On optimal control and machine learning

Tutorial at the Workshop on New Frontiers in Learning, Control, and Dynamical Systems

Brandon Amos • Meta AI (FAIR) NYC

slides

github.com/bamos/presentations

International Conference On Machine Learning

Disclaimer

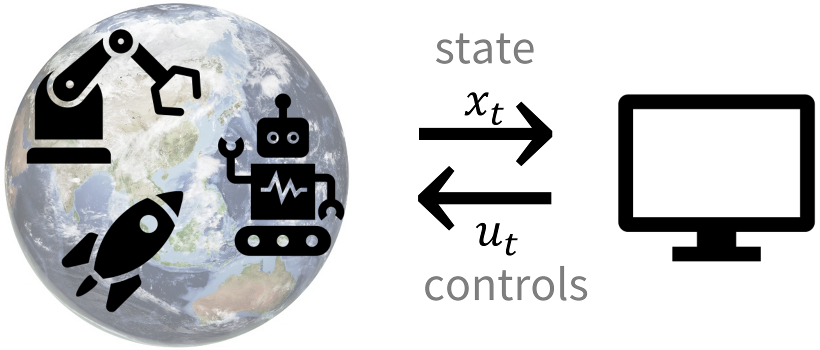
This is an **introduction to optimal control for machine learning researchers**

- **Perspective:** a starting map of ideas rather than a comprehensive coverage
- A tour through some of my favorite **ideas**, **foundations**, and **recent papers**
- Will emphasize the **engineering** side — concepts most useful for building systems

Focus also on **continuous** control, but many concepts transfer to discrete settings

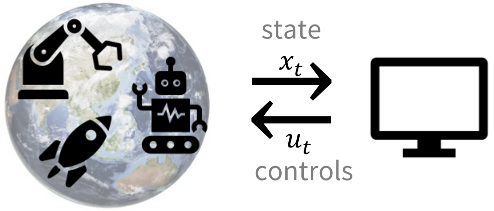
What is optimal control?

Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model



The brachistochrone problem

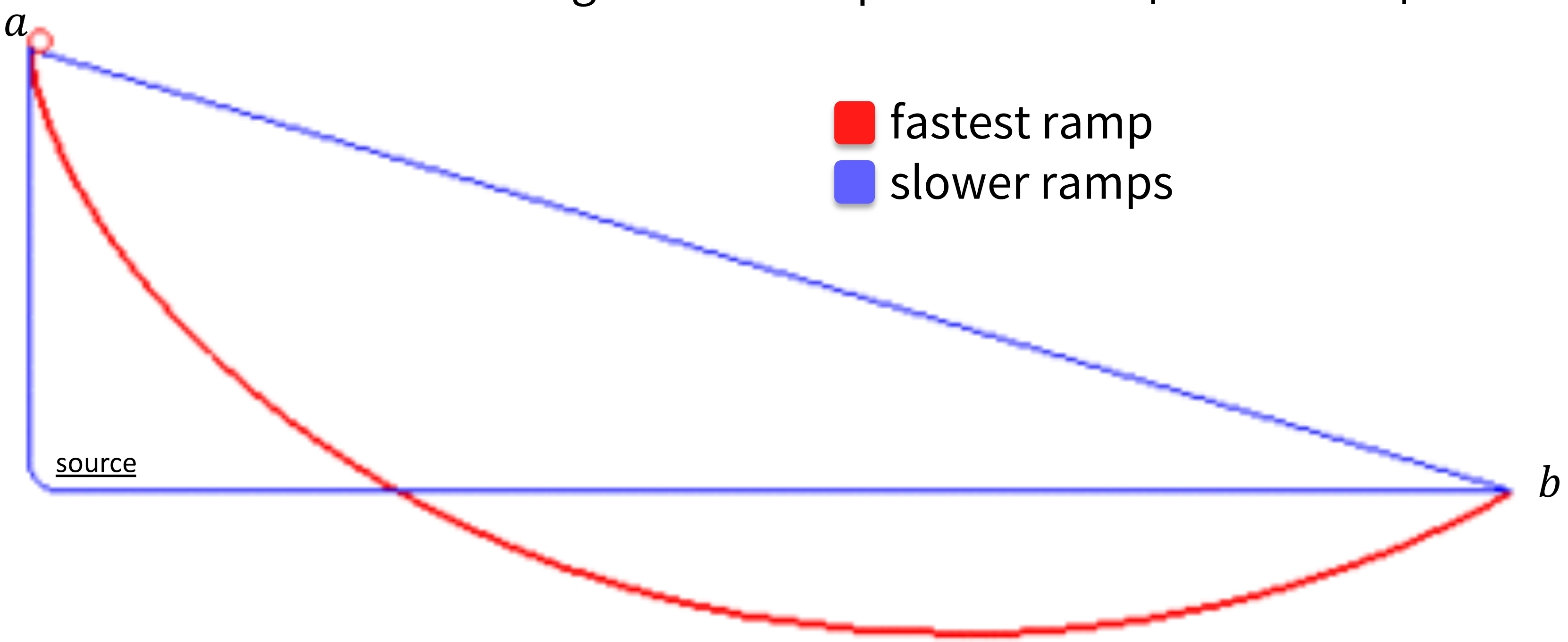
Johann Bernoulli, 1696



Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

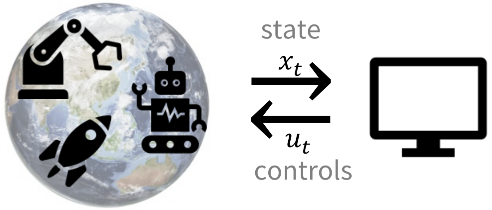
ball rolling down a ramp

shape of the ramp



Source code examples in Dymos and APMonitor

Optimal control in robotics

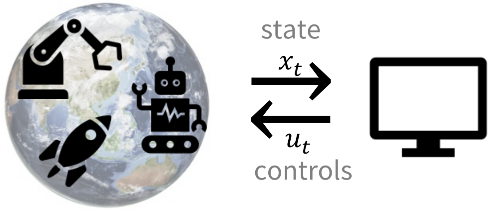


Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion $M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking

Optimal control in robotics



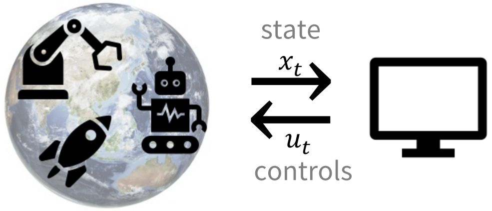
Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion $M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking

📚 *Learning Agile and Dynamic Motor Skills for Legged Robots*. Hwangbo et al., Science Robotics 2019.

Optimal control in robotics



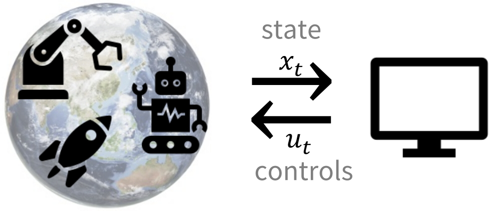
Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion $M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking

📚 *RMA: Rapid Motor Adaptation for Legged Robots*. Ashish Kumar et al., RSS 2021.

Optimal control in robotics



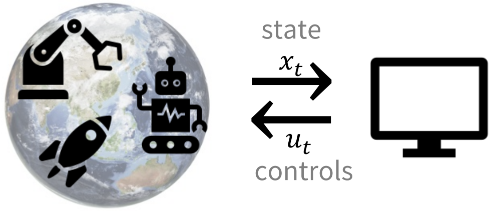
Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion $M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking

📚 *Advanced skills through multiple adversarial motion priors in reinforcement learning.* Vollenweider et al., ICRA 2023.

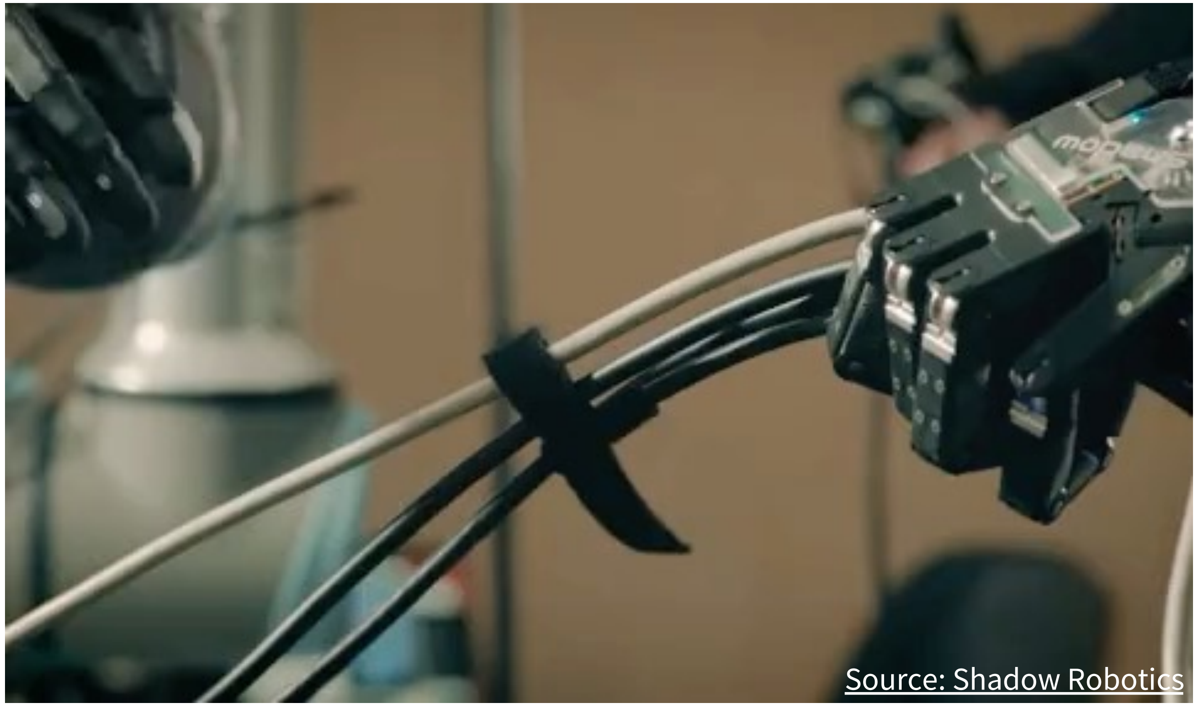
Optimal control in robotics



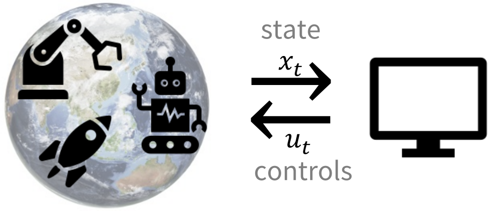
Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion
$M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking


Source: Shadow Robotics

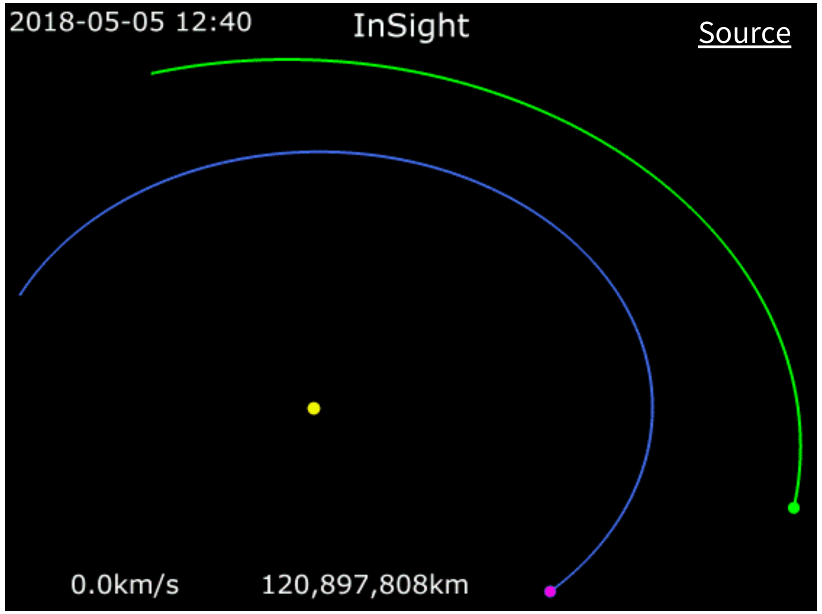
Optimal control in robotics



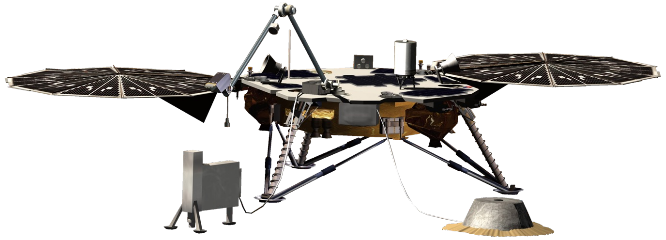
Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion $M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

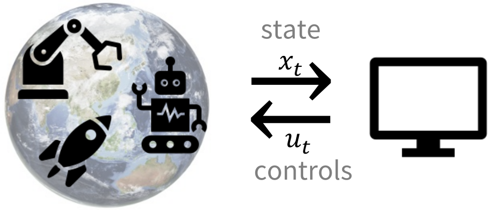
actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking



A Hohmann transfer orbit. See also:
Dynamical Systems, the Three-Body Problem and Space Mission Design. Koon et al., 1999.



Optimal control in robotics



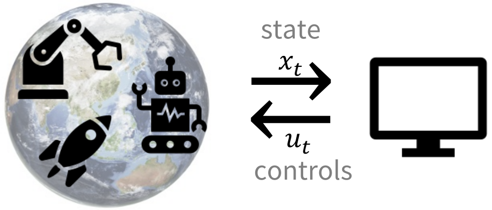
Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion $M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking

Convex Programming Approach to Powered Descent Guidance for Mars Landing. Açıkmeşe and Ploen, 2007.
Minimum-landing-error powered-descent guidance for Mars landing using convex optimization. Blackmore et al., 2010.
Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem. Açıkmeşe et al., 2013.

Optimal control in robotics

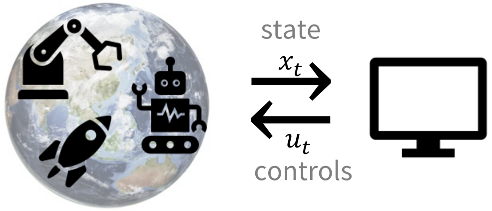


Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion $M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking

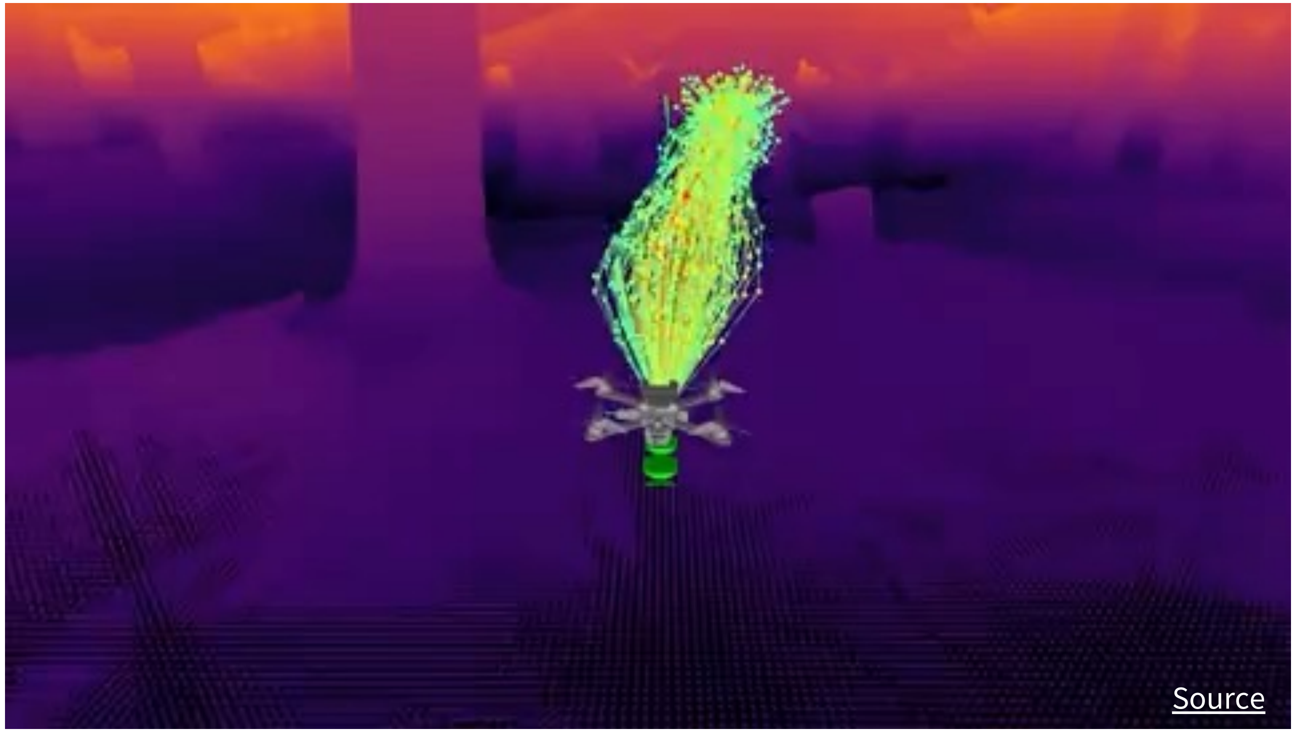
Optimal control in robotics



Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

the robotic system — e.g., the Newton-Euler equations of motion $M(q_t)\ddot{q}_t + n(q_t, \dot{q}_t) = \tau(q_t) + Bu_t$

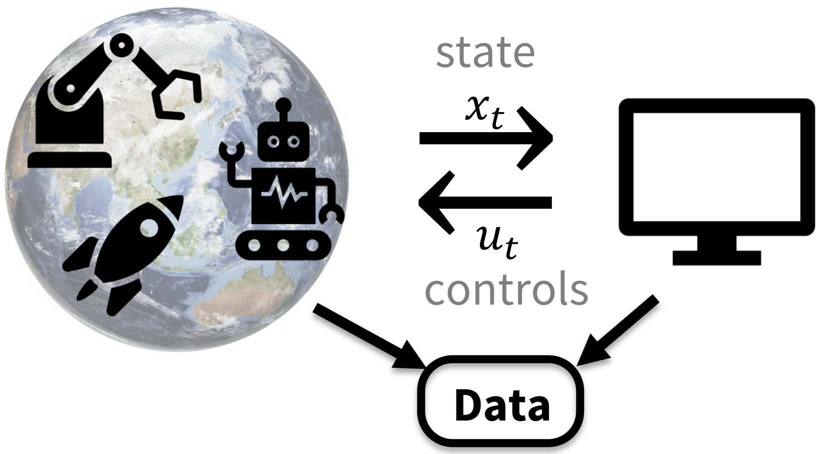
actuators — e.g., torques on the joints, thrusters, steering, acceleration, braking



📚 *Learning high-speed flight in the wild.* Loquercio et al., Science Robotics 2021.

Where does machine learning fit in?

Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model



Machine learning (ML) is about using data to 1) **create abstractions**, and 2) **make predictions**

[ML→Control] learn how to model and interact with the world from data (e.g., reinforcement learning)
! **[Control→ML]** interpret ML problems as control problems, solve with control methods

e.g., RL from human feedback for language models

This talk: machine learning ⇄ optimal control

1. Modeling and learning dynamics

2. Machine learning for optimal control
+ Reinforcement learning (policy, value, and model learning)
+ Differentiable control

3. Optimal control for machine learning
+ Perspective on diffusion and optimal transport models
+ RL-based updates for machine learning models (e.g., RLHF)

How do we (passively) model the world?

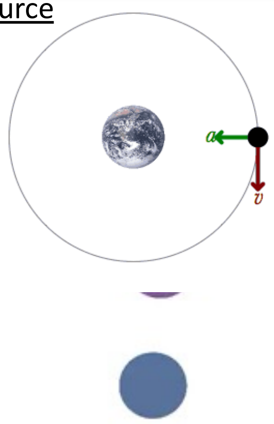
> *Mechanics is the paradise of the mathematical sciences, because by means of it one comes to the fruits of mathematics.*
> da Vinci (1459-1519), Notebooks, v. 1, ch. 20.

📚 Quote also given at the beginning of *Geometric Control of Mechanical Systems*, Bullo and Lewis, 2000.

1. Make observations

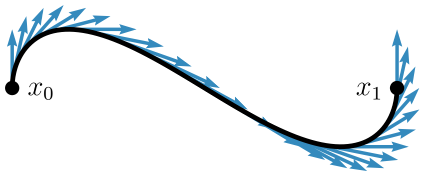
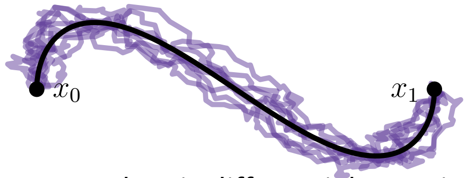
Source

Source



📚 *Learning Neural Event Functions for Ordinary Differential Equations.* Ricky T. Q. Chen, Brandon Amos, Maximilian Nickel, ICLR 2021.

2. Come up with a theory

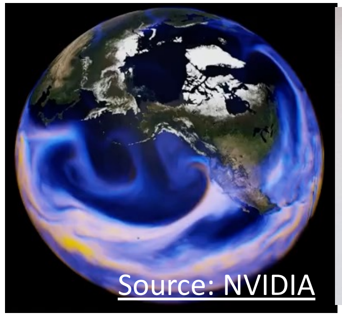
	Continuous time	Discrete time
Deterministic	$\dot{x}_t = f(x_t)$  e.g., differential equations (ODE/PDEs)	$x_{t+1} = f(x_t)$ e.g., Turing machines, games
Stochastic	$\mathrm{d}x_t = f(x_t)\mathrm{d}t + F(x_t)\mathrm{d}B_t$  e.g., stochastic differential equations	$x_{t+1} = f(x_t, w_t)$ $w_t \sim p(w)$ e.g., Markov chains

Machine learning way of learning dynamics

1. Collect data of the system 2. Throw neural networks at it

Deterministic

Continuous time

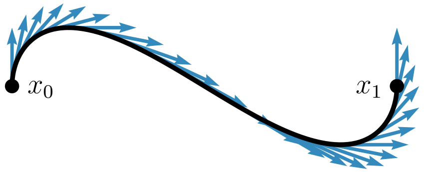


FourCastNet, Pathak et al., 2022.

Learning Neural Constitutive Laws. Ma et al., ICML 2023.

$$\dot{x}_t = f_\theta(x_t)$$

a neural network



e.g., Neural ODEs/PDEs, neural operators

Discrete time

$$x_{t+1} = f_\theta(x_t)$$

a neural network

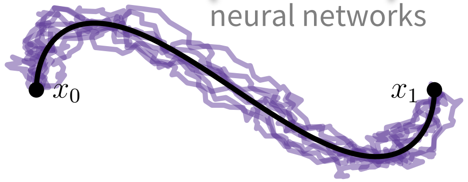
e.g., RNNs, LSTMs, Transformers
for language and other discrete-time sequential data

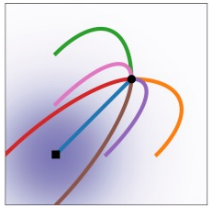
Stochastic

Continuous time

$$\mathrm{d}x_t = f_\theta(x_t)\mathrm{d}t + F_\theta(x_t)\mathrm{d}B_t$$

neural networks





e.g., Neural SDEs, diffusion models, flow matching

- *Deep unsupervised learning using nonequilibrium thermodynamics.* Sohl-Dickstein et al., ICML 2015.
- *Score-Based Generative Modeling through Stochastic Differential Equations*. Song et al., ICLR 2021.
- *Flow Matching for Generative Modeling*. Lipman et al., ICLR 2023.
- *Stochastic Interpolants*. Albergo et al., ICLR 2023.

Discrete time

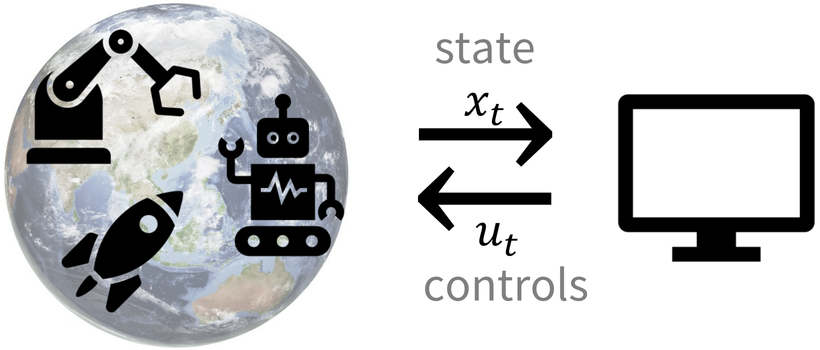
$$x_{t+1} = f_\theta(x_t, w_t)$$
$$w_t \sim p_\theta(w)$$

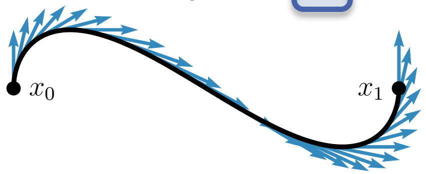
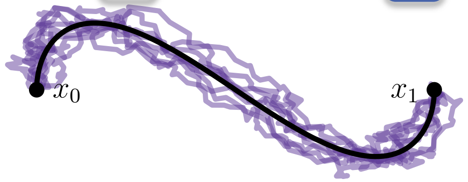
neural networks

e.g., RNNs with stochastic states

- *A Recurrent Latent Variable Model for Sequential Data.* Chung et al., NeurIPS 2015.
- *Sequential Neural Models with Stochastic Layers*. Fraccaro et al., NeurIPS 2016.

Adding interactions — controlled dynamics



	Continuous time	Discrete time
Deterministic	$\dot{x}_t = f(x_t, \boxed{u_t})$ 	$x_{t+1} = f(x_t, \boxed{u_t})$
Stochastic	$\mathrm{d}x_t = f(x_t, \boxed{u_t})\mathrm{d}t + F(x_t, \boxed{u_t})\mathrm{d}B_t$ 	$x_{t+1} = f(x_t, \boxed{u_t}, w_t)$ $w_t \sim p(w)$

This talk: machine learning ⇄ optimal control

1. Modeling and learning dynamics

2. Machine learning for optimal control
+ Reinforcement learning (policy, value, and model learning)
+ Differentiable control

3. Optimal control for machine learning
+ Perspective on diffusion and optimal transport models
+ RL-based updates for machine learning models (e.g., RLHF)

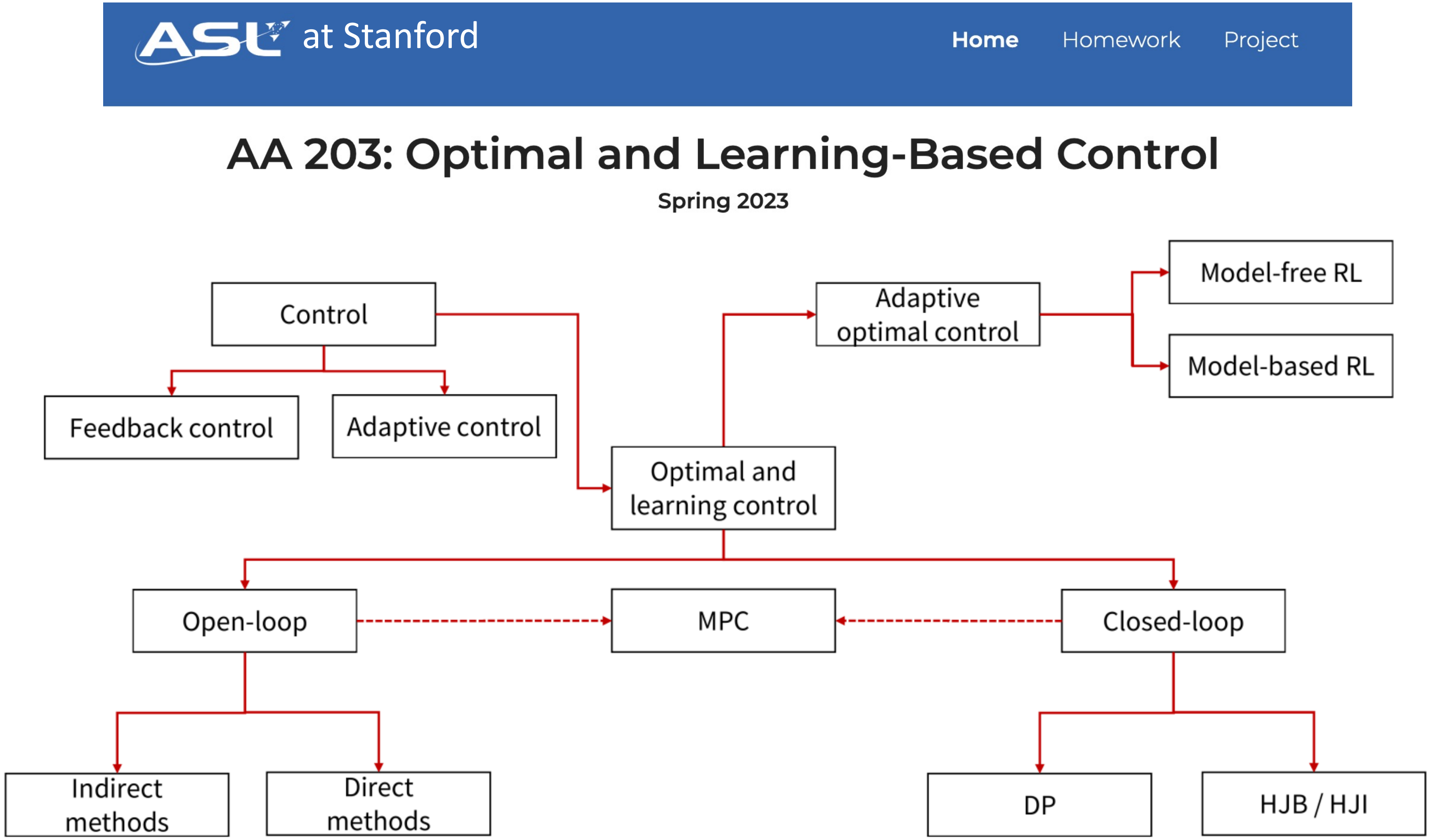
Formulating basic optimal control problems

*Can add many more constraints/variations

Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

	Continuous time	Discrete time
Deterministic	$\min_u C_H(x_H) + \int_{t=0}^{H} c_t(x_t, u_t)$ subject to $\dot{x} = f(x_t, u_t)$, x_0 given	$\min_{u_{0:H-1}} C_H(x_H) + \sum_{t=0}^{H-1} c_t(x_t, u_t)$ subject to $x_{t+1} = f(x_t, u_t)$, x_0 given
Stochastic	$\min_u \mathbb{E}\left\{C_H(x_H) + \int_{t=0}^{H} c_t(x_t, u_t)\right\}$ subject to $\mathrm{d}x_t = f(x_t, u_t)\mathrm{d}t + F(x_t, u_t)\mathrm{d}B_t$, x_0 given	$\min_{u_{0:H-1}} \mathbb{E}\left\{C_H(x_H) + \sum_{t=0}^{H-1} c_t(x_t, u_t)\right\}$ subject to $x_{t+1} = f(x_t, u_t, w_t)$, $w_t \sim p(w)$, x_0 given

Solving optimal control problems



Instructors

Spencer M. Richards

Daniele Gammelli

Analyzing controllers

ASL at Stanford

Home Homework Project

AA 203: Optimal and Learning-Based Control

Spring 2023

Traditional feedback control balances the following desiderata.

Stability	The system output does not diverge or "blow up".
Tracking	The system output converges to a desired reference.
Disturbance rejection	The system is insensitive to disturbances and noise.
Robustness	The controller performs well despite some model misspecification.

Instructors

Spencer M. Richards

Daniele Gammelli

Analyzing controllers

ASL at Stanford   **Home**   Homework   Project

AA 203: Optimal and Learning-Based Control

Spring 2023

This course also incorporates and focuses on the following objectives.

Performance The controller achieves an optimal trade-off between various metrics.

Constraints The controller does not cause the system to violate safety restrictions or inherent (e.g., physical) limitations.

Planning An appropriate reference trajectory is computed and given to the controller for tracking.

Learning The controller can adapt to an unknown or time-varying system.

Instructors

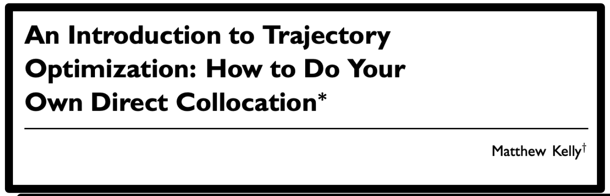
Spencer M. Richards

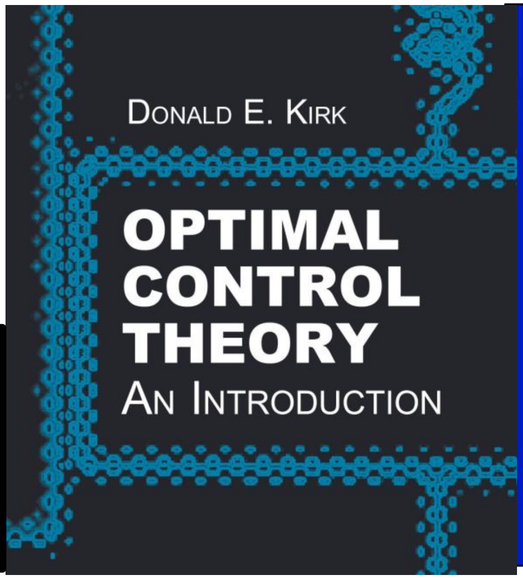
Daniele Gammelli

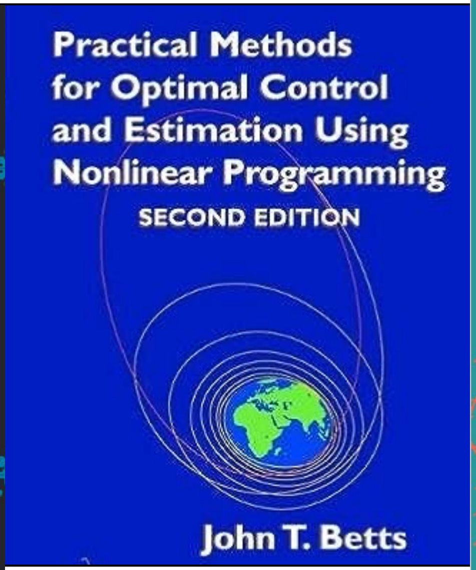
Other optimal control courses & books

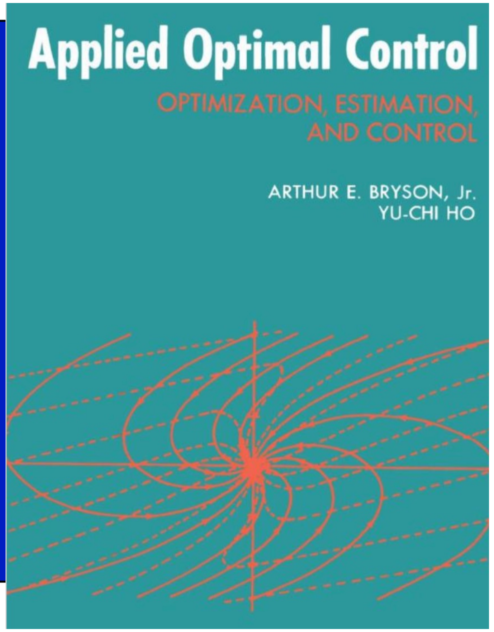
1. *Adaptive and nonlinear control,* Maxim Raginsky [UIUC ECE 517]
2. *SDEs in Optimization, Control, and Learning*, Maxim Raginsky [UIUC ECE 586]
3. *Robotic manipulation*, Russ Tedrake [MIT 6.4210]
4. *Underactuated robotics*, Russ Tedrake [MIT 6.8210]
5. *Optimal control*, Zac Manchester [CMU 16-745]
6. *Optimal and Learning-Based Control*, Spencer M. Richards and Daniele Gammelli [Stanford AA 203]

… and many others









Optimal control as a function

(no learning yet, all optimization)

$$x, c_t, C_h, f \longrightarrow \boxed{\begin{array}{l} \underset{u_{0:H-1}}{\operatorname{argmin}} \; C_H(x_H) + \sum_{t=0}^{H-1} c_t(x_t, u_t) \\ \quad \text{subject to} \;\; x_{t+1} = f(x_t, u_t) \end{array}} \longrightarrow \pi(x; c_t, C_H, f)$$

Where does machine learning fit in?

differentiable control $\frac{\partial}{\partial f}\pi(x)$

cost learning

value learning

$$x, c_t, C_h, f \longrightarrow \begin{array}{l} \operatorname*{argmin}_{u_{0:H-1}} \; C_H(x_H) + \sum_{t=0}^{H-1} c_t(x_t, u_t) \\ \text{subject to} \;\; x_{t+1} = f(x_t, u_t) \end{array} \longrightarrow \pi(x; c_t, C_H, f)$$

dynamics learning

policy learning
(amortized optimization)

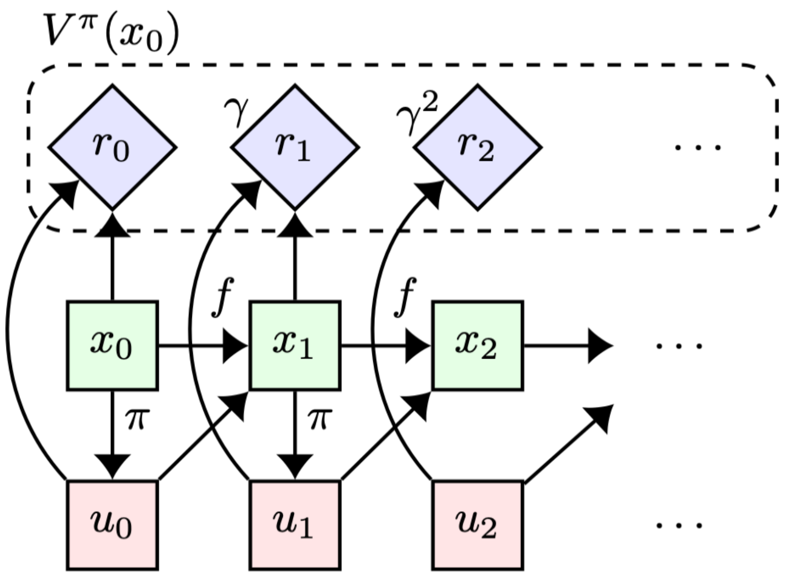
Reinforcement learning (/approximate dynamic programming)
usually value and policy learning, especially when f is unknown

Markov decision processes and RL

A **Markov decision process (MDP)** is a stochastic control process:

- **State** space $\mathcal{X}$, **control** space $\mathcal{U}$
- **Transition dynamics** $f(x, u)$
- **Reward** function $r(x, u)$

A **policy** $\pi: \mathcal{X} \rightarrow \mathcal{U}$ maps a state to a reward.
Goal: find the optimal policy to **maximize the value**

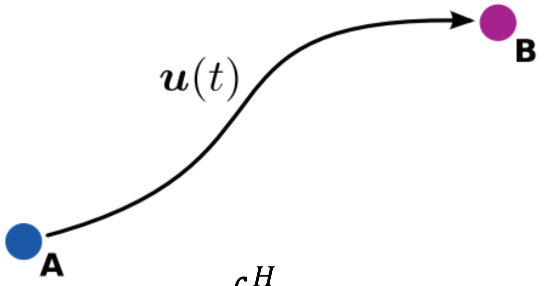


Comparing the MDP to optimal control

MDPs focus strongly on **infinite horizon/time-invariant policies** $\pi(x)$
- May not exist for all OC problems

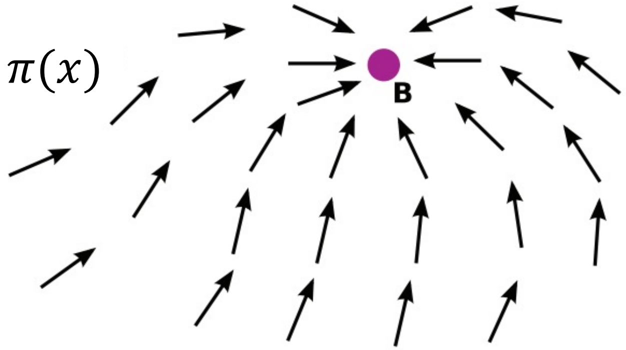
MDPs usually **more difficult** to add constraints/extra terms to
- E.g., goal conditioning or time-varying constraints

Optimal control (open-loop)



$$\min_{u} \int_{t=0}^{H} \|x_t - B\|^2$$
$$\text{subject to} \quad \dot{x} = f(x_t, u_t), \qquad x_0 = A$$

MDP policies (and closed loop control)

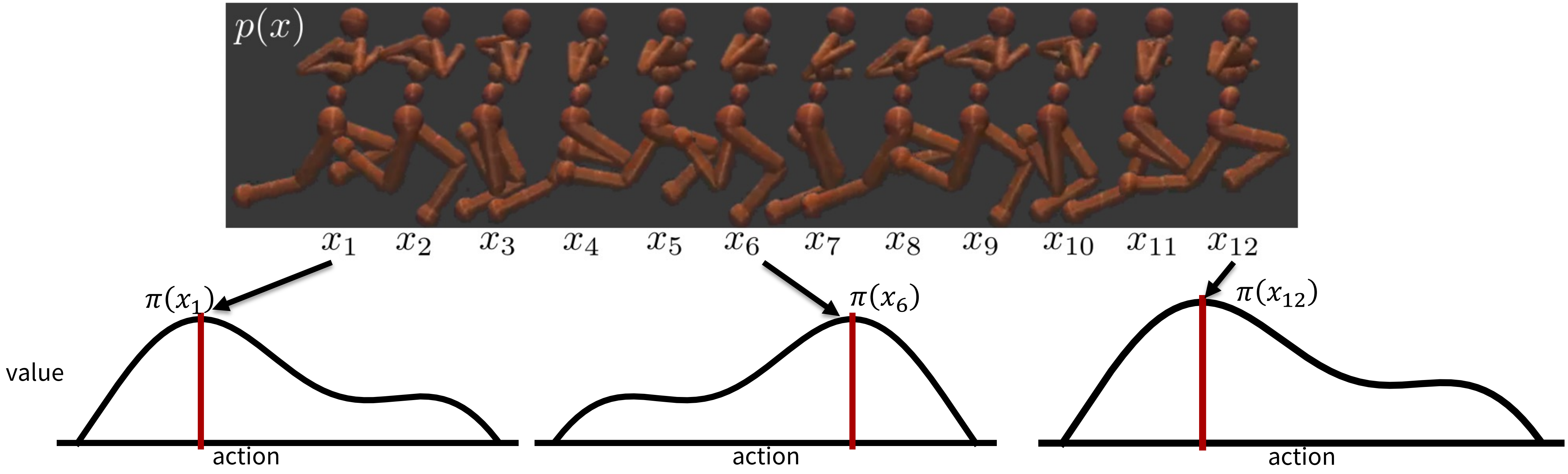


$$\min_{\theta} \mathbb{E}\, Q(x, \pi_\theta(x))$$

📚 *An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation.* Matthew Kelley, SIAM Review, 2017.

Repeatedly optimizing for computing a policy

$$\pi(x) = \underset{u}{\operatorname{argmax}}\{Q(x,u) \coloneqq r(x,u) + \mathbb{E}\,V(x')\}$$



Policy learning and amortized optimization

- *Continuous control with deep reinforcement learning.* Lillicrap et al., ICLR 2016.
- *Learning Continuous Control Policies by Stochastic Value Gradient.* Heess et al., NeurIPS 2015.
- *Tutorial on amortized optimization for learning to optimize over continuous domains*. Amos, Foundations and Trends in Machine Learning 2023.

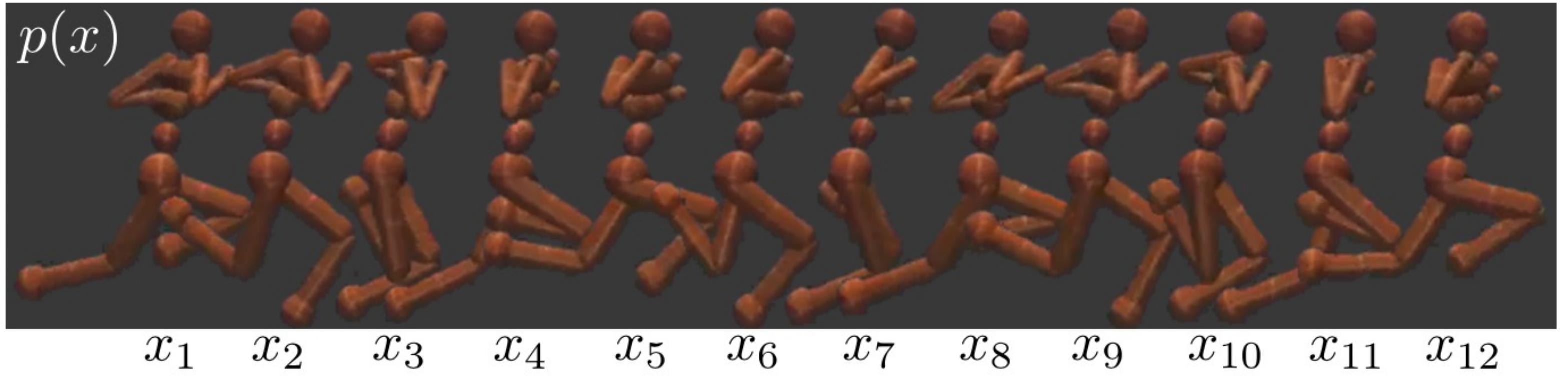
Independently solve

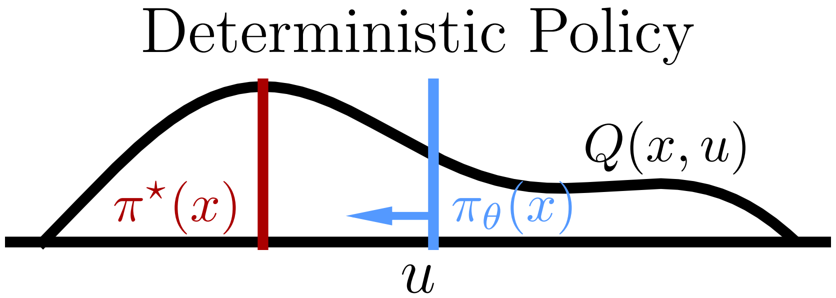
$$\pi(x) = \operatorname*{argmax}_u Q(x, u)$$

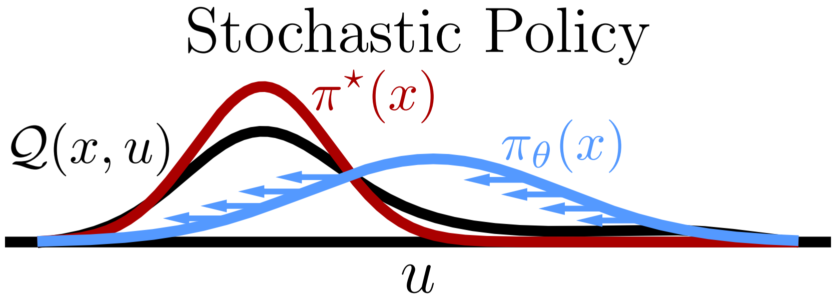
$\longrightarrow$

Learn a policy to predict the solution

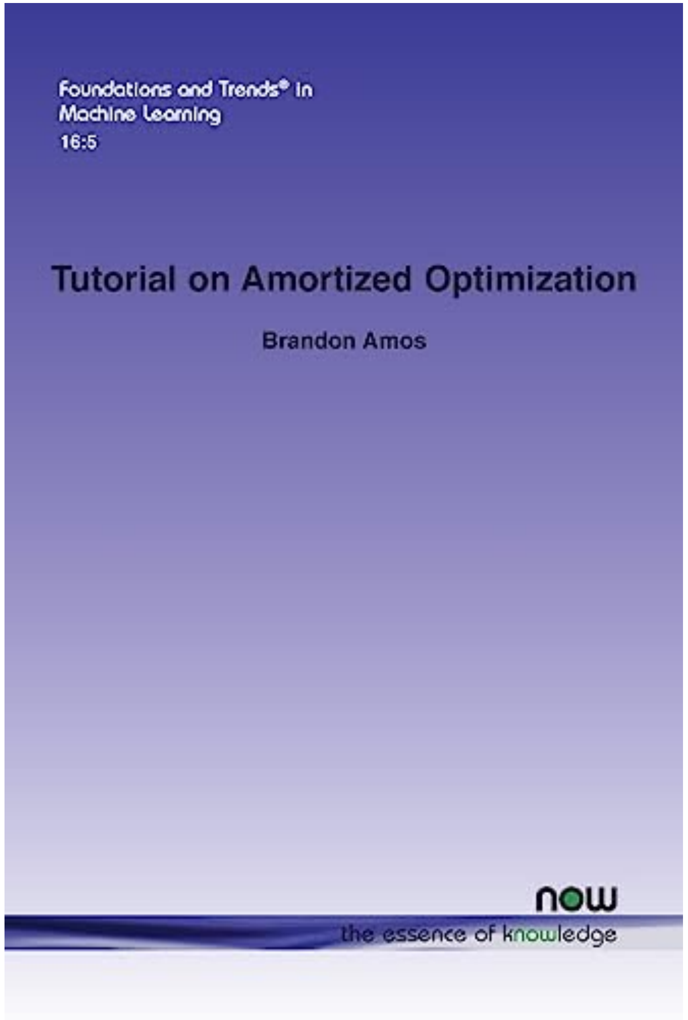
$$\operatorname*{argmax}_\theta \mathbb{E}_{p(x)} Q(x, \pi_\theta(x))$$







Further reading on amortized optimization



Reinforcement learning and **control** (actor-critic methods, SAC, DDPG, GPS, BC)

Variational inference (VAEs, semi-amortized VAEs)

Meta-learning (HyperNets, MAML)

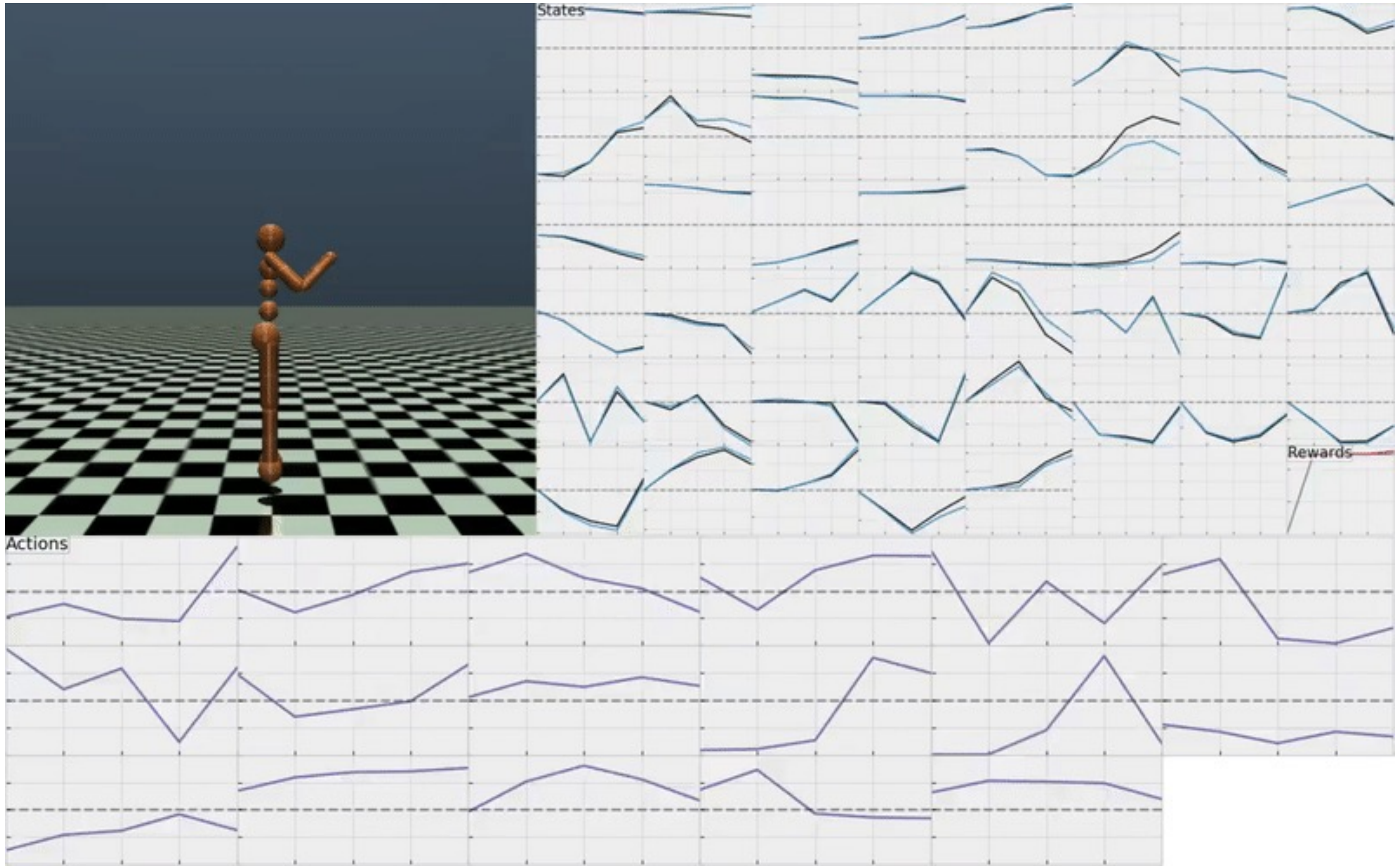
Sparse coding (PSD, LISTA)

Roots, fixed points, and convex optimization (NeuralDEQs, RLQP, NeuralSCS)

Optimal transport (slicing, conjugation, Meta Optimal Transport)

Model-based stochastic value gradients

📚 *Learning Continuous Control Policies by Stochastic Value Gradient.* Heess et al., NeurIPS 2015.
📚 *Imagined value gradients.* Byravan et al., CoRL 2020.
📚 *On the model-based stochastic value gradient for continuous reinforcement*. B. Amos et al., L4DC 2021.



✓ value learning   ✓ cost learning

$$\underset{u_{0:H-1}}{\operatorname{argmin}} \; C_H(x_H) + \sum_{t=0}^{H-1} c_t(x_t, u_t)$$

$$\text{subject to} \quad x_{t+1} = f(x_t, u_t)$$

✓ dynamics learning

$$\downarrow$$

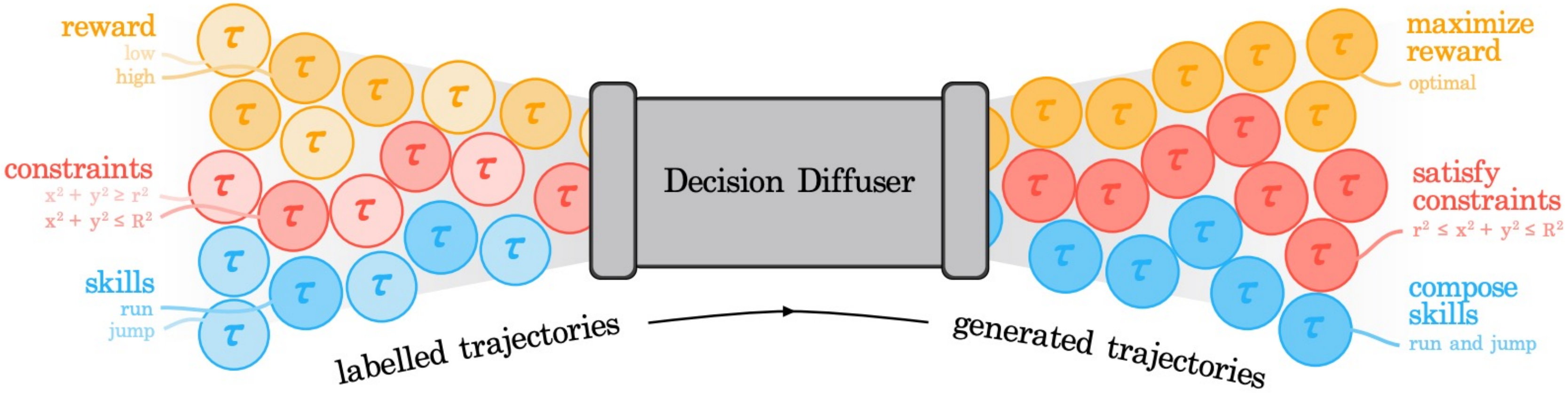
$$\pi(x; c_t, C_H, f)$$

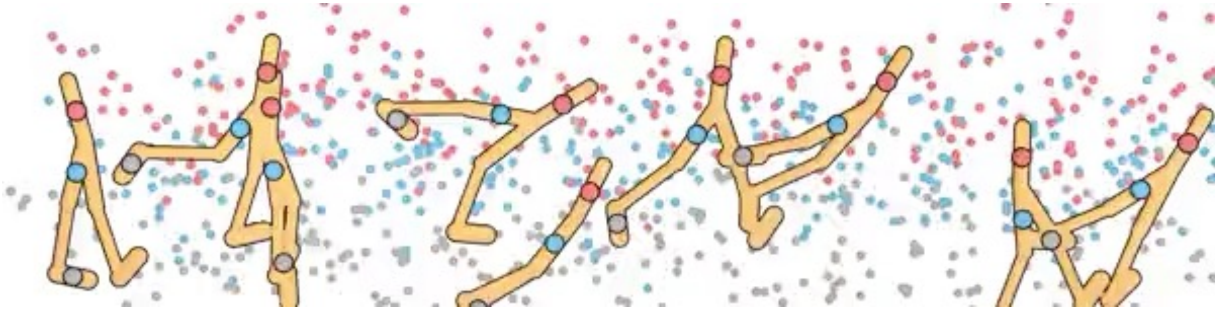
$\pi(x; c_t, C_H, f)$ policy learning

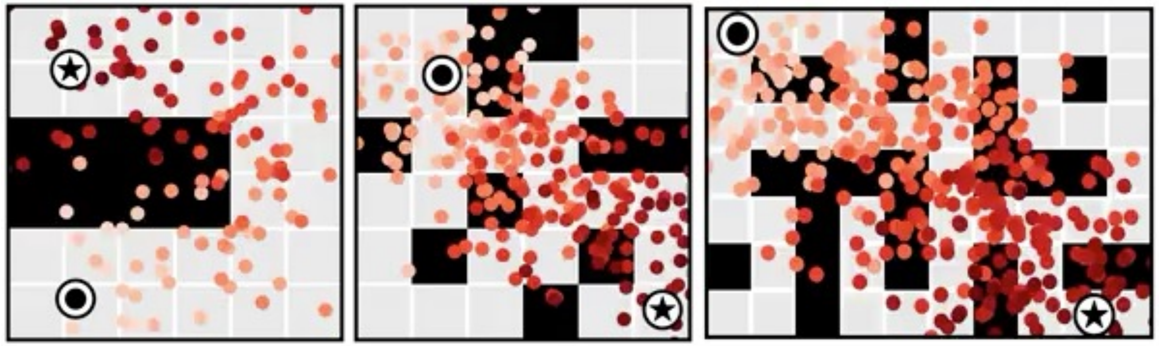
Diffusion for control and RL

📚 Planning with Diffusion for Flexible Behavior Synthesis. Janner*, Du*, et al., ICML 2022.
📚 Is Conditional Generative Modeling all you need for Decision-Making? Ajay*, Du* et al., ICML 2023.

Predicting 1) **dynamics**, 2) **rewards**, and 3) **optimal trajectories/policies**





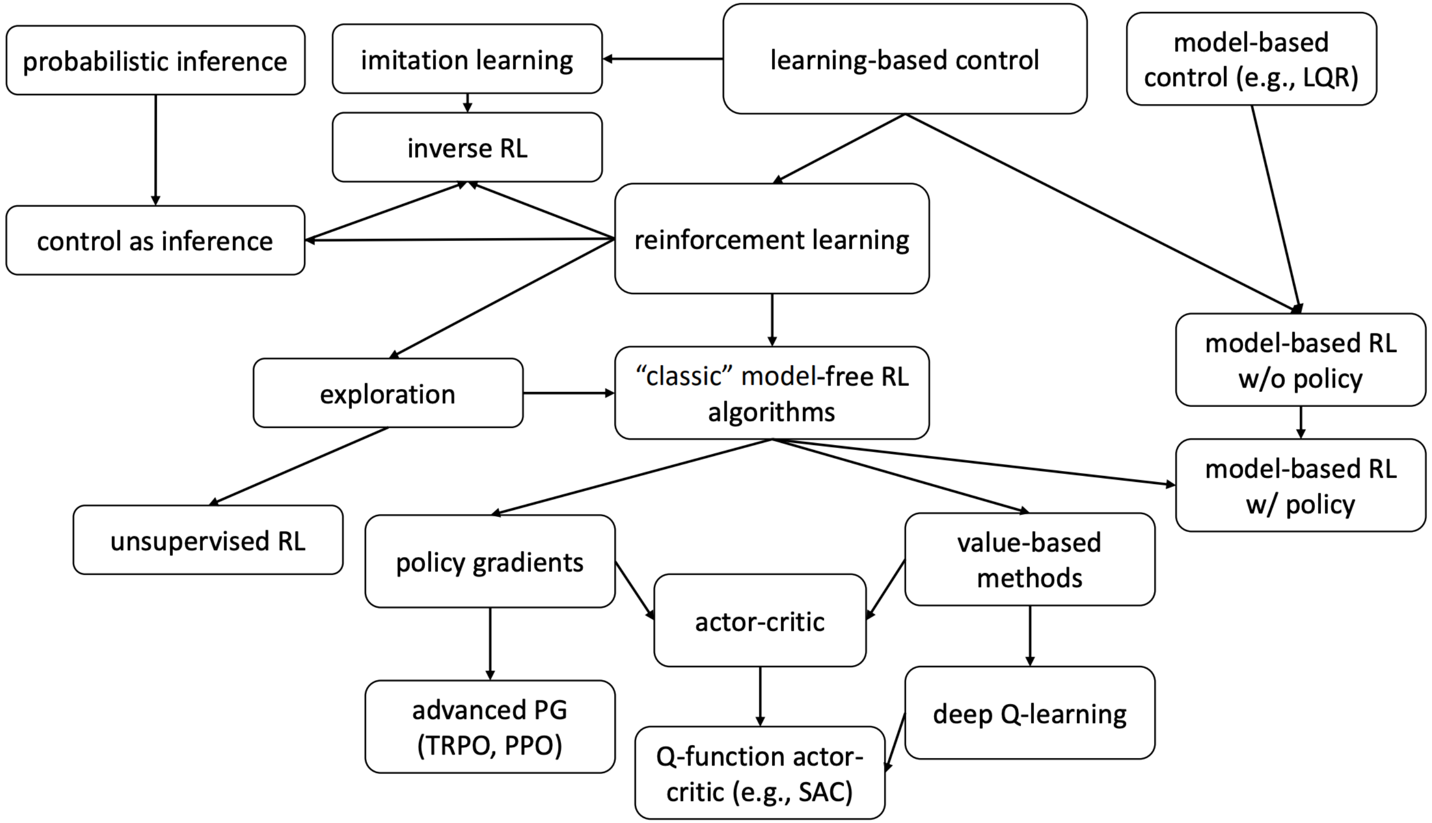


Reinforcement learning topics

CS 285 at UC Berkeley

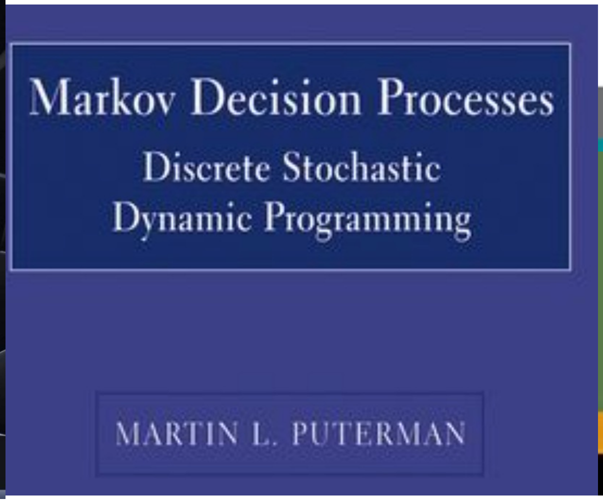
Deep Reinforcement Learning

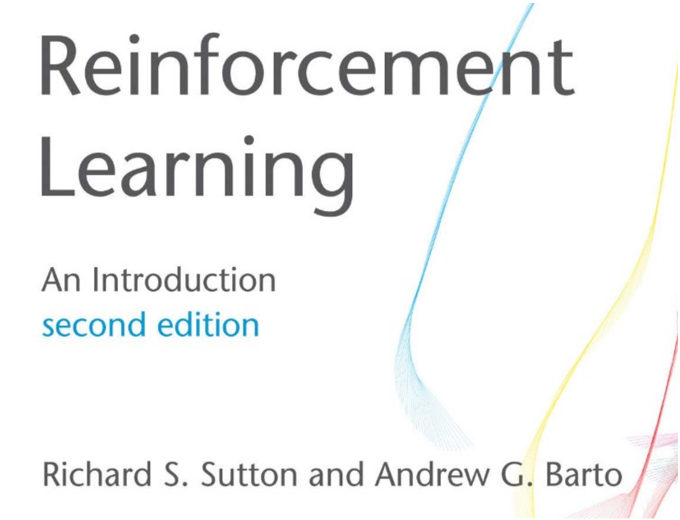
Instructor Sergey Levine
svlevine@eecs.berkeley.edu
Office Hours: After lecture



Control & reinforcement learning resources

1. *A Tour of Reinforcement Learning: The View from Continuous Control*. Benjamin Recht, 2019.
2. *Deep Reinforcement Learning.* Sergey Levine. [Berkeley CS 285]
3. *Reinforcement learning.* David Silver [UCL]
4. *Deep Reinforcement Learning.* Katerina Fragkiadaki [CMU 10-703]





This talk: machine learning ⇄ optimal control

1. Modeling and learning dynamics

2. Machine learning for optimal control
+ Reinforcement learning (policy, value, and model learning)
+ Differentiable control

3. Optimal control for machine learning
+ Perspective on diffusion and optimal transport models
+ RL-based updates for machine learning models (e.g., RLHF)

Controllers don't live in isolation

We can often measure the **downstream performance** induced by the controller

Idea: optimize (i.e., tune/learn) the parameters for a **downstream performance metric**

Controller-design loop is **not** a new idea and has been extensively used over the past century

---

θ controller **parameters**
(design space of cost and model)

solve control optimization problem

$$x^\star_{1:T}, u^\star_{1:T} \in \underset{x_{1:T}, u_{1:T}}{\operatorname{argmin}} \sum_t \underbrace{C_\theta(x_t, u_t)}_{\text{cost}} \;\text{s.t.}\; \underbrace{x_1 = x_{\text{init}}}_{\text{initial state}} \quad \underbrace{x_{t+1} = f_\theta(x_t, u_t)}_{\text{dynamics}} \quad \underbrace{u_t \in \mathcal{U}}_{\text{constraints}}$$

Full notation: $u^\star_{1:T}(x_{\text{init}}, \theta)$

$\pi_\theta(x) \stackrel{\text{def}}{=} u^\star_1(x, \theta)$ **execute** control policy on the system

$\mathcal{J}(\pi_\theta)$ **measure downstream performance** of the controller

Differentiate the controller!

We can often measure the **downstream performance** induced by the controller
Idea: optimize (i.e., tune/learn) the parameters for a **downstream performance metric** by **differentiating through the control optimization problem**

θ controller **parameters**
(design space of cost and model)

solve control optimization problem

$$x^\star_{1:T}, u^\star_{1:T} \in \operatorname*{argmin}_{x_{1:T}, u_{1:T}} \sum_t \underbrace{C_\theta(x_t, u_t)}_{\text{cost}} \;\text{s.t.}\; \underbrace{x_1 = x_{\text{init}}}_{\text{initial state}} \;\; \underbrace{x_{t+1} = f_\theta(x_t, u_t)}_{\text{dynamics}} \;\; \underbrace{u_t \in \mathcal{U}}_{\text{constraints}}$$

Full notation: $u^\star_{1:T}(x_{\text{init}}, \theta)$

$\pi_\theta(x) \stackrel{\text{def}}{=} u_1^\star(x, \theta)$ **execute** control policy on the system

$\mathcal{J}(\pi_\theta)$ **measure downstream performance** of the controller

differentiable control

$\nabla_\theta \mathcal{J}(\pi_\theta)$

or $\mathrm{D}_\theta(\pi_\theta(x))$

Derivatives in RL and control

The policy (or value) **gradient**

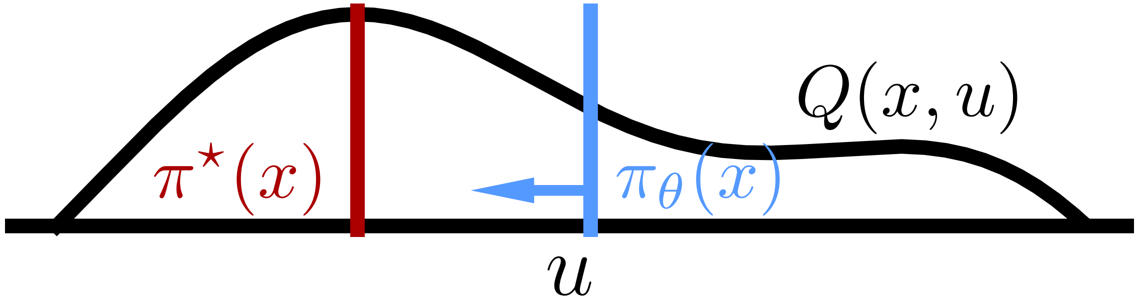
Derivative of **value** w.r.t. a **parameterized policy**:

$$\nabla_\theta \mathbb{E}_{x_t}\left[Q\big(x_t, \pi_\theta(x_t)\big)\right]$$

For policy learning via **amortized optimization**

Q-value can be model-based or model-free

Works for deterministic and stochastic policies

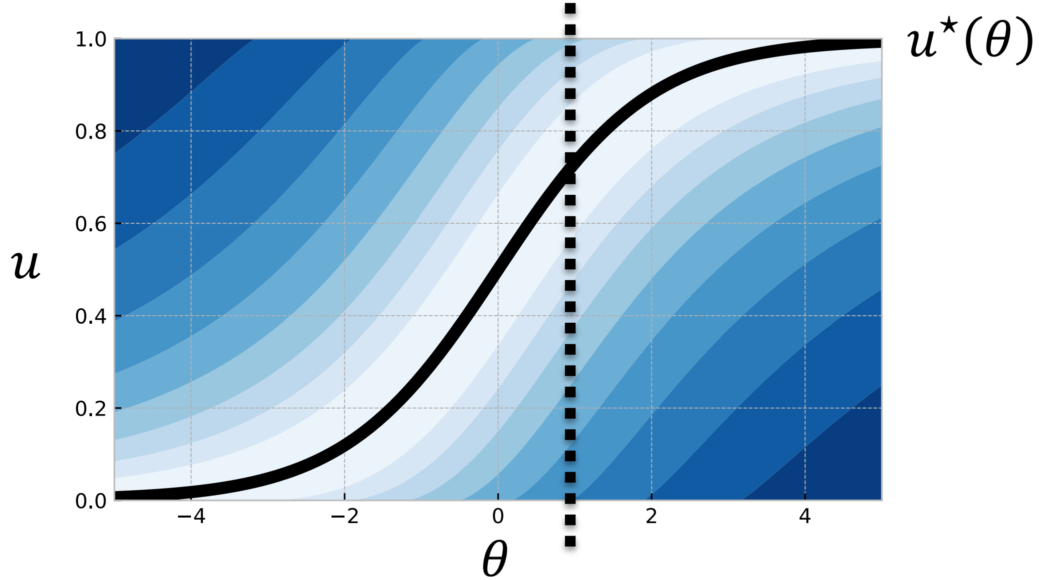


Differentiable control

Derivative of **actions** w.r.t. **controller parameters**:

$$\partial u^\star_{1:T}(\theta)/\partial\theta$$

Controller induces a **model-based policy**



Each vertical slice is a control problem

How to differentiate the controller?

Differentiable MPC. Amos et al., NeurIPS 2018; *Differentiable convex optimization layers.* Agrawal*, Amos*, et al., NeurIPS 2019; *The differentiable cross-entropy method.* ICML 2020; Learning Convex Optimization Control Policies, Agrawal* et al., L4DC 2020.

Unrolling or autograd

$$\hat{u}_\theta^0 \rightarrow \hat{u}_\theta^1 \rightarrow \cdots \rightarrow \hat{u}_\theta^K \rightarrow \hat{\pi}_\theta(x) \rightarrow \mathcal{J}$$

Idea: Implement controller, let **autodiff** do the rest
Like MAML's unrolled gradient descent

Ideal when **unconstrained** with a **short horizon**
Does **not** require a fixed-point or optimal solution
Instable and resource-intensive for large horizons

Can unroll algorithms **beyond gradient descent**
The differentiable cross-entropy method

Implicit differentiation

$$\mathrm{D}_\theta u^\star(\theta) = -\mathrm{D}_u g\big(\theta, u^\star(\theta)\big)^{-1} \mathrm{D}_\theta g\big(\theta, u^\star(\theta)\big)$$

Idea: Differentiate controller's optimality conditions

Agnostic of the control algorithm
Ill-defined if controller gives **suboptimal solution**
Memory and **compute** efficient: free in some cases

End-to-end model learning starting references

… among many others!

Using a Financial Training Criterion Rather than a Prediction Criterion*

Yoshua Bengio†

Gnu-RL: A Precocial Reinforcement Learning Solution for Building HVAC Control Using a Differentiable MPC Policy

Bingqing Chen
Carnegie Mellon University
Pittsburgh, PA, USA
bingqinc@andrew.cmu.edu

Zicheng Cai
Dell Technologies
Austin, TX, USA
zicheng.cai@dell.com

Mario Bergés
Carnegie Mellon University
Pittsburgh, PA, USA
mberges@andrew.cmu.edu

Smart "Predict, then Optimize"

Adam N. Elmachtoub
Department of Industrial Engineering and Operations Research and Data Science Institute, Columbia University, New York, NY 10027, adam@ieor.columbia.edu

Paul Grigas
Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, pgrigas@berkeley.edu

Task-based End-to-end Model Learning in Stochastic Optimization

Priya L. Donti
Dept. of Computer Science
Dept. of Engr. & Public Policy
Carnegie Mellon University
Pittsburgh, PA 15213
pdonti@cs.cmu.edu

Brandon Amos
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bamos@cs.cmu.edu

J. Zico Kolter
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
zkolter@cs.cmu.edu

Learning Convex Optimization Control Policies

Akshay Agrawal AKSHAYKA@CS.STANFORD.EDU
Shane Barratt SBARRATT@STANFORD.EDU
Stephen Boyd BOYD@STANFORD.EDU
450 Serra Mall, Stanford, CA, 94305

Bartolomeo Stellato* STELLATO@MIT.EDU
77 Massachusetts Ave, Cambridge, MA, 02139

This talk: machine learning ⇄ optimal control

1. Modeling and learning dynamics

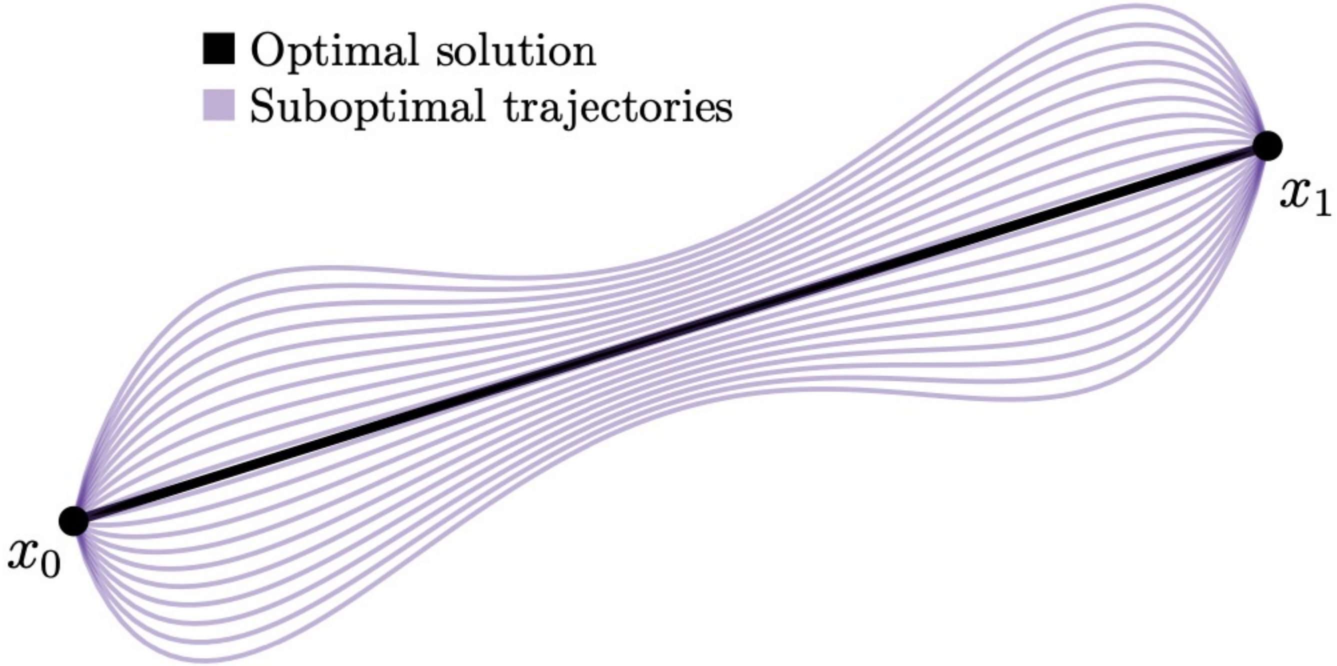
2. Machine learning for optimal control
+ Reinforcement learning (policy, value, and model learning)
+ Differentiable control

3. Optimal control for machine learning
+ Perspective on diffusion and optimal transport models
+ RL-based updates for machine learning models (e.g., RLHF)

Euclidean distance as optimal control

📚 *On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint.* Chen et al., 2016. (Section II.B)

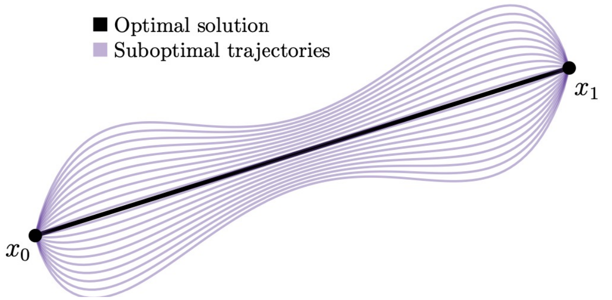
$$\|x_0 - x_1\|^2 = \min_{u_t, x_t} \int_0^1 \|u_t\|^2 \mathrm{d}t \quad \text{subject to} \quad \dot{x}_t = u_t, \quad x_0, x_1 \text{ given}$$



Optimal control for optimal transport

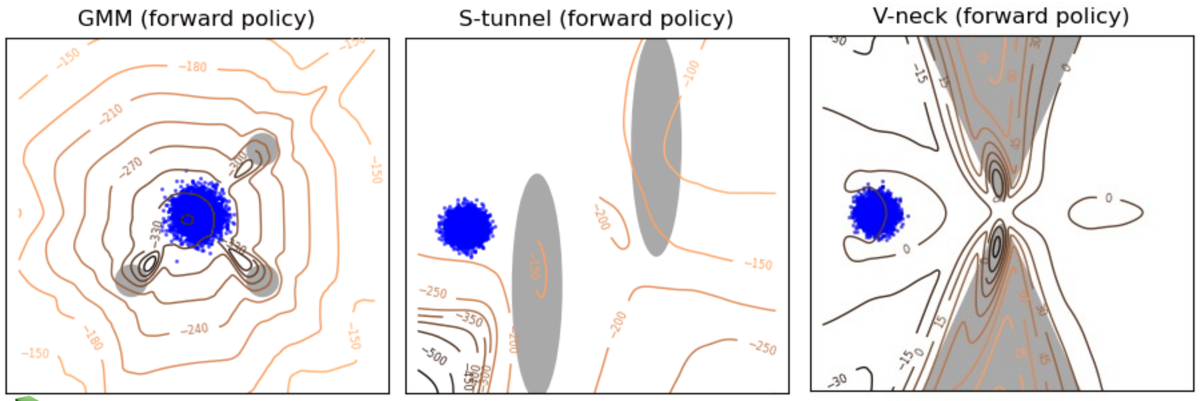
On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint. Chen et al., 2016

Euclidean path
optimal control between two points

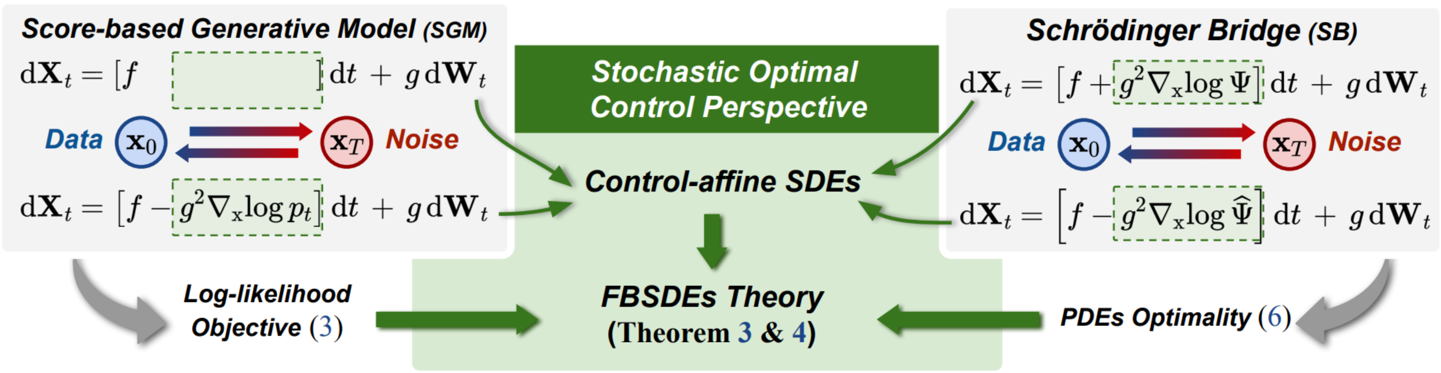


$$\|x_0 - x_1\|^2 = \min_{u_t, x_t} \int_0^1 \|u_t\|^2 \mathrm{d}t \quad \text{subject to} \quad \dot{x}_t = u_t, \quad x_0, x_1 \text{ given}$$

(Entropic) optimal transport path
stochastic optimal control between two measures



Deep Generalized Schrödinger Bridge. Liu et al., NeurIPS 2022.



Likelihood Training of Schrödinger Bridge using Forward-Backward SDEs Theory. Chen*, Liu*, and Theodorou, ICLR 2022.

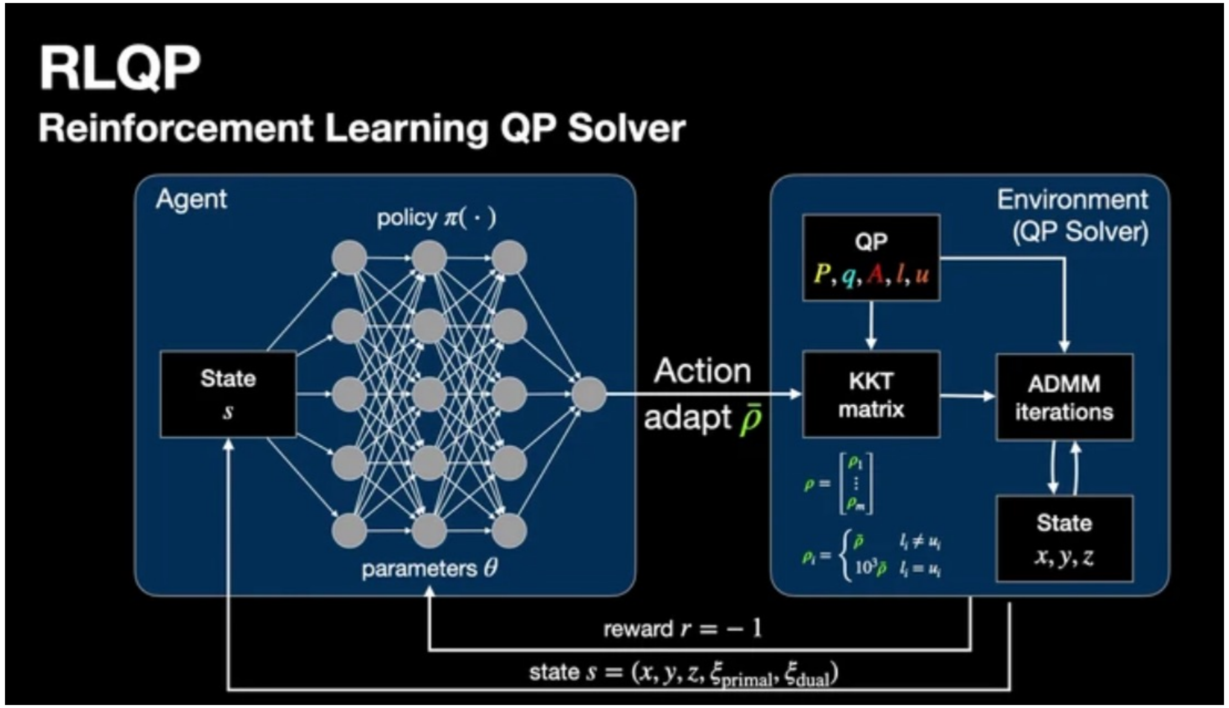
Optimal control+RL for solving QPs

📚*Accelerating quadratic optimization with reinforcement learning.* Ichnowski et al., NeurIPS 2021.

Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

Solving a quadratic program

Solver parameters during training



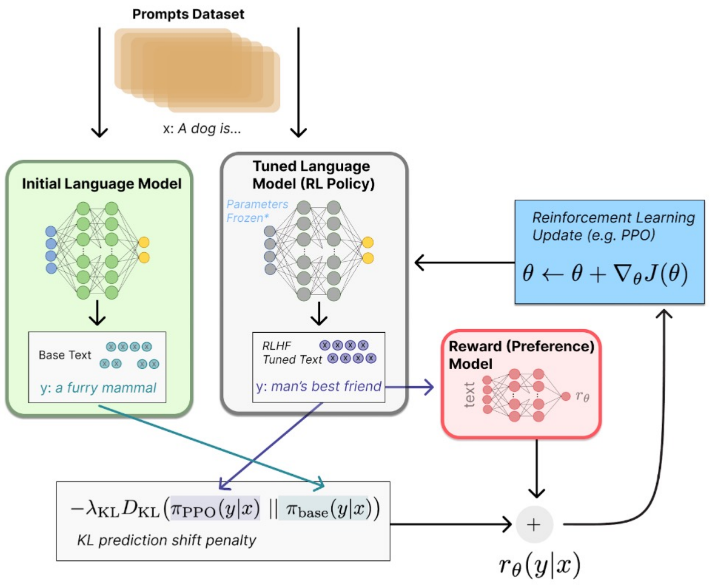
Reinforcement learning from human feedback

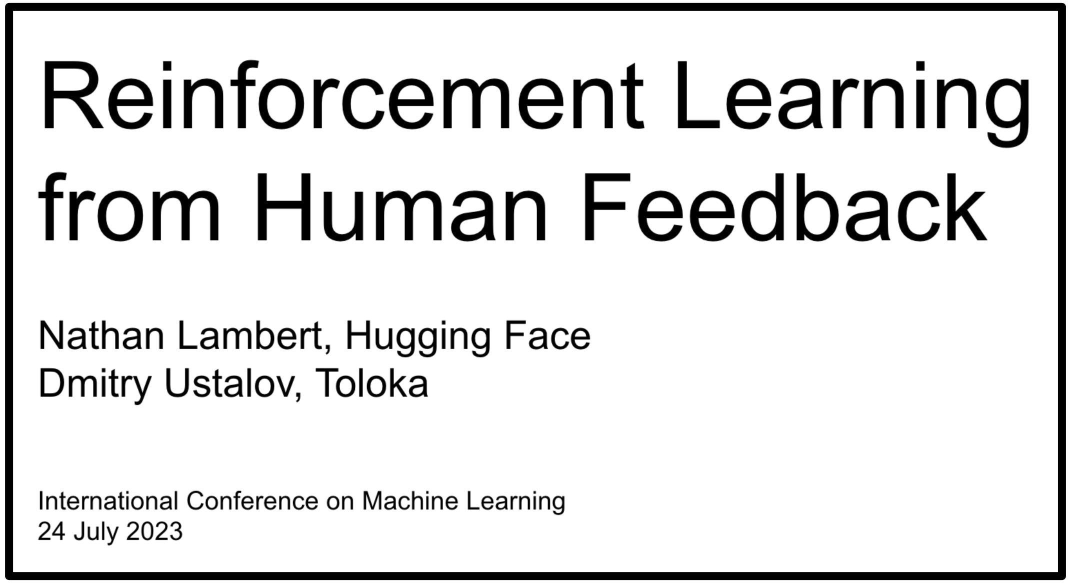
📚 *Deep reinforcement learning from human preferences.* Christiano et al., NeurIPS 2017; *Fine-tuning language models from human preferences.* Ziegler et al., 2019. *Learning to summarize with human feedback*. Stiennon et al., NeurIPS 2020. *Training language models to follow instructions with human feedback.* Ouyang et al., NeurIPS 2022.

Optimal control is about 1) **modeling** part of the world and 2) **interacting** with that model

language preferences

a language model



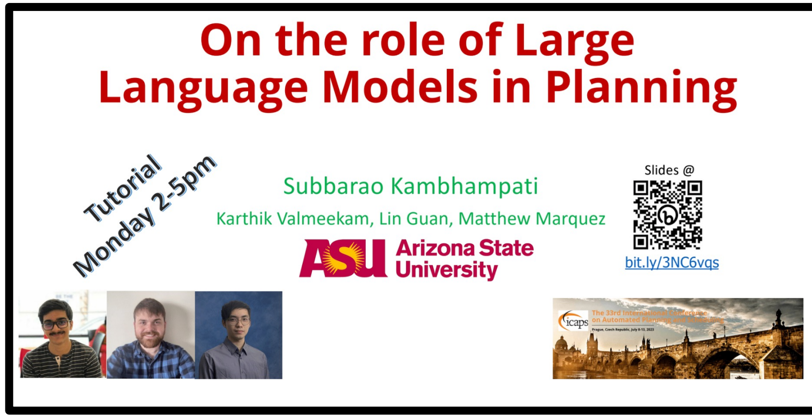

Reinforcement Learning from Human Feedback

Nathan Lambert, Hugging Face
Dmitry Ustalov, Toloka

International Conference on Machine Learning
24 July 2023

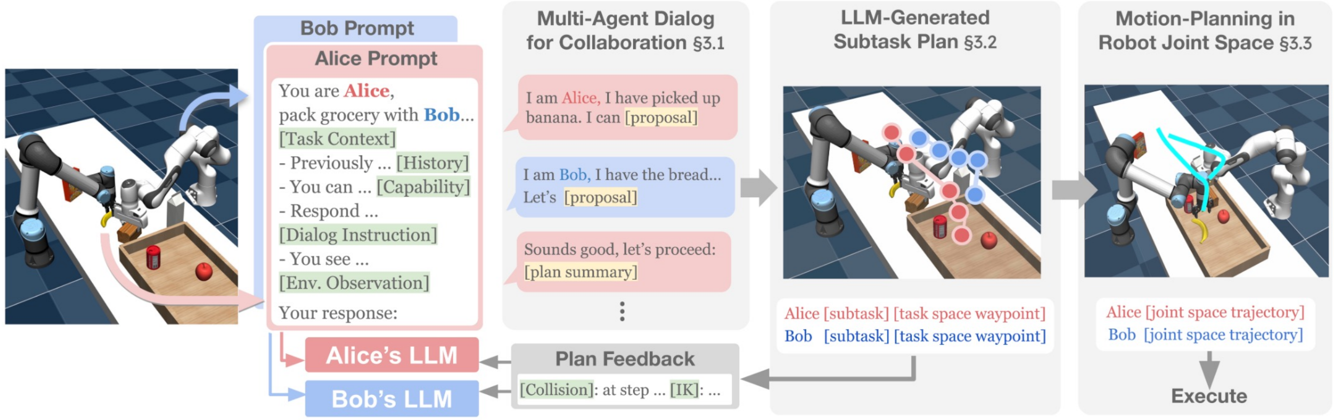
Ongoing and future directions in ML+control

Modeling and controlling isolated systems is relatively well-understood
Challenge: understanding and interacting with other parts of the world

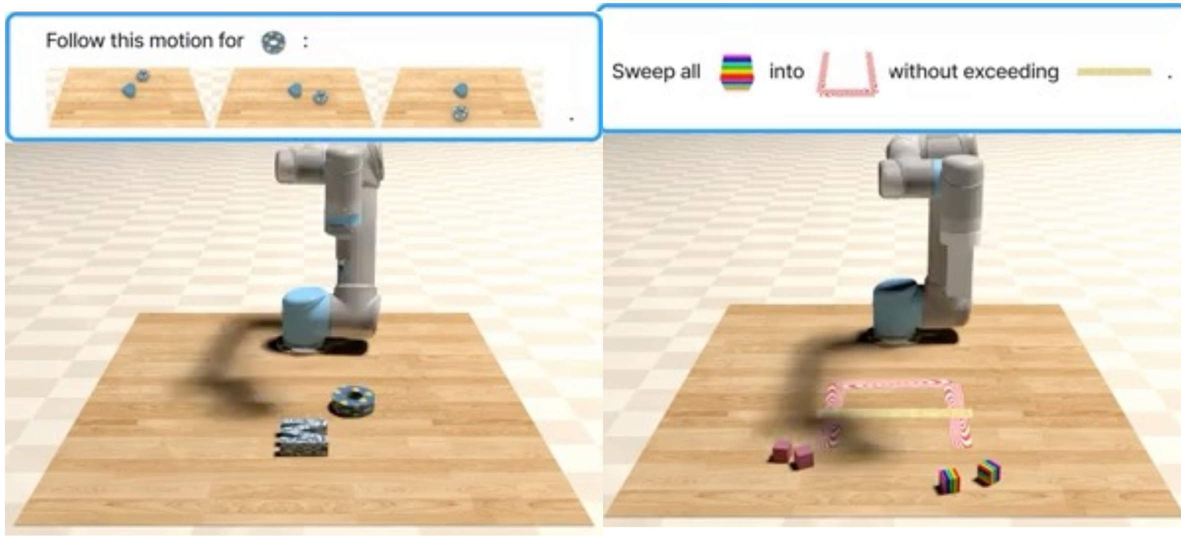


Use language to specify tasks, goals, actions

- 📚 *VIMA: General Robot Manipulation with Multimodal Prompts.* Jiang et al., ICML 2023
- 📚 *RoCo: Dialectic Multi-Robot Collaboration with Large Language Models.* Zhao et al., 2023.



📚 *RoCo: Dialectic Multi-Robot Collaboration with Large Language Models.* Zhao et al., 2023.



📚 *VIMA: General Robot Manipulation with Multimodal Prompts.* Jiang et al., ICML 2023

Ongoing and future directions in ML+control

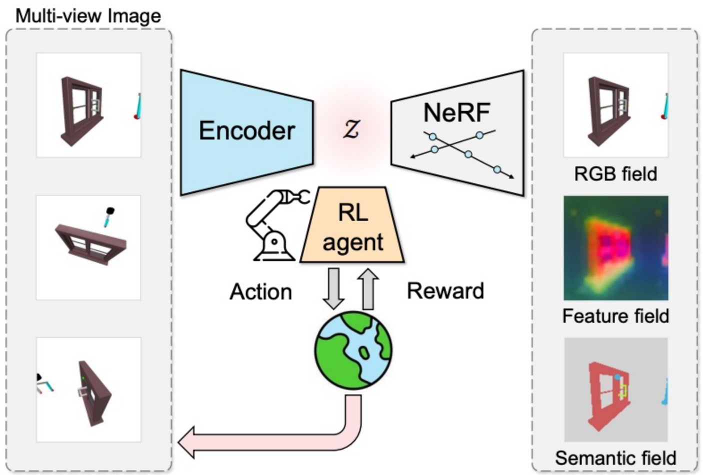
Modeling and controlling isolated systems is relatively well-understood
Challenge: understanding and interacting with other parts of the world

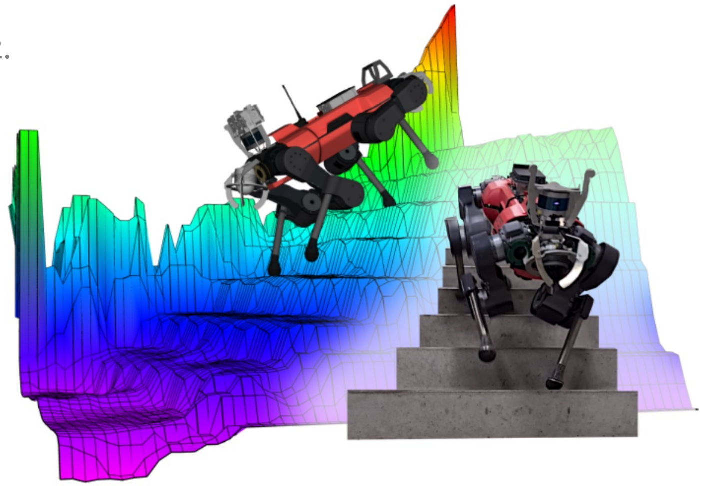
Use language to specify tasks, goals, actions

- 📚 *VIMA: General Robot Manipulation with Multimodal Prompts.* Jiang et al., ICML 2023
- 📚 *RoCo: Dialectic Multi-Robot Collaboration with Large Language Models.* Zhao et al., 2023.

Incorporating non-trivial **geometries** and **terrains**

- 📚 *TAMOLS: Terrain-Aware Motion Optimization for Legged Systems.* Jenelten et al., Transactions on Robotics, 2022.
- 📚 *Vision-only robot navigation in a neural radiance world.* Adamkiewicz et al., IEEE Robotics and Automation Letters 2022.
- 📚 *SNeRL: Semantic-aware Neural Radiance Fields for Reinforcement Learning*. Shim*, Lee*, and Kim, ICML 2023.





Ongoing and future directions in ML+control

Modeling and controlling isolated systems is relatively well-understood
Challenge: understanding and interacting with other parts of the world

Use language to specify tasks, goals, actions

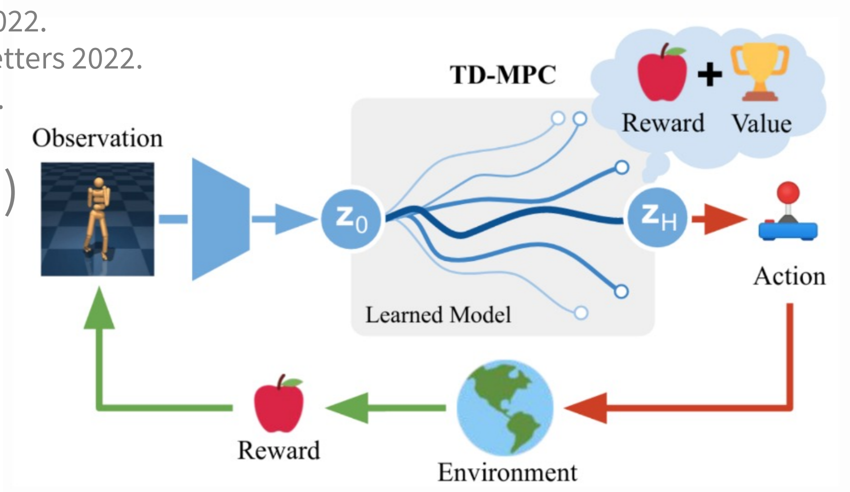
- 📚 *VIMA: General Robot Manipulation with Multimodal Prompts.* Jiang et al., ICML 2023
- 📚 *RoCo: Dialectic Multi-Robot Collaboration with Large Language Models.* Zhao et al., 2023.

Incorporating non-trivial **geometries** and **terrains**

- 📚 *TAMOLS: Terrain-Aware Motion Optimization for Legged Systems.* Jenelten et al., Transactions on Robotics, 2022.
- 📚 *Vision-only robot navigation in a neural radiance world.* Adamkiewicz et al., IEEE Robotics and Automation Letters 2022.
- 📚 *SNeRL: Semantic-aware Neural Radiance Fields for Reinforcement Learning*. Shim*, Lee*, and Kim, ICML 2023.

Learning **control-aware** dynamics (task-oriented, decision-focused)

- 📚 *Objective Mismatch in Model-based Reinforcement Learning*. Lambert et al., L4DC 2020.
- 📚 *Temporal Difference Learning for Model Predictive Control.* Hansen et al., ICML 2022.
- 📚 *Learning Control-Oriented Dynamical Structure from Data.* Richards et al., ICML 2023.



Ongoing and future directions in ML+control

Modeling and controlling isolated systems is relatively well-understood
Challenge: understanding and interacting with other parts of the world

Use language to specify tasks, goals, actions

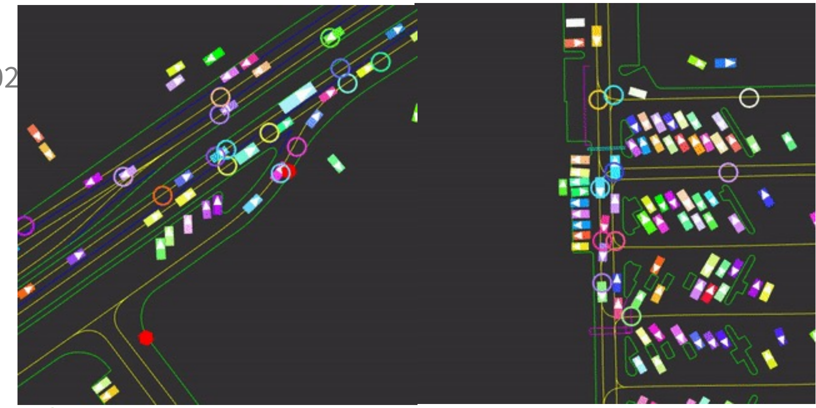
- *VIMA: General Robot Manipulation with Multimodal Prompts.* Jiang et al., ICML 2023
- *RoCo: Dialectic Multi-Robot Collaboration with Large Language Models.* Zhao et al., 2023.

Incorporating non-trivial **geometries** and **terrains**

- *TAMOLS: Terrain-Aware Motion Optimization for Legged Systems.* Jenelten et al., Transactions on Robotics, 2022.
- *Vision-only robot navigation in a neural radiance world.* Adamkiewicz et al., IEEE Robotics and Automation Letters 202
- *SNeRL: Semantic-aware Neural Radiance Fields for Reinforcement Learning*. Shim*, Lee*, and Kim, ICML 2023.

Learning **control-aware** dynamics (task-oriented, decision-focused)

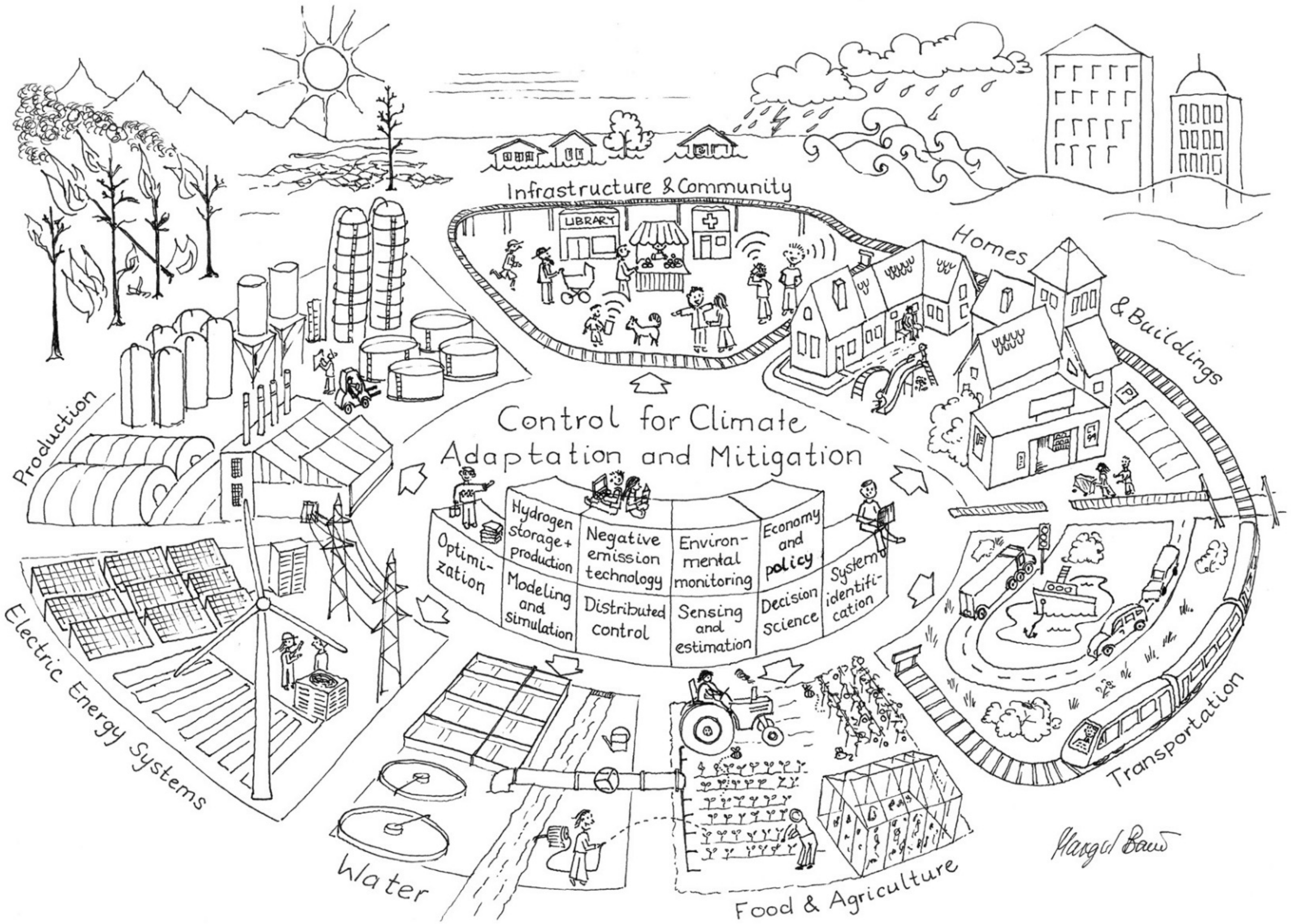
- *Objective Mismatch in Model-based Reinforcement Learning*. Lambert et al., L4DC 2020.
- *Temporal Difference Learning for Model Predictive Control.* Hansen et al., ICML 2022.
- *Learning Control-Oriented Dynamical Structure from Data.* Richards et al., ICML 2023.



Nocturne: a driving benchmark for multi-agent learning. Vinitsky et al., NeurIPS Datasets and Benchmarks 2022

Multi-agent control and **game theory**

Optimal control and societal challenges



This talk: machine learning ⇄ optimal control

1. Modeling and learning dynamics

2. Machine learning for optimal control
+ Reinforcement learning (policy, value, and model learning)
+ Differentiable control

3. Optimal control for machine learning
+ Perspective on diffusion and optimal transport models
+ RL-based updates for machine learning models (e.g., RLHF)

On optimal control and machine learning

Tutorial at the Workshop on New Frontiers in Learning, Control, and Dynamical Systems

Brandon Amos • Meta AI (FAIR) NYC

slides

github.com/bamos/presentations

International Conference On Machine Learning