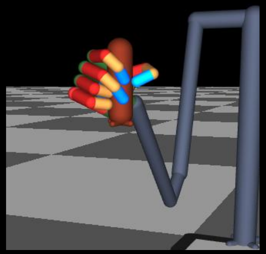
Yuval Tassa, DeepMind

Why MPC works ?

1. Fast premature updates (better than slow converged updates)
    - Optimum is never achieved. Why solve for it?
    - Drags the solution closer to the minimum with each update



2. Partial plans
    - Partial policy for a shorter horizon



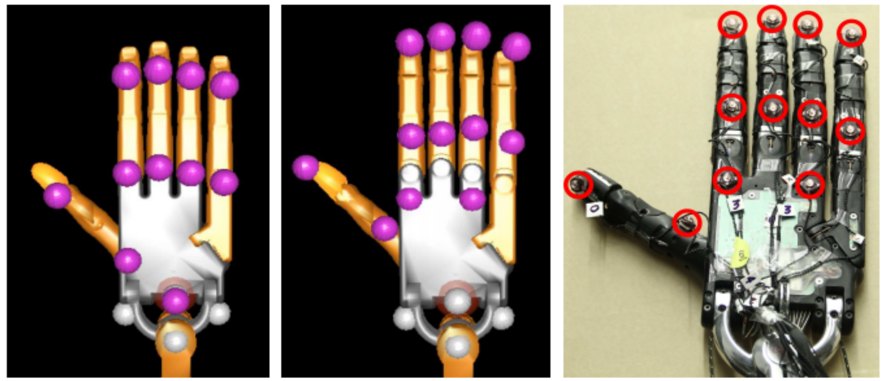
Challenges with MPC on hardware

1. Sensing
    - Space constraints
    - Low quality sensors
    - Mocap - occlusion and confusion
    - Partial observability

2. Calibration and Estimation
    - Manual – calibration jigs ineffective
    - Optimization – misguided

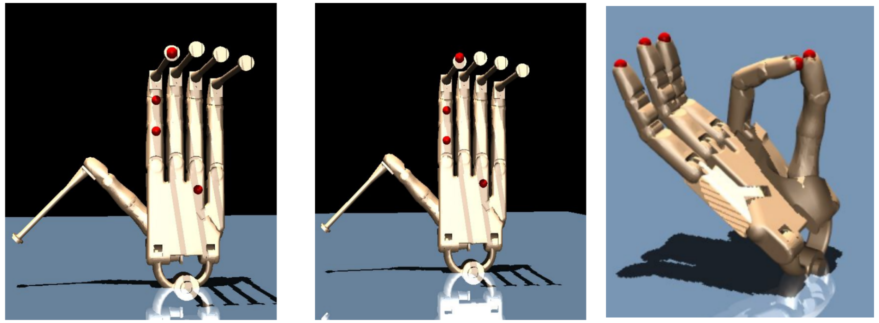
3. Modelling and identification
    - Never be able to replicate reality

STAC: Simultaneous tracking & calibration

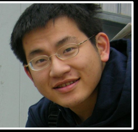


(a) initial pos (b) calibrated (c) ground truth

Kinematic Extensions



(a) Before Identifications (b) Identified site locations



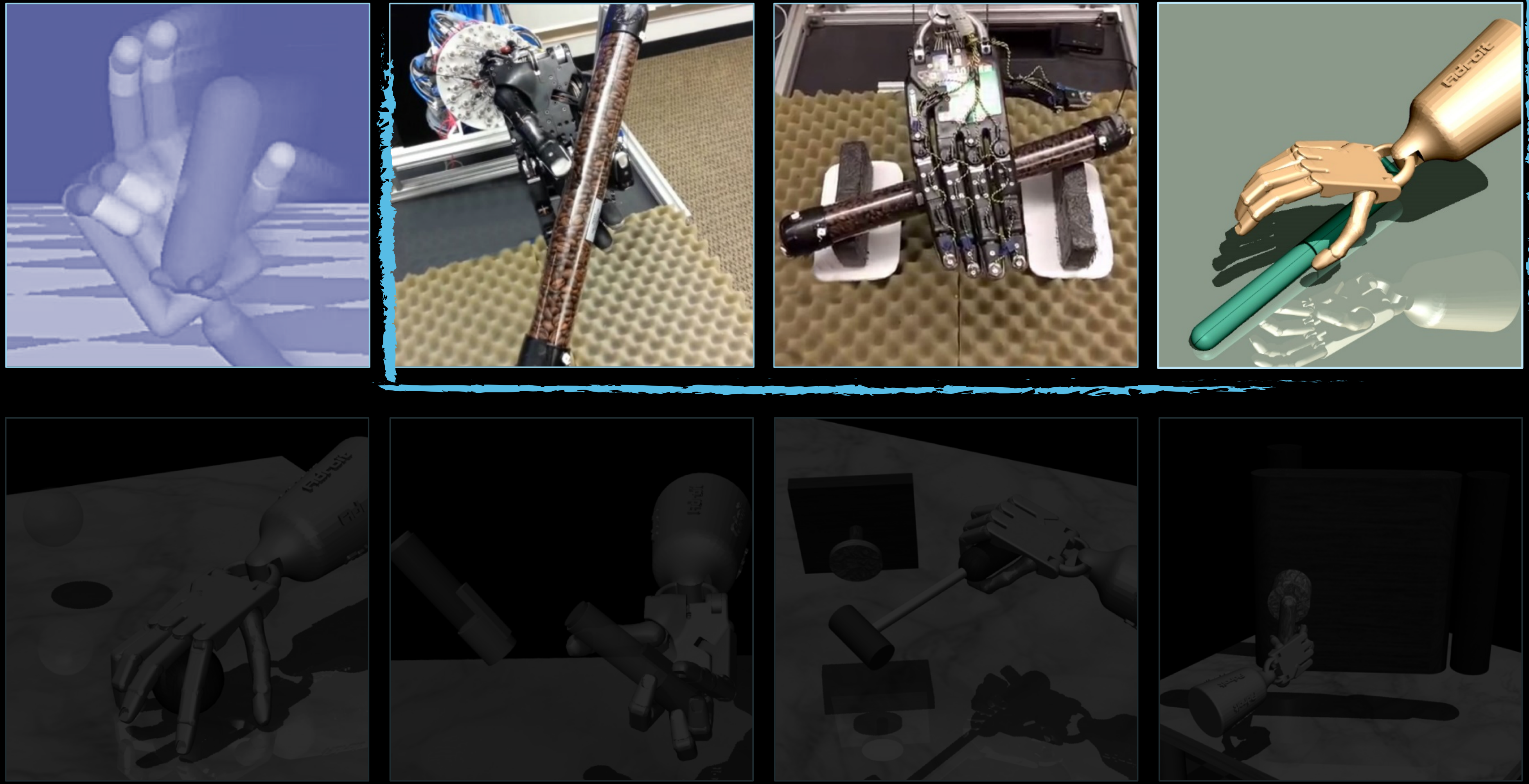
TingFan Wu,
UCSD/ iHMC

Visak, C.
UW/ GTech.

Algorithmic Paradigms: Learned Partial Model

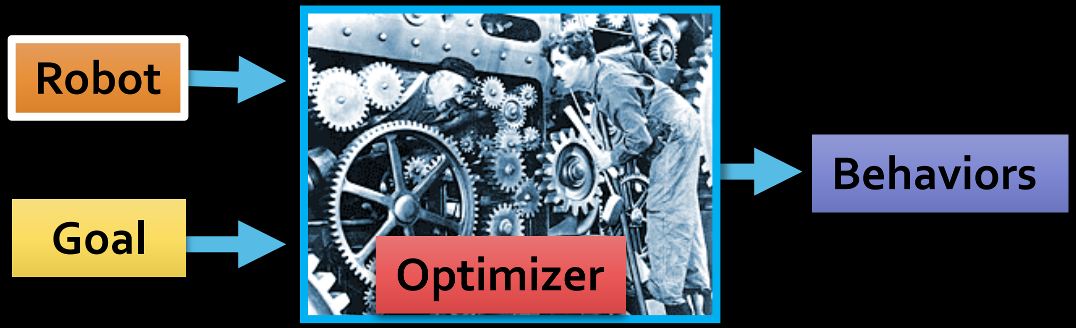
Model Based

Model Free



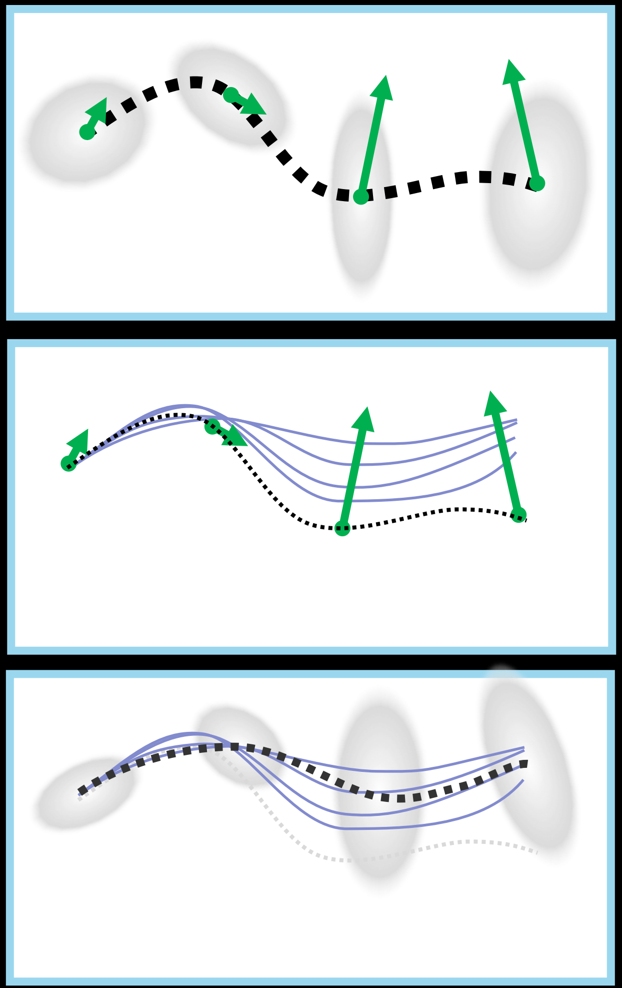
Learning Partial Models From Experience

~~Global Physics Model~~



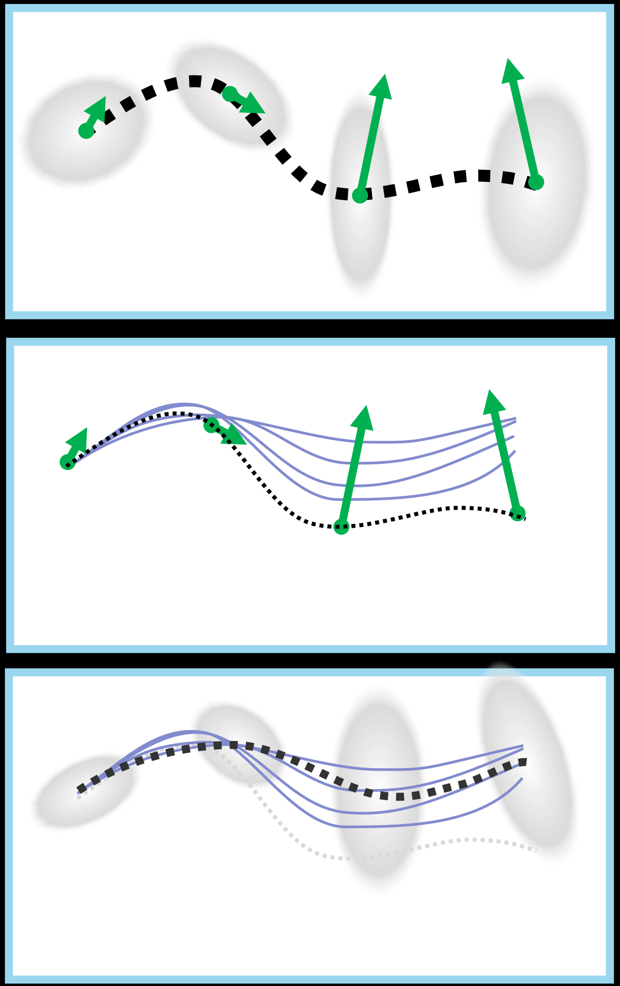
Partial model

- Time-varying linear model
- Parameterized directly by sensor data
- Adapt as we go



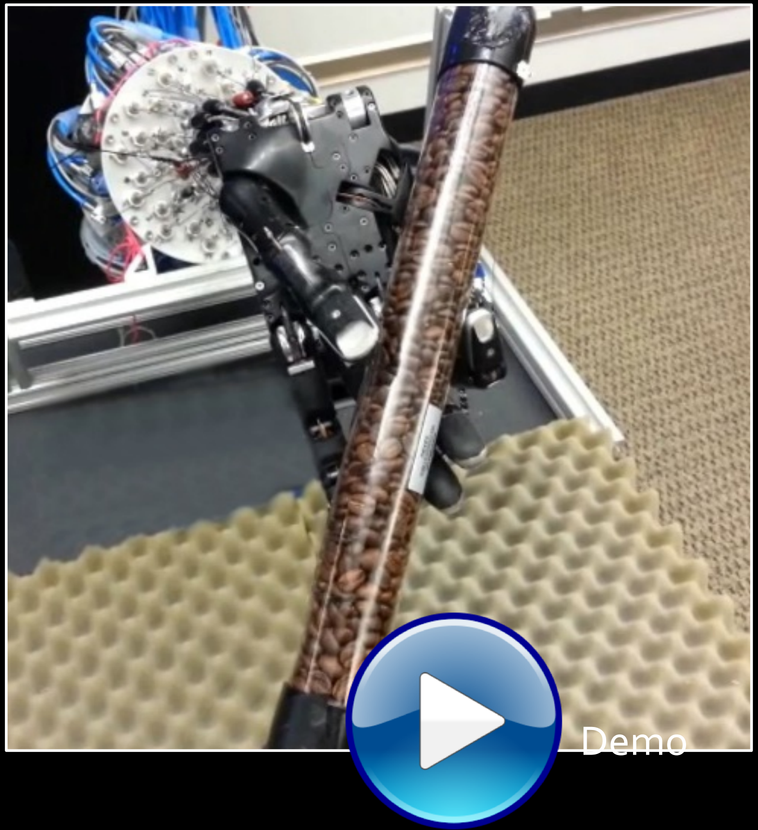
Optimal control with learned local models

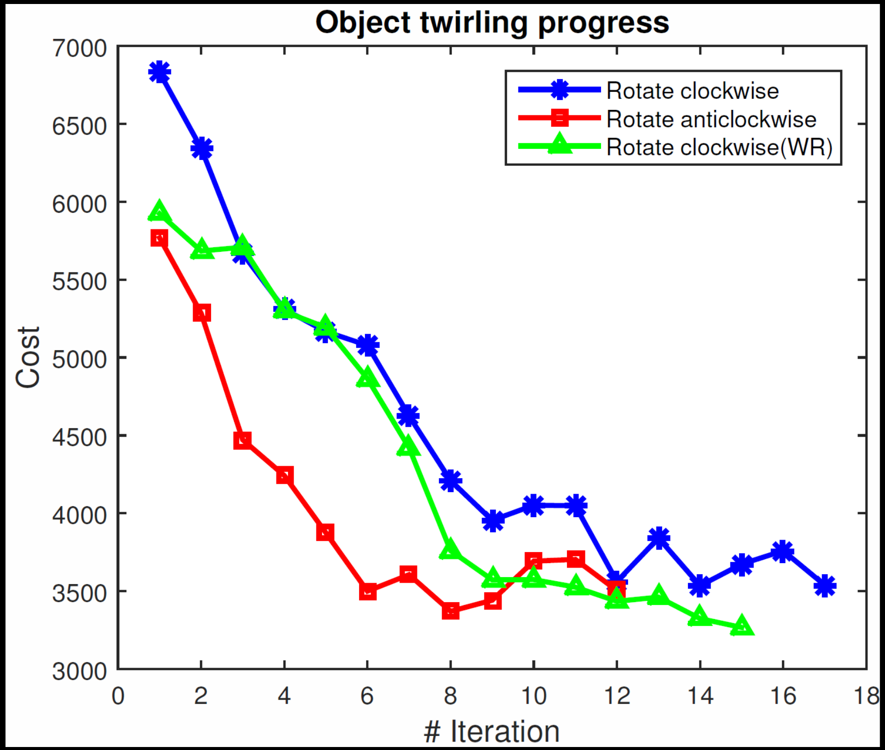
initialize $p(\mathbf{u}_t|\mathbf{x}_t)$
for iteration $k = 1$ to K **do**
  run $p(\mathbf{u}_t|\mathbf{x}_t)$ to collect trajectory samples $\{\tau_i\}$
  fit dynamics $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$ to $\{\tau_j\}$ using linear regression with GMM prior
  fit $p = \arg\min_p E_{p(\tau)}[\ell(\tau)]$ s.t. $D_{\mathrm{KL}}(p(\tau)\|\hat{p}(\tau)) \le \epsilon$
end for



Best Manipulation Paper Award, ICRA 2016

Optimal control with learned local models





- sample efficient
- effective with intermittent contacts
- effective on physical hardware

Best Manipulation Paper Award, ICRA 2016

Learning from Experience and Imitation

$$\ell(\mathbf{x}_t, \mathbf{u}_t) = \alpha_1 ||q_t - q^*||^2 + \alpha_2 ||\mathbf{u}_t||^2 + \alpha_3 ||q_t^{pos} - q^{pos*}||^2 + \alpha_4 ||q_t^{rot} - q^{rot*x}||^2$$

Observed Behaviors

- Random exploration with no progress
- Object flies away form manipulatable workspace
- Interaction with no progress (local minima)

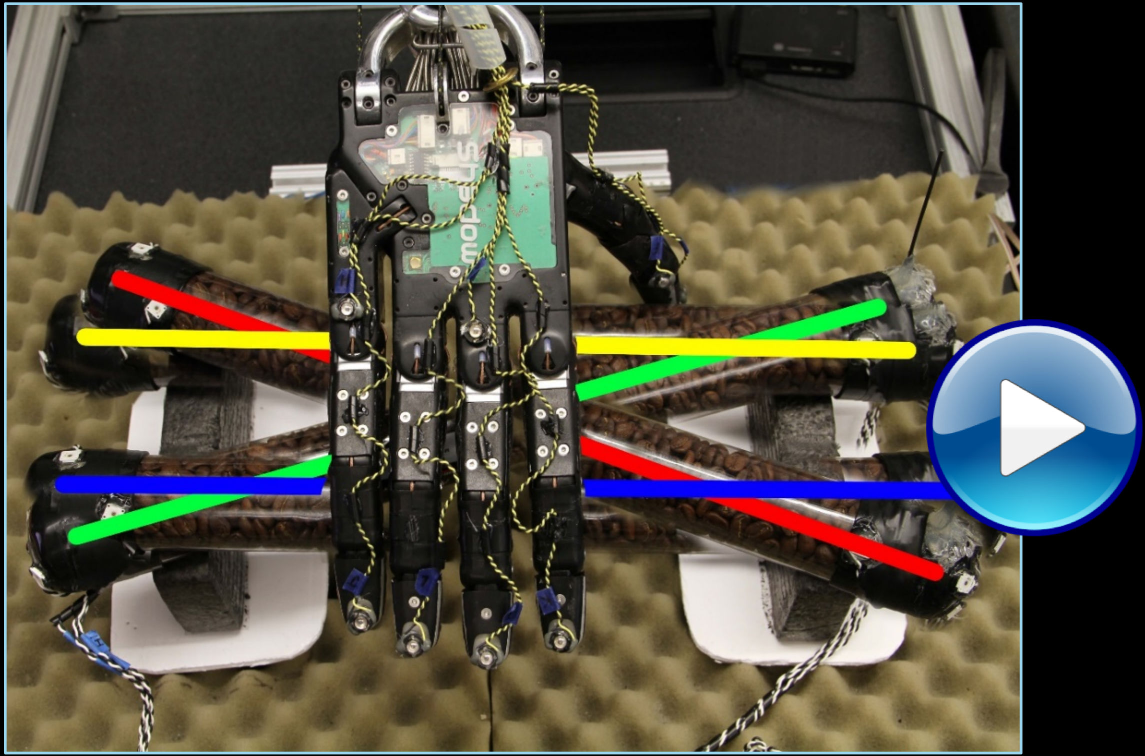
- Reward delayed in the future
- Ineffective exploration

Poor sample complexity

Reducing Sample Complexity with Demos

Use demonstrations collected in VR to guide exploration in task-relevant part of the state space

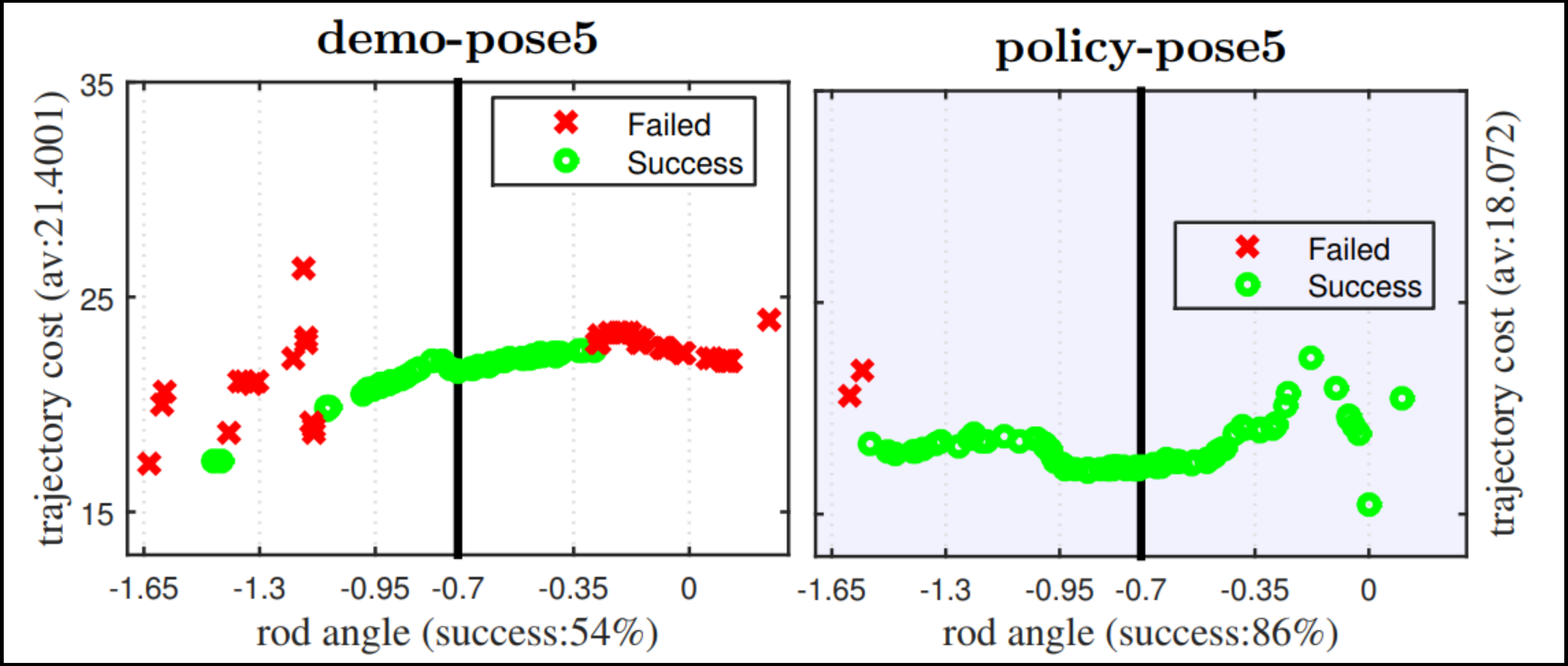
Expert demonstration
(collected with action noise)



$$\ell(\mathbf{x}_t, \mathbf{u}_t) = ||\mathbf{q}_t - \hat{\mathbf{q}}_t||^2 + 0.1||\mathbf{u}_t||^2 + 50||q_t^{posZ} - 0.12||^2$$

Imitation + synthesis

Surpassing experts

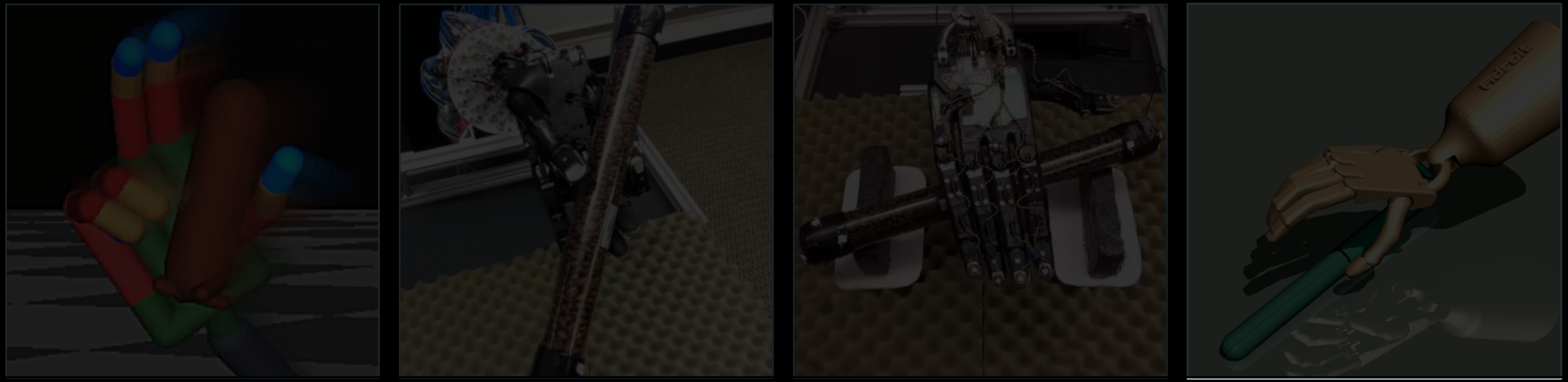


Model Based Algorithmic Paradigm

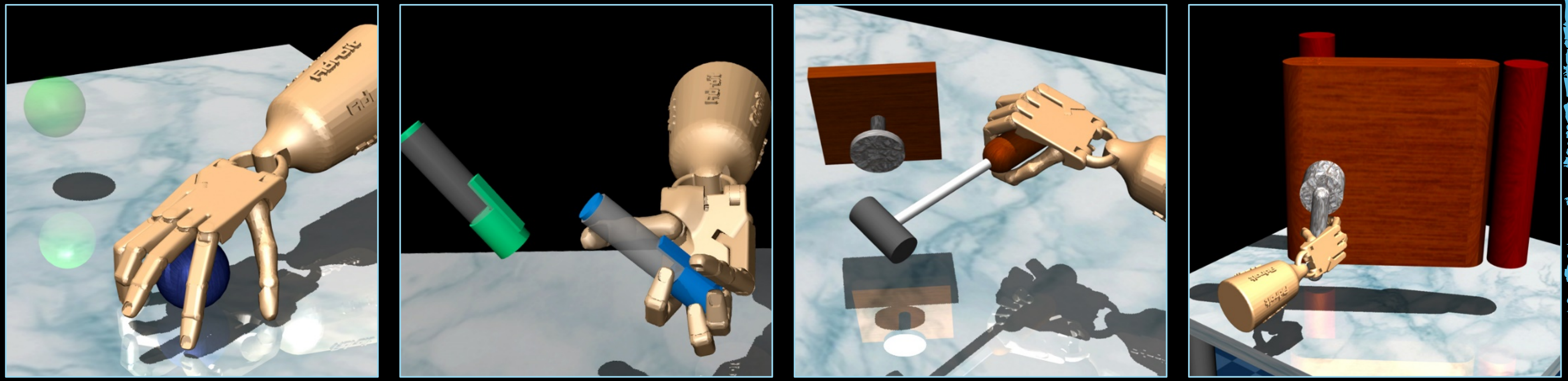
- Sample efficient
- Effective with intermittent contact rich behaviors
- Resulting policies are local
- Effective only if test distribution and training distribution are close
- Reward needs to be differentiable
- Ineffective with sparse reward

Algorithmic Paradigms

Model
Based

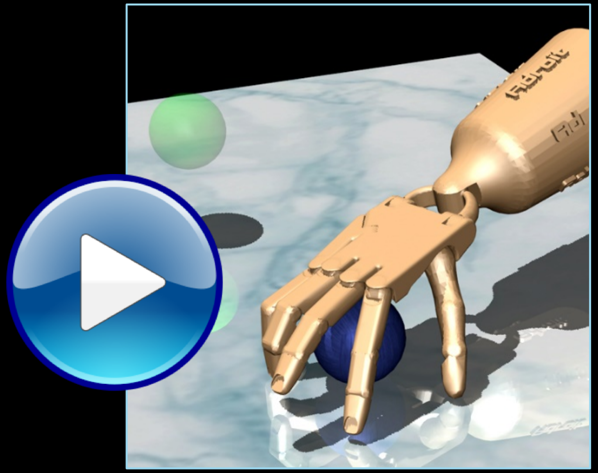


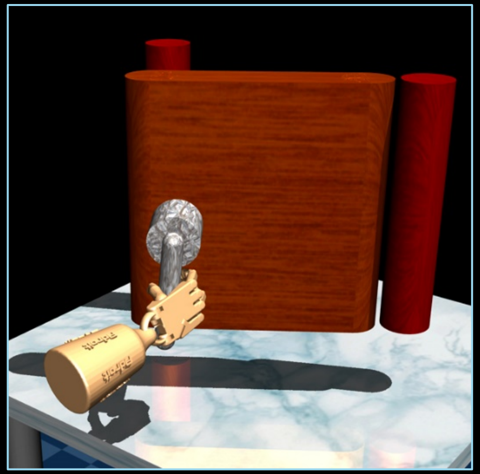
Model
Free



Behavior Cloning (# demonstrations: 25)

$$\text{maximize}_\theta \sum_{(s,a^*)\in\rho_D} \ln \pi_\theta(a^*|s)$$





- Large # of demos needed for high DoF
- State distribution mismatch
- Leads to compounding error

Behavior Cloning with RL (# demonstrations: 25)

- Behavior cloning doesn't work well by itself

  Solving for the wrong objective
  - tries to produce optimal action under demo state distribution instead of induced state distribution
  - Compounding errors
  - Cascading failures

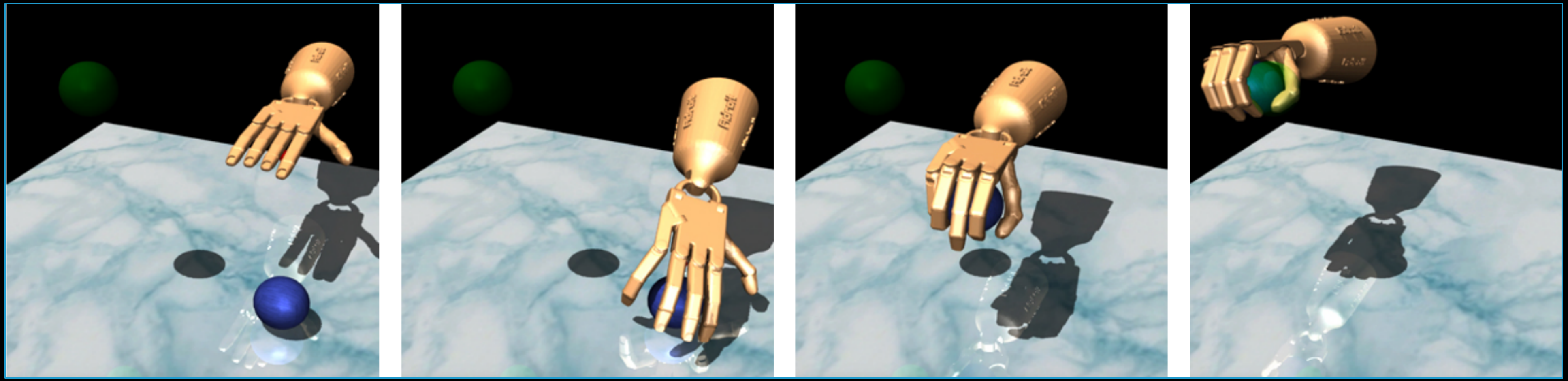
- Good initialization for RL fine-tuning

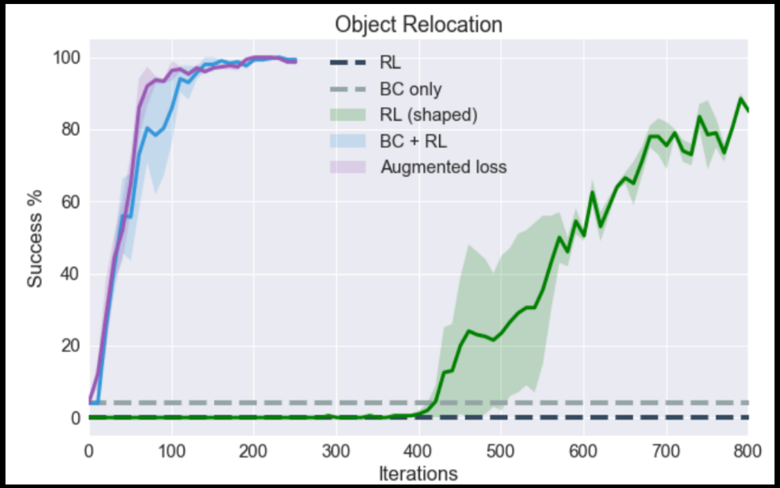
Auxiliary Objective

- Demo contains so much more information than the BC initialized network
- Different parts of the demonstration data is useful in different learning stages

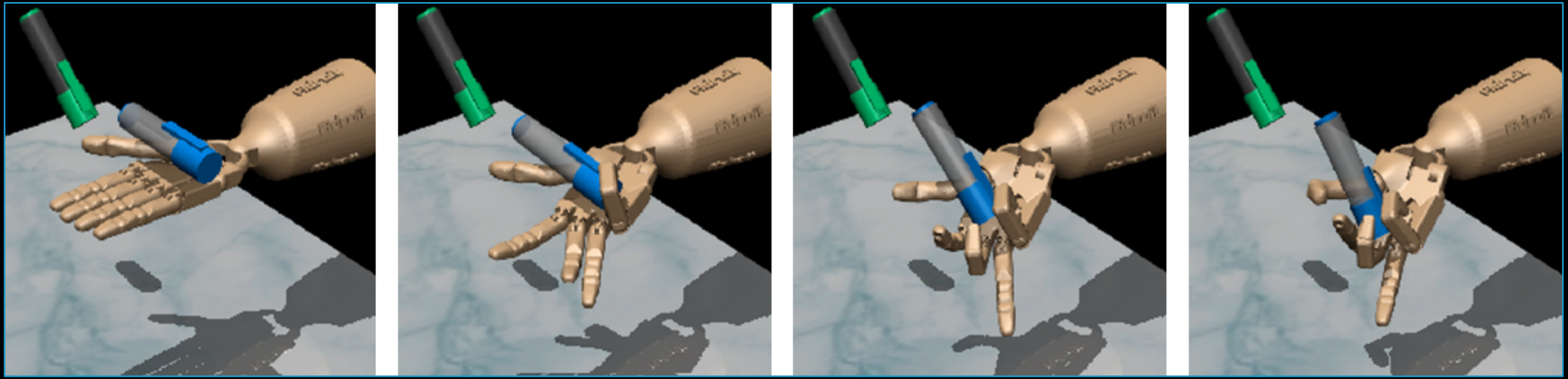
$$g_{aug} = \sum_{(s,a)\in\rho_\pi} \nabla_\theta \ln \pi_\theta(a|s) A^\pi(s,a) + \sum_{(s,a^*)\in\rho_D} \nabla_\theta \ln \pi_\theta(a^*|s) w(s,a^*)$$

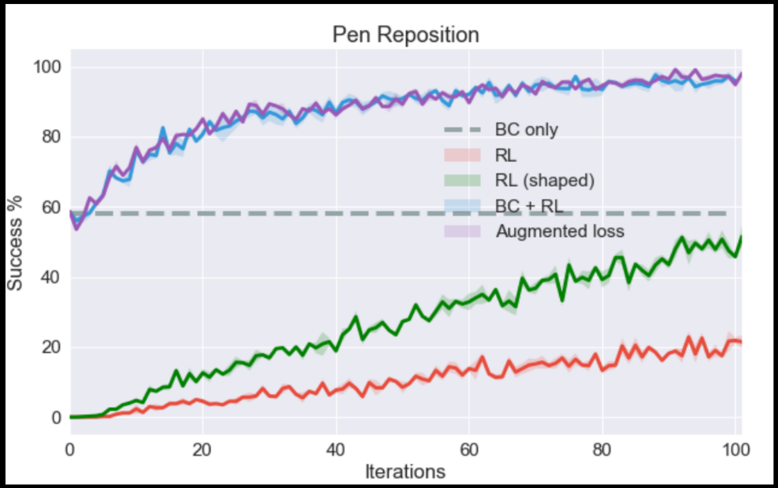
Tasks: Relocation



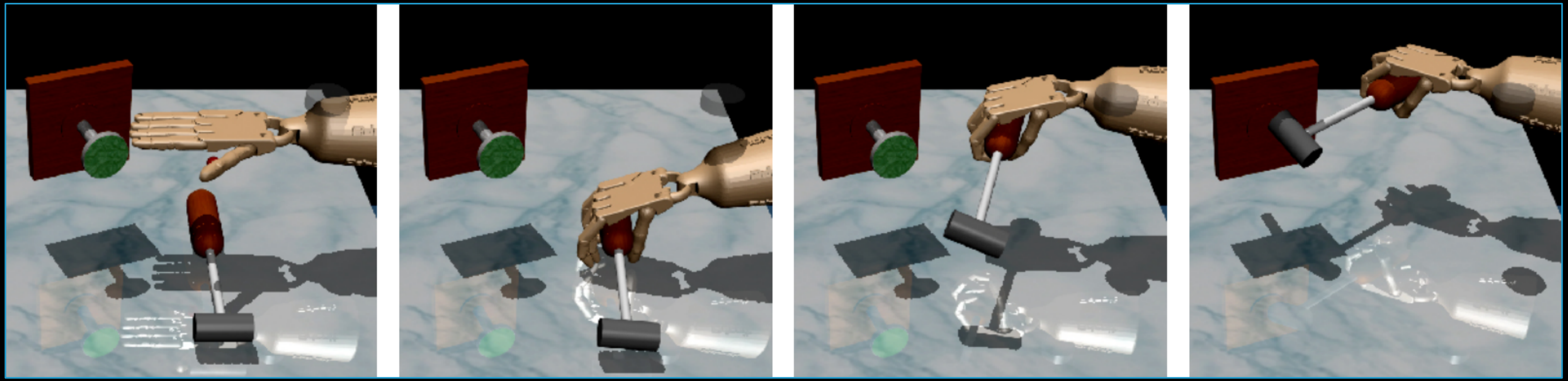


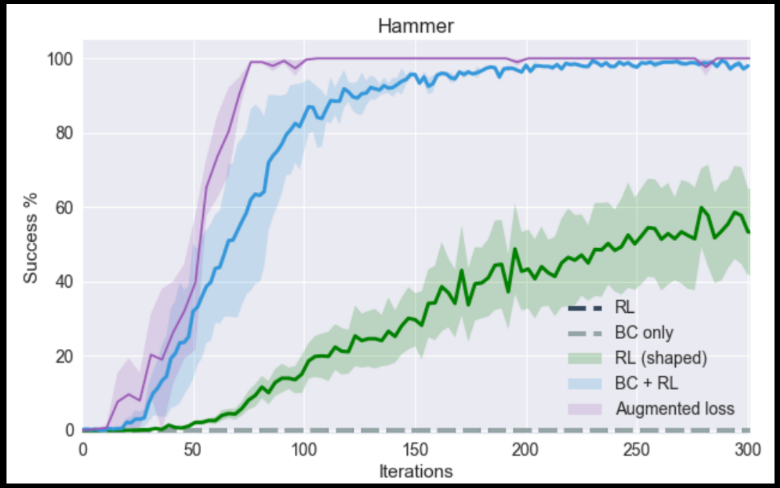
Tasks: in-hand manipulation



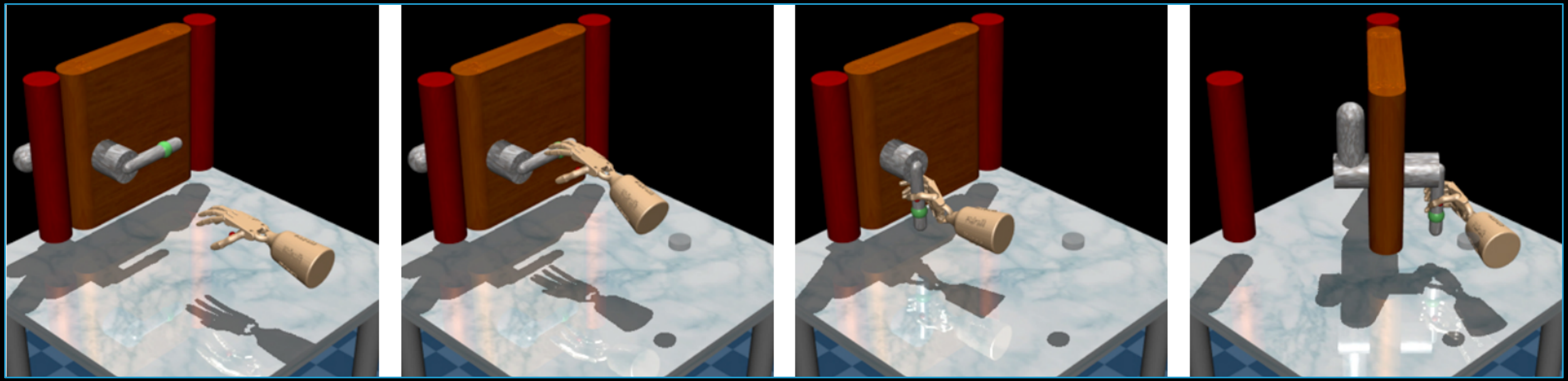


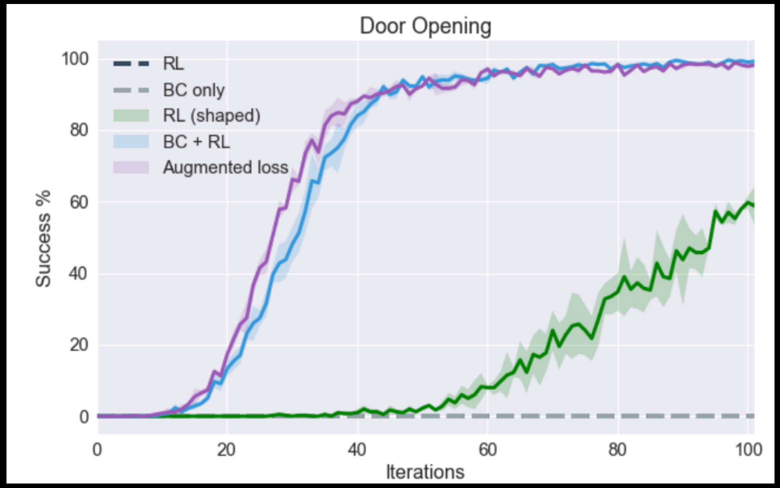
Tasks: Tool usage





Tasks: Environment interaction

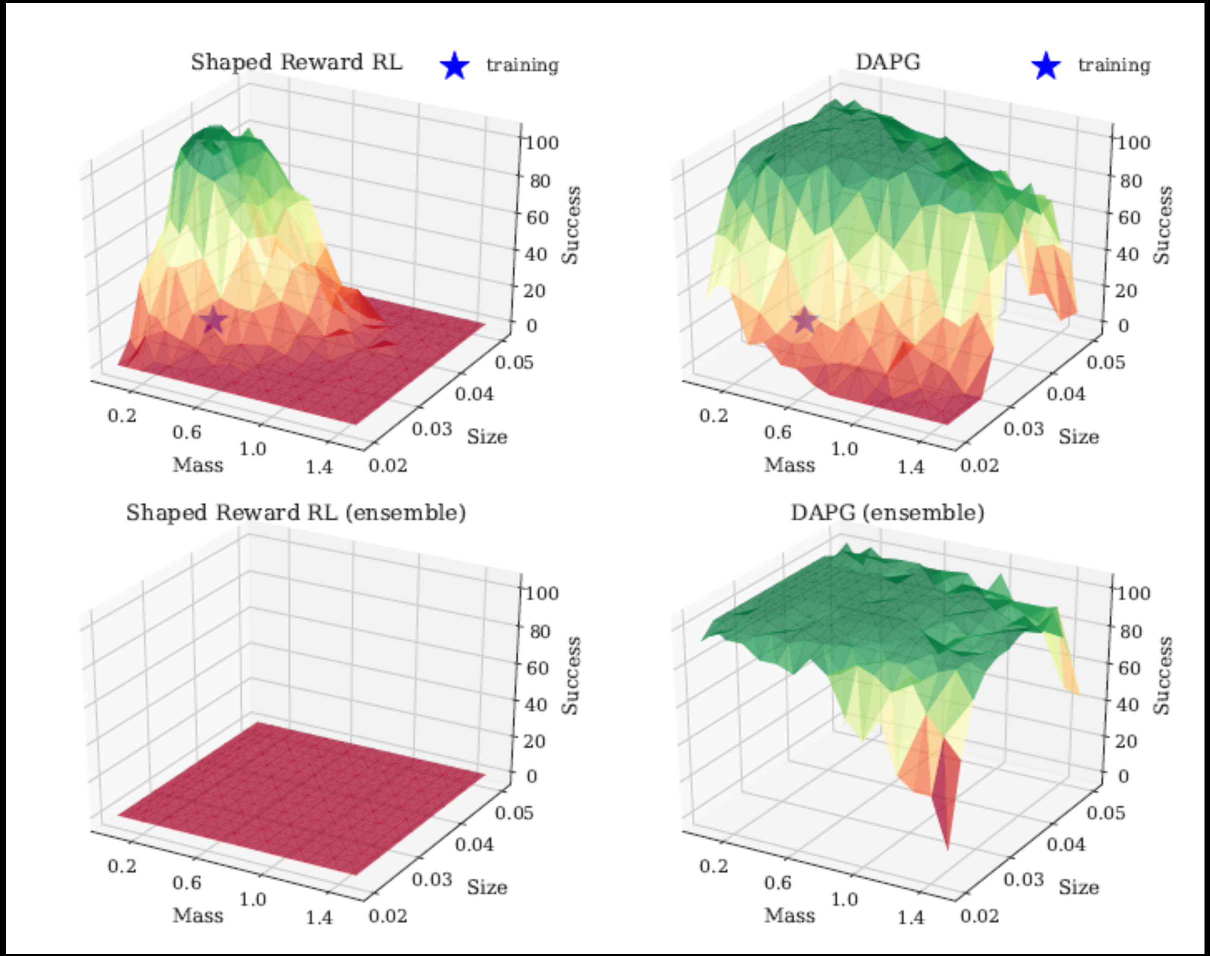




Model-free paradigm: results

Method	Ours		RL (sh)		RL(sp)	
Task	N	Hours	N	Hours	N	Hours
Obj Relocation	52	5.77	880	98	∞	∞
Hammer	55	6.1	448	50	∞	∞
Door	42	4.67	146	16.2	∞	∞
Pen	30	3.33	864	96	2900	322

Model-free paradigm: results



Future Directions

Movements

Skill

Bag of Skills

- Adaptation
- Transfer
- Sequencing
- Composition

End to End

- Vision
- Haptic
- Multi-agent
- Human-in-loop
- NLP

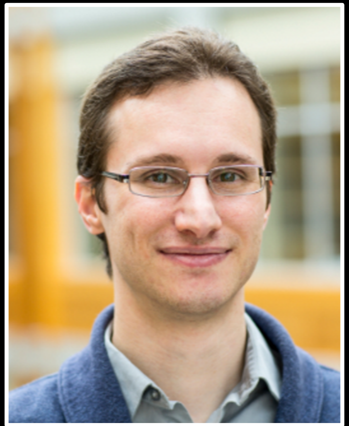
Key Insights

Caching & Recall

Synthesis via
Optimal Control

Learning via
Experience & Imitation

Collaborators & Institutions





UNIVERSITY OF WASHINGTON

OpenAI

Berkeley
UNIVERSITY OF CALIFORNIA

Google Brain

Thank you (homes.cs.washington.edu/~vikash/)