Improving Model Predictive Control in Model-based Reinforcement Learning

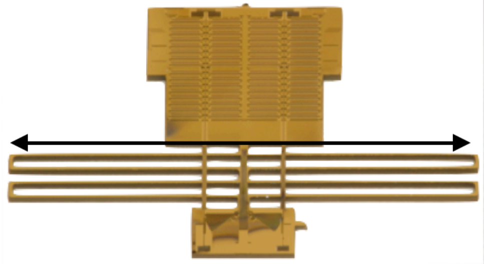
Nathan Lambert (Cornell ECE ’17, go Big Red)

Advised by:
Kristofer S.J. Pister, UC Berkeley EECS
Roberto Calandra, Facebook AI Research

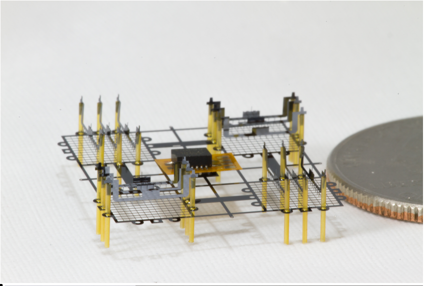
Novel robotic platforms

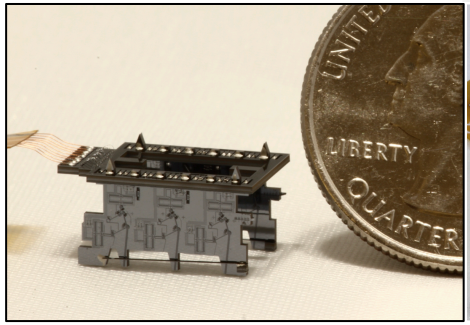
Microrobots

Jumper [1]

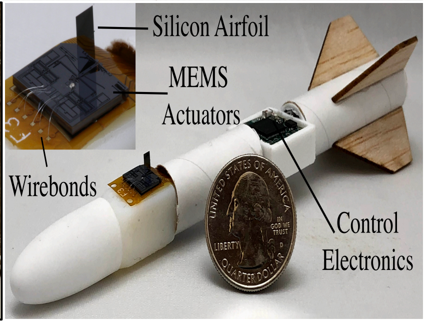


Ionocraft [2]



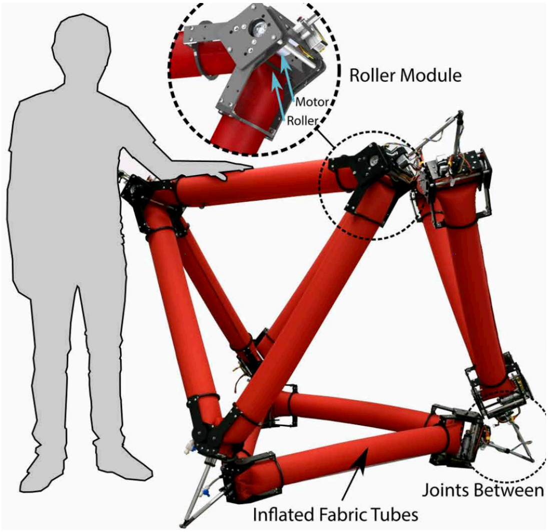


Hexapod [3]



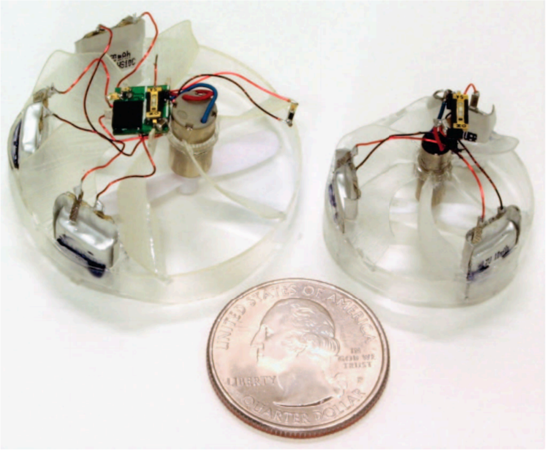
Rocket [4]

Isoperimetric soft robot [5]

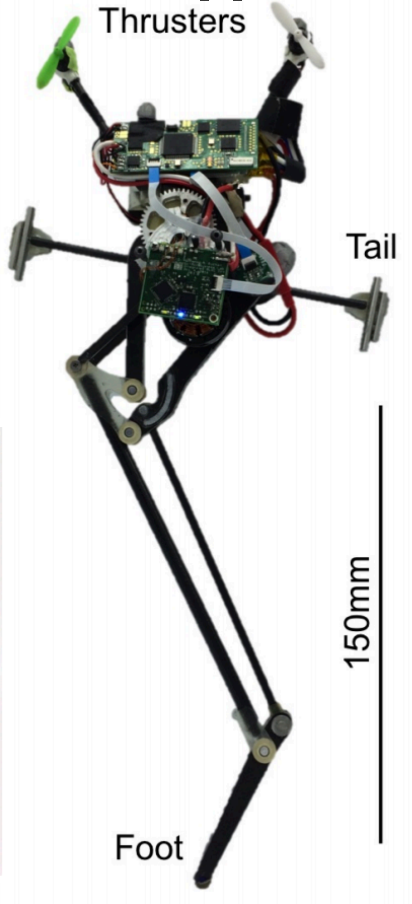


Found Objects [8]

Picolissimo [7]



SALTO [6]

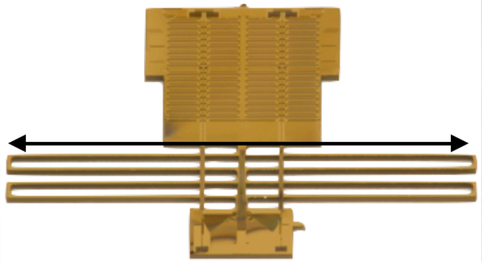


[1] C. B. Schindler, J. T. Greenspun, H. C. Gomez and K. S. J. Pister, "A Jumping Silicon Microrobot with Electrostatic Inchworm Motors and Energy Storing Substrate Springs," *2019 20th International Conference on Solid-State Sensors, Actuat Microsystems & Eurosensors XXXIII (TRANSDUCERS & EUROSENSORS XXXIII)*, Berlin, Germany, 2019, pp. 88-91.
[2] Drew, Daniel S., et al. "Toward controlled flight of the ionocraft: a flying microrobot using electrohydrodynamic thrust with onboard sensing and no moving parts." *IEEE Robotics and Automation Letters* 3.4 (2018): 2807-2813.
[3] Contreras, Daniel S., Daniel S. Drew, and Kristofer SJ Pister. "First steps of a millimeter-scale walking silicon robot." *2017 19th International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS)*. IEEE, 2017.
[4] Rauf, Ahad M., et al. "Towards Aerodynamic Control of Miniature Rockets with MEMS Control Surfaces." *2020 IEEE 33rd International Conference on Micro Electro Mechanical Systems (MEMS)*. IEEE, 2020.
[5] Usevitch, Nathan S., et al. "An untethered isoperimetric soft robot." Science Robotics 5.40 (2020).
[6] Yim, Justin K., and Ronald S. Fearing. "Precision jumping limits from flight-phase control in salto-1p." *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.
[7] Piccoli, Matthew, and Mark Yim. "Piccolissimo: The smallest micro aerial vehicle." *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.
[8] Maekawa, Azumi, et al. "Improvised Robotic Design with Found Objects."

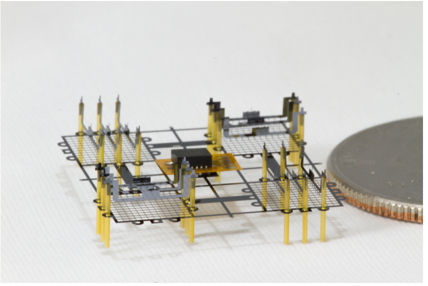
Novel robotic platforms

Microrobots

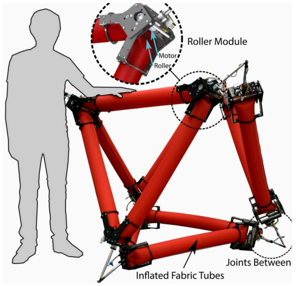
Jumper [1]



Ionocraft [2]

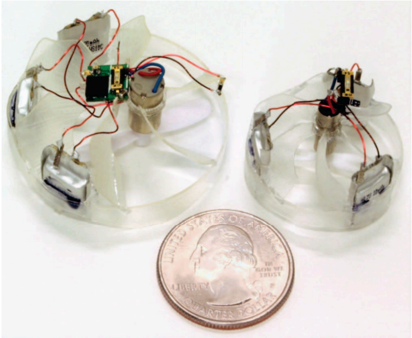


Isoperimetric soft robot [5]



Found Objects [8]

Picolissimo [7]



Demonstrating control for the first time:

1. No strong prior on robot dynamics
2. High cost-per-test

The method for control needs to:

1. Manage uncertainty
2. Be sample efficient

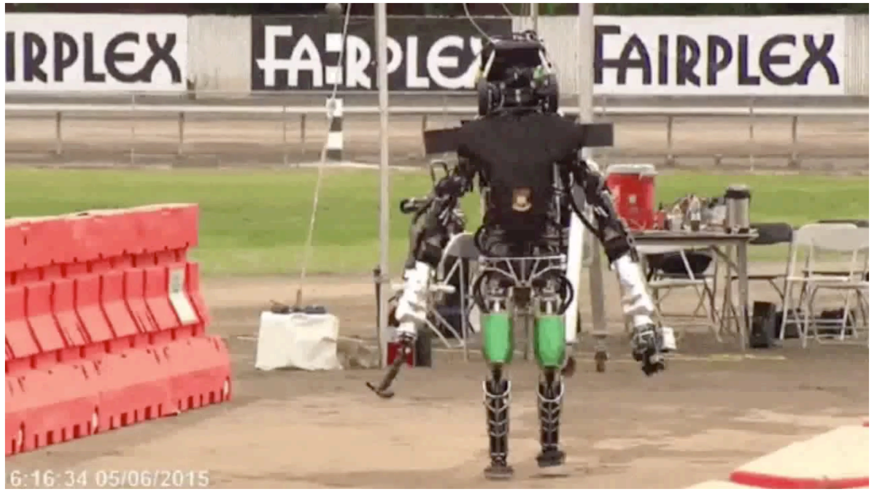
[1] C. B. Schindler, J. T. Greenspun, H. C. Gomez and K. S. J. Pister, "A Jumping Silicon Microrobot with Electrostatic Inchworm Motors and Energy Storing Substrate Springs," *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems & Eurosensors XXXIII (TRANSDUCERS & EUROSENSORS XXXIII)*, Berlin, Germany, 2019, pp. 88-91.
[2] Drew, Daniel S., et al. "Toward controlled flight of the ionocraft: a flying microrobot using electrohydrodynamic thrust with onboard sensing and no moving parts." *IEEE Robotics and Automation Letters* 3.4 (2018): 2807-2813.
[5] Usevitch, Nathan S., et al. "An untethered isoperimetric soft robot." Science Robotics 5.40 (2020).
[7] Piccoli, Matthew, and Mark Yim. "Piccolissimo: The smallest micro aerial vehicle." *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.
[8] Maekawa, Azumi, et al. "Improvised Robotic Design with Found Objects."

“Minimum data” controller synthesis for high-cost robotic systems

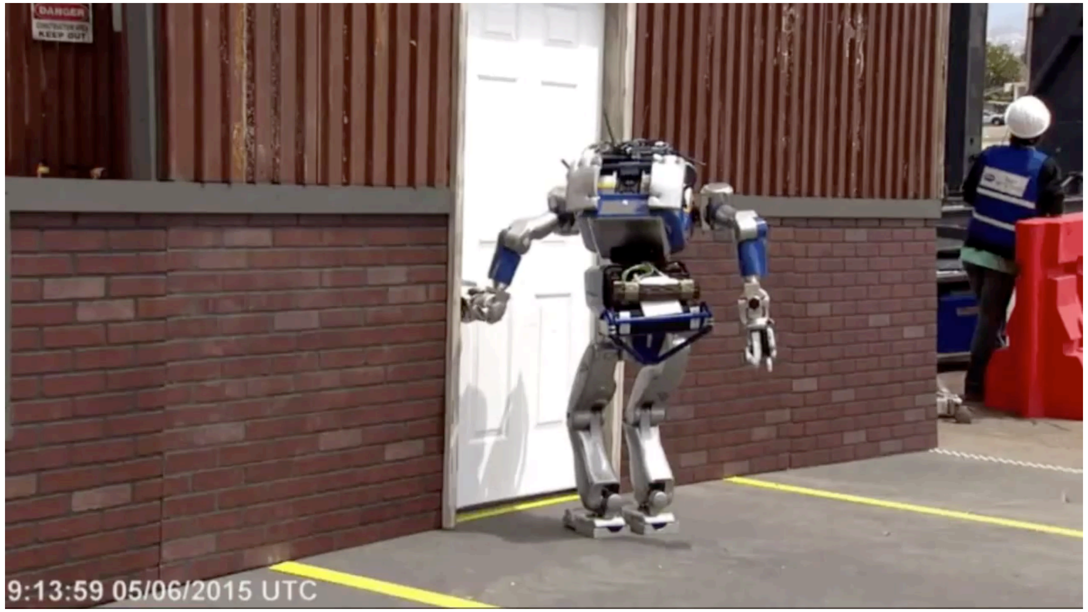
This talk

1. Motivation for model-based reinforcement learning (MBRL)
2. Pairing of model-controller optimization in MBRL
3. Dynamics model design for model predictive control (MPC) in MBRL

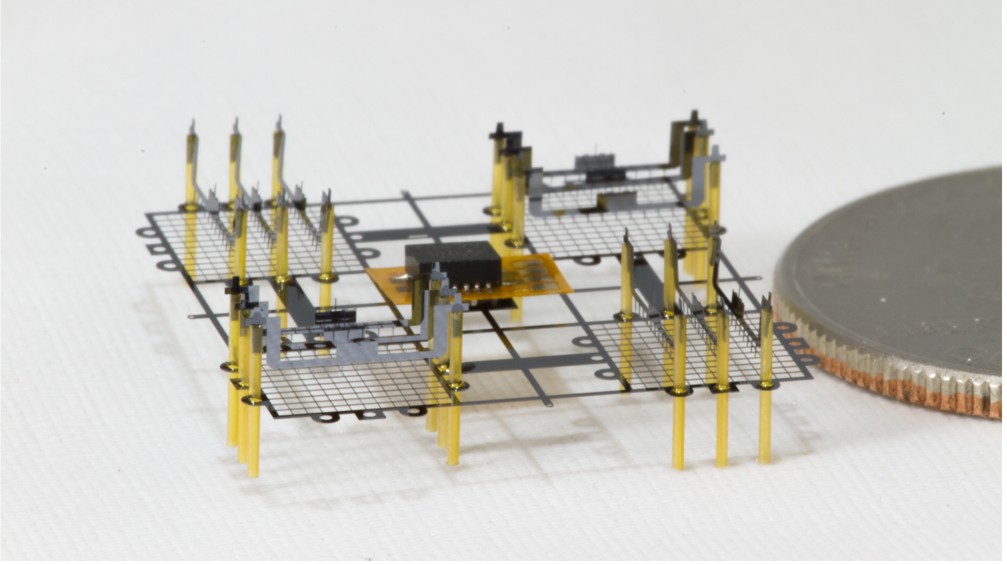
Why use machine learning for robotics?



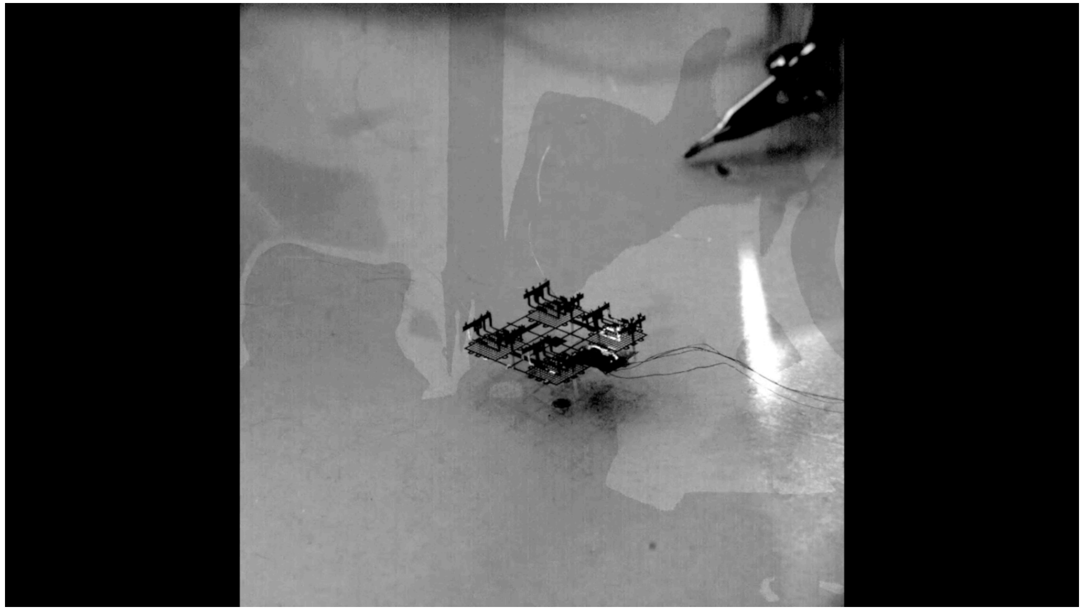
Some famous examples from
DARPA Robotics Challenge (2015)



Why did I start using machine learning?

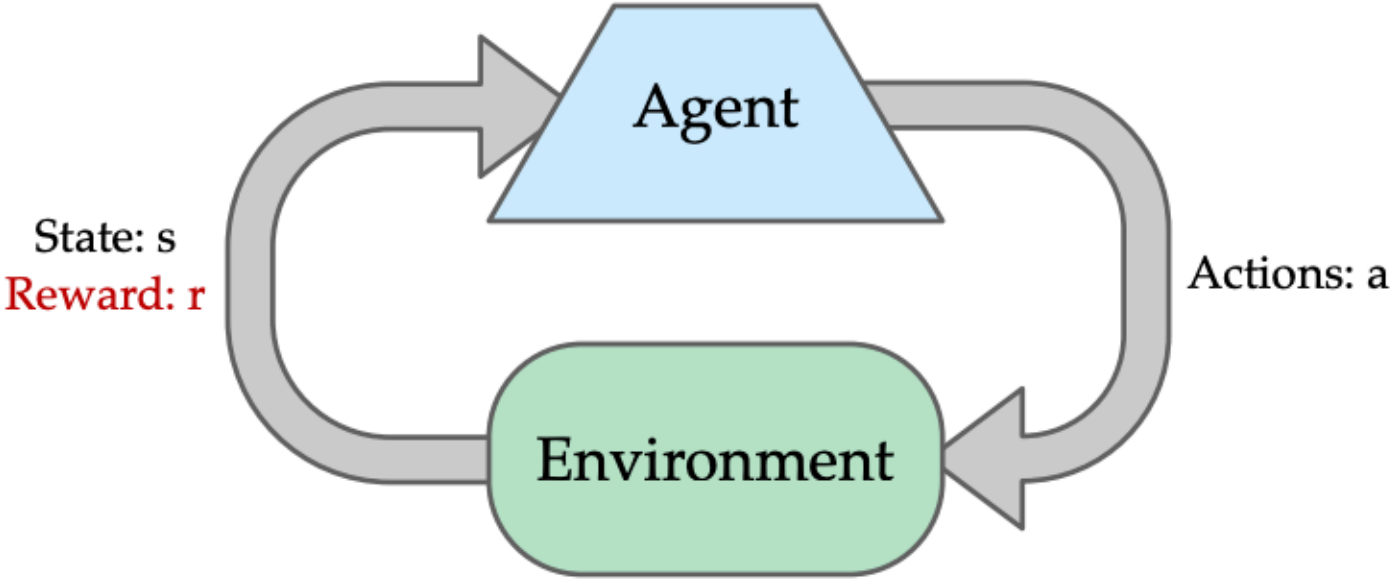


The Ionocraft



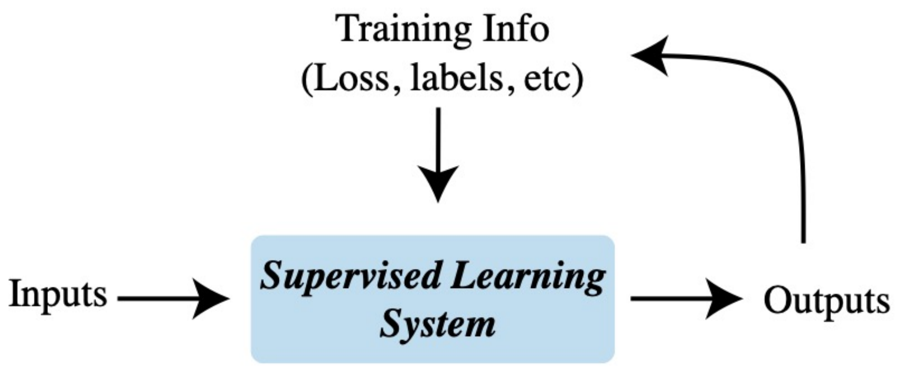
Drew, Daniel S., et al. "Toward controlled flight of the ionocraft: a flying microrobot using electrohydrodynamic thrust with onboard sensing and no moving parts." *IEEE Robotics and Automation Letters* 3.4 (2018): 2807-2813.

Why use reinforcement learning?



[CS 188, UC Berkeley]

Supervised learning vs. reinforcement learning



- Closed system
- Stationary

- Broader (open) system specification
- Added uncertainty from interacting with world

Model-based vs. model-free

Model-based methods (RL, system-identification, PID-tuning):

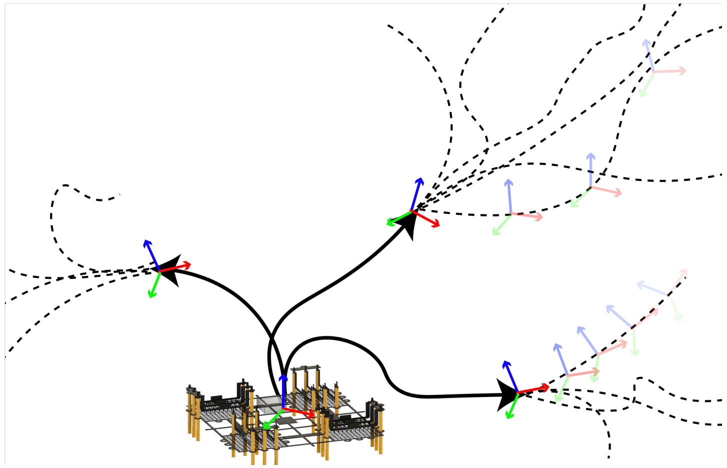
- Offline planning capabilities
- Generalization
- Sample-efficient
- Difficult to implement
- Computationally intensive to train

Model-free reinforcement learning:

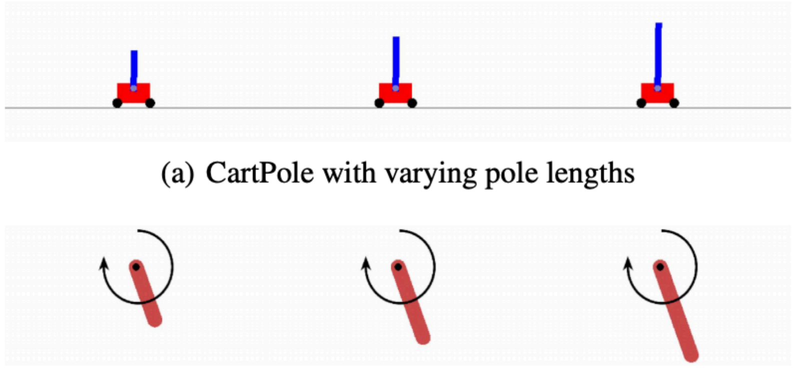
- Reactive policies
- Task-specific
- Data hungry
- Simple to implement
- Computationally light

Why may we want to use models?

Now: Data-efficient



Soon: Generalizable

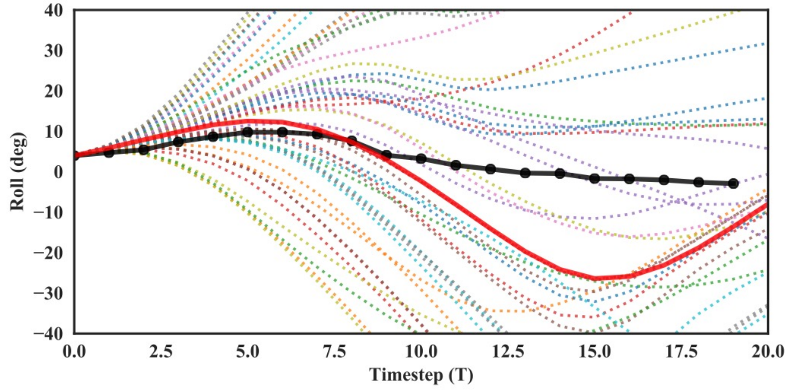


(a) CartPole with varying pole lengths

(b) Pendulum with varying pendulum lengths

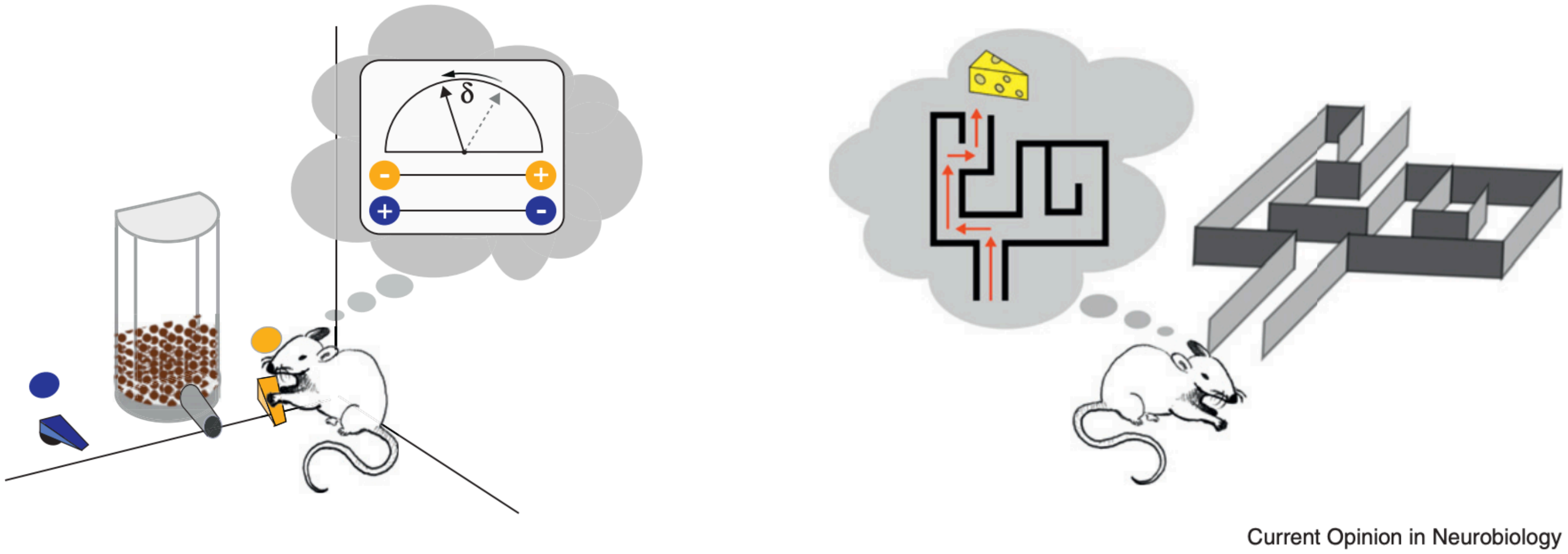
Lee, Kimin, et al. "Context-aware dynamics model for generalization in model-based reinforcement learning." *International Conference on Machine Learning*. PMLR, 2020.

Future: Interpretable



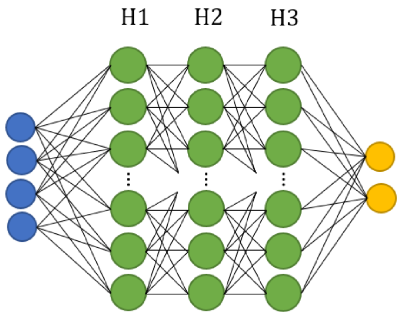
Why was this action chosen?

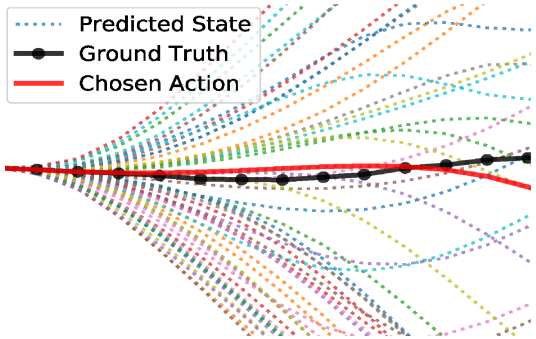
Why may we want to use models?



Doll, Bradley B., Dylan A. Simon, and Nathaniel D. Daw. "The ubiquity of model-based reinforcement learning." *Current opinion in neurobiology* 22.6 (2012): 1075-1081.

Model-based Reinforcement Learning (MBRL)



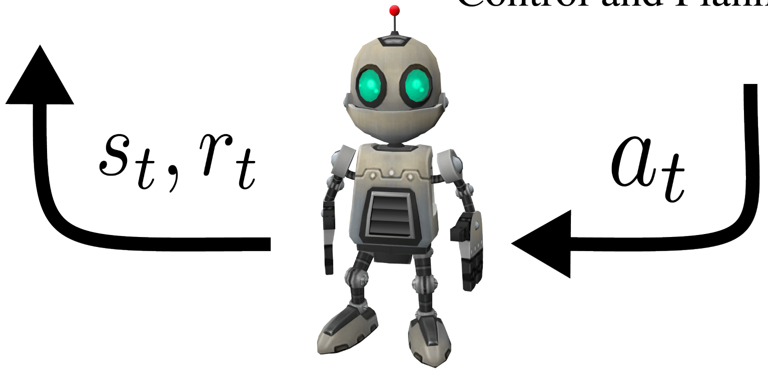


$$s_{t+1} = f_\theta(s_t, a_t)$$

Dynamics Model

$$\arg\max_{u(t)} \sum_{t=0}^{T} r(s_t, a_t)$$

Control and Planning



While improving:

1. Agent acts in environment
2. Learn model of dynamics

$$p_\theta = \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(s_{t+1} | s_t, a_t)$$

3. Plan actions to maximize reward

$$a^* = \arg\max_a \sum_{t=0}^{T} \gamma^t r(s_t, a_t)$$

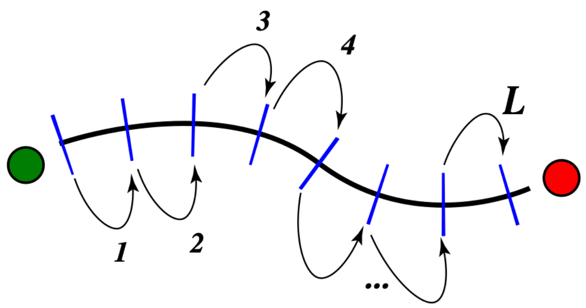
$$s.t.\ s_{t+1} \sim p_\theta(s_{t+1} | s_t, a_t)$$

Feedforward Dynamics Models

Learn model of dynamics

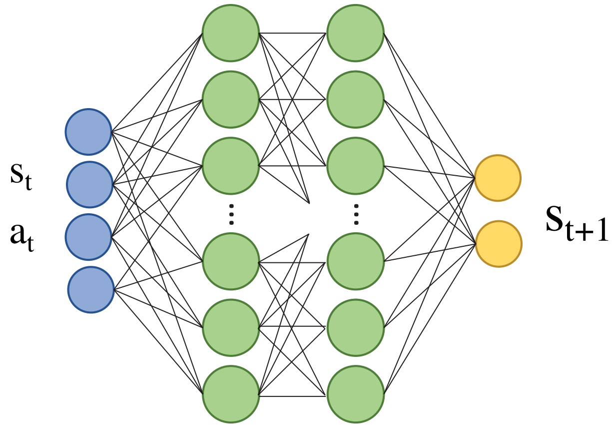
Problem setup:

$$s_{t+1} = s_t + f_\theta(s_t, a_t)$$



Training:

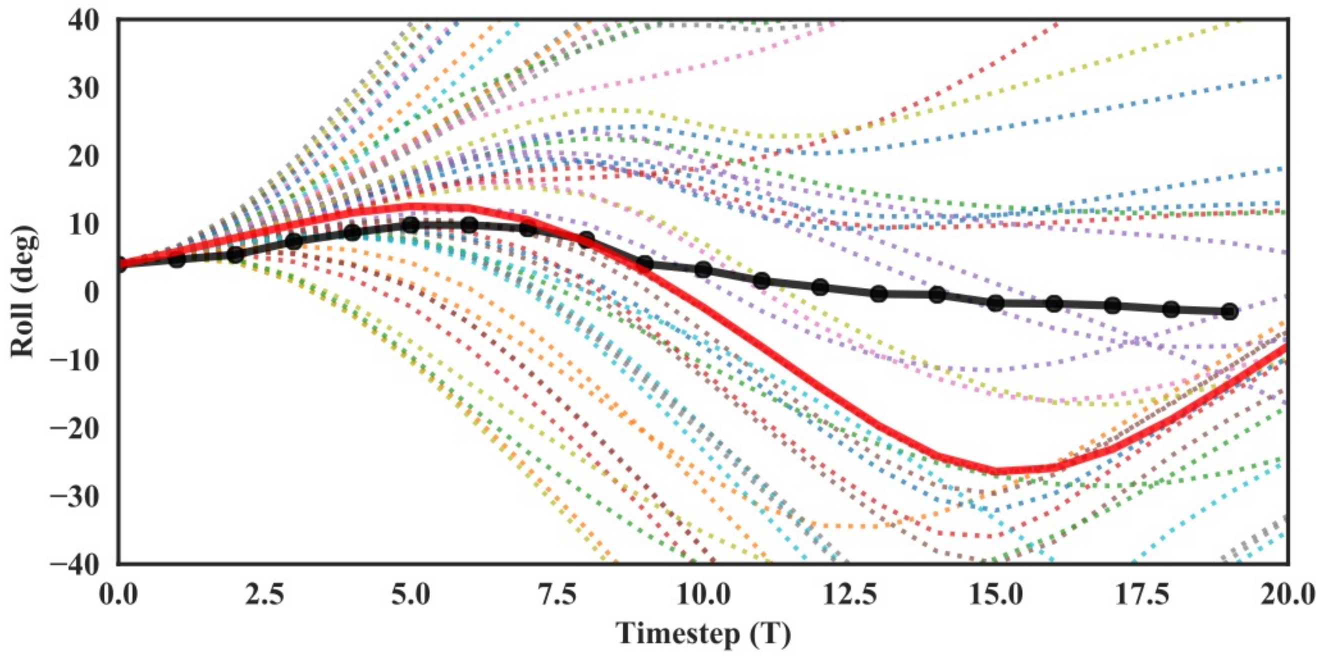
$$p_\theta = \arg\max_\theta \sum_{i=1}^{N} \log\, p_\theta(s_{t+1} | s_t, a_t)$$



Predicting trajectories

$$s_T = f_\theta\Big(f_\theta(\cdots f_\theta(s_i, a_i)\cdots)\Big)$$

Many compounded network passes!



Sample-based Model Predictive Control (MPC)

Planning with a model to maximize reward

Optimization:

$$a^* = \arg\max_a \sum_{t=0}^{T} \gamma^t r(s_t, a_t)$$
$$s.t. \; s_{t+1} \sim p_\theta(s_{t+1} | s_t, a_t)$$

- Sample actions from distribution P
- Plan to horizon h (need to tune)
- Computationally intensive planning trajectories

An example using MBRL

Lambert, Nathan O., et al. "Low-level control of a quadrotor with deep model-based reinforcement learning." *IEEE Robotics and Automation Letters* 4.4 (2019): 4224-4230.

- Task: minimize Euler angles

$$r(s) = -(\theta^2 + \phi^2)$$
$$r(s) = -c(s)$$

- Onboard state values

$$s_t = [\theta\ \phi\ \psi\ \ddot{x}\ \ddot{y}\ \ddot{z}\ \dot{\omega}_x\ \dot{\omega}_y\ \dot{\omega}_z]$$

- Direct motor PWM application

$$a_t = [PWM_1\ \ PWM_2\ \ PWM_3\ \ PWM_4] \in [0, 65535]$$

- Internal controllers off (MPC update at 25/50 Hz)

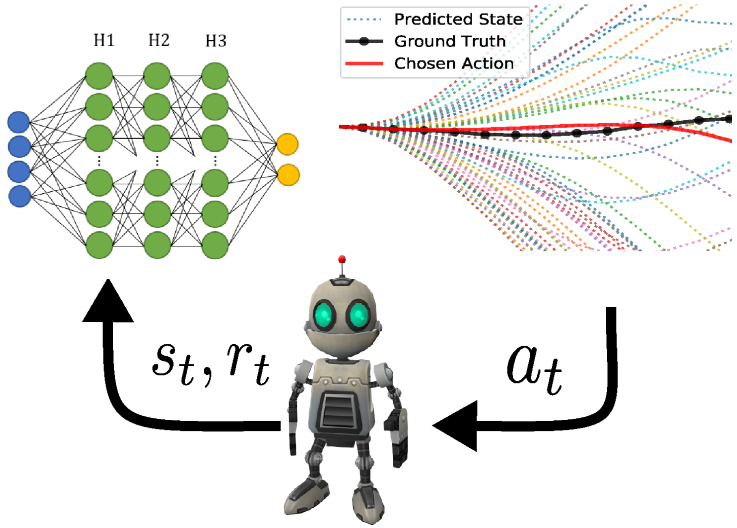
Limitations & challenges of MBRL

Theoretical

- Optimizing model for control
- Modelling accuracy is limited
- Stochasticity of sample-based control

Practical

- Computational limits
- Getting useful data



This talk

1. Motivation for model-based reinforcement learning (MBRL)
2. Pairing of model-controller optimization in MBRL
3. Dynamics model design for model predictive control (MPC) in MBRL

Key Assumption

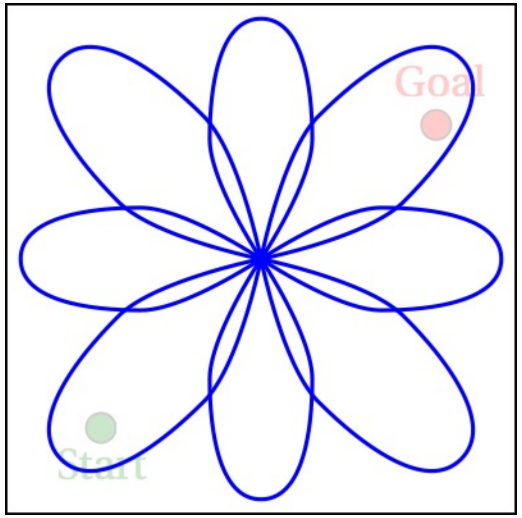
Optimizing dynamics model for control

$$\max \text{log-likelihood} \leftrightarrow \max \text{episode reward}$$

Model learning for control: origins

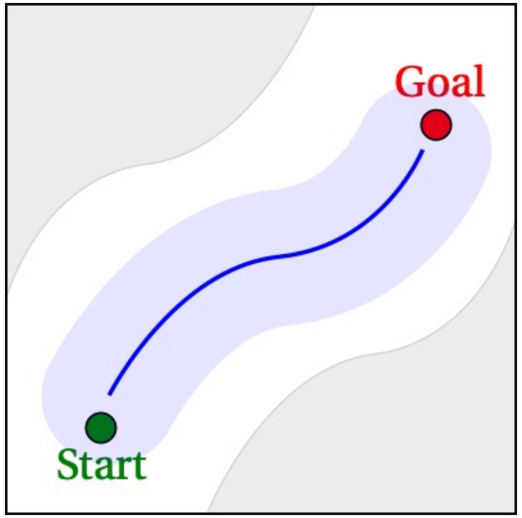
System Identification

- Obtain a task-agnostic (sometimes global) model
- Then learn control



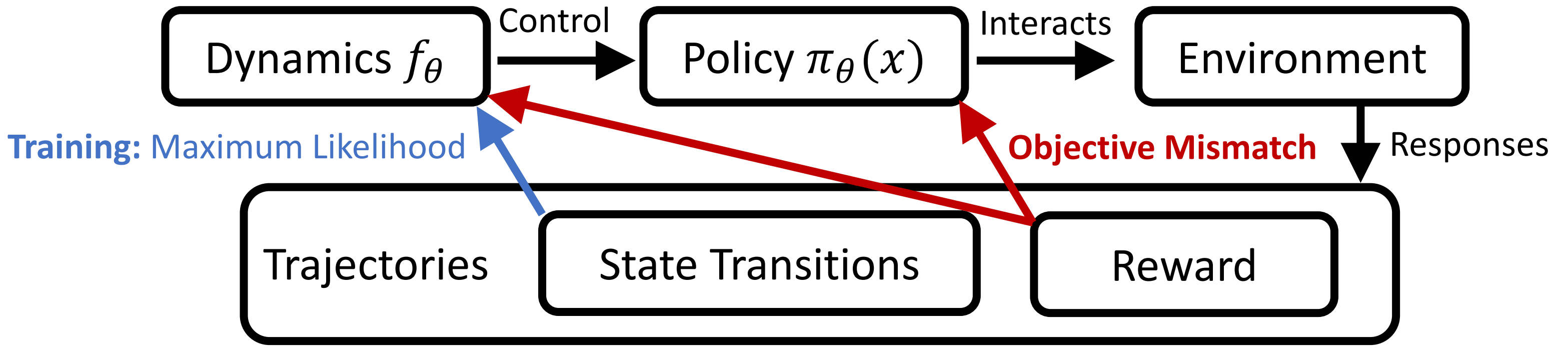
Reinforcement learning

- Observe task-specific data subset
- Iteratively learn model, control



Lambert, N., Amos, B., Yadan, O. & Calandra, R.. (2020). Objective Mismatch in Model-based Reinforcement Learning. *Proceedings of the 2nd Conference on Learning for Dynamics and Control, in PMLR* 120:761-770

Revisiting MBRL



Lambert, N., Amos, B., Yadan, O. & Calandra, R.. (2020). Objective Mismatch in Model-based Reinforcement Learning. *Proceedings of the 2nd Conference on Learning for Dynamics and Control, in PMLR* 120:761-770

A dual optimization

$$\textbf{Training:}\ \arg\max_{\theta} \sum_{i=1}^{N} \log p_\theta(s_i'|s_i, a_i), \quad \textbf{Control:}\ \arg\max_{a_{t:t+T}} \mathbb{E}_{\pi_\theta(s_t)} \sum_{i=t}^{t+T} r(s_i, a_i)$$

Objective Mismatch

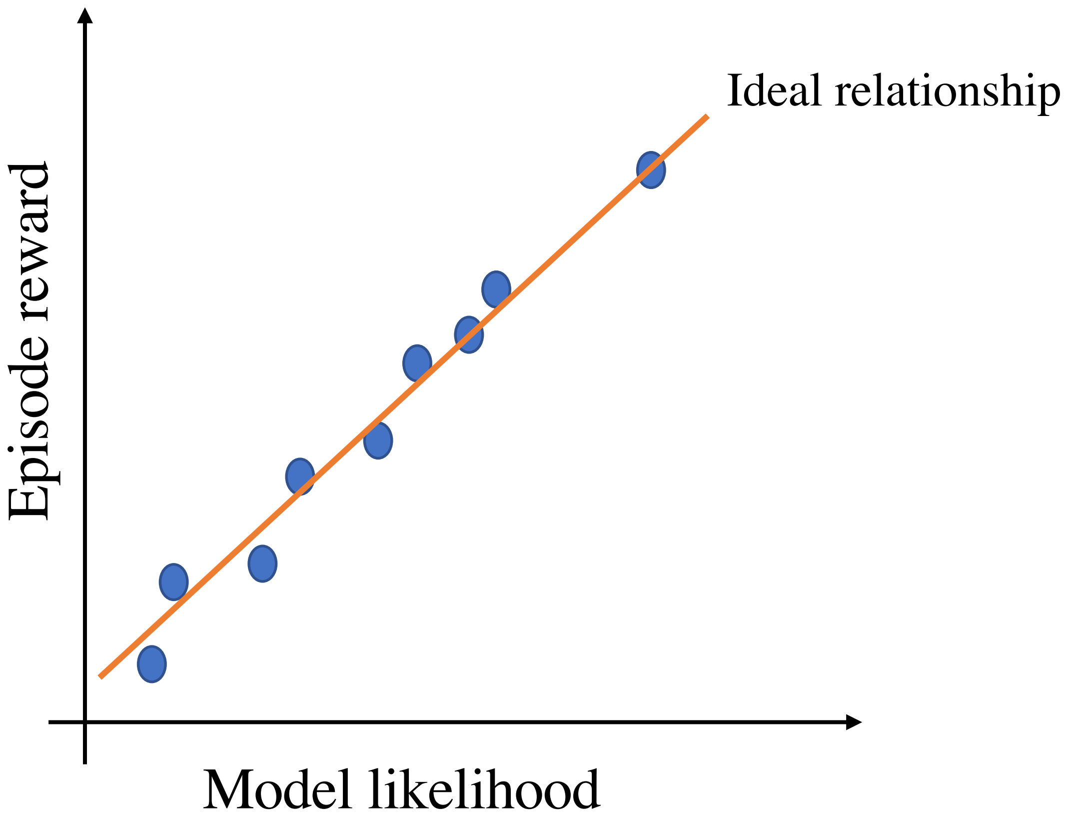
Lambert, N., Amos, B., Yadan, O. & Calandra, R.. (2020). Objective Mismatch in Model-based Reinforcement Learning. *Proceedings of the 2nd Conference on Learning for Dynamics and Control, in PMLR* 120:761-770

Underlying assumption of model learning

$$\max \text{log-likelihood} \overset{?}{\leftrightarrow} \max \text{episode reward}$$

Lambert, N., Amos, B., Yadan, O. & Calandra, R.. (2020). Objective Mismatch in Model-based Reinforcement Learning. *Proceedings of the 2nd Conference on Learning for Dynamics and Control, in PMLR* 120:761-770

Model validation likelihood vs episode reward

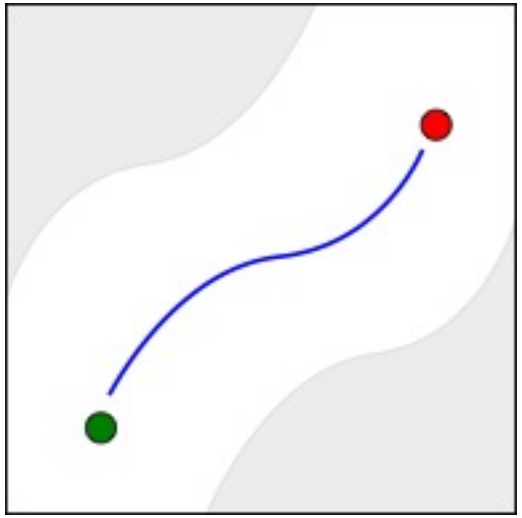


Lambert, N., Amos, B., Yadan, O. & Calandra, R.. (2020). Objective Mismatch in Model-based Reinforcement Learning. *Proceedings of the 2nd Conference on Learning for Dynamics and Control, in PMLR* 120:761-770

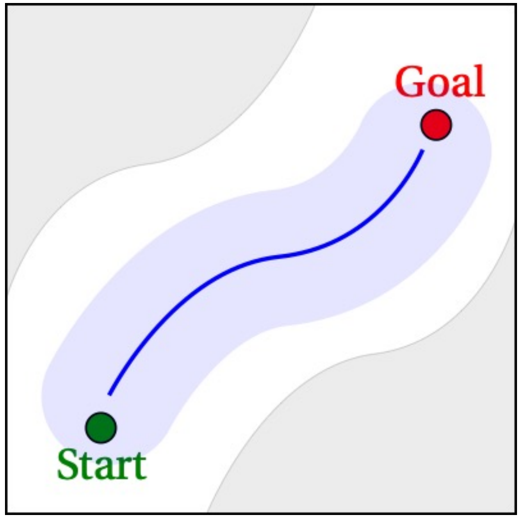
Correlation on different datasets?

$$\max \text{log-likelihood} \overset{?}{\leftrightarrow} \max \text{episode reward}$$

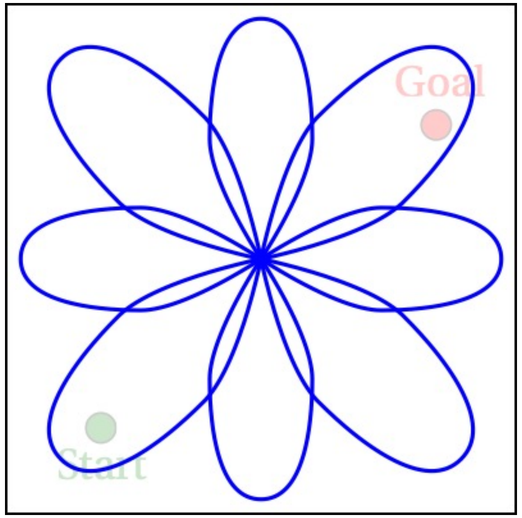
Expert



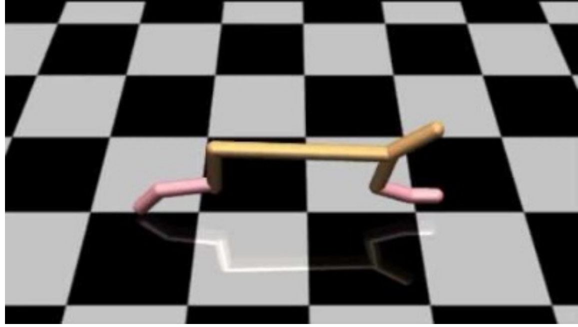
On-policy

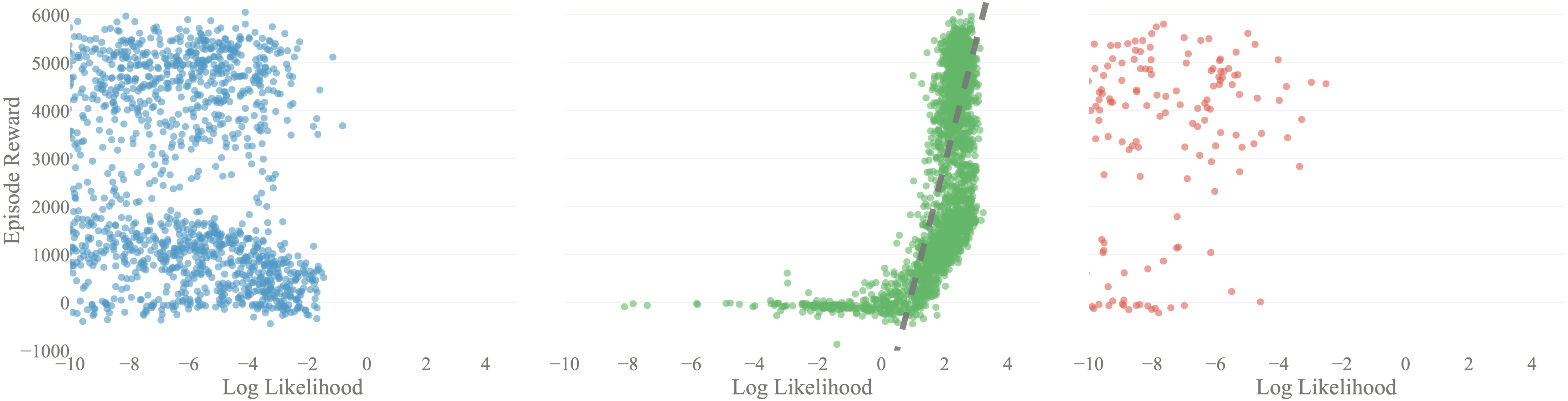


Global



Model Likelihood vs reward





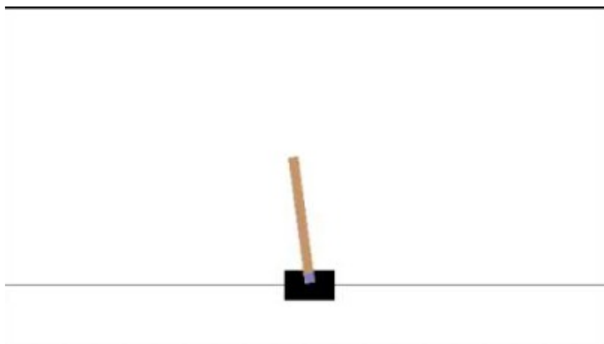
Expert (ϱ=0.07) On-Policy (ϱ=0.46) Global (ϱ =0.19)

ϱ: *Pearson Correlation Coefficient*

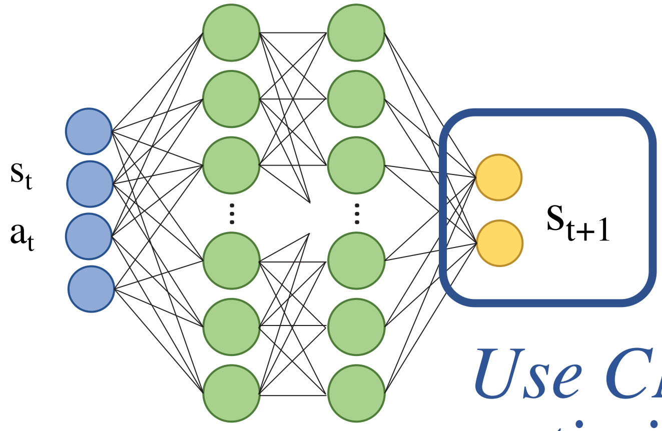
Lambert, N., Amos, B., Yadan, O. & Calandra, R.. (2020). Objective Mismatch in Model-based Reinforcement Learning. *Proceedings of the 2nd Conference on Learning for Dynamics and Control, in PMLR* 120:761-770

Adversarial attack on a dynamics model

Task: Cartpole



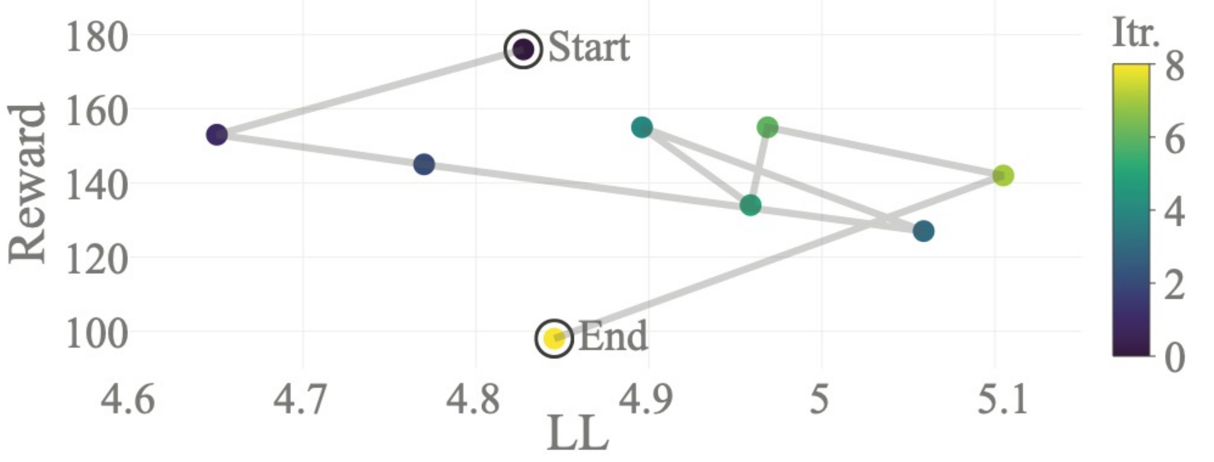
$$p_\theta = \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(s_{t+1}|s_t, a_t)$$



Use CMA-ES to optimize output layer.

Goal, model on cartpole with

- High accuracy (log-likelihood of transitions, LL)
- Low mean reward with MPC



Adversarial attack on a dynamics model

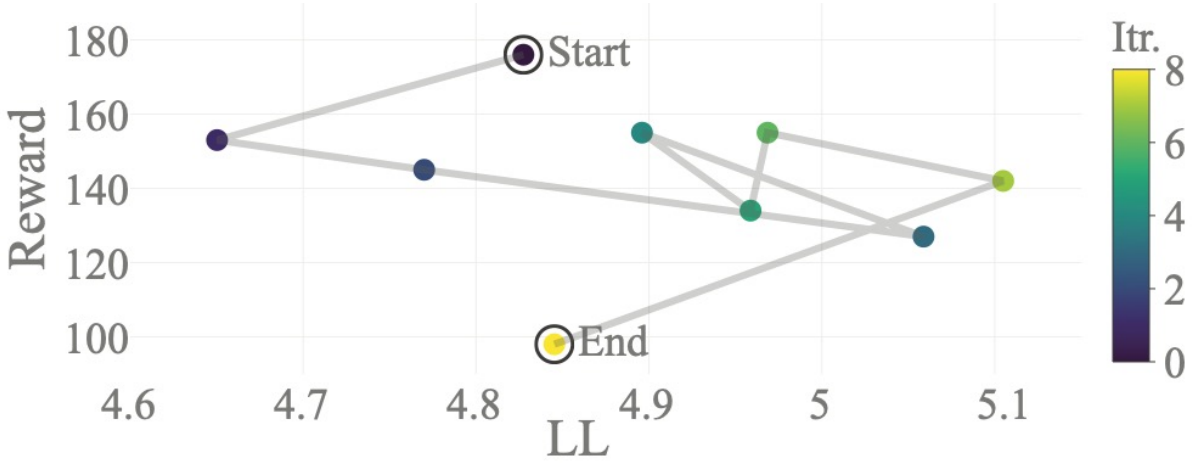
Intuition

- Lose model accuracy on area of interest
- Gain model accuracy on unimportant areas of the state-space

Hard phenomena to measure!

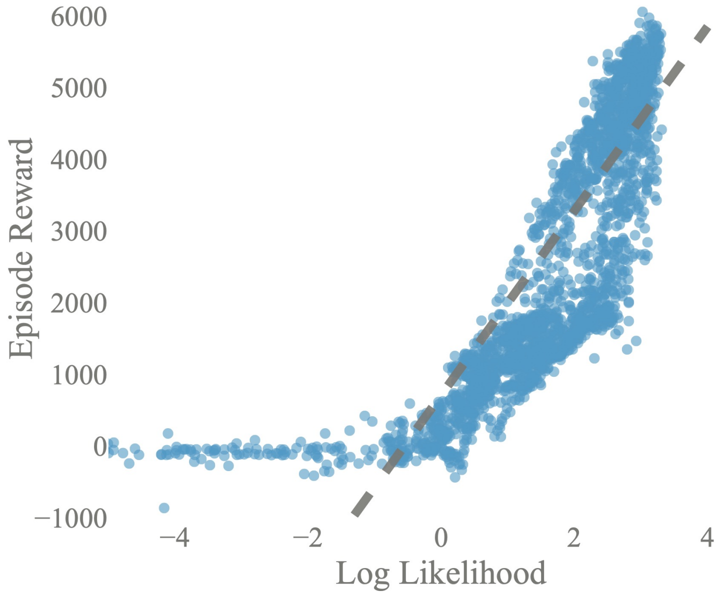
Goal, model on cartpole with

- High accuracy (log-likelihood of transitions, LL)
- Low mean reward with MPC



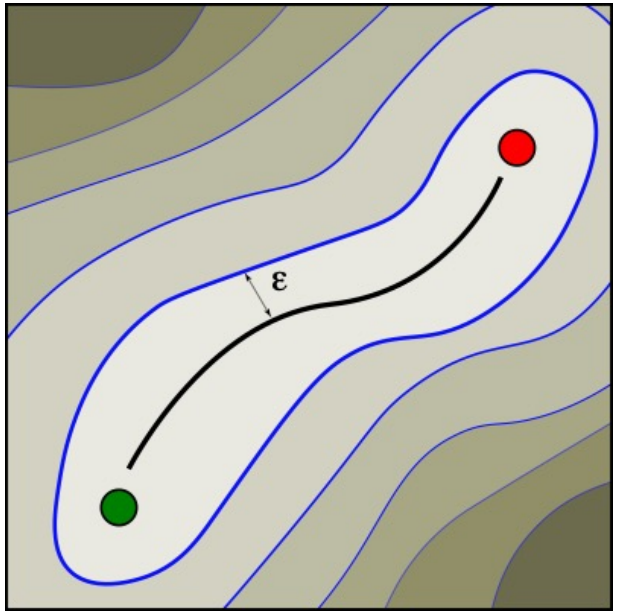
Ways to mitigate "objective mismatch"

1. Train models to predict trajectories



(a) HC traj. loss ($\rho = 0.63$)

2. Re-weight dynamics data around task of interest



From one-step training on trajectories to a model *designed* for prediction trajectories!

This talk

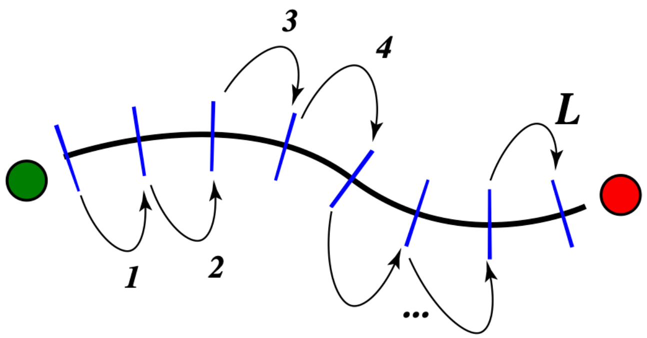
1. Motivation for model-based reinforcement learning (MBRL)
2. Pairing of model-controller optimization in MBRL
3. Dynamics model design for model predictive control (MPC) in MBRL

A model for predicting trajectories

Standard one-step lookahead

- Compounding predictions

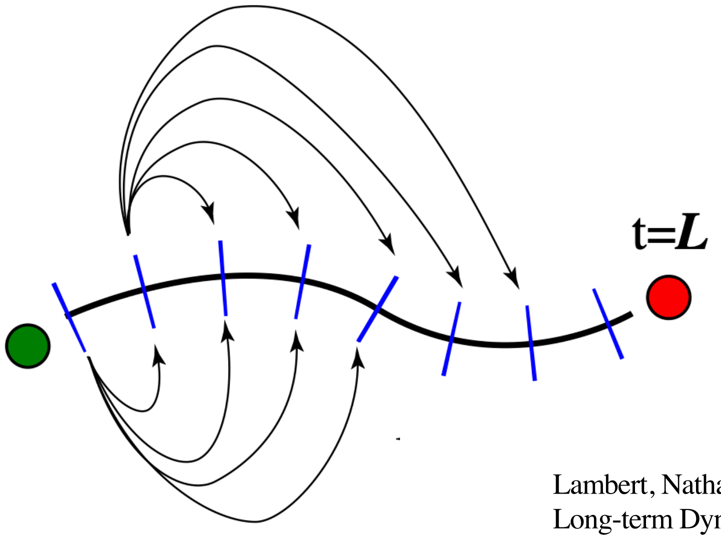
$$s_{t+1} = s_t + f_\theta(s_t, a_t)$$



Trajectory-based models

- Time dependent prediction

$$s_{t+h} = f_\theta(s_t, h, \theta_\pi)$$

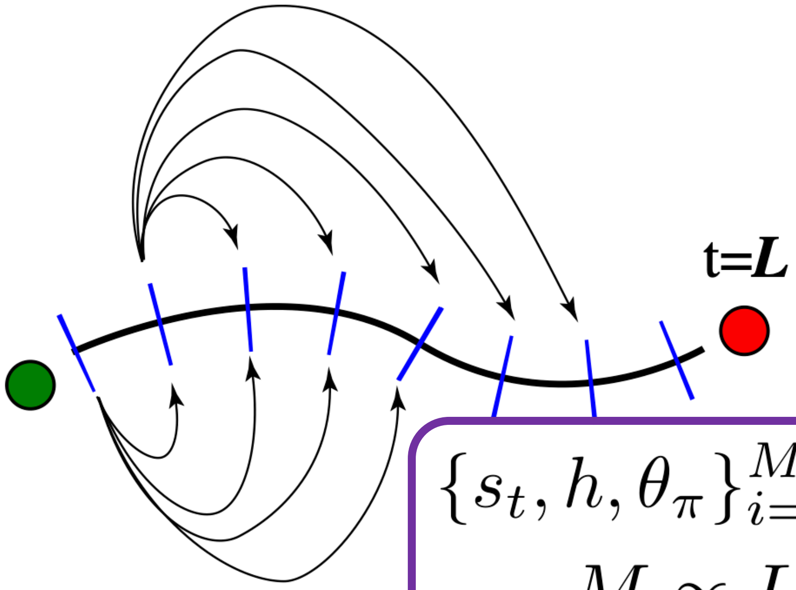


Lambert, Nathan O., et al. "Learning Accurate Long-term Dynamics for Model-based Reinforcement Learning." *arXiv preprint arXiv:2012.09156* (2020)

“Trajectory-based” dynamics model

Trajectory-based models

Control conditioned, time indexed



$$\{s_t, h, \theta_\pi\}_{i=1}^{M}$$
$$M \propto L^2$$

Supervised learning samples (more later)

Advantages

- Long-term prediction accuracy
- Collects datapoints at rate of L^2
- Computationally efficient planning
- Stable uncertainty propagation

Control parameters

Starting state

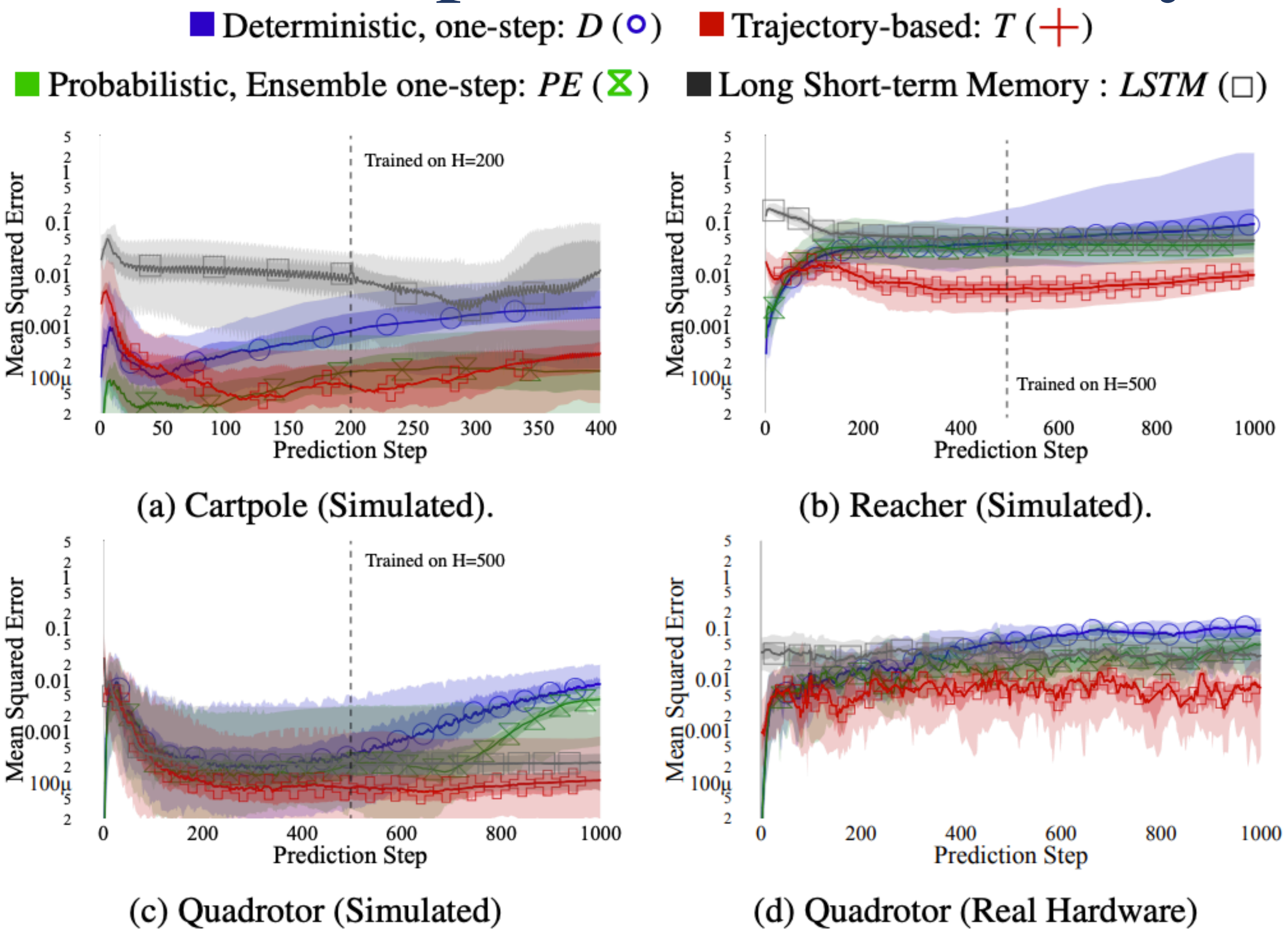
$$s_{t+h} = f_\theta(\boxed{s_t}, \boxed{h}, \boxed{\theta_\pi})$$

Prediction horizon

Lambert, Nathan O., et al. "Learning Accurate Long-term Dynamics for Model-based Reinforcement Learning." *arXiv preprint arXiv:2012.09156* (2020)

Trajectory-based Model Benefits

Benefits – prediction accuracy



Lambert, Nathan O., et al. "Learning Accurate Long-term Dynamics for Model-based Reinforcement Learning." *arXiv preprint arXiv:2012.09156* (2020)

Benefits – efficient planning

- More labelled data

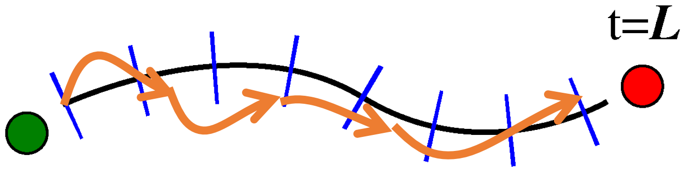
$$N_{\text{train}} = n \sum_{t=1}^{L} t = n \frac{(L)(L-1)}{2} \approx nL^2$$

- Predict with time index (rather then recursive trajectory)

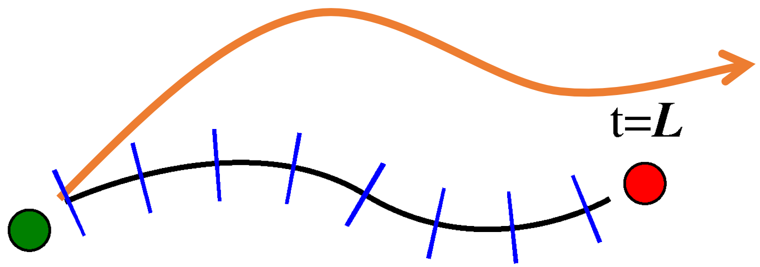
Trajectory-based prediction

$$s_{t+h} = f_\theta(s_t, h, \theta_\pi)$$

Recall: one-step prediction

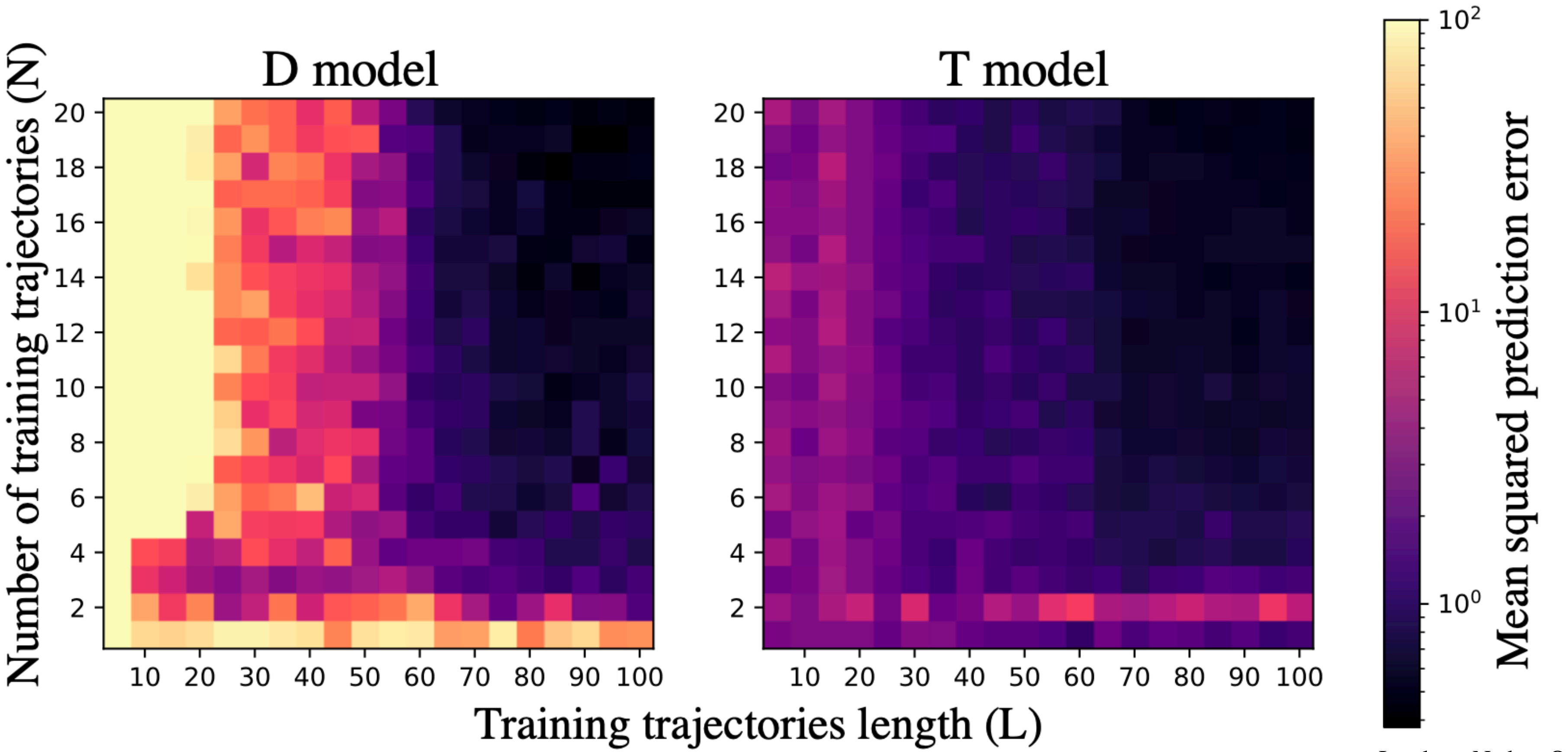


Parallel pass in $t = [1\ 2\ 3\ 4\ \cdots\ L]$



Benefits – sample efficiency

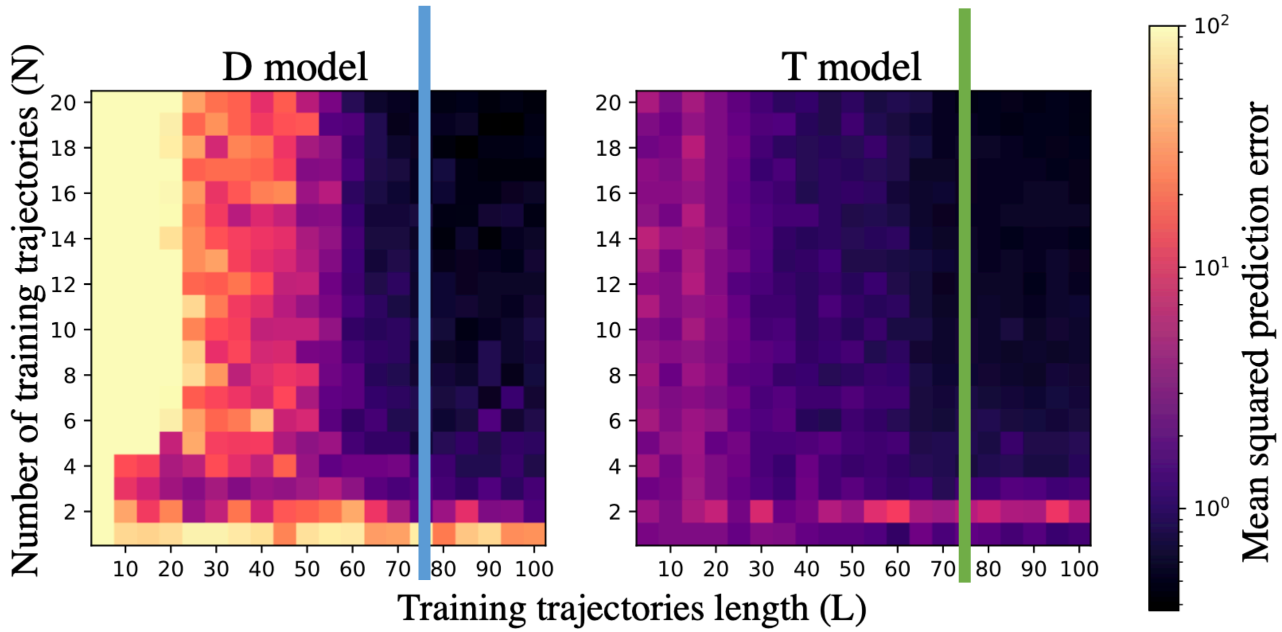
$$N_{\text{train}} = n\sum_{t=1}^{L} t = n\frac{(L)(L-1)}{2} \approx nL^2$$

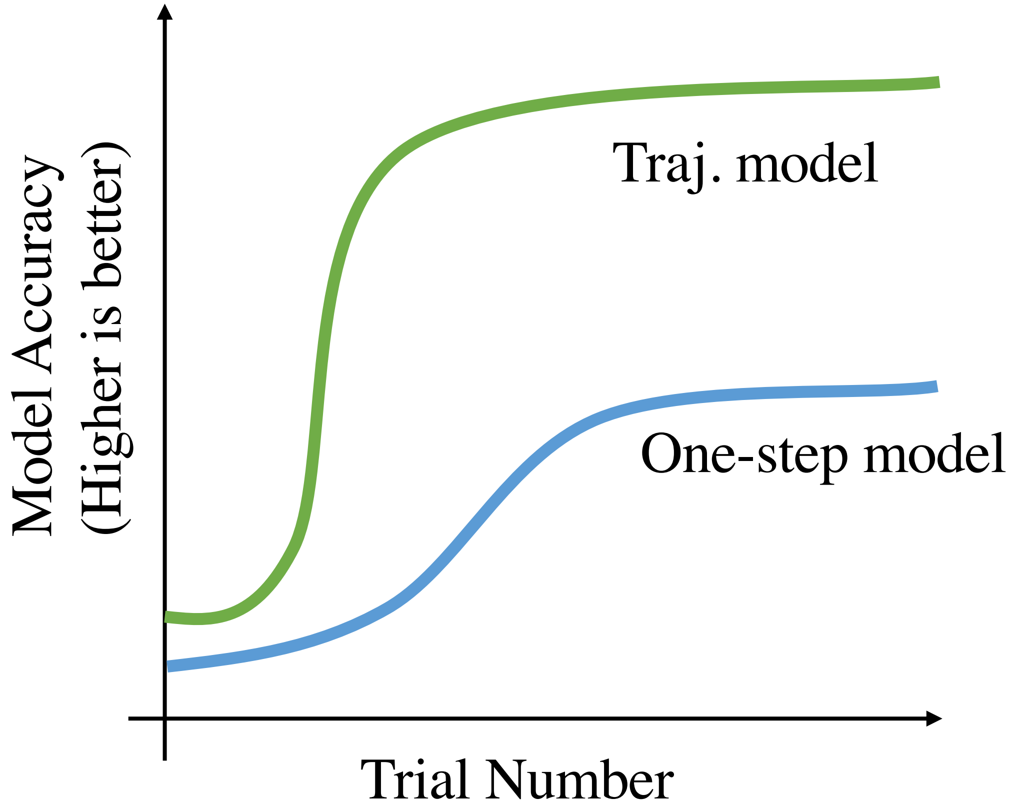


Lambert, Nathan O., et al. "Learning Accurate Long-term Dynamics for Model-based Reinforcement Learning." *arXiv preprint arXiv:2012.09156* (2020)

Benefits – sample efficiency

What is a slice of this heatmap?





Model accuracy over trials

Using the Trajectory-based model in MPC

One-step model planning:

$$u_t^* = \arg\max_{u_{t:t+\tau}} \sum_{i=0}^{\tau} r(\hat{x}_{t+i}, u_{t+i}),$$
$$s.t. \quad \hat{x}_{t+1} = f_\theta(\hat{x}_t, u_t).$$

Trajectory-based model planning:

Plan over control parameter (θ_π) space

$$\theta_{\pi,t}^* = \arg\max_{\theta_{\pi,t:t+\tau}} \sum_{i=0}^{\tau} r(\hat{x}_{t+i}, u_{t+i})$$
$$s.t. \quad \hat{x}_{t+\tau} = f_\theta(\hat{x}_t, \theta_{\pi,t}, t+\tau), \quad u_t^* = \theta_\pi^*(t).$$

Lambert, Nathan O., et al. "Learning Accurate Long-term Dynamics for Model-based Reinforcement Learning." *arXiv preprint arXiv:2012.09156* (2020)

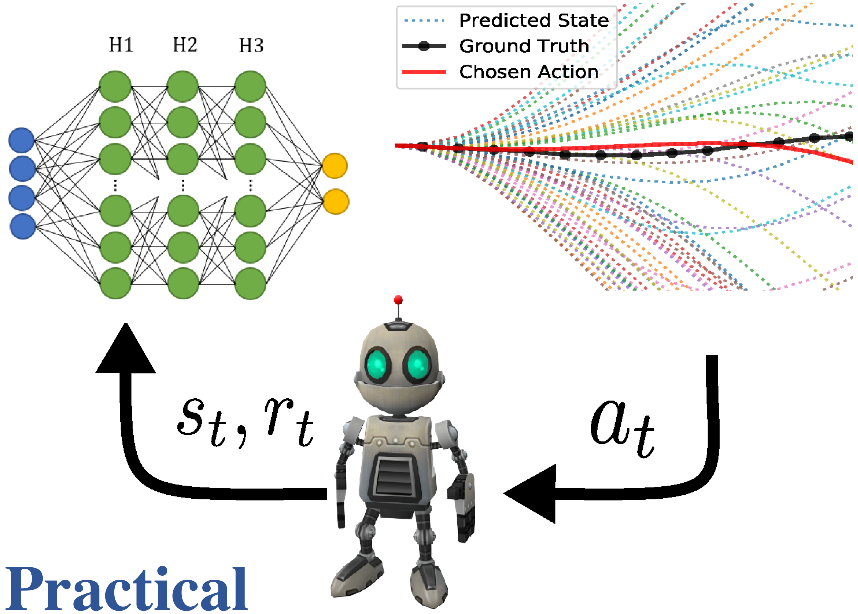
“No free lunch” and dynamics models

- Long term prediction accuracy, but needs controller parametrization
- One-step models are broadly applicable (*so not specialized!*)

Recap & future directions in MBRL

Theoretical

- Optimizing both model and controller
- Modelling accuracy is limited
- Stochasticity of sample-based control



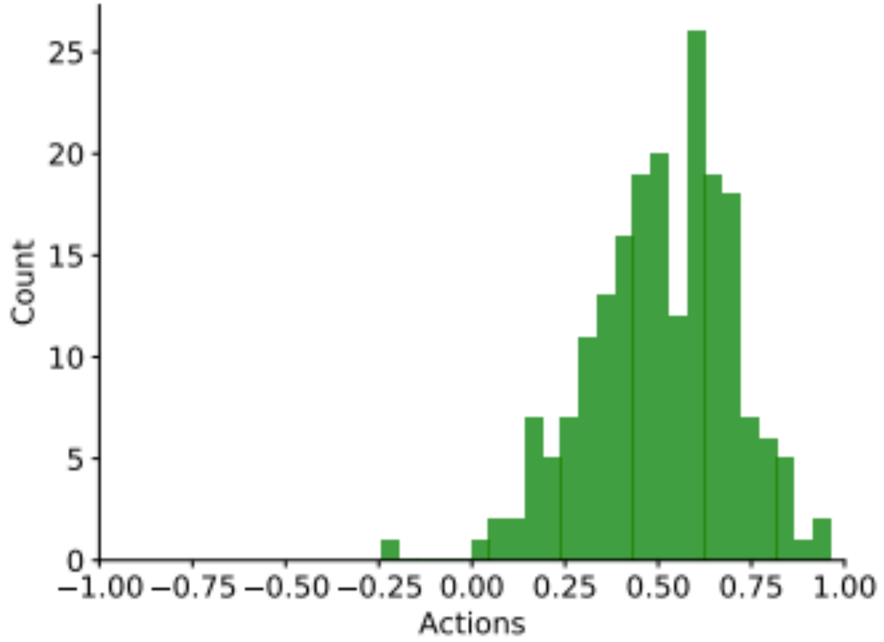
Practical

- Computational limits
- Getting useful data

Future work: MPC distillation

Can we reduce MPC compute to a feedforward policy?

- Imation learning,
- Managing uncertainty of sample-based planning,
- Huge potential upside to hardware robots!



Example: action distribution when re-running MPC at a given state (learned model, cartpole)

“Minimum data” controller synthesis for robotics:

- There is not time to get data to perfectly understand the world
- RL allows one to build structures to optimize for what matters

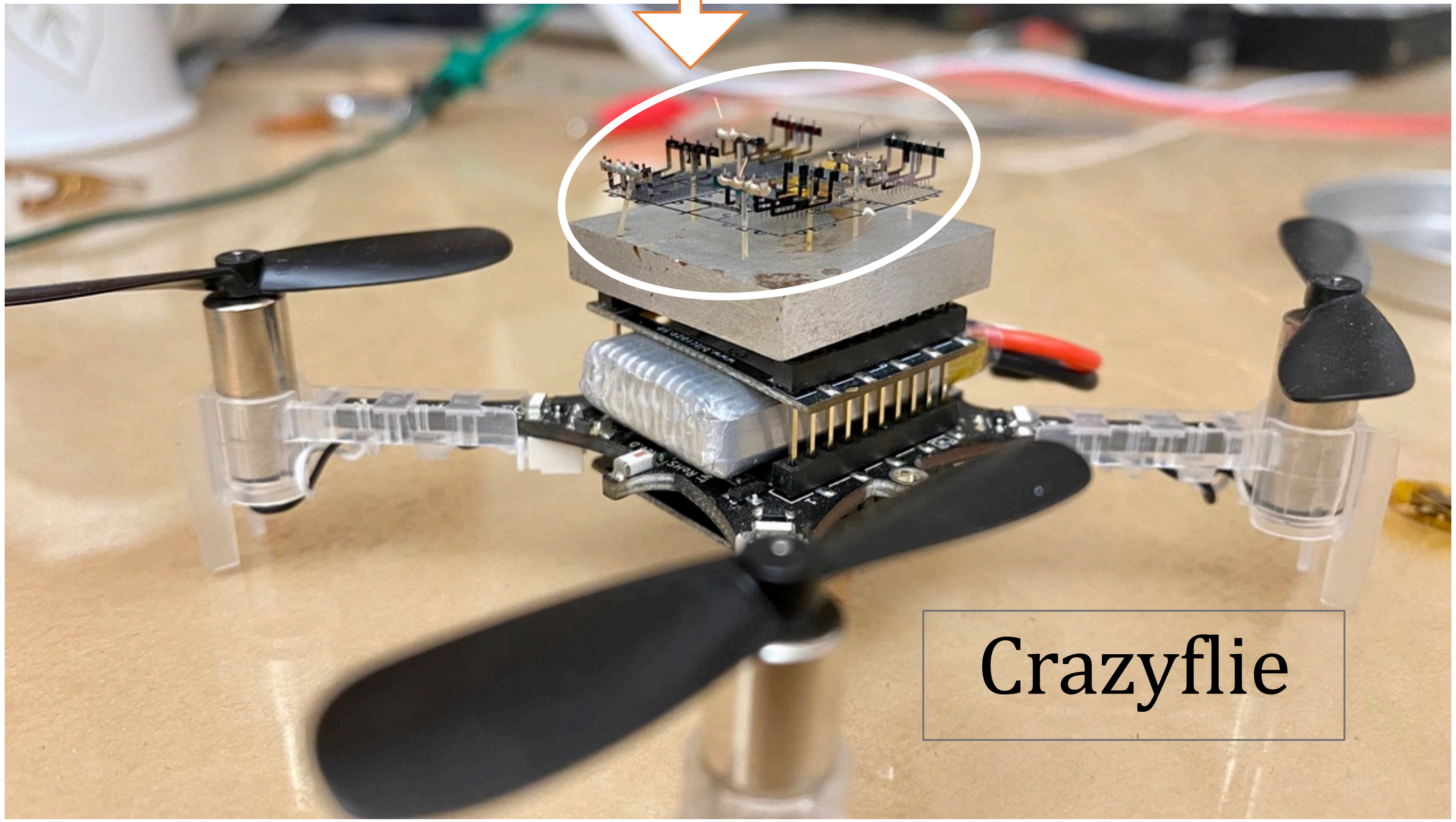
Collaborators!

Roberto Calandra

Kris Pister

Brandon Amos, Daniel Drew, Craig Schindler, Sergey Levine, Luis Pineda, Albert Wilcox, Joseph Yaconelli, Omry Yadan, Howard Zhang

Thanks!



Nathan Lambert, nol@berkeley.edu, natolambert.com

facebook research

BSAC