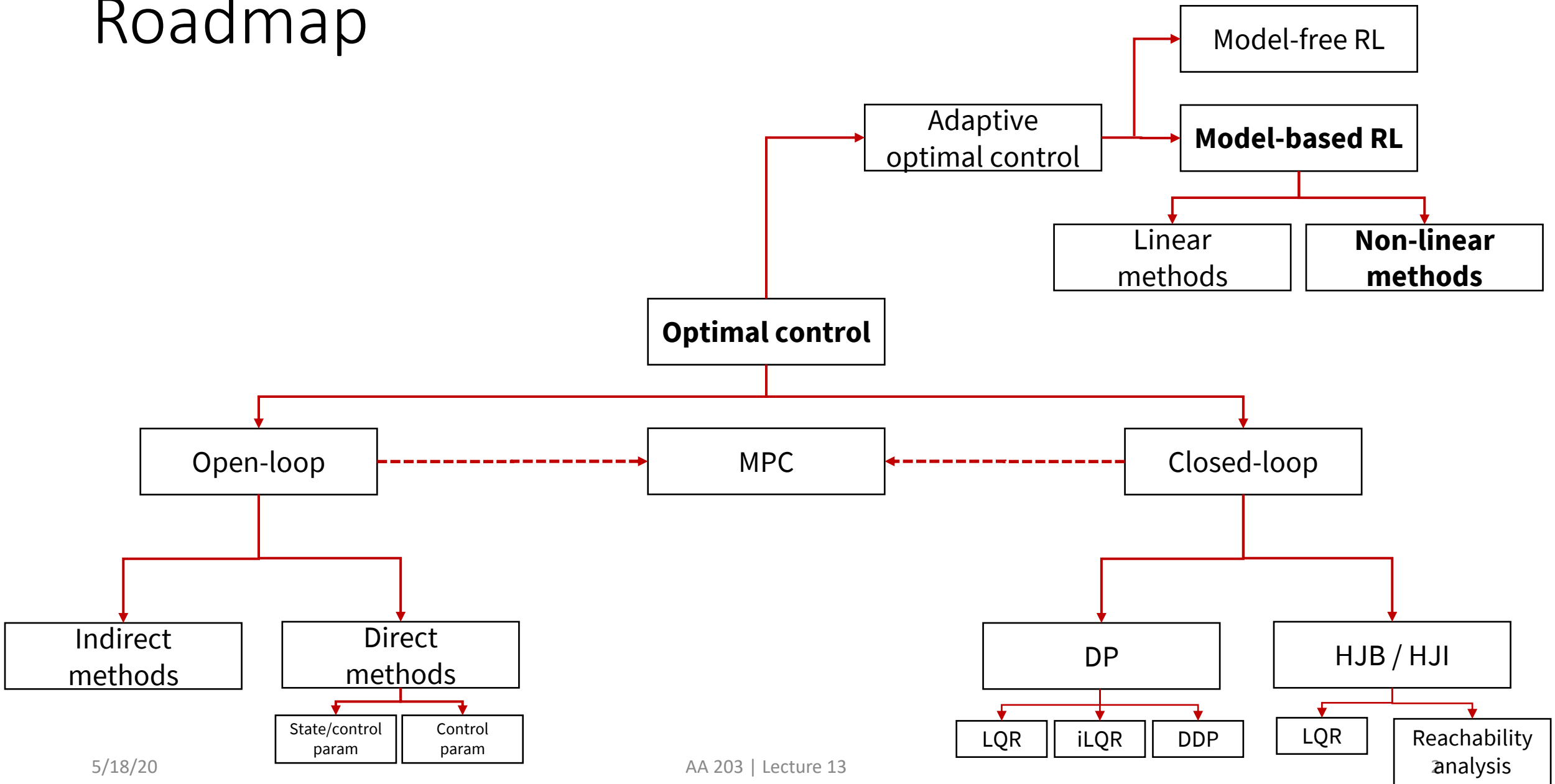
# AA203
# Optimal and Learning-based Control

Model-based reinforcement learning

Stanford University

ASL
Autonomous Systems Lab

# Roadmap


Model-free RL
Adaptive optimal control
Model-based RL
Linear methods
Non-linear methods
Optimal control
Open-loop
MPC
Closed-loop
Indirect methods
Direct methods
State/control param
Control param
DP
HJB / HJI
LQR
iLQR
DDP
LQR
Reachability analysis

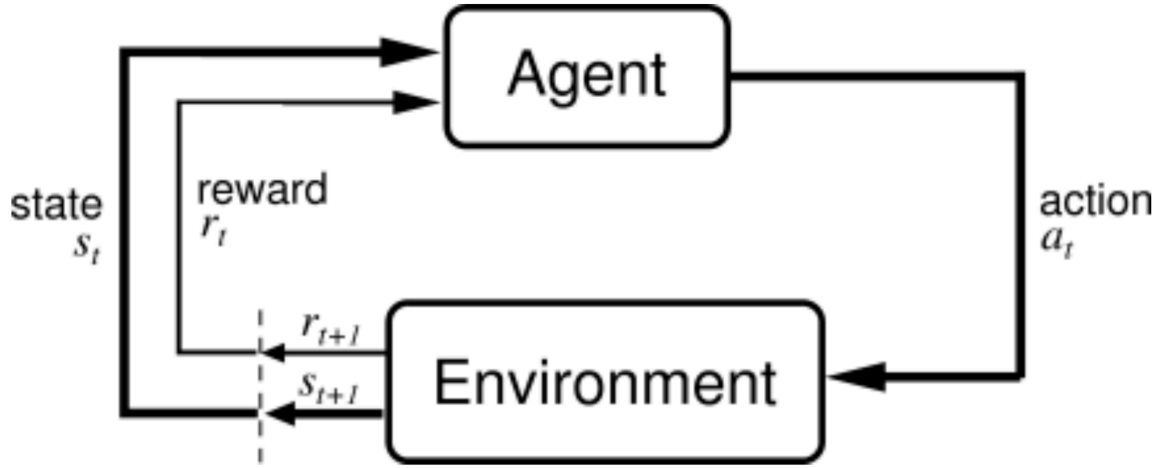
# Agenda

- Reviewing the reinforcement learning problem statement
- Tabular model-based RL
- Continuous model-based RL

- Readings:
  - M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, *Bayesian Reinforcement Learning: A Survey,* Foundations and Trends in ML, 2016.
  - R. Sutton and A. Barto. *Reinforcement Learning: An Introduction,* 2018.
  - M. Kochenderfer. *Decision Making Under Uncertainty,* 2015.
  - K. Chua, R. Calandra, R. McAllister, and S. Levine. *Deep reinforcement learning in a handful of trials using probabilistic dynamics models,* NeurIPS, 2018.

# Reinforcement learning assumptions and information patterns

- Previously:
  - Intro model-free RL: Q-learning, Policy gradient
    - Multiple episodes, interleave data collection and policy improvement
    - Tabular (exact) or continuous dynamics (approximate)
  - System Identification
    - Batch, offline data collection
    - Primarily linear dynamics
  - Adaptive Control
    - Intra-episode/online adaptation
    - Primarily linear dynamics



# How are these different?

- Short answer: different historical developments, thus different standard assumptions
- Typically not clearly stated
- Thus difficult to:
  - connect similar ideas in different fields
  - know what works, and when
  - know what you should use for your problem
- Currently, all are increasingly overlapping

# Core assumptions

- Linear vs. nonlinear dynamics
- Known vs. unknown cost function
- Episodic interaction vs. single episode (online)  vs. batch offline data
  - Related: which policy was used to collect data?

# Breaking down assumptions

| | System Identification | Adaptive Control | Model-based RL | Model-free RL |
|---|---|---|---|---|
| **Dynamics** | Usually linear | Usually linear | Discrete or nonlinear continuous | Discrete or nonlinear continuous |
| **Reward knowledge?** | N/A | Designed (thus known) | Typically assumed known (not always) | Typically assumed unknown, provided by environment |
| **Data collection/ episodic structure** | Dataset provided | One episode | Typically, repeated episodes | Typically, repeated episodes |
| **What do we learn?** | Dynamics model | Usually policy (MRAC) or model (MIAC) | Dynamics model, sometimes reward model, sometimes policy | Policy (or Q function) |

Caveat: there are exceptions to **all** of the above.

# Generalization and exploration

- LTI dynamics: if dataset generated with sufficient excitation, gives **global** knowledge
- Nonlinear dynamics: extrapolation is difficult and can be misleading
  - As AC/RL moves to more complex systems, have to consider uncertainty, exploration, and data collection process

# Tabular model-based RL

- Discrete state/action space with stochastic transitions
- If model is known, can use value iteration/policy iteration/Monte Carlo policy improvement/etc.
- Model unknown: want to build approximate model from observed transitions

# Tabular MBRL outline

- Assume initial policy
- Loop forever:
  - Take some number of actions, resulting in transition/reward data
  - Improve dynamics model
  - Choose actions/policy

- Approaches for action selection:
  - Dynamic programming/VI/DP on approximate model
    - Expensive, gives optimal policy for model
  - Plan suboptimal sequence of actions via online control opt (focus of today's lecture)

# Learning a tabular model from data

- States $(x_1, x_2, \dots, x_n)$
- Actions $(u_1, u_2, \dots, u_m)$
- Want to learn $p(x_i | x_j, u_k)$ for all $i, j, k$

- We will discuss both max likelihood point estimation and fully Bayesian approaches

# Max likelihood for tabular MBRL

- Categorical likelihood: $p\left(x_i \middle| x_j, u_k, \theta\right) = \theta_{ijk}$
- Assume data $D = \{(x, u, x')\}_{i=1}^{d}$
- Max likelihood:

$$\max_{\theta \in \Theta} \sum_{D} \log p(x' | x, u, \theta)$$

- Gives MLE $\theta_{ijk} = N(x_j, u_k, x_i)/N\left(x_j, u_k\right)$ where $N(\cdot,\cdot)$ is the empirical count

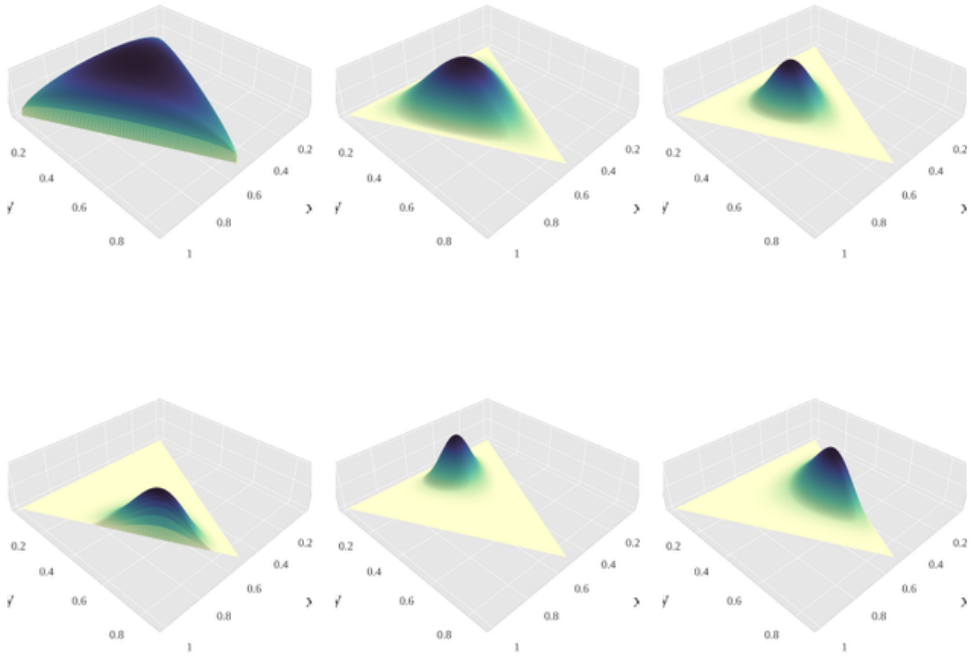
# Max likelihood for tabular MBRL

- $\theta_{ijk} = N(x_j, u_k, x_i)/N(x_j, u_k)$
- Problem: what if $N(x_j, u_k) = 0$?
  - For example, if we are starting with zero information, this model estimation scheme breaks
- Simple solution:
  - Store $N(x_j, u_k, x_i)$; note that $N(x_j, u_k) = \sum_{x_i} N(x_j, u_k, x_i)$
  - Replace $N(x_j, u_k, x_i)$ with $N(x_j, u_k, x_i) + 1$
  - Gives $\theta_{ijk} = (N(x_j, u_k, x_i) + 1)/(N(x_j, u_k) + n)$

- We will see: corresponds to weak prior over transition mass function

# Bayesian inference of transition probabilities

- Fix *Dirichlet distribution* prior
  - Corresponds to a probability distribution *over* discrete probability distributions
  - Write $\mathrm{Dir}(\boldsymbol{x}, \alpha) = p(x_1, \ldots, x_n | \alpha_1, \ldots, \alpha_n)$
- Dirichlet is *conjugate* with categorical distribution:
  - Dirichlet prior with parameters $\alpha_1, \ldots, \alpha_n$, plus Categorical distribution gives Dirichlet posterior $\mathrm{Dir}(\boldsymbol{x}, \alpha + c)$
  - $c = (c_1, \ldots, c_n)$ is counts of data



For details on derivation of posterior, see: *The Dirichlet-Multinomial and Dirichlet-Categorical models for Bayesian inference,* Stephen Tu (available online).

# Bayesian posterior

- Prior parameter $\alpha$ corresponds to number of prior observations
- For $(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n}) \sim Dir(\alpha)$, $\mathrm{E}[x_i] = \alpha_i / \sum_j \alpha_j$
  - Posterior predictive is $\mathrm{p}(x'|x, u, D, \alpha) = \frac{N(x,u,x') + \alpha'(x,u,x')}{\sum_{x'} N(x,u,x') + \alpha'(x,u,x')}$
- Choose $\alpha = (1, \ldots, 1)$ gives our previous correction

- But, we have more than just the point estimate of our model. How can we use this?

# Bayes-adaptive MDPs

- Have model parameters $\theta$, which summarizes "counts" for Dirichlet posterior for every state/action/next state combination.
- *Bayes-adaptive* MDP, with hyperstate $(x, \theta)$
  - So have transition dynamics $\mathrm{p}(x', \theta' | x, u, \theta)$
  - Factor as $\mathrm{p}(x' | x, u, \theta)\ \mathrm{p}(\theta' | x', x, u, \theta)$
    - $\mathrm{p}(x' | x, u, \theta) = \frac{N(x,u,x') + \alpha'(x,u,x')}{\sum_{x'} N(x,u,x') + \alpha'(x,u,x')}$
    - Model parameter count increases by 1 for corresponding transition count

- Problem: state space grows infinitely, so can not do dynamic programming

Good reference on review of Bayes-adaptive RL and approximate methods: Michael Duff's PhD thesis, 2003.

# Exploration heuristics: Thompson sampling

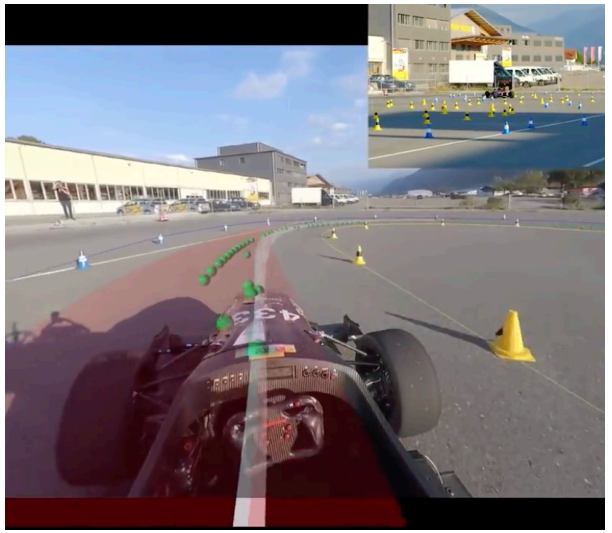
- Even in tabular and linear MDPs, dual control/Bayes-adaptive MDP intractable
- So turn to heuristics to explore
  - Epsilon greedy, noise addition (persistent excitation)
- Simple approach using posterior over models: Thompson sampling
  - Sample MDP from posterior
  - Act optimally w.r.t. this MDP for episode
  - Update model posterior and loop

# Example

# Continuous MBRL

- Will now consider general non-LTI continuous models
- Many possible model choices:
  - Nonlinear features in linear regression
  - Time varying linear dynamics
  - Gaussian processes
  - Neural networks
- Many possible control choices:
  - MPC (gradient-based vs. sampling, with/without final cost)
  - Direct methods (e.g. iLQR/DDP)
  - Directly optimize policy (next week)

Abbeel et al., NeurIPS 2008



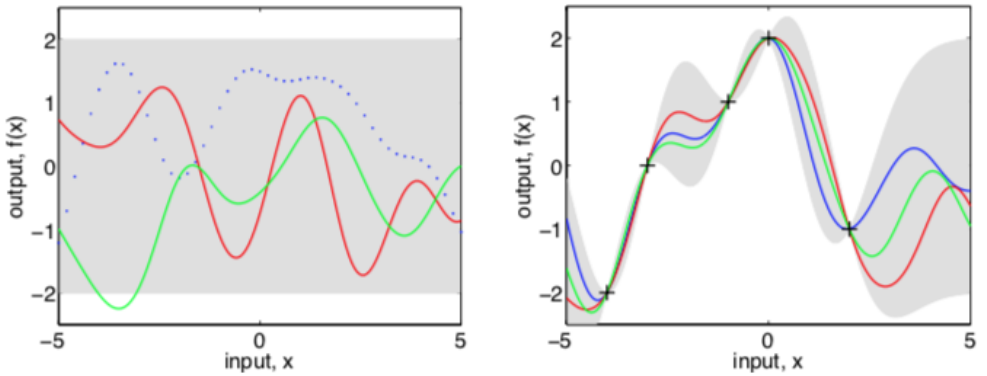
Kabzan et al., RA-L 2019

# Gaussian process models

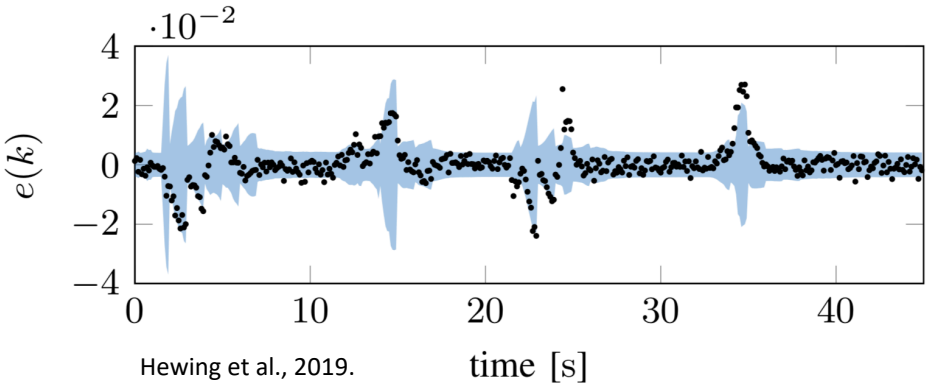
- Place prior over dynamics, $f \sim GP\big(m(\cdot), \mathrm{k}(\cdot,\cdot)\big)$
  - Corresponds to infinite dimensional gaussian distribution, prior over functions

- Strengths
  - Data efficient
  - Exact posterior
  - Predictable behavior via kernel choice

- Weaknesses
  - High computational complexity
  - Assume Gaussian measurement error
  - Can not learn expressive features



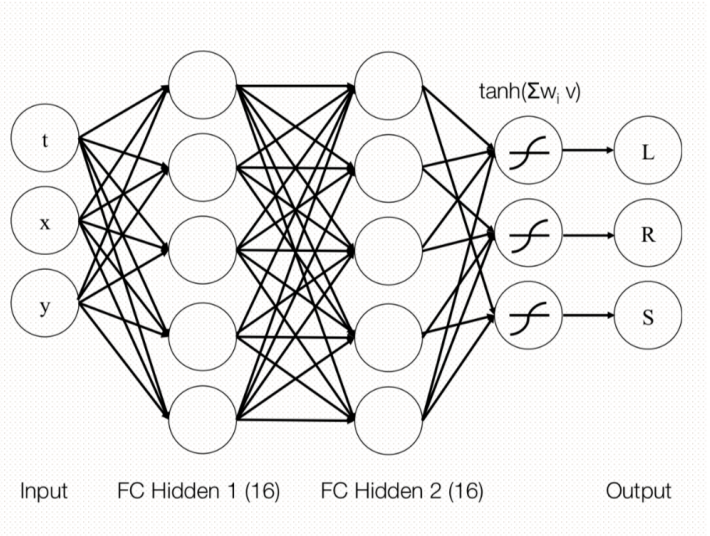

Rasmussen and Williams, 2006.

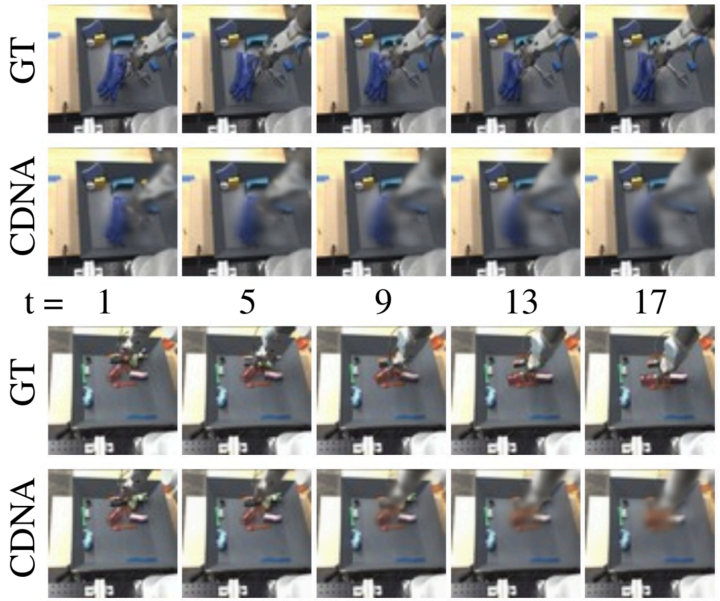



Hewing et al., 2019.

# Neural network models

- Parameterize model using neural network
- Strengths
  - Can learn complex, expressive features
  - Can be paired with arbitrary loss functions
- Weaknesses
  - Data inefficient
  - Difficult to represent uncertainty
  - Unpredictable extrapolation



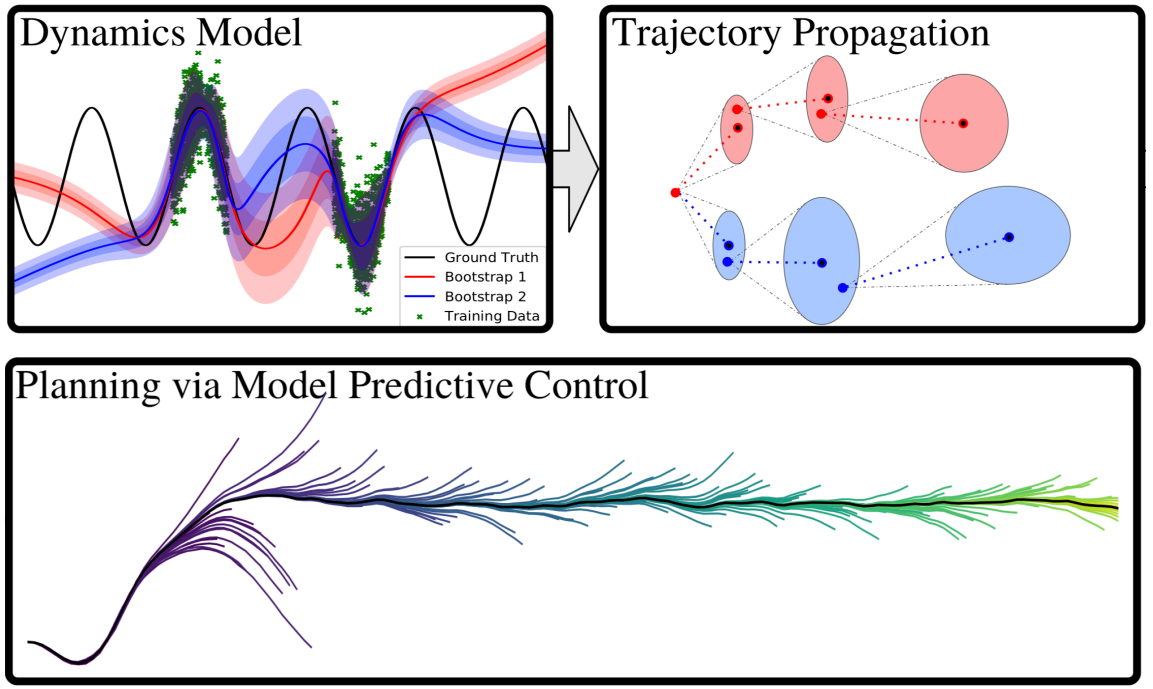

Finn et al., 2017.

# Case study: PETS

- Key idea:
  - Use *ensemble* (collection) of NNs to approximate posterior over model
  - Incorporate model uncertainty into control
- Ensembling:
  - Initialize several networks with different weights
  - Will agree where there is a lot of data, disagree elsewhere

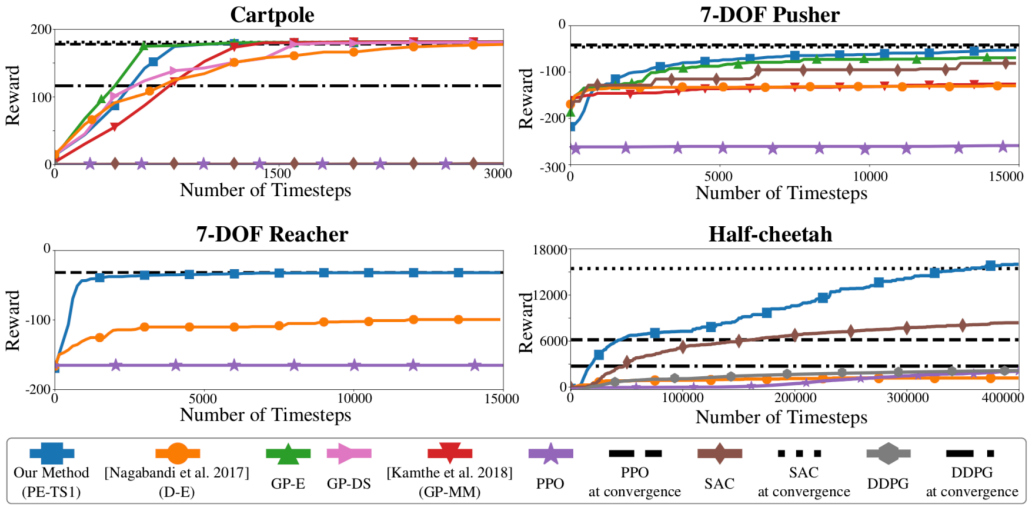
**Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models**

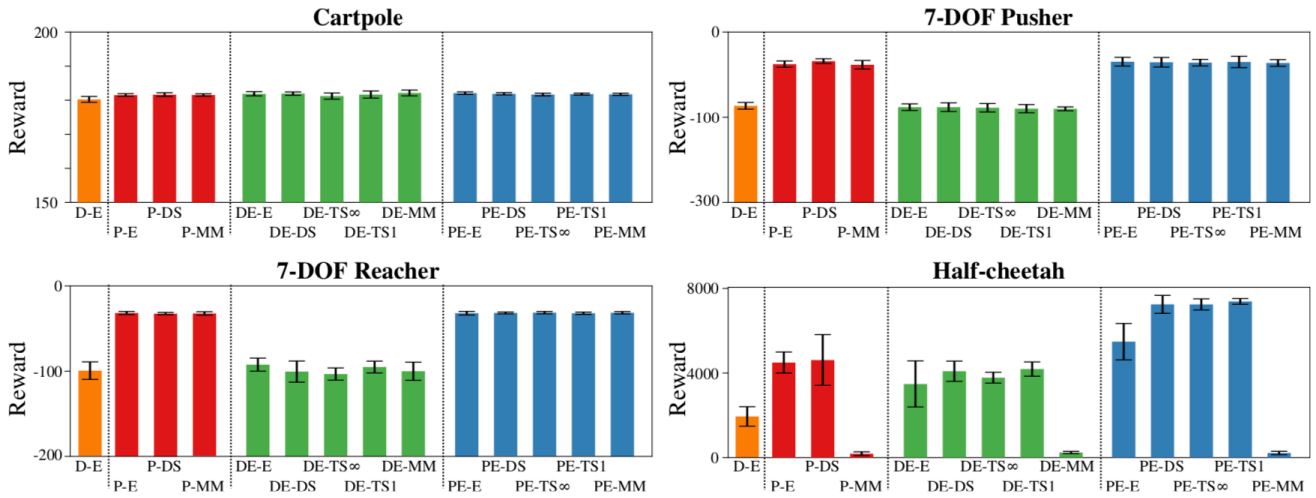
**Kurtland Chua** **Roberto Calandra** **Rowan McAllister** **Sergey Levine**
Berkeley Artificial Intelligence Research
University of California, Berkeley
{kchua, roberto.calandra, rmcallister, svlevine}@berkeley.edu



# PETS findings

- Consider both probabilistic network (outputs mean + variance) and deterministic
- Use particle-based MPC controller (random action sampling)
- Either re-sample dynamics at each time, or keep fixed



# Why model-based?

- Advantages
  - Transitions give strong signal
  - Data efficiency, improved multi-task performance, generalization

- Weaknesses
  - Optimizing the wrong objective
  - May be very difficult/intractable for systems with high dimensional observations/states

# Next time

- Actor-critic and advanced policy gradient