# Modularity and Coordination
## for Planning and Reinforcement Learning

Ph.D. Defense

Jayesh K. Gupta

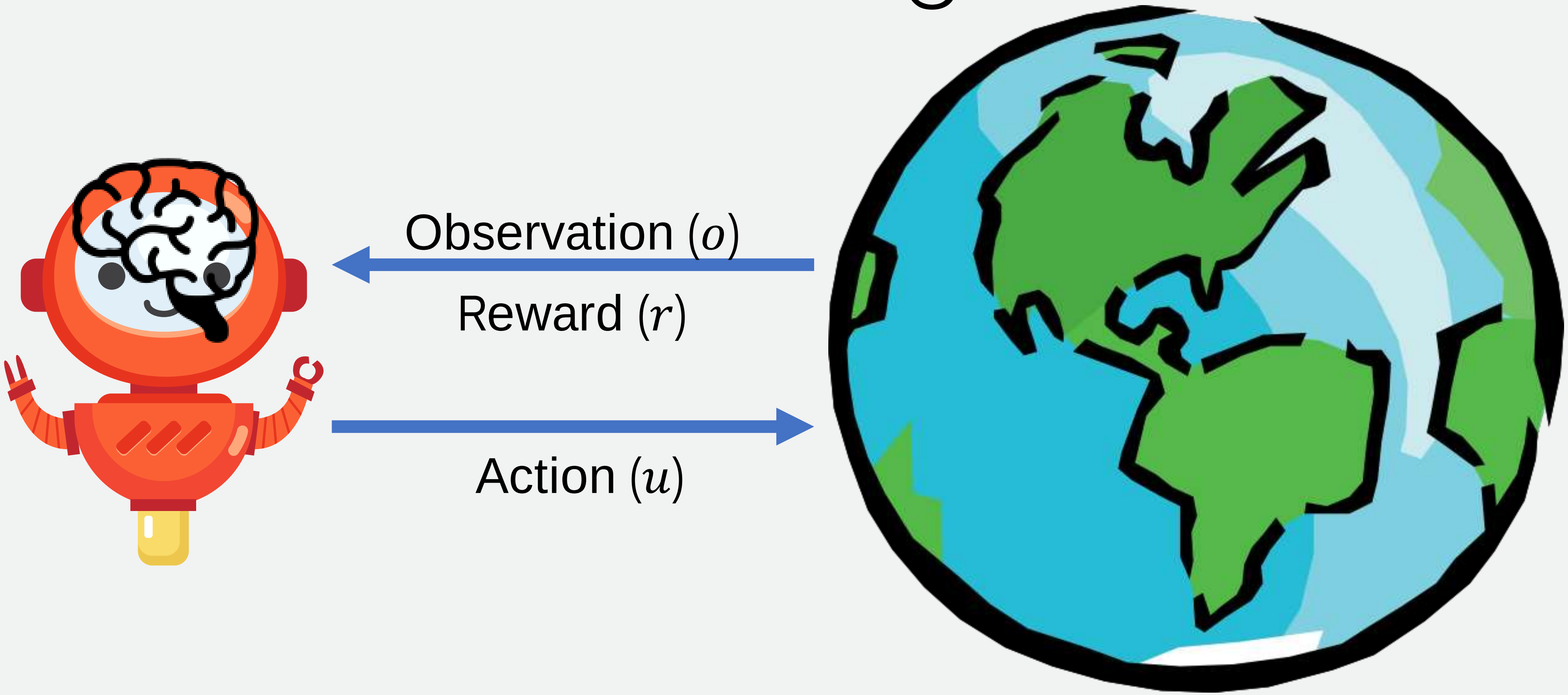
Department of Computer Science
Stanford University

SISL
Stanford Intelligent Systems Laboratory

**Advisor:** Mykel Kochenderfer

# Autonomy

# Reinforcement Learning

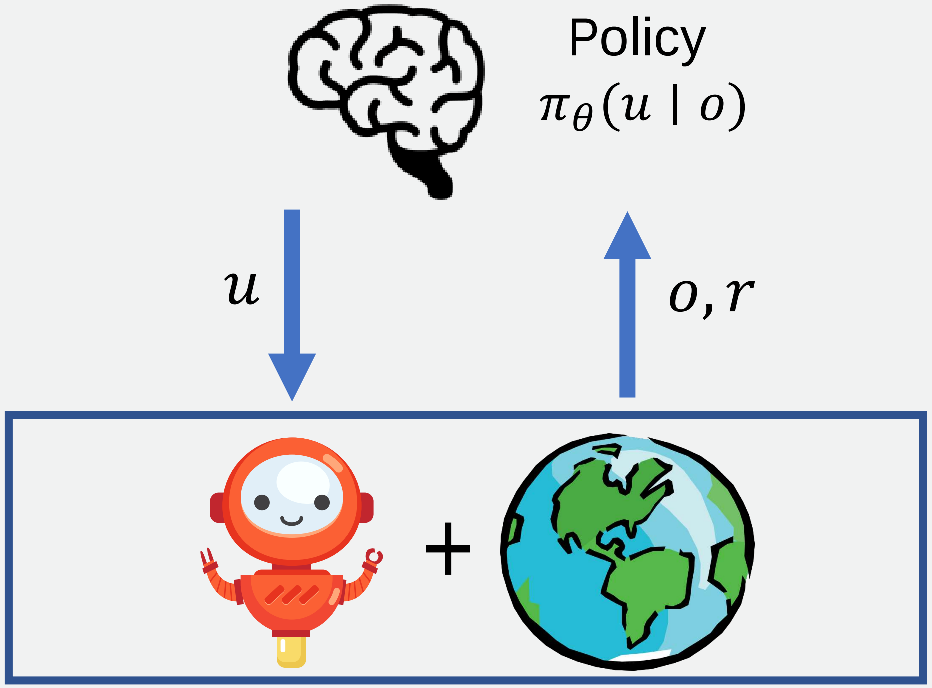


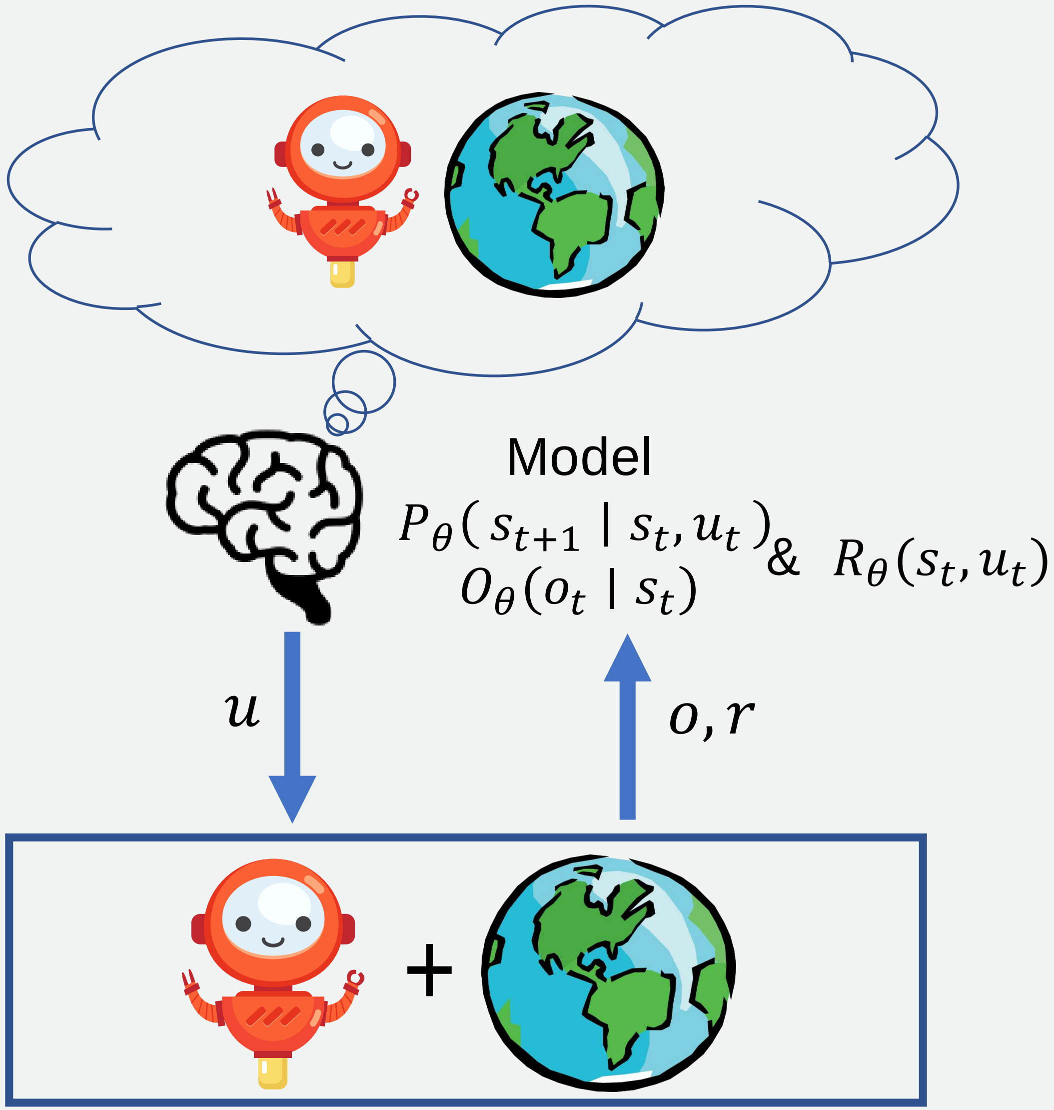

$$s_{t+1} \sim P(\cdot \mid s_t, u_t)$$

Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

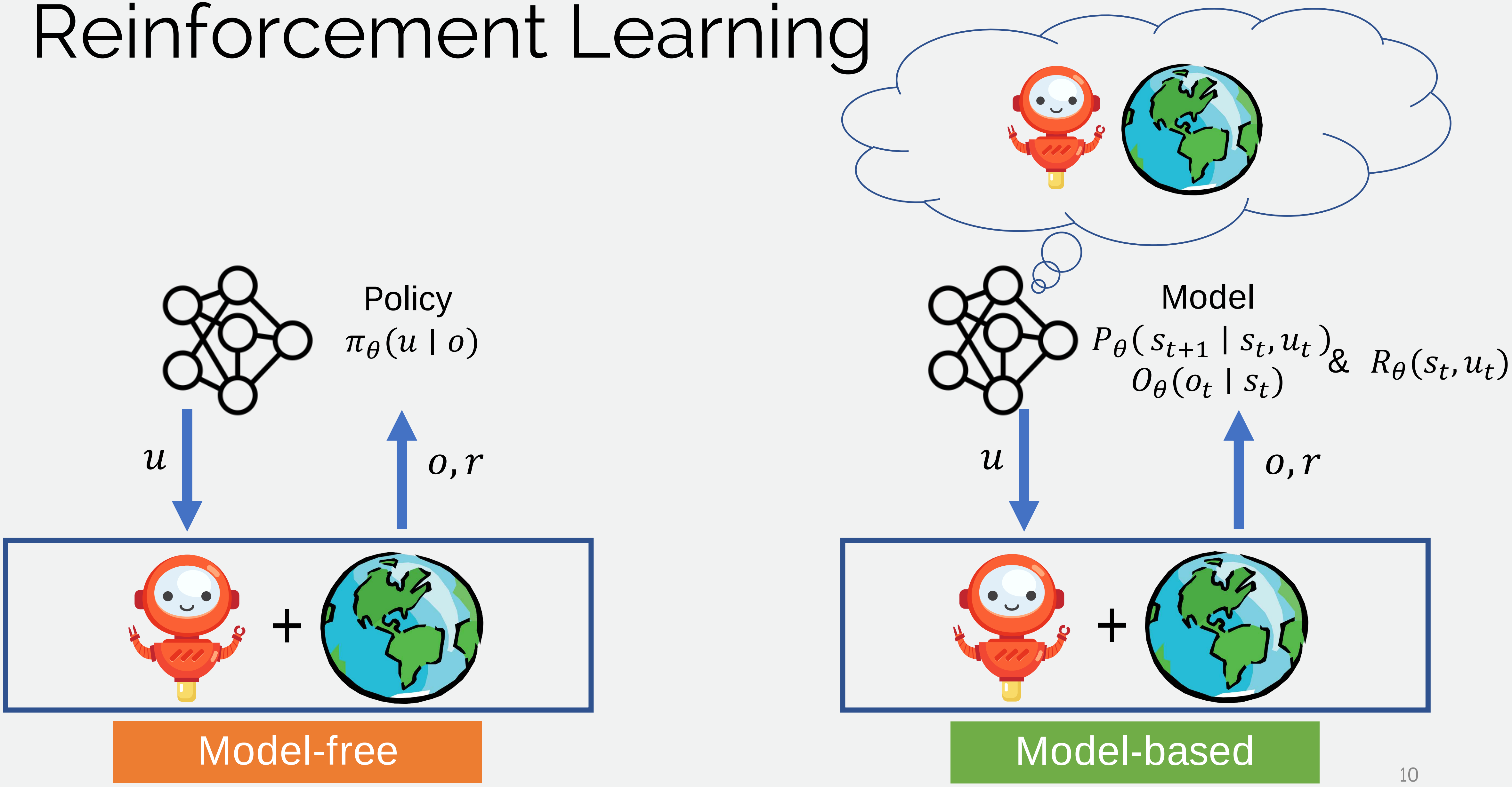
Goal is to maximize *total* return per episode: $\sum_t \gamma^t r_t$

# Thinking Fast & Slow

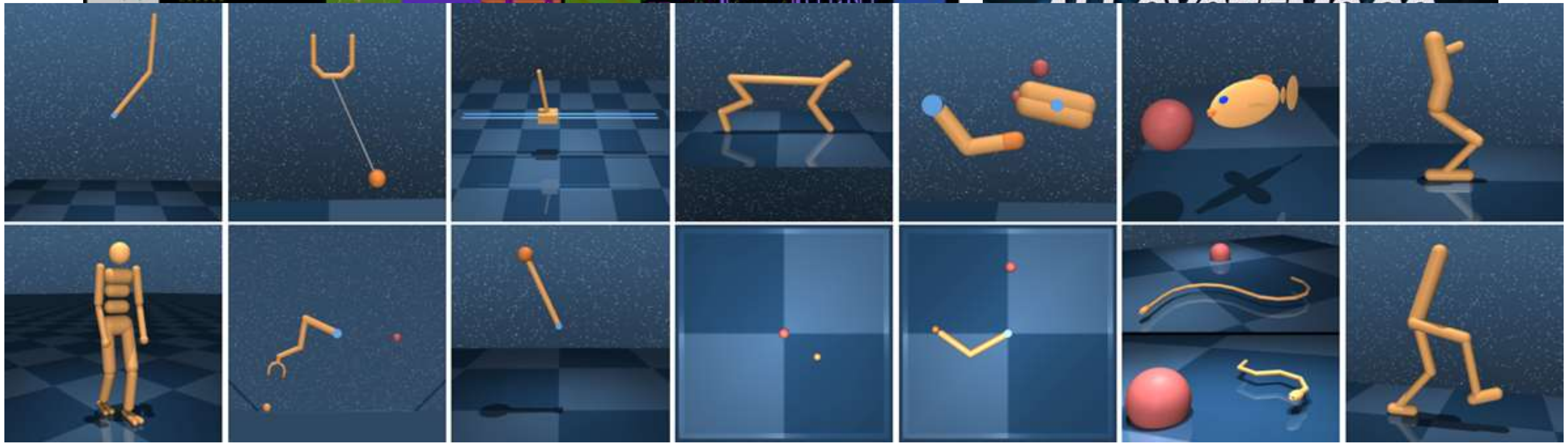


# Reinforcement Learning



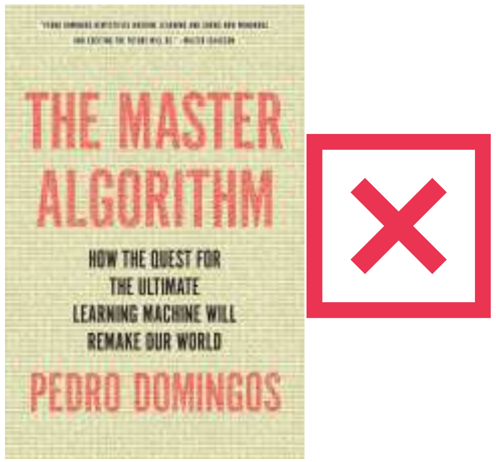
# Substantial Progress

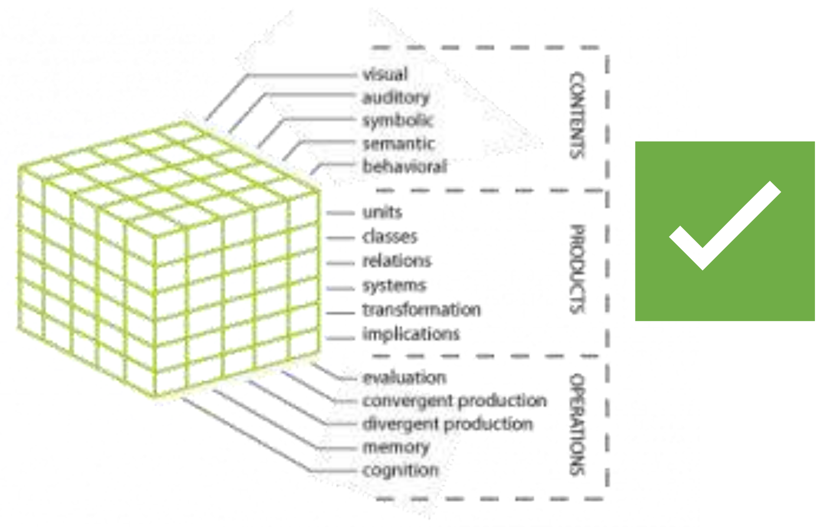



# Real World Requirements

Small sample complexity

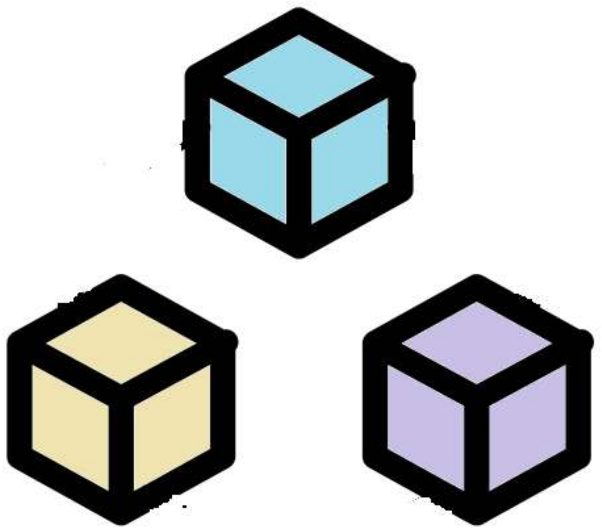
Generalization
to related tasks



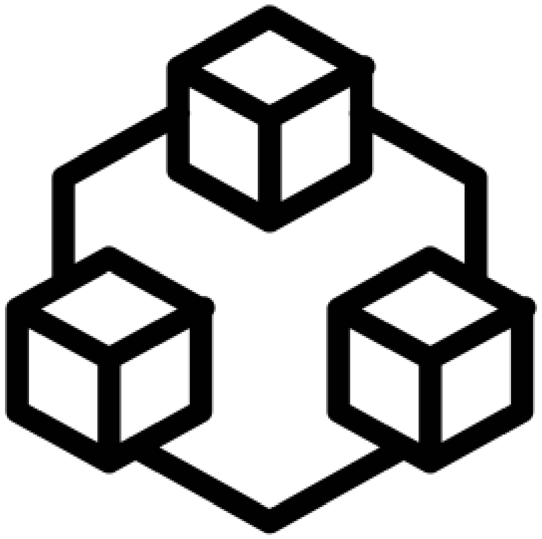




# Decision-making entity as a system of coordinated modules

# Modularity

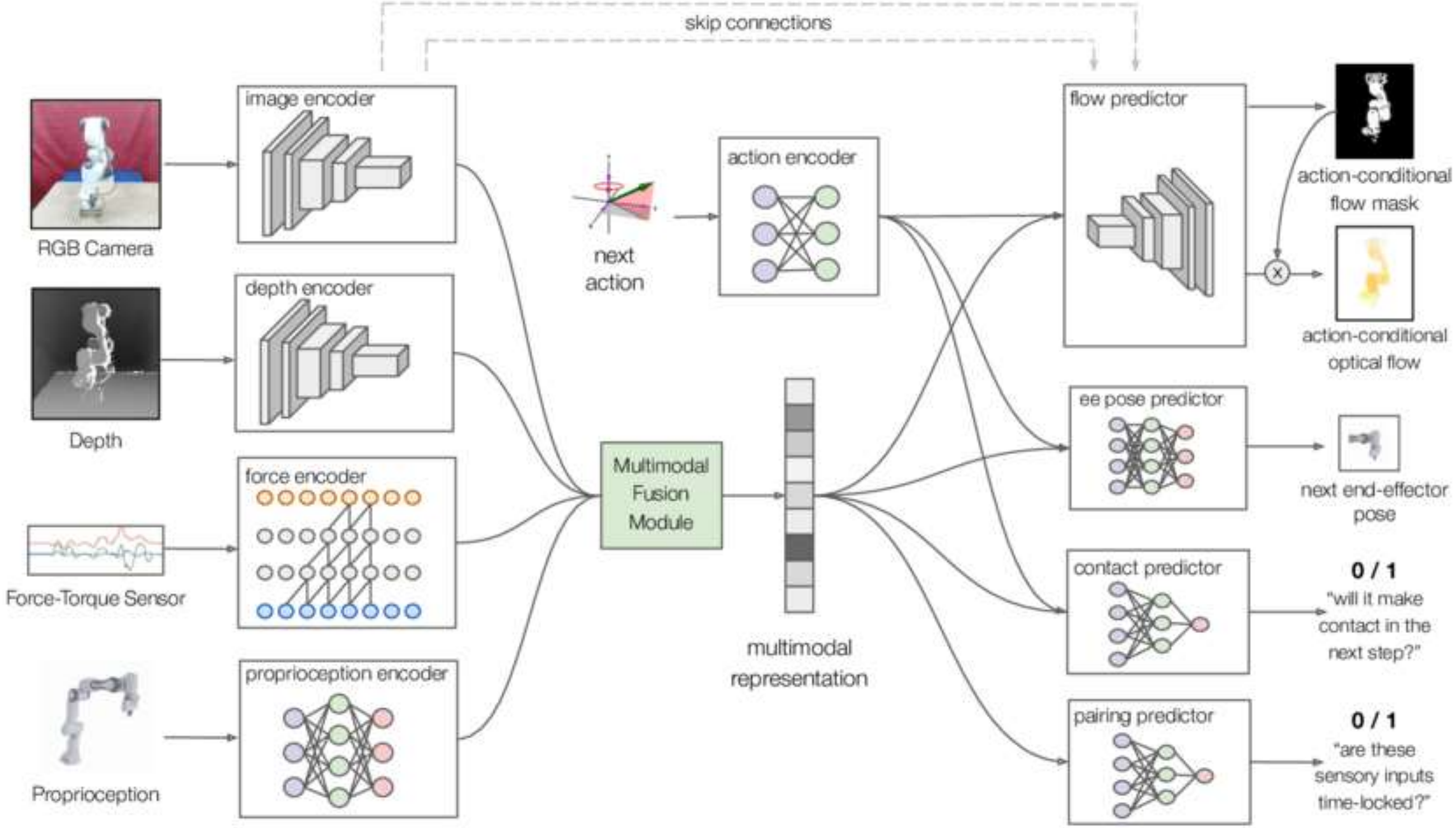


Information Encapsulation

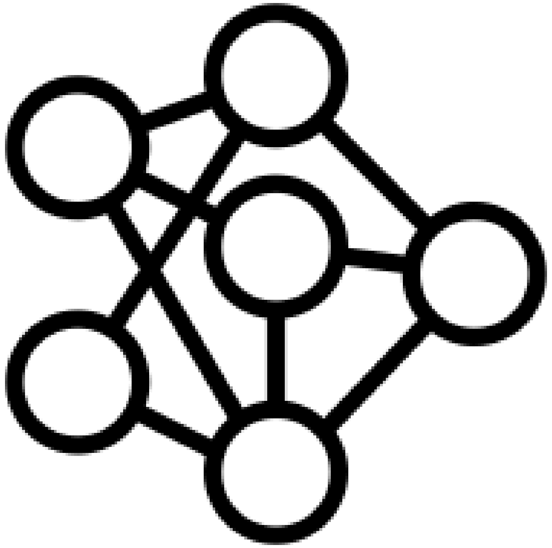


Coordination Framework

# Modularity

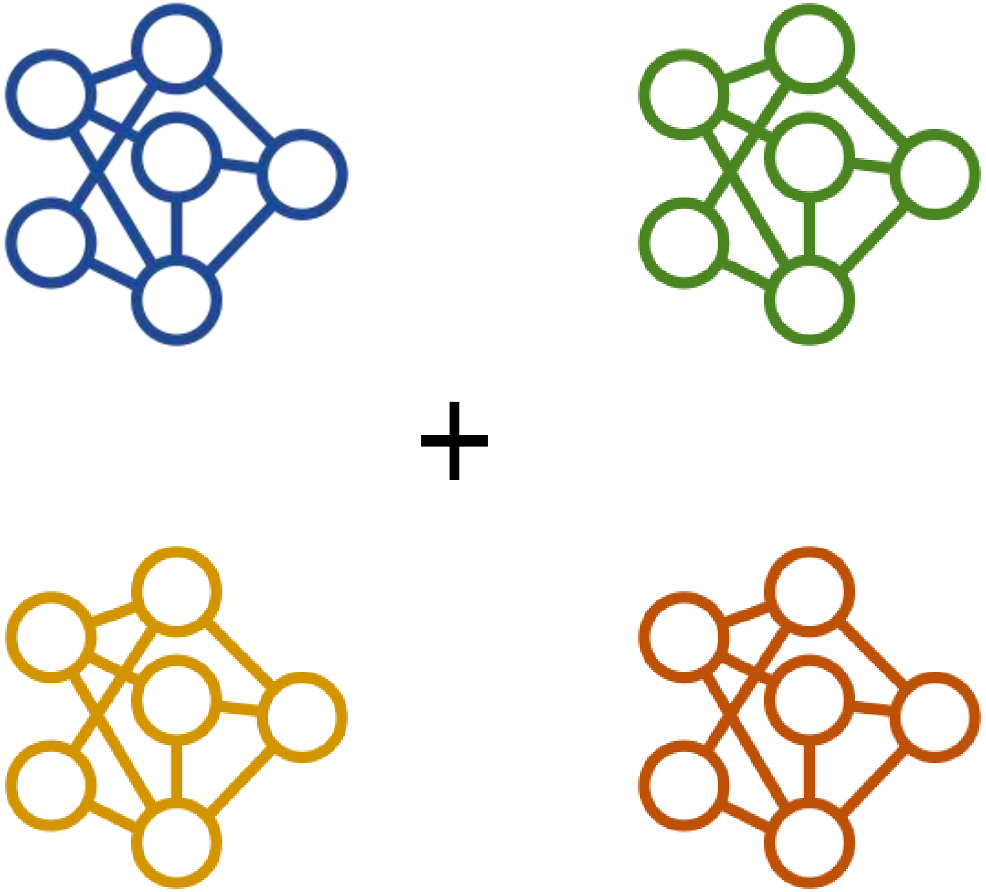

skip connections
image encoder
depth encoder
force encoder
proprioception encoder
RGB Camera
Depth
Force-Torque Sensor
Proprioception
next
action
action encoder
Multimodal
Fusion
Module
multimodal
representation
flow predictor
ee pose predictor
contact predictor
pairing predictor
action-conditional
flow mask
action-conditional
optical flow
next end-effector
pose
0 / 1
"will it make
contact in the
next step?"
0 / 1
"are these
sensory inputs
time-locked?"

# Modularity

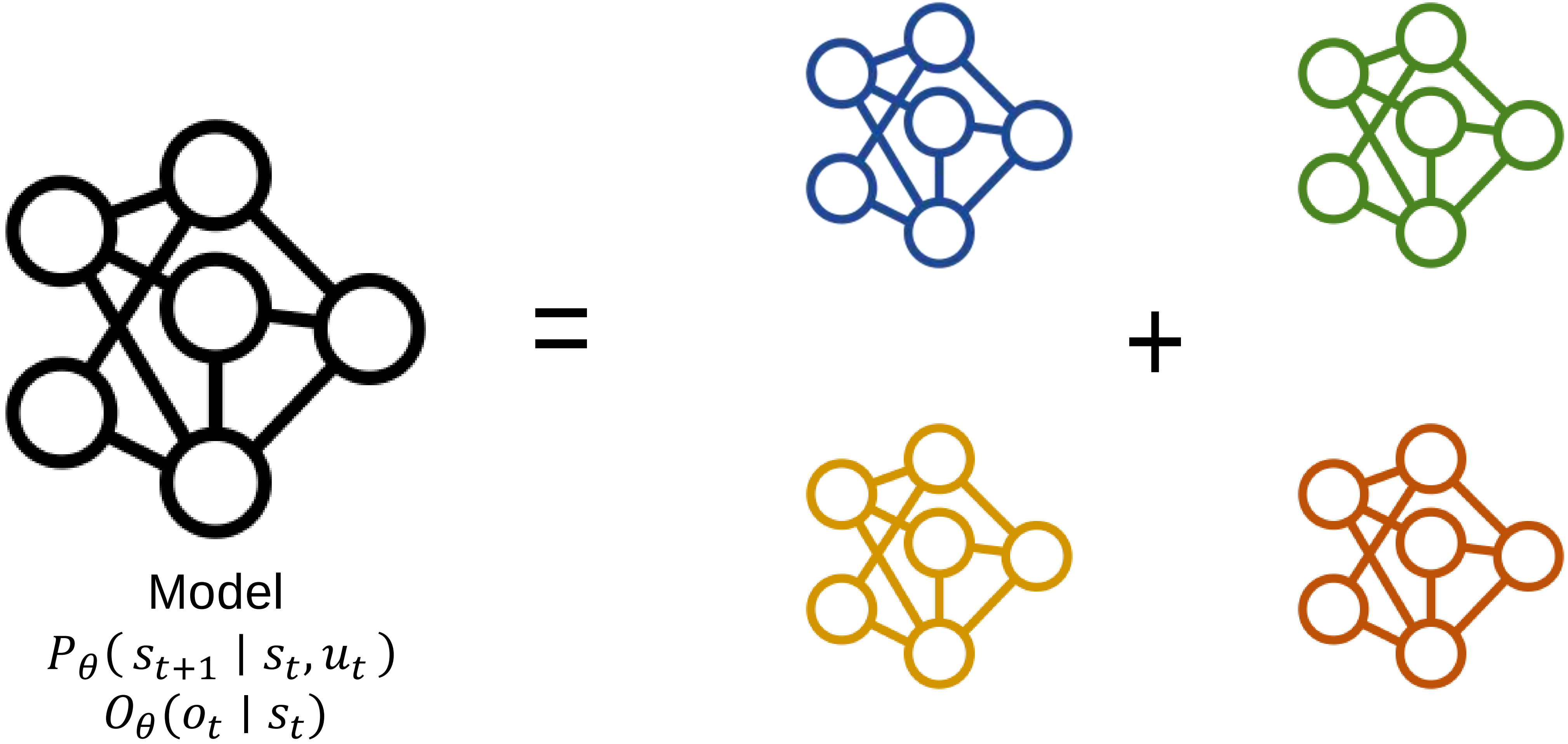


Policy
$\pi_\theta(u \mid o)$

=



# Modularity



## THE ARCHITECTURE OF COMPLEXITY

HERBERT A. SIMON*

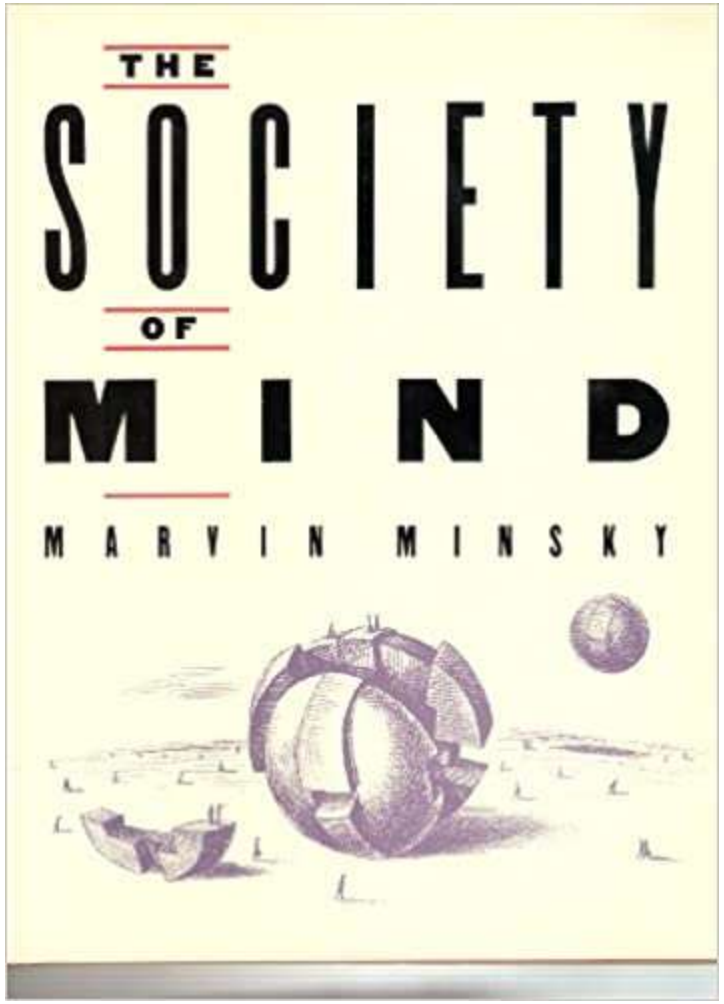
Professor of Administration, Carnegie Institute of Technology

*(Read April 26, 1962)*

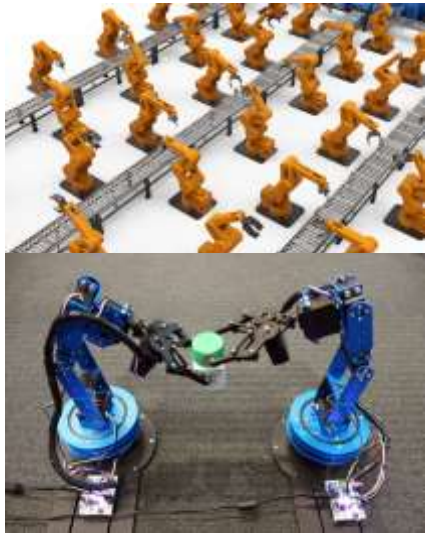
A NUMBER of proposals have been advanced in recent years for the development of "general systems theory" which, abstracting from properties peculiar to physical, biological, or social systems, would be applicable to all of them.[1] We might well feel that, while the goal is laudable, systems of such diverse kinds could hardly be expected to have any nontrivial properties in common. Metaphor and analogy can be helpful, or they can be misleading. All depends on whether the similarities the metaphor captures are significant or superficial.

It may not be entirely vain, however, to search for common properties among diverse kinds of complex systems. The ideas that go by the name of cybernetics constitute, if not a theory, at least a point of view that has been proving fruitful over a wide range of applications.[2] It has been useful to look at the behavior of adaptive systems in terms of the concepts of feedback and homeostasis, and to analyze adaptiveness in terms of the theory of selective information.[3] The ideas of feedback and information provide a frame of reference for viewing a wide range of situations, just as do the ideas of evolution, of relativism, of axiomatic method, and of operationalism.

In this paper I should like to report on some things we have been learning about particular kinds of complex systems encountered in the behavioral sciences. The developments I shall discuss arose in the context of specific phenomena, but the theoretical formulations themselves make little reference to details of structure. Instead they refer primarily to the complexity of the systems under view without specifying the exact content of that complexity. Because of their abstractness, the theories may have relevance—application would be too strong a term—to other kinds of complex systems that are observed in the social, biological, and physical sciences.

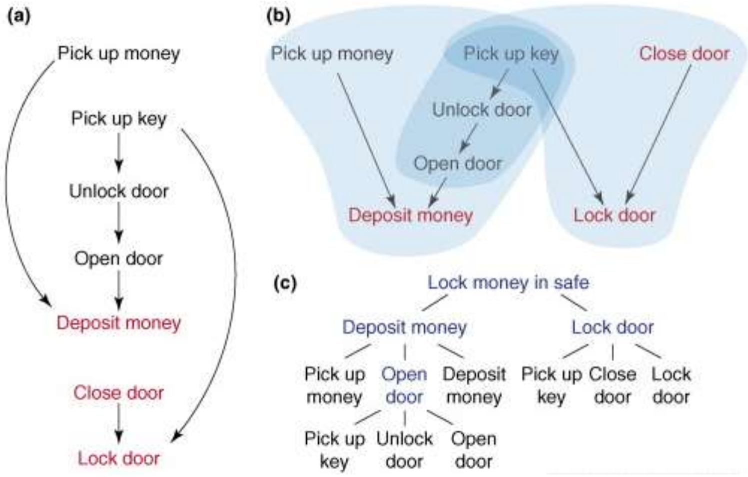


# Modularity for Intelligence



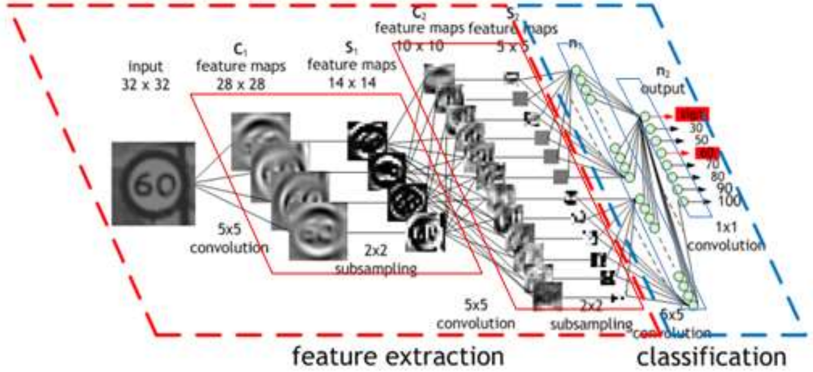
Ref: https://www.media.mit.edu/publications/3d-backscatter/

**Functional Modularity**




Ref: 10.1016/j.tics.2008.02.009

**Temporal Modularity**

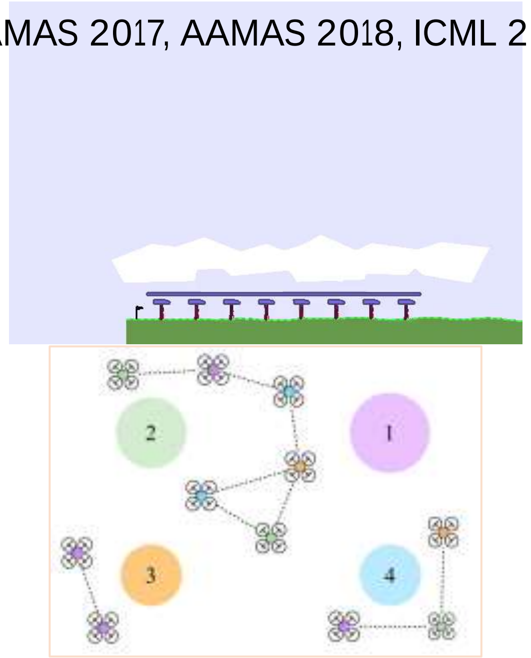



**Architectural Modularity**

1. Fodor, J. A. (1983). The Modularity of the Mind. The Massachusetts Institute of Technology.
2. Bryson, Joanna Joy. Intelligence by design: principles of modularity and coordination for engineering complex adaptive agents. Diss. MIT, 2001.

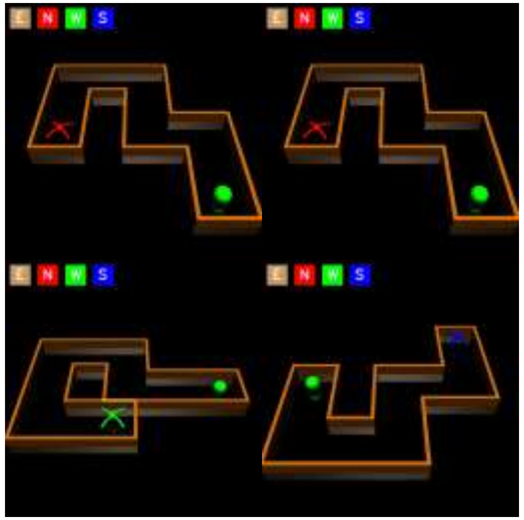
# Contributions

AAMAS 2017, AAMAS 2018, ICML 2018




**Functional Modularity**
Agent-informed modules

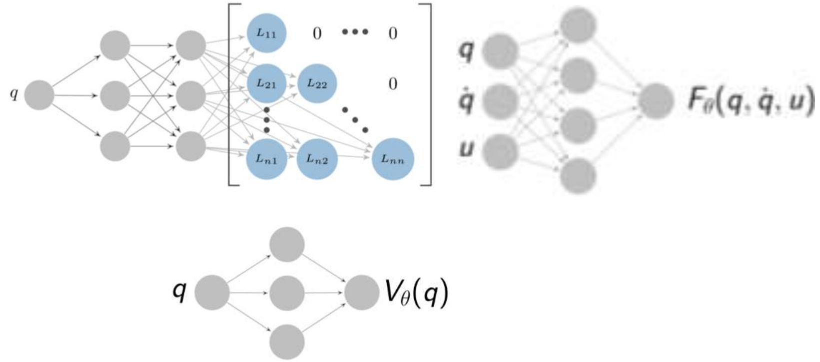
AAMAS 2019




**Temporal Modularity**
Task-informed modules

L4DC 2020

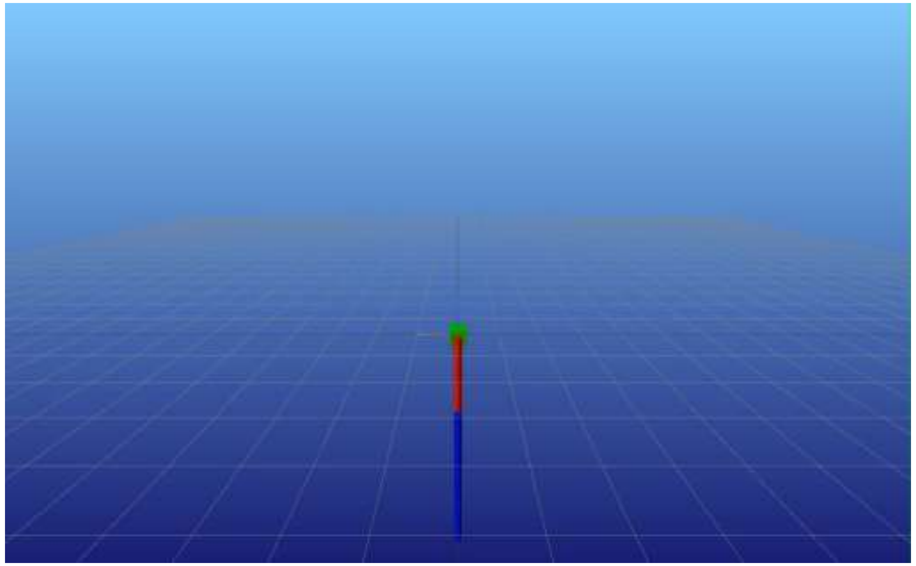
$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = F(q, \dot{q}, u)$$




**Architectural Modularity**
Physics-informed modules

1. Identification of specific ways modularity comes into play during design of decision-making systems
2. Application of these modular design principles, reduces sample complexity and improves generalization

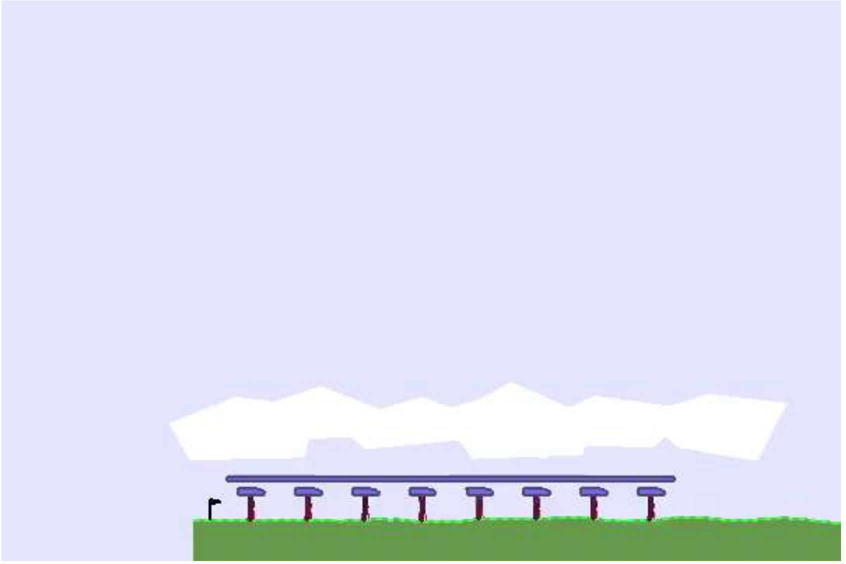
1



Structured Mechanical Models
(L4DC 2020)

**Architectural Modularity**
Physics informed modules

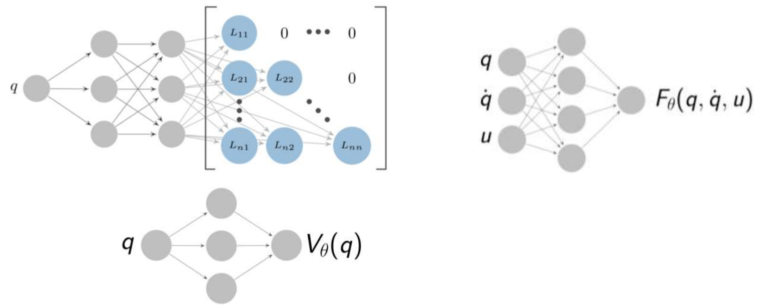
2



Scaling Deep RL to Large MAS
(AAMAS 2017)

**Functional Modularity**
Agent informed modules

# Physics informed modules




# Structured Learning of Mechanical Systems

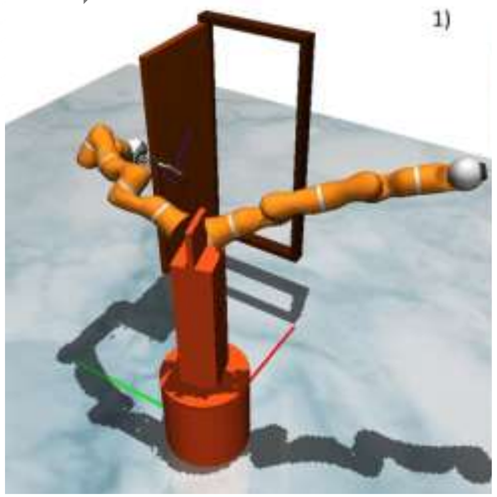
Joint work with Kunal Menda, Zac Manchester, Mykel J. Kochenderfer

L4DC 2020

# Why model dynamical systems?

**Prediction**

**Model-based Control**

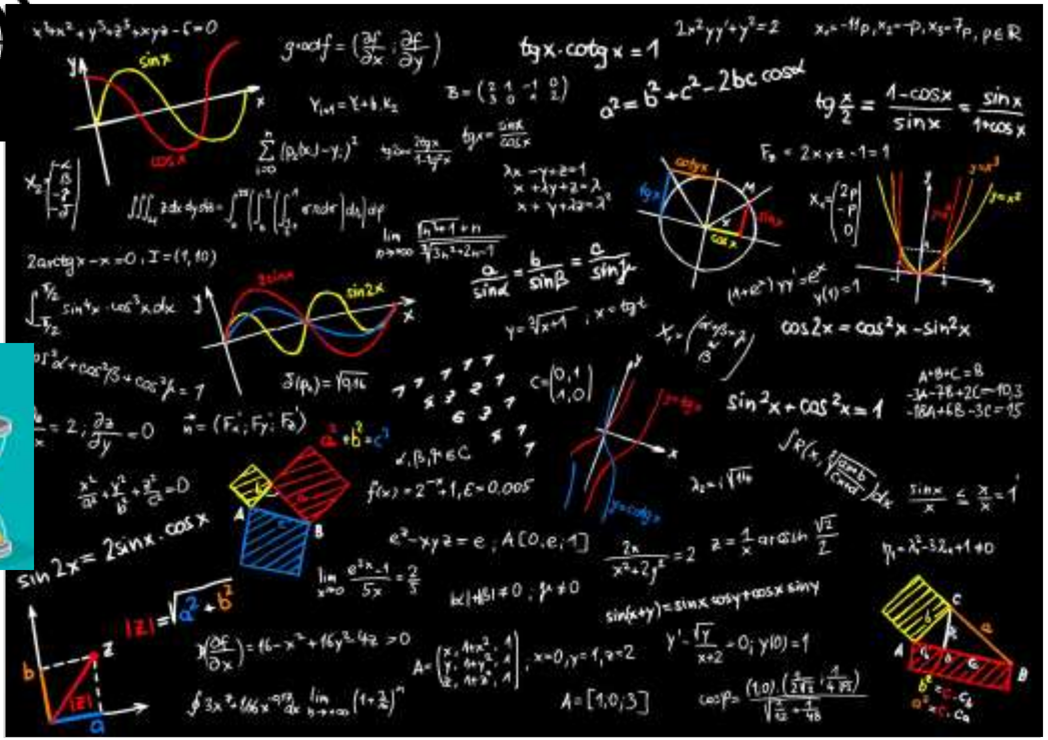



Models generalize better

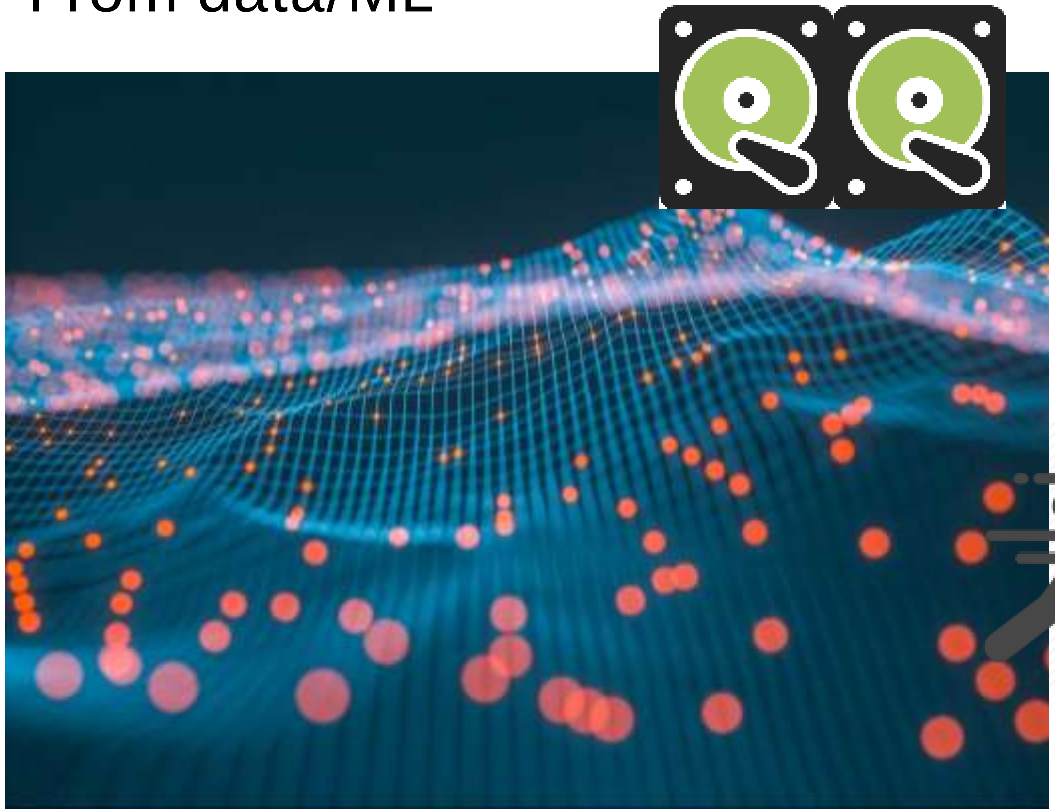
Ref: 10.1109/ICAR.2017.8023522

# Current Approaches

## From first-principles

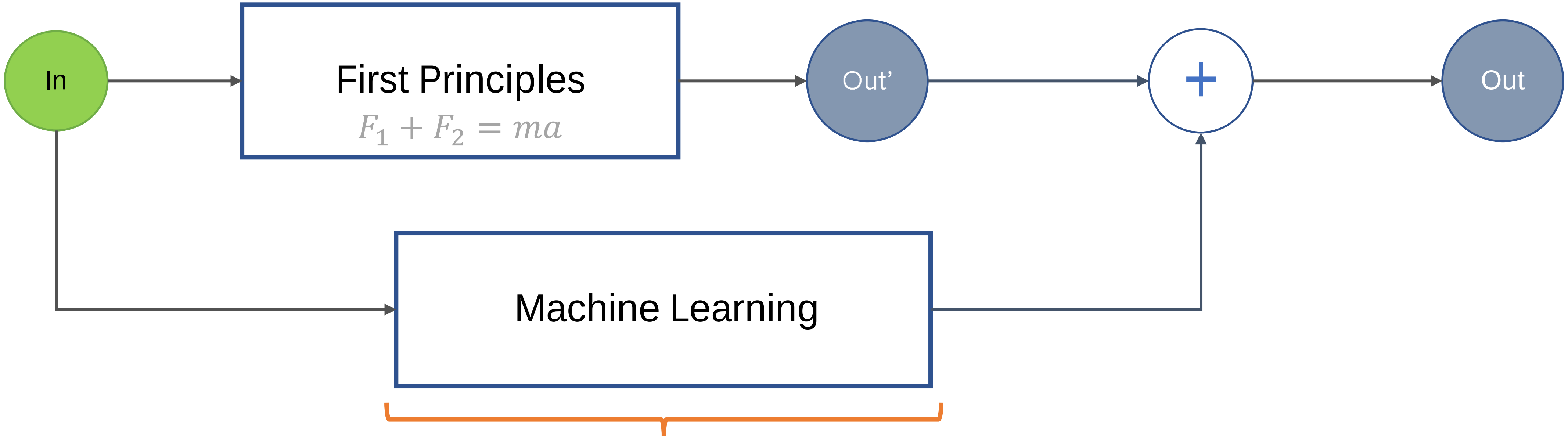


## From data/ML



# Current Approaches

- From first principles + residual from data/ML

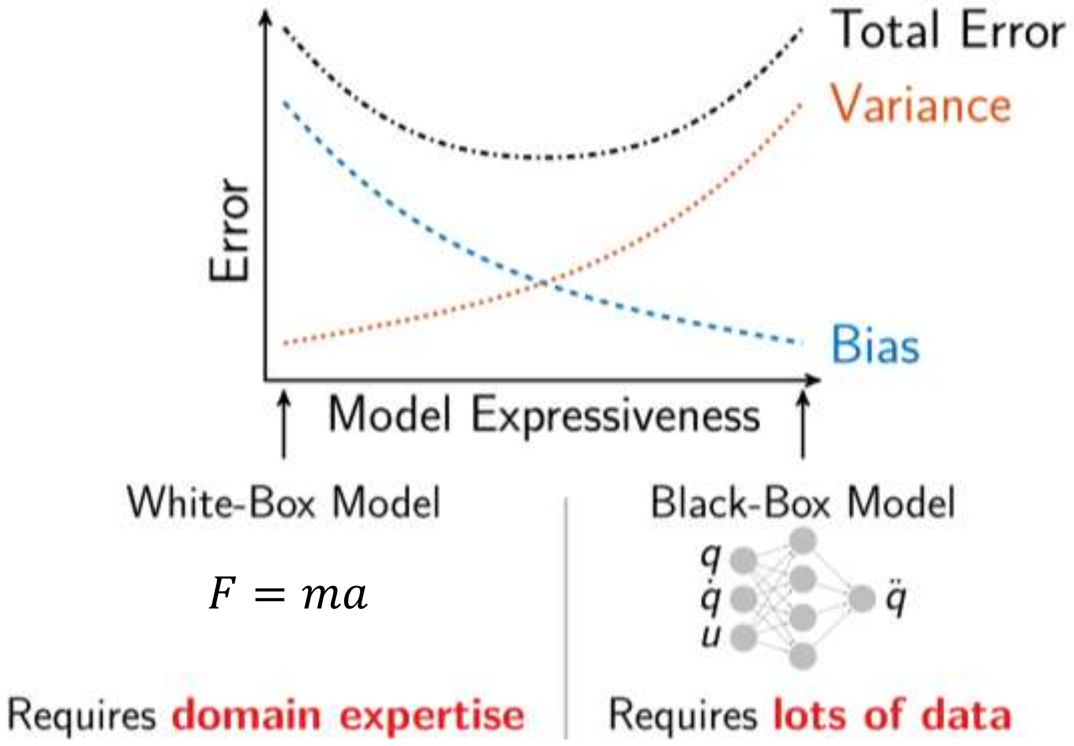



No physical constraints on the machine learning model

e.g. conservation of energy etc.

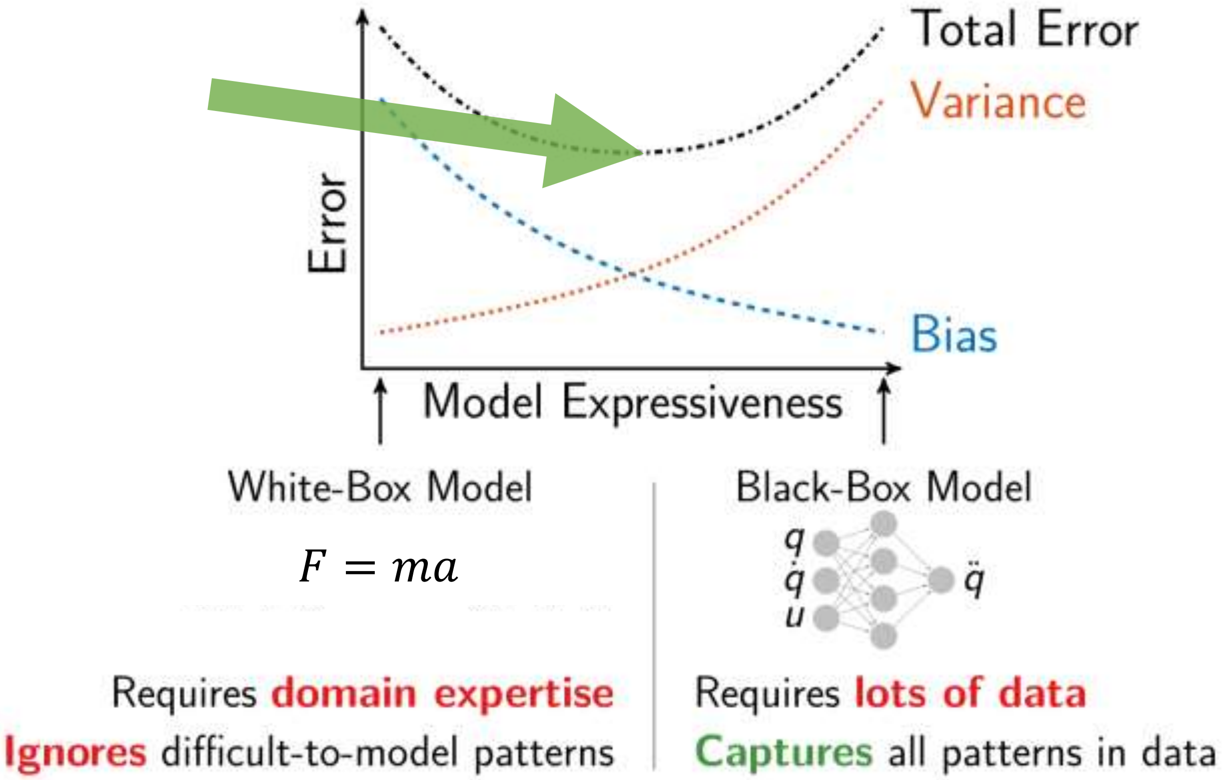
# Common Issue

Bias-Variance Tradeoff



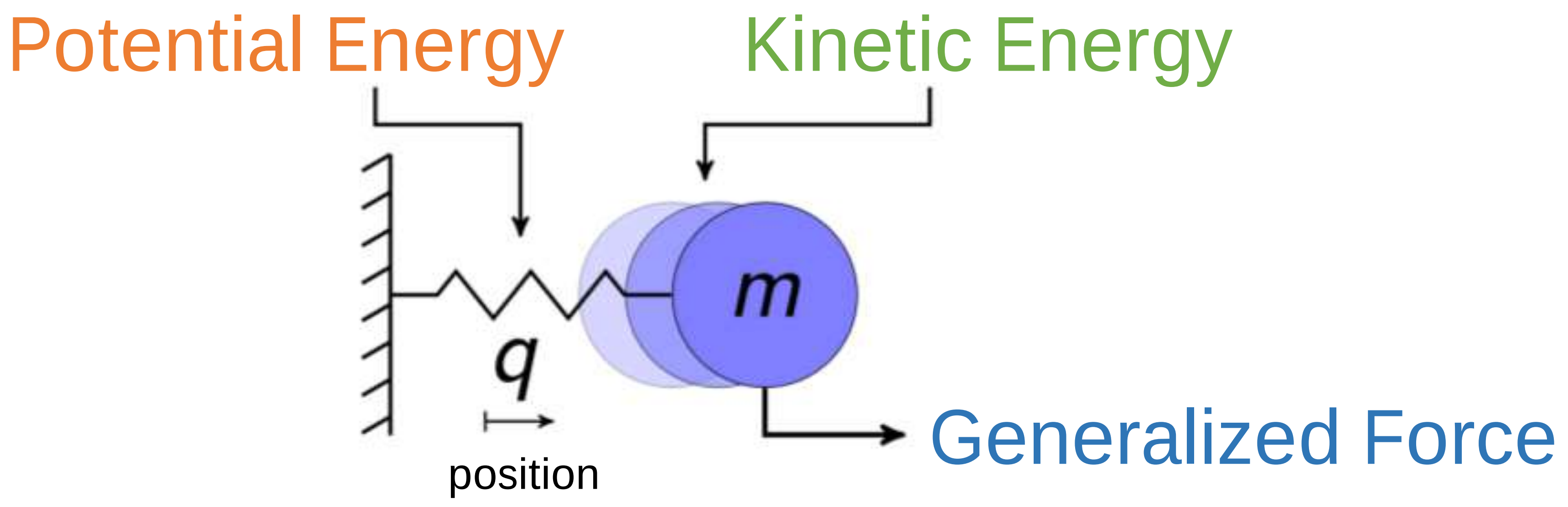
# Goal

- Allowing domain experts to make the bias-variance trade-off

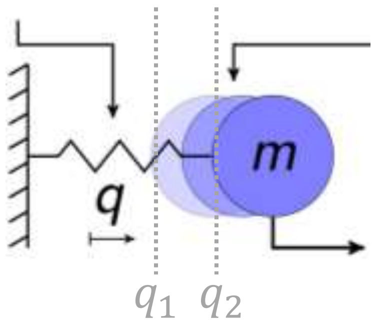


# Modeling Mechanical Systems

$$F = ma$$



# Principle of Least "Action"



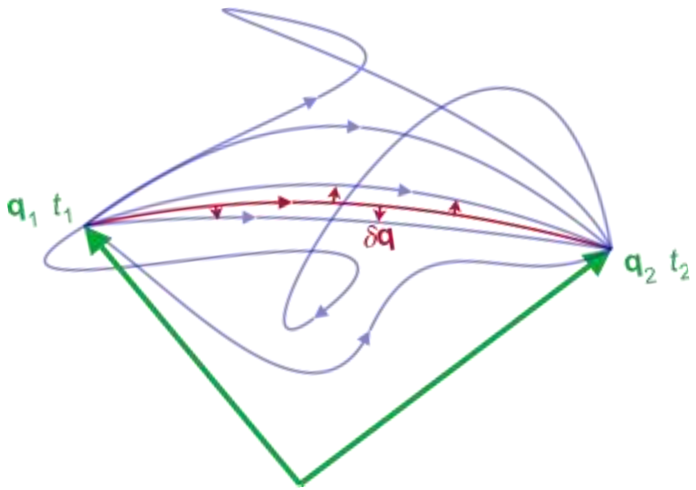

"Action"

minimize

$$S = \int_{t_1}^{t_2} \underbrace{KE(\cdot) - PE(\cdot)}_{\text{Lagrangian, } \mathcal{L}} dt$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = F(q, \dot{q}, u)$$

Euler-Lagrange equation




Ref: Wikipedia Commons

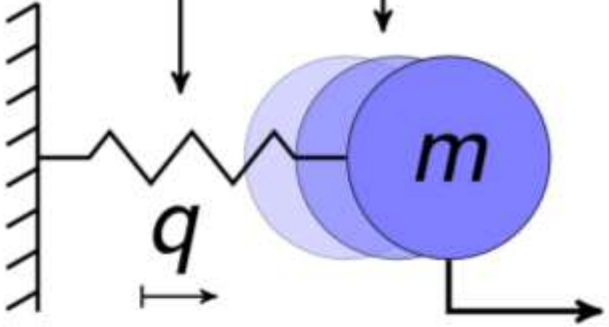
Only one path minimizes nature's cost function $S$

# Lagrangian Dynamics

Mass-matrix

$PE(\cdot) = V(q)$ $KE(\cdot) = \frac{1}{2}\dot{q}^T \mathbf{M}(q)\dot{q}$

Velocity




Force

$F(q, \dot{q}, u)$

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2}\dot{q}^T \mathbf{M}(q)\dot{q} - V(q)$$

# Manipulator Equation

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q,\dot{q})\dot{q} - \nabla_q V(q) = F(q,\dot{q},u)$$

$ma = F$

$\mathbf{M}(q)\ddot{q}$

Acceleration

Integrate to make next state predictions

# Quick Physics Summary

| If you know | Given | You can find |
|---|---|---|
| Mass-matrix $\mathbf{M}(q)$ | Current | Next |
| Potential Energy $V(q)$ | velocity $q_t, \dot{q}_t$ position | velocity $q_{t+1}, \dot{q}_{t+1}$ position |
| Generalized Force $F(q, \dot{q}, u)$ | | |

# Quick Physics Summary

| If you know | Given | You can find |
| --- | --- | --- |
| Mass-matrix $\mathbf{M}_{\boldsymbol{\theta}}(q)$ | Current | Next |
| Potential Energy $V_\theta(q)$ | $q_t, \dot{q}_t$ | $q_{t+1}, \dot{q}_{t+1}$ |
| Generalized Force $F_\theta(q, \dot{q}, u)$ | | |

$$f_\theta \begin{cases} \mathbf{M}_{\boldsymbol{\theta}}(q) \\ V_\theta(q) \\ F_\theta(q, \dot{q}, u) \end{cases}$$

# Structured Modeling Constraints

Mass-matrix

$\mathbf{M}_{\boldsymbol{\theta}}(q)$

Potential Energy

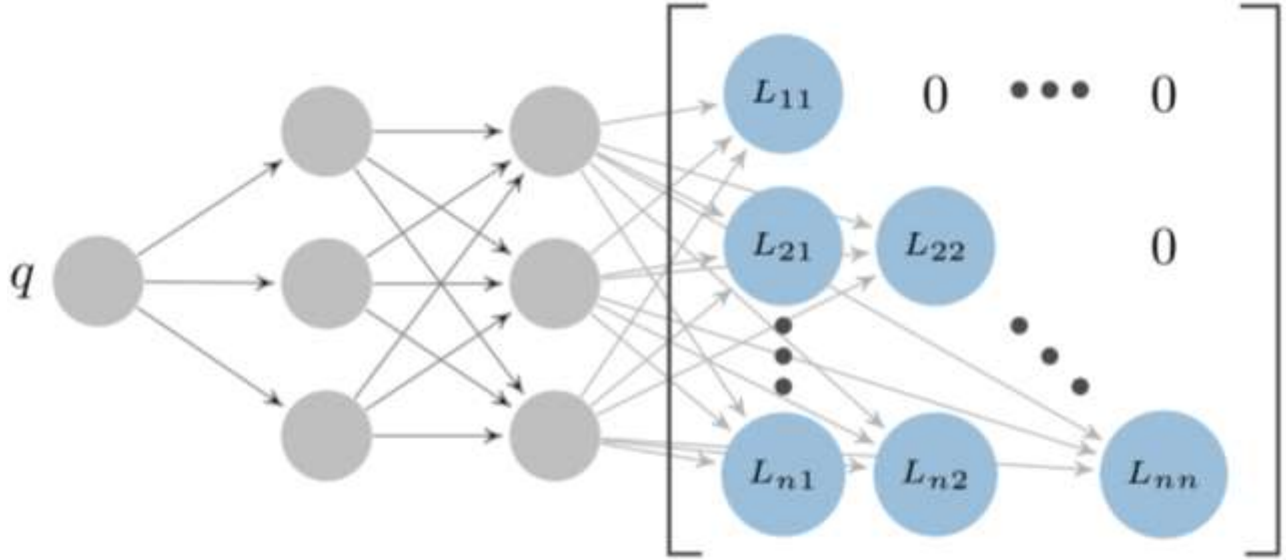
$V_\theta(q)$

Generalized Force

$F_\theta(q, \dot{q}, u)$

- Constraint: Ensure $\mathbf{M}(q) \succ 0$
- Solution: Predict the Cholesky Factor

$$\mathbf{M}(q) = \mathbf{L}(q)\mathbf{L}^T(q)$$




$\mathbf{L}_\theta(q)$

# Structured Modeling Constraints

Mass-matrix

$\mathbf{M}_{\boldsymbol{\theta}}(q)$

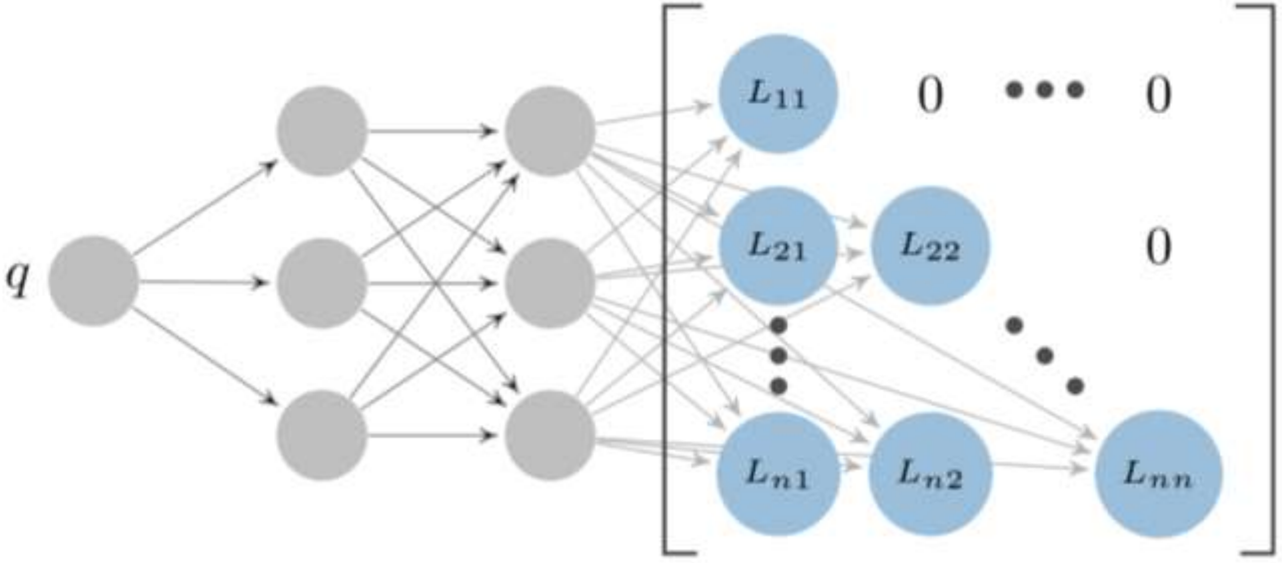
Potential Energy

$V_\theta(q)$

Generalized Force

$F_\theta(q, \dot{q}, u)$

- Constraint: Ensure $\mathbf{M}(q)$ is invertible
- Solution: Bias the diagonal terms to be larger




$+ \text{bias} \cdot \mathbf{I}$

$\mathbf{L}_\theta(q)$

# Structured Modeling Constraints

Mass-matrix

$\mathbf{M}_\theta(q)$

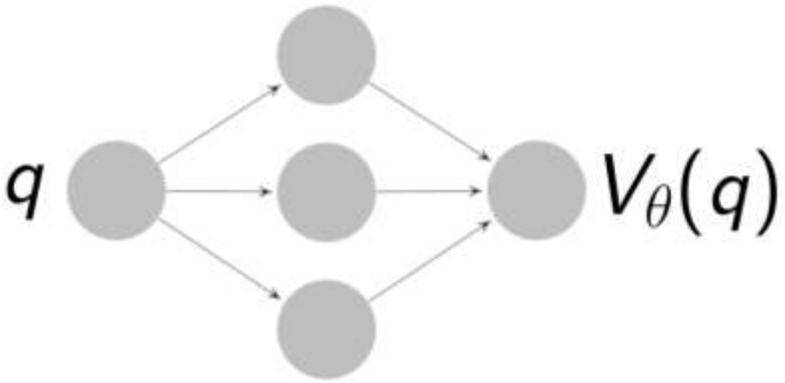
Potential Energy

$V_\theta(q)$

Generalized Force

$F_\theta(q, \dot{q}, u)$

- Constraint: Ensure $V_\theta(q)$ is at least $\mathcal{C}^2$
- Solution: Use appropriate activation functions




| | |
|---|---|
| ReLU | ✗ |
| tanh | ✓ |
| sigmoid | ✓ |
| ⋮ | |

# Structured Modeling Constraints

Mass-matrix

$\mathbf{M}_{\boldsymbol{\theta}}(q)$

Potential Energy

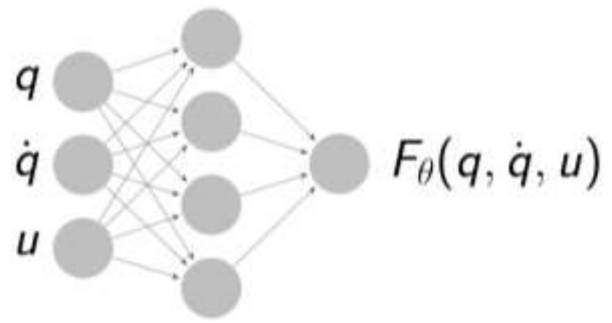
$V_\theta(q)$

Generalized Force

$F_\theta(q, \dot{q}, u)$

- Constraint: No constraints
- Solution: If you know nothing, use a black box function approximator

Black-box



# Structured Modeling Constraints

Mass-matrix

$$\mathbf{M}_{\boldsymbol{\theta}}(q)$$

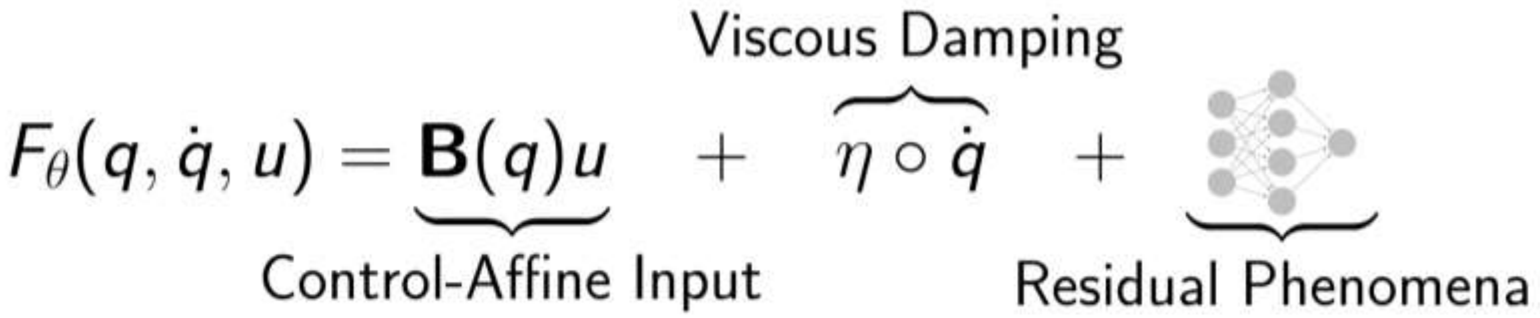
Potential Energy

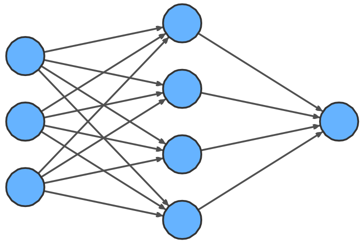
$$V_{\theta}(q)$$

Generalized Force

$$F_{\theta}(q, \dot{q}, u)$$

- Constraint: Have prior knowledge
- Solution:

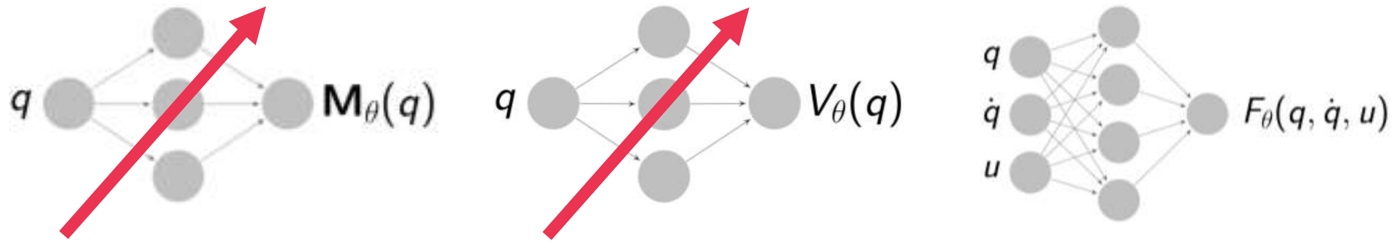


# Representation Capacity

**Black-box**

**Structured Black-box**


$$\ddot{q} = f(q, \dot{q}, u)$$

$$\ddot{q} = \mathbf{M}^{-1}(q)\big[F(q, \dot{q}, u) \; - \mathbf{C}(q, \dot{q})\dot{q} + \nabla_q V(q)\big]$$

Structured Neural Network has the same representation capacity as the Black-box Neural Network

# Learning

- Given $\{\dots, q_t, \dot{q}_t, u_t, q'_t, \dot{q}'_t, \dots\}$

- Minimize

$$L(\theta) = \sum_t \underbrace{(\hat{q}'_t - q'_t)^2}_{\text{Position Discrepancy}} + \lambda \underbrace{(\hat{\dot{q}}'_t - \dot{q}'_t)}_{\text{Velocity Discrepancy}}$$

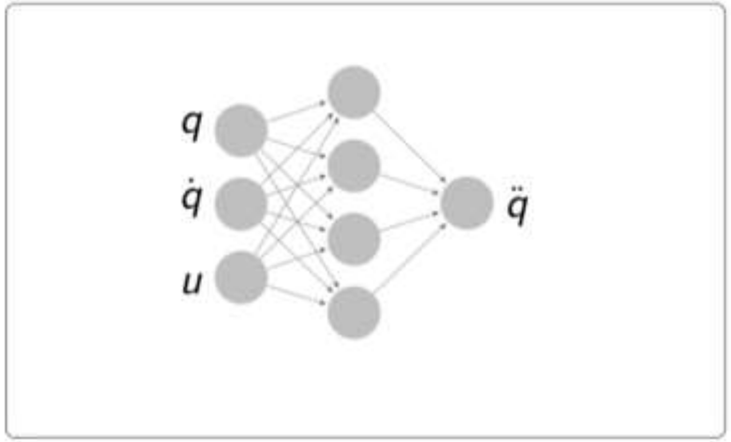
- Where

$$\int_t^{t+\Delta t} f_\theta\big(q(\tau), \dot{q}(\tau), u(\tau)\big) d\tau \to \hat{q}_{t+1}, \hat{\dot{q}}_{t+1}$$

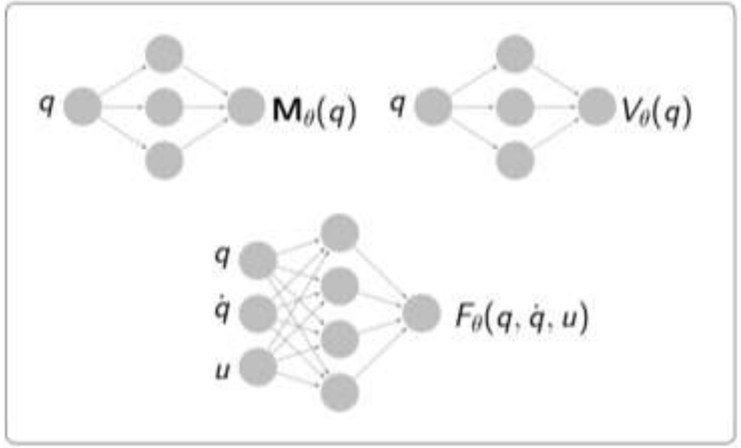
Train via SGD

# Prior Knowledge and Generalization

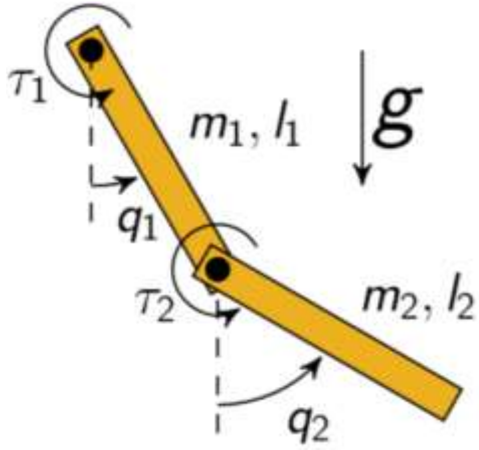
Naïve Black-Box




Structured Black-box




White-box

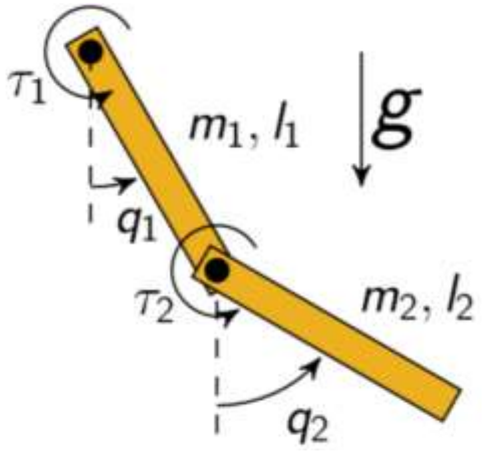



Everything in between

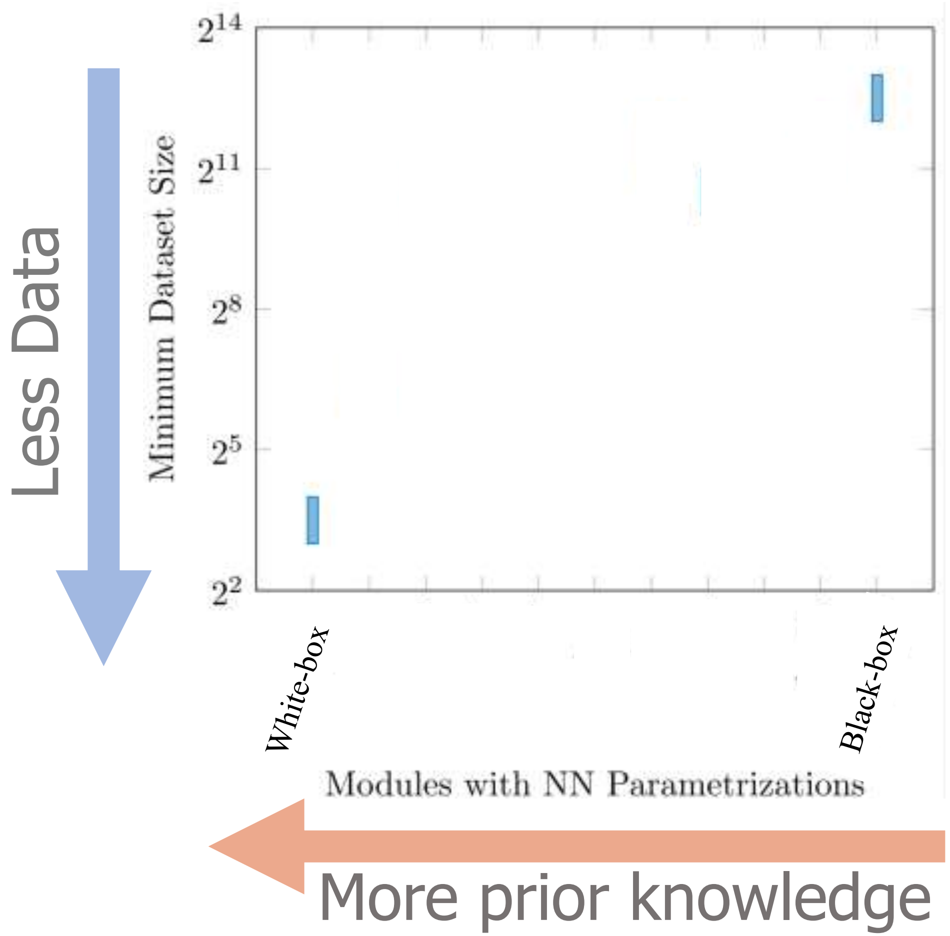
# Experiments

How much data does a given model need to generalize as well as the Naïve model?

# Data Requirements



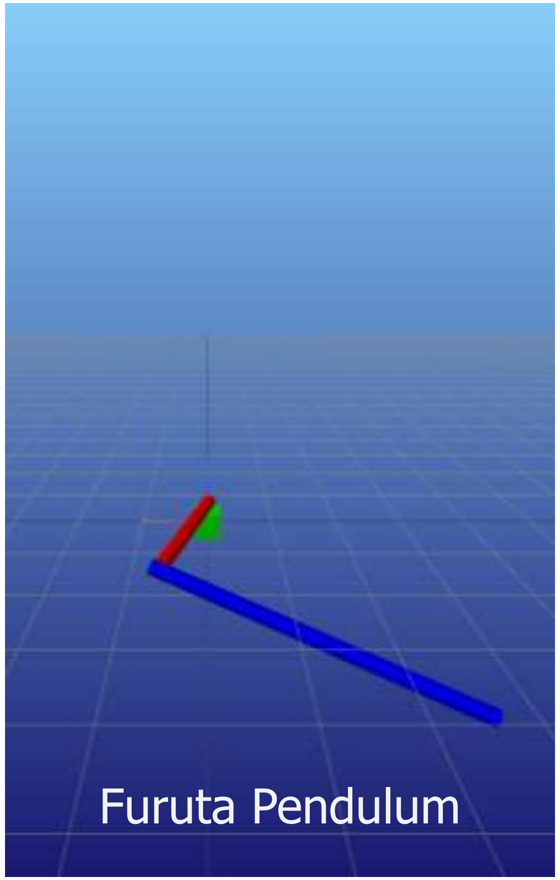

Can get to same performance as a standard NN with orders of magnitude of less data

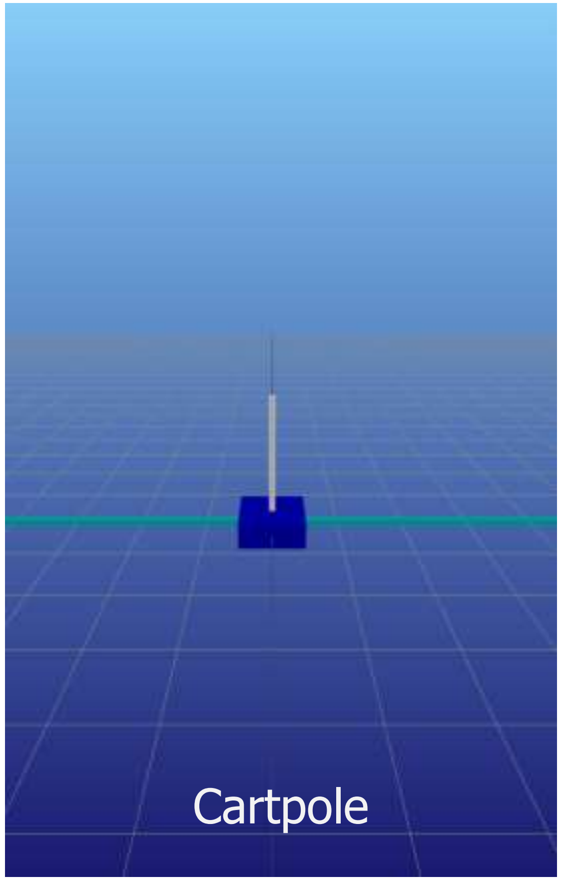


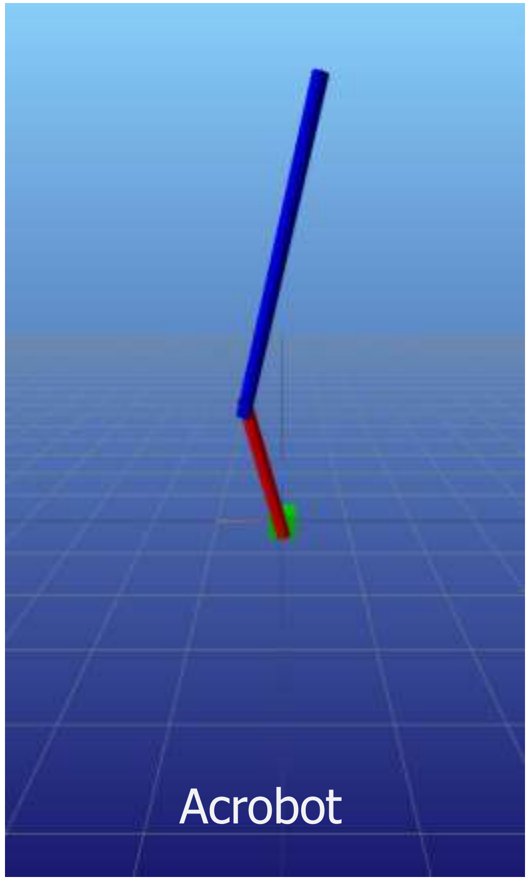
# Experiments

1. Data requirements on Underactuated Dynamics
2. Model-based Control

# Domains



# Recommendation for Mechanical Systems
# **SMM-C**

Mass-matrix

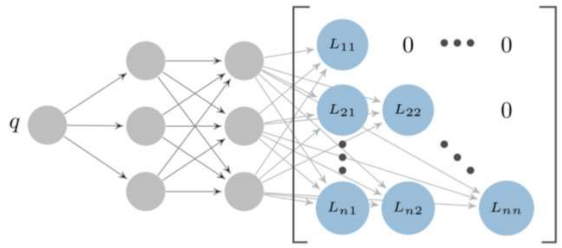
$$\mathbf{M}_{\boldsymbol{\theta}}(q)$$

Potential Energy

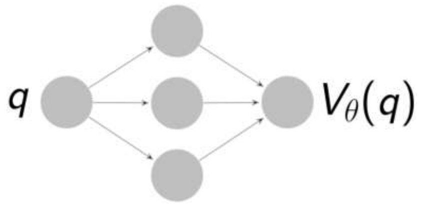
$$V_{\theta}(q)$$

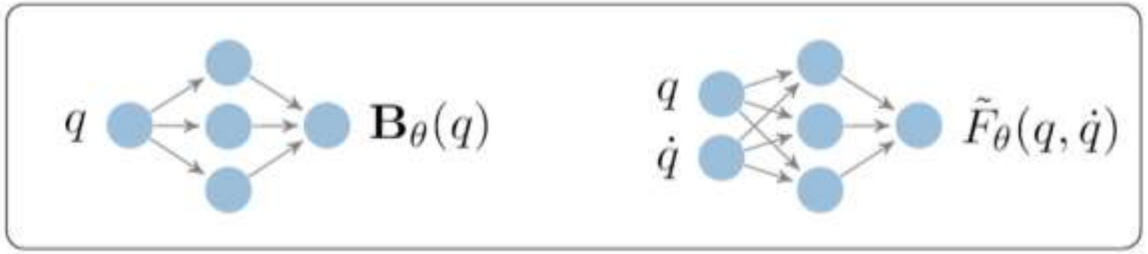
Generalized Force

$$F_{\theta}(q, \dot{q}, u)$$




$$\mathbf{M}_{\boldsymbol{\theta}}(q) = \mathbf{L}_{\boldsymbol{\theta}}(q)\mathbf{L}_{\boldsymbol{\theta}}^{T}(q)$$

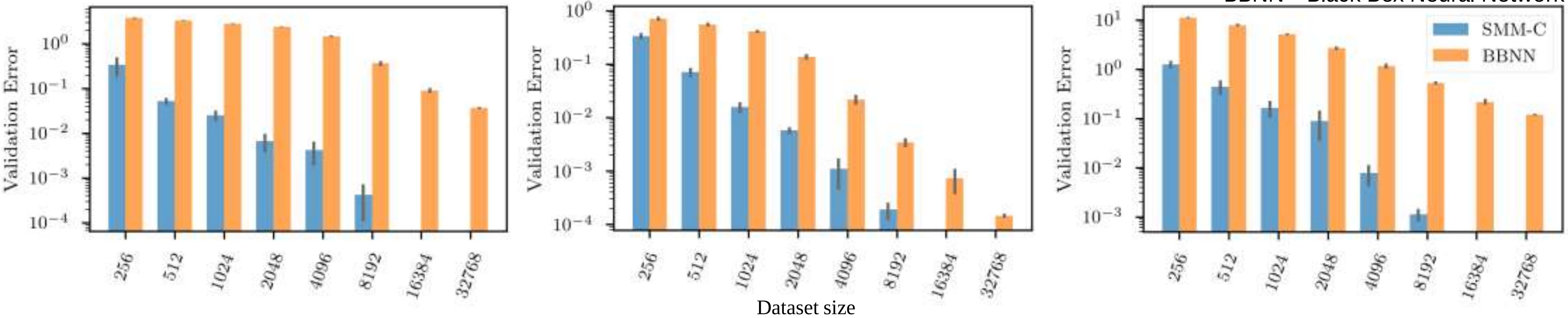


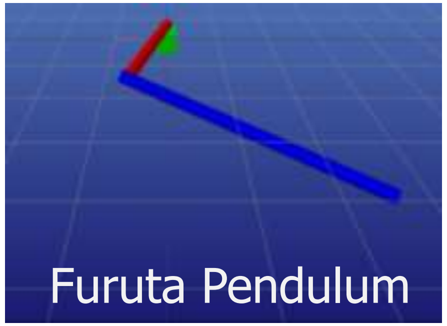



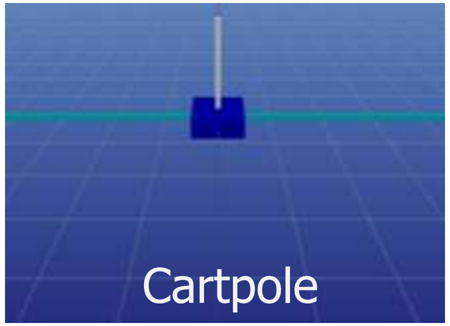

$$F_{\theta}(q, \dot{q}, u) = \boldsymbol{B}_{\theta}(q)u + \tilde{F}_{\theta}(q, \dot{q})$$

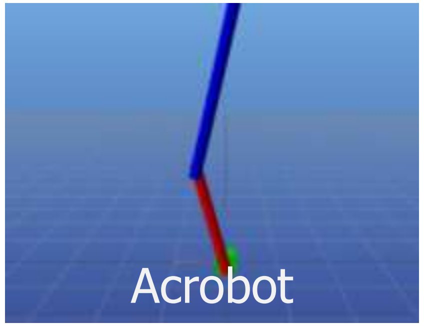
# Underactuated Dynamics

BBNN = Black-Box Neural Network




SMM-C requires a lot fewer samples to achieve same generalization error across multiple domains

# Model-based Control

velocity

$$\ddot{q} = f_\theta(q, \dot{q}, u)$$

position control action

**Trajectory Planner**

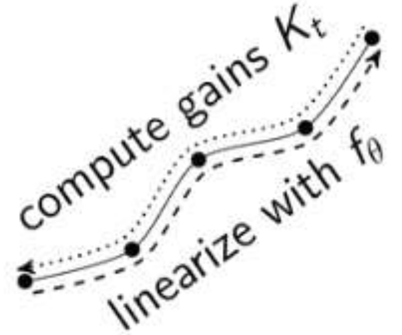
$$\underset{(q_t, \dot{q}_t), u_t}{\text{maximize}} \sum_{t=0}^{T} R(q_t, \dot{q}_t, u_t)$$

$$\xrightarrow[\bar{u}]{\bar{x} = (\bar{q}, \dot{\bar{q}})}$$

Nominal Trajectory

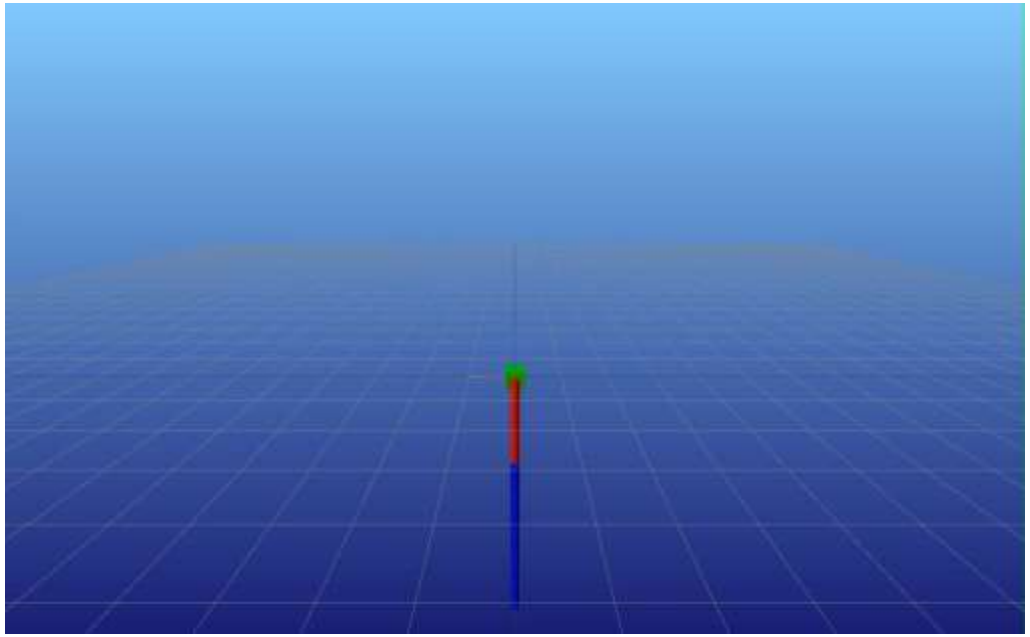
**Trajectory Follower**

$$u_t = \pi(x_t) = \bar{u}_t - K_t(x_t - \bar{x}_t)$$

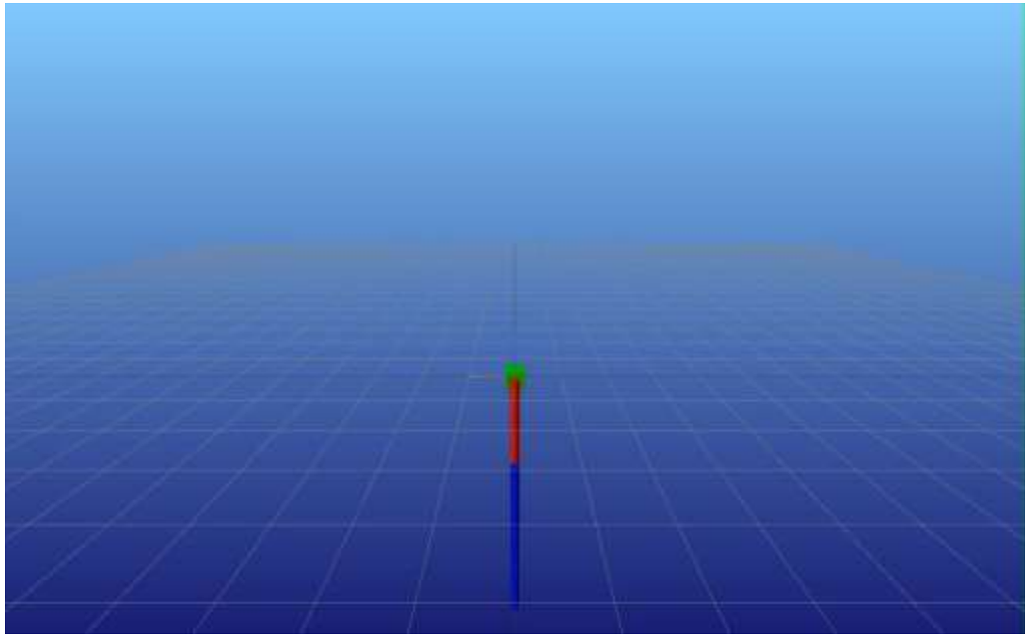



+
other constraints

# Reliable Model-based Control

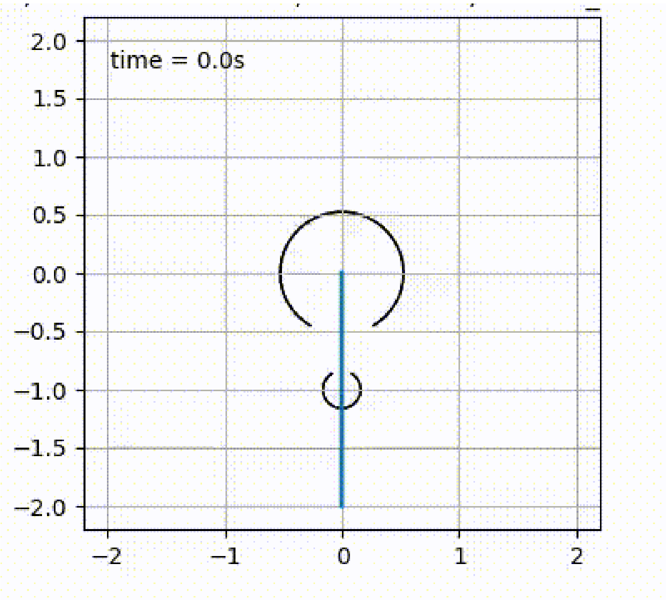


Black-box



SMM-C

For the same generalization error SMM-C model leads to more reliable control than Black-box model

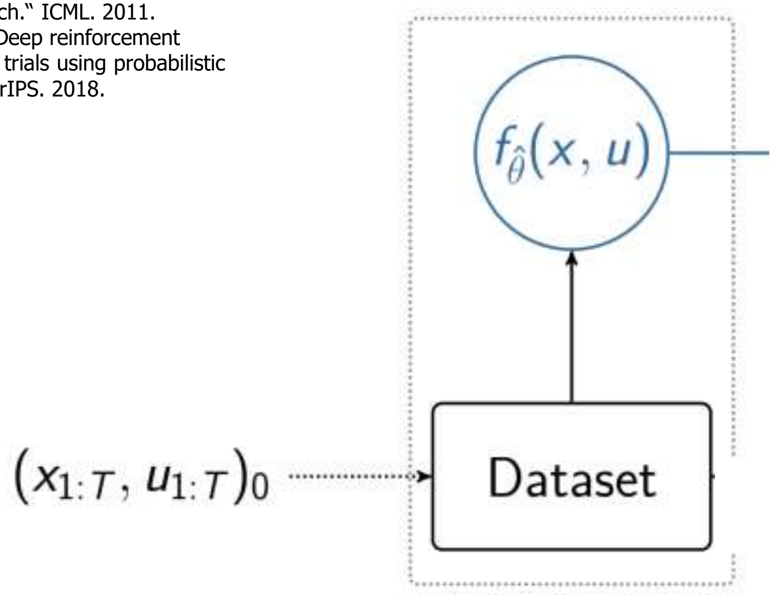



# Experiments

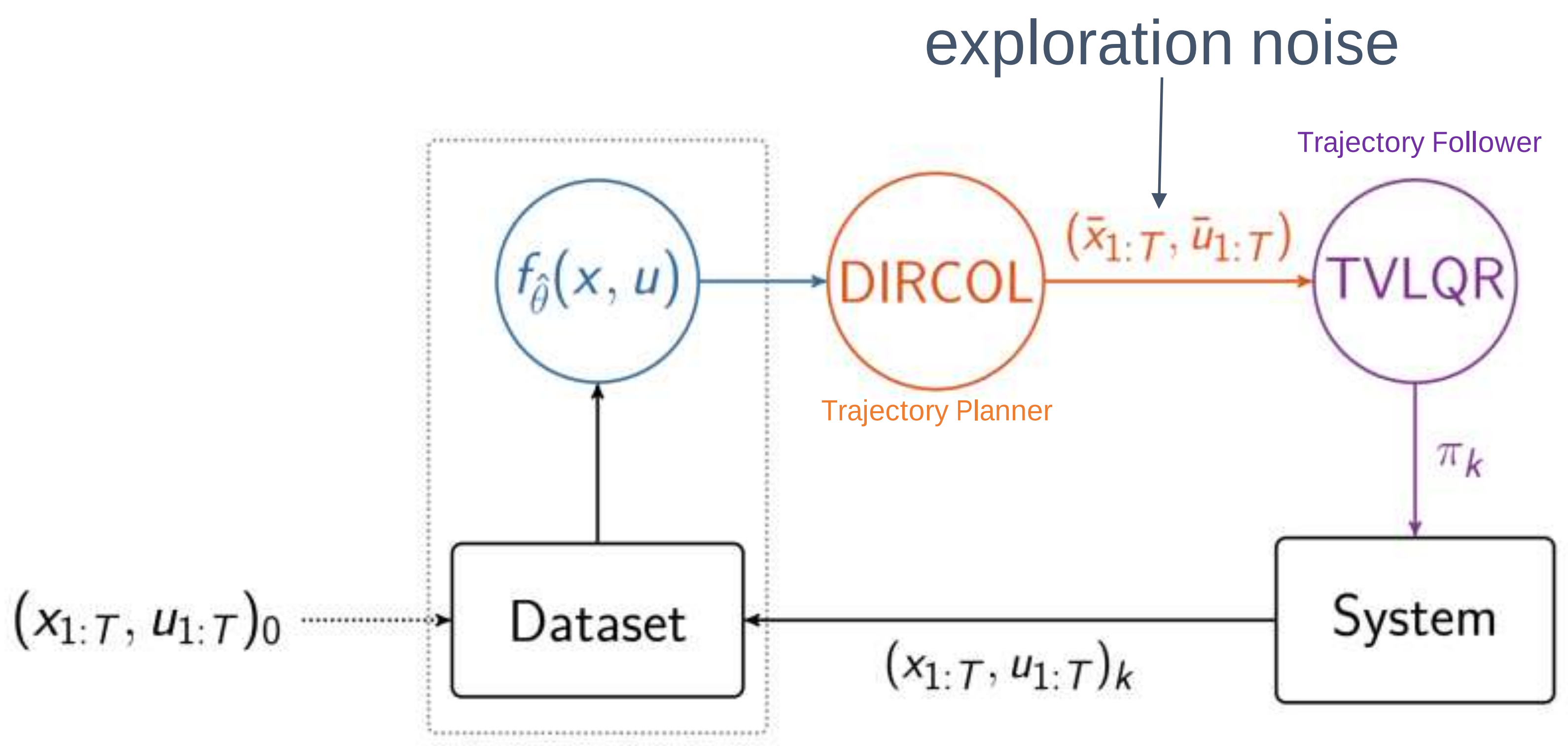
3. Model-based Reinforcement Learning

# Model-based RL Loop


1. Sutton, Richard S. "Integrated architectures for learning, planning, and reacting based on approximating dynamic programming." Machine Learning. 1990.
2. Paduraru, Cosmin. "Planning with Approximate and Learned Models of Markov Decision Processes." These de maıtre, University of Alberta. 2007.
3. Deisenroth, Marc, and Carl E. Rasmussen. "PILCO: A model-based and data-efficient approach to policy search." ICML. 2011.
4. Chua, Kurtland, et al. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." NeurIPS. 2018.

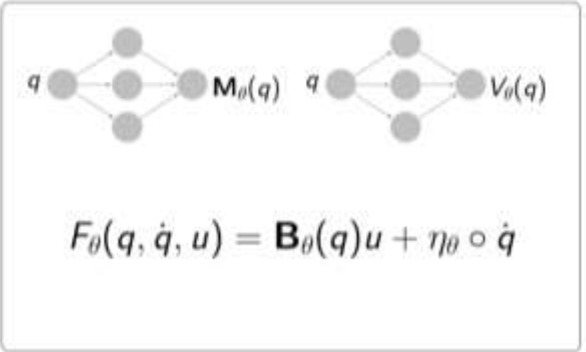



# Completing the Loop

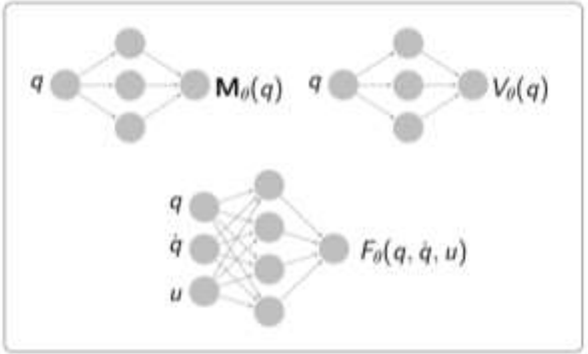


# Comparison Baselines

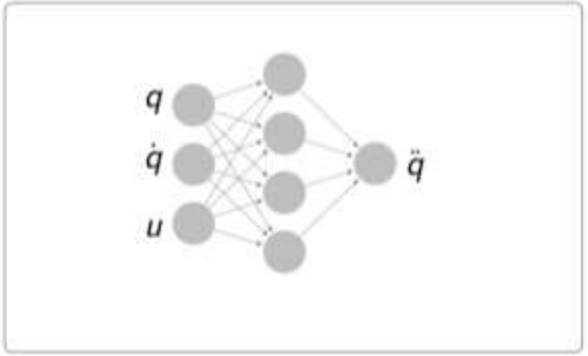
SMM-C




Structured Black-box



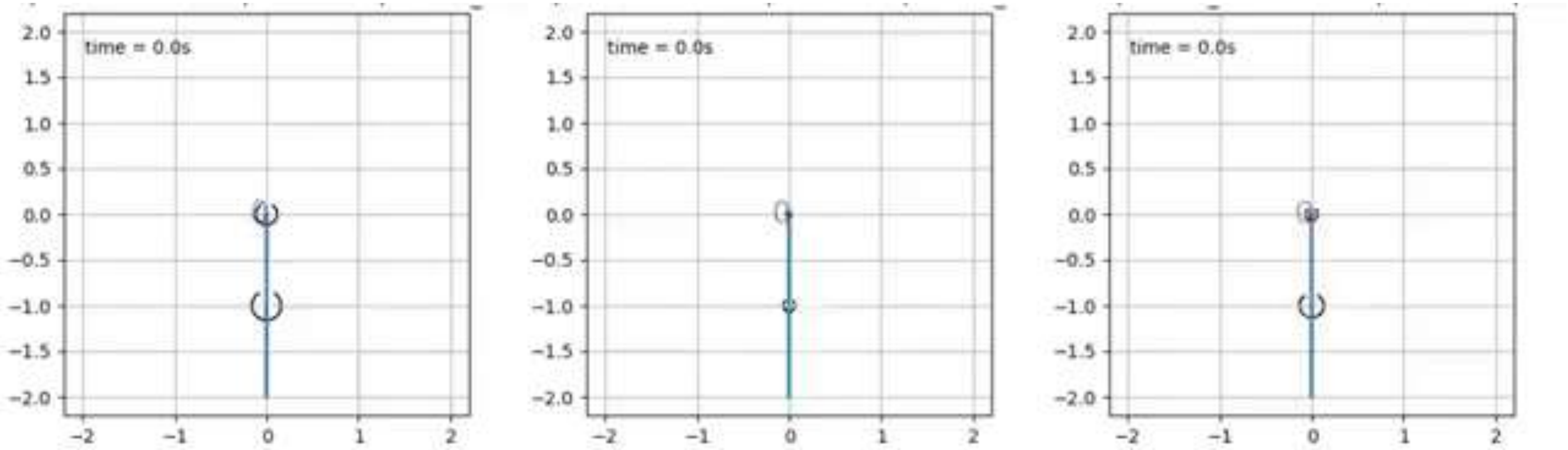

Black-box




More prior knowledge

# Results

Red: Plan from DIRCOL based on Learned Model
Purple: Actual Trajectory
Black Curve at joints: Applied Force Magnitude



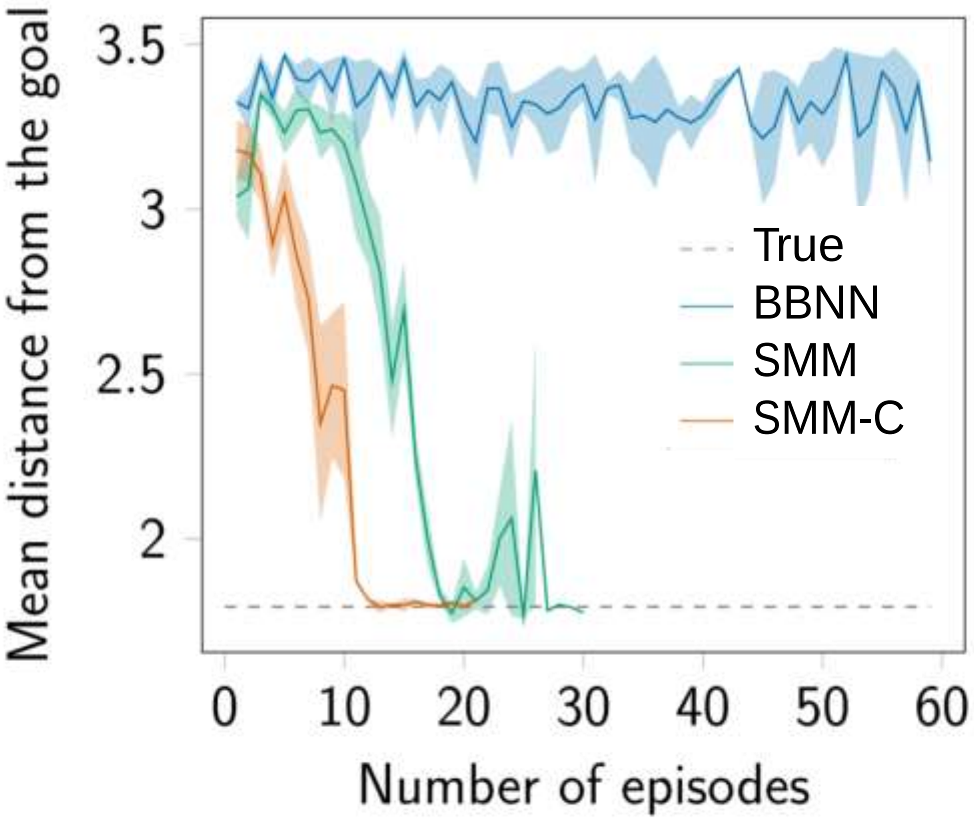

SMM-C

Structured Black-box

Black-box

SMM-C learns the quickest, and then Structured Black-box.
Naïve Black-box isn't able to explore to learn a good model

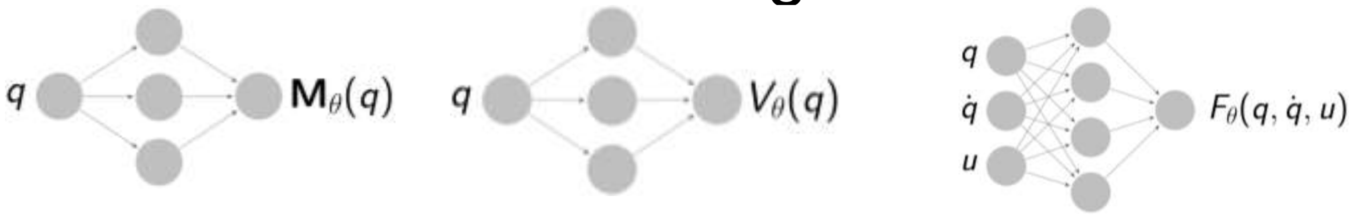
# Results



Mean distance from the goal
3.5
3
2.5
2
0
10
20
30
40
50
60
Number of episodes
True
BBNN
SMM
SMM-C

# Take-aways

- Structured black-box model for general mechanical systems

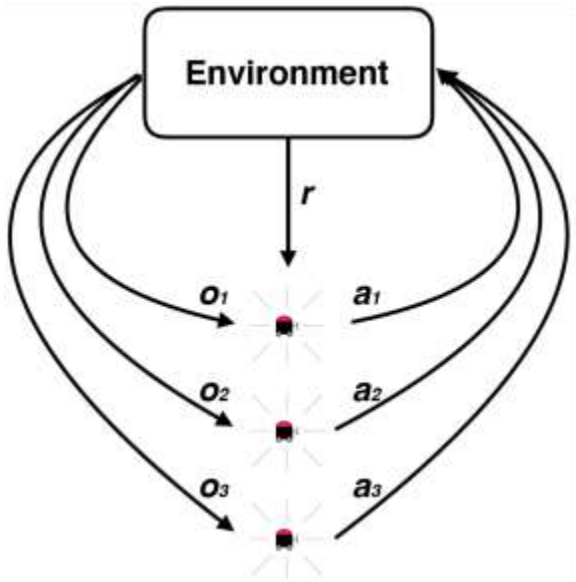



- Easily incorporate prior knowledge

$$F_\theta(q, \dot{q}, u) = \underbrace{\mathbf{B}(q)u}_{\text{Control-Affine Input}} + \overbrace{\eta \circ \dot{q}}^{\text{Viscous Damping}} + \underbrace{\qquad}_{\text{Residual Phenomena}}$$

- Model parametrization effective for long term planning
- Significant sample efficiency gains with model-based RL

2 Agent-informed modules

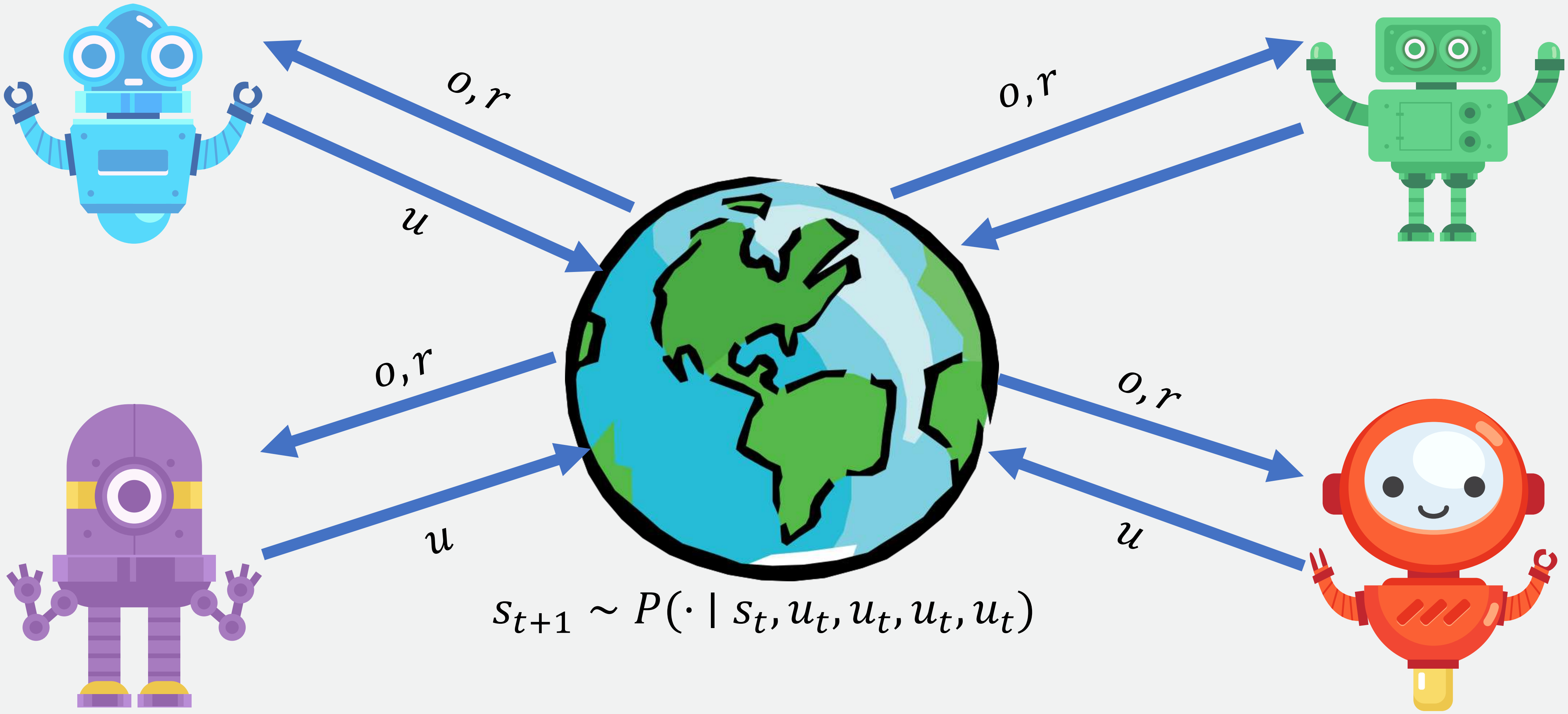



# Learning in Teams

Joint work with Maxim Egorov and Mykel Kochenderfer

AAMAS2017

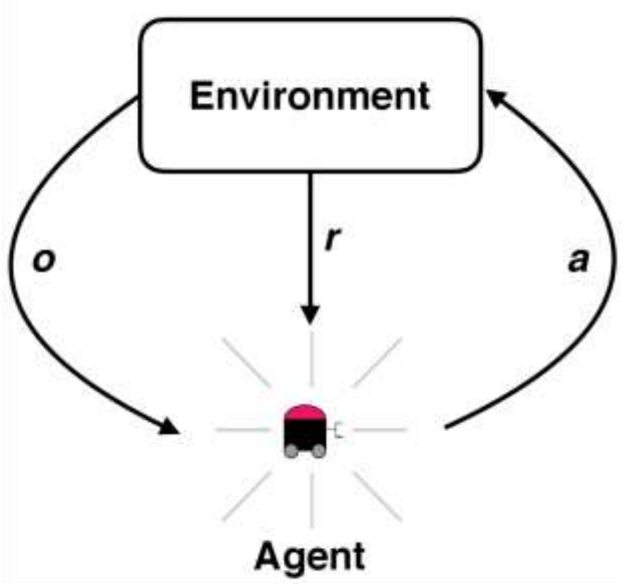
# Multi-agent Systems



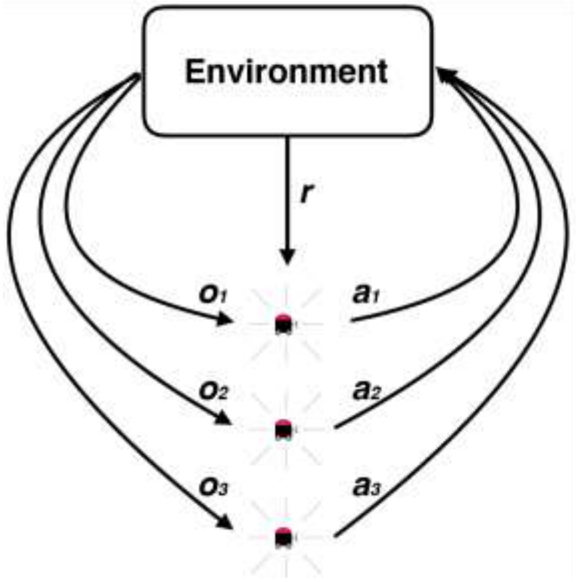

$$s_{t+1} \sim P(\cdot \mid s_t, u_t, u_t, u_t, u_t)$$

# Problem

Single Agent

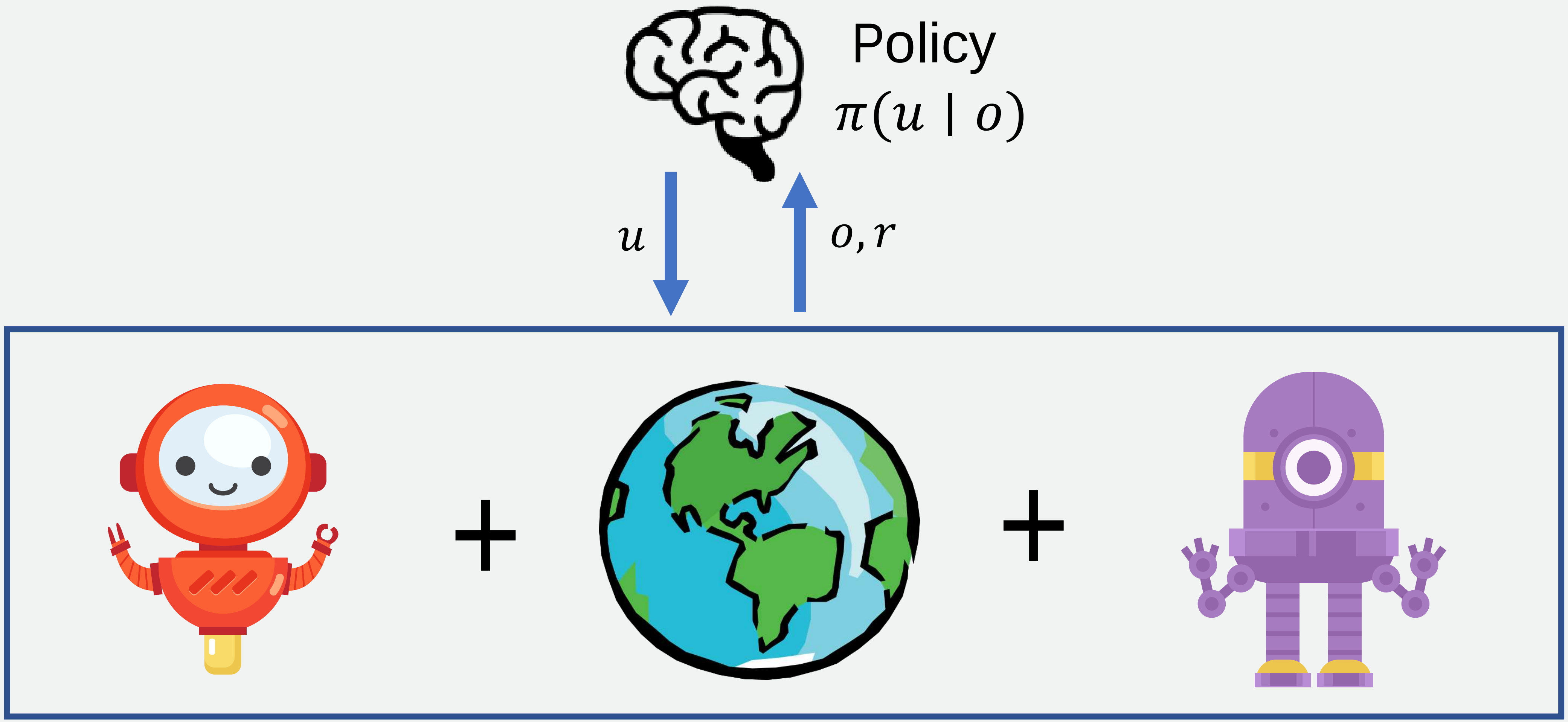



Multi-Agent




Multiple agents coordinate to achieve a shared objective

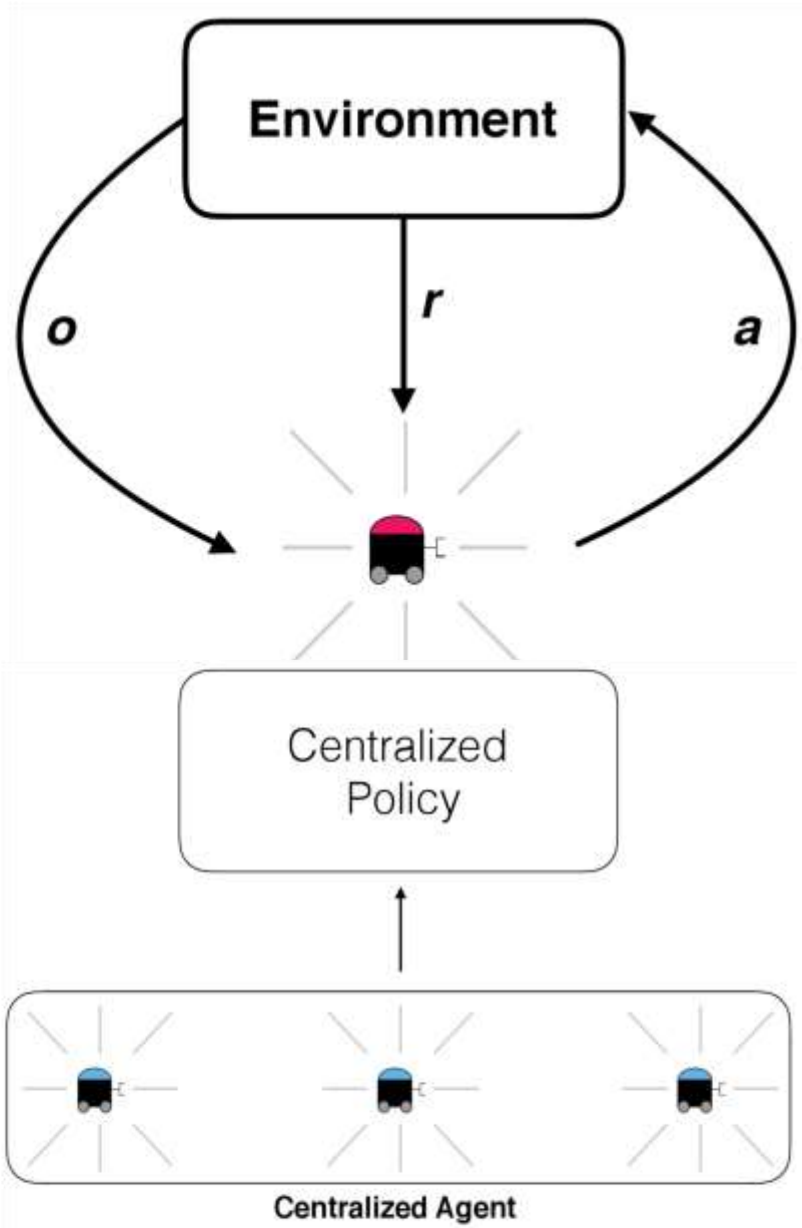
# Team Decision-making



# Previous Work

## Centralized policy

- Reduce the multi-agent problem to a single agent problem with the joint action space



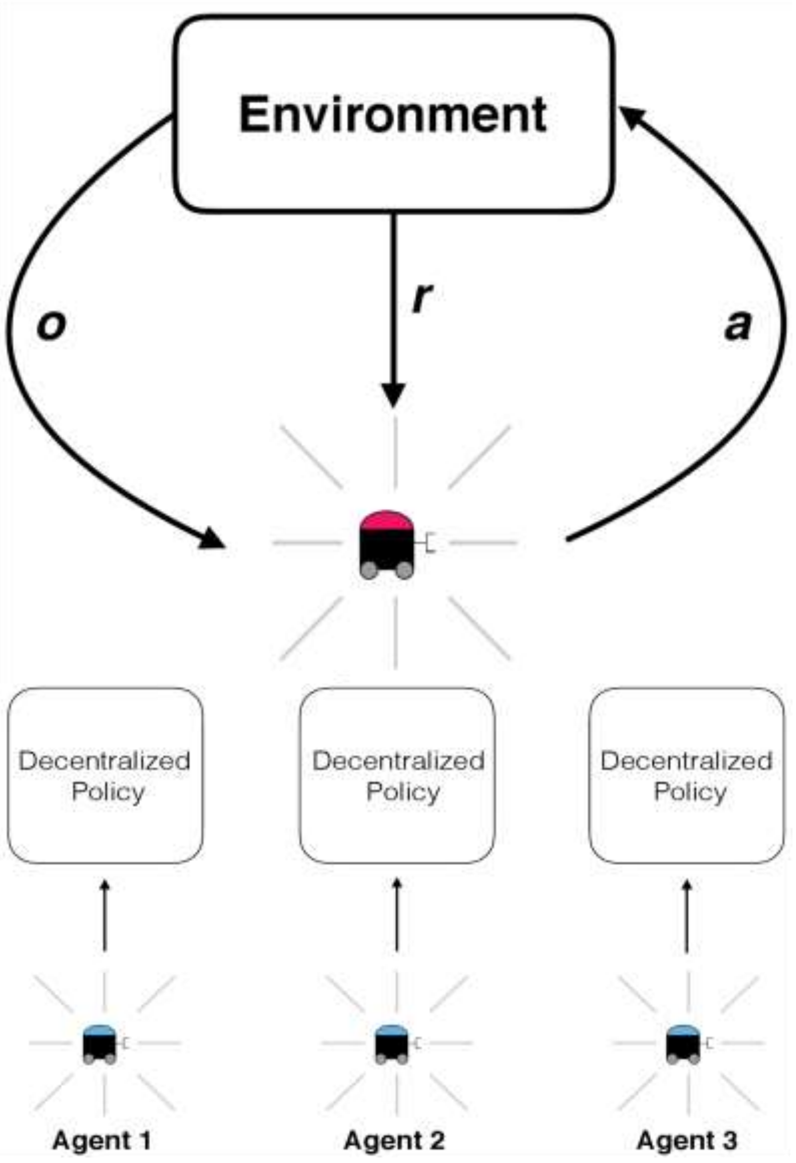

Action space exponential in the number of agents

1. Claus, Caroline, and Craig Boutilier. "The dynamics of reinforcement learning in cooperative multiagent systems." AAAI/IAAI 1998.

# Agents Modules
# Learning to Coordinate

- Each agent receives observations about environment as well as other agents
- Each agent executes local actions

- Coordination emerges from trying to achieve the shared global objective




**Non-stationary dynamics make coordination difficult**


1. Tan, Ming. "Multi-agent reinforcement learning: Independent vs. cooperative agents." ICML. 1993.
2. Sen, Sandip, Mahendra Sekaran, and John Hale. "Learning to coordinate without sharing information." AAAI. 1994.
3. Claus, Caroline, and Craig Boutilier. "The dynamics of reinforcement learning in cooperative multiagent systems." AAAI/IAAI 1998.

# Modeling Non-stationarity as Information Loss

- Modularity is about information encapsulation

- Decentralization causes non-stationarity
  - Individual agent modules need to capture information about other agents' changing behavior

- Let information after learning step $t$ for agent $i$ be $\mathcal{I}_i(t)$

Terry, Justin K., et al. "Parameter Sharing is Surprisingly Useful for Multi-Agent Deep Reinforcement Learning." *arXiv preprint arXiv:2005.13625* (2020).

# Modeling Non-stationarity as Information Loss

- Change in information after a learning step

$$\Delta\mathcal{I}_i(t) = \Delta^{\uparrow}\mathcal{I}_{i,env}(t) + \sum(\Delta^{\uparrow}\mathcal{I}_{i,j}(t) - \Delta^{\downarrow}\mathcal{I}_{ij}(t))$$

Env info learned

Other agent info learned

Other agent info to unlearn

$$\Delta^{\uparrow}\mathcal{I}_{i,j}(t) \propto \text{degree of centralization } \times (\text{coordination required } - \mathcal{I}_{i,j}(t-1))$$

$$\frac{1}{\Delta^{\downarrow}\mathcal{I}_{ij}(t)} = \frac{1}{\Delta^{\uparrow}\mathcal{I}_{i,j}(t)} + \frac{1}{\mathcal{I}_{i,j}(t-1)}$$

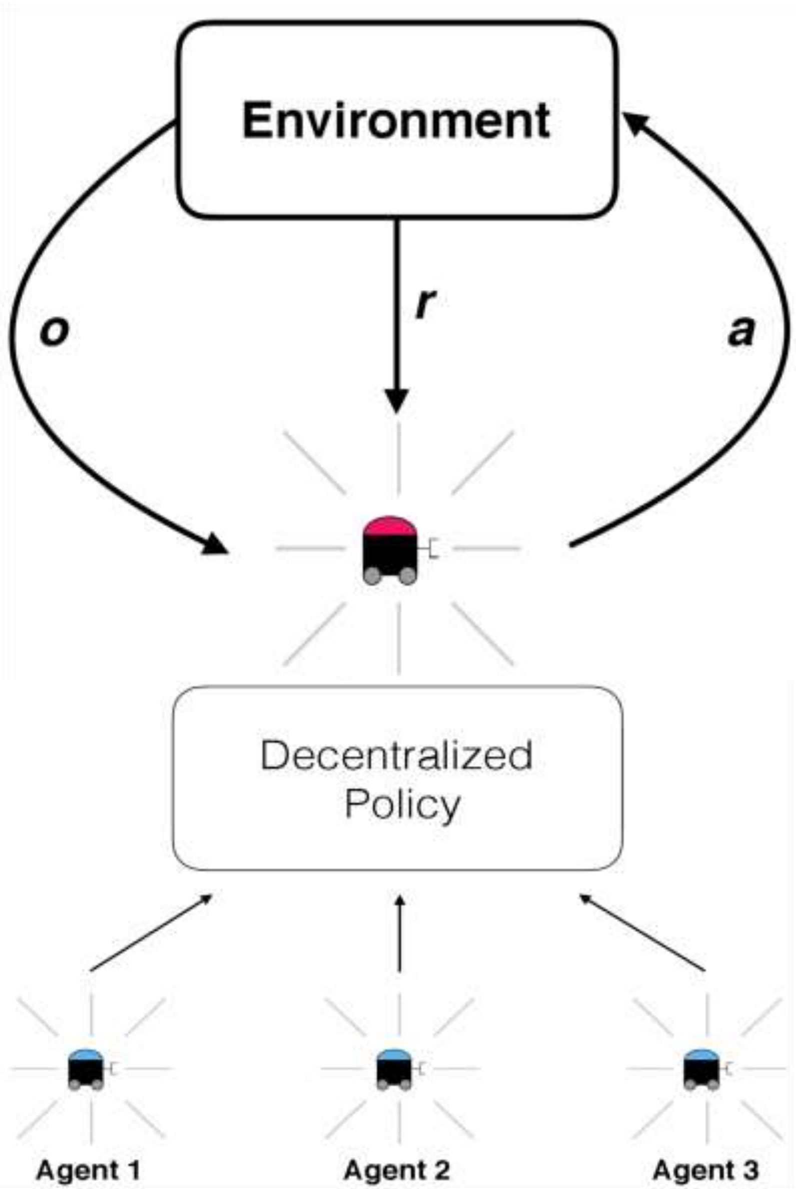
How to centralize this information without full centralization of the policy?

# Parameter Sharing Modules

- Each agent receives observations about environment as well as other agents
- Each agent executes local actions

- Assumption: Homogenous agents

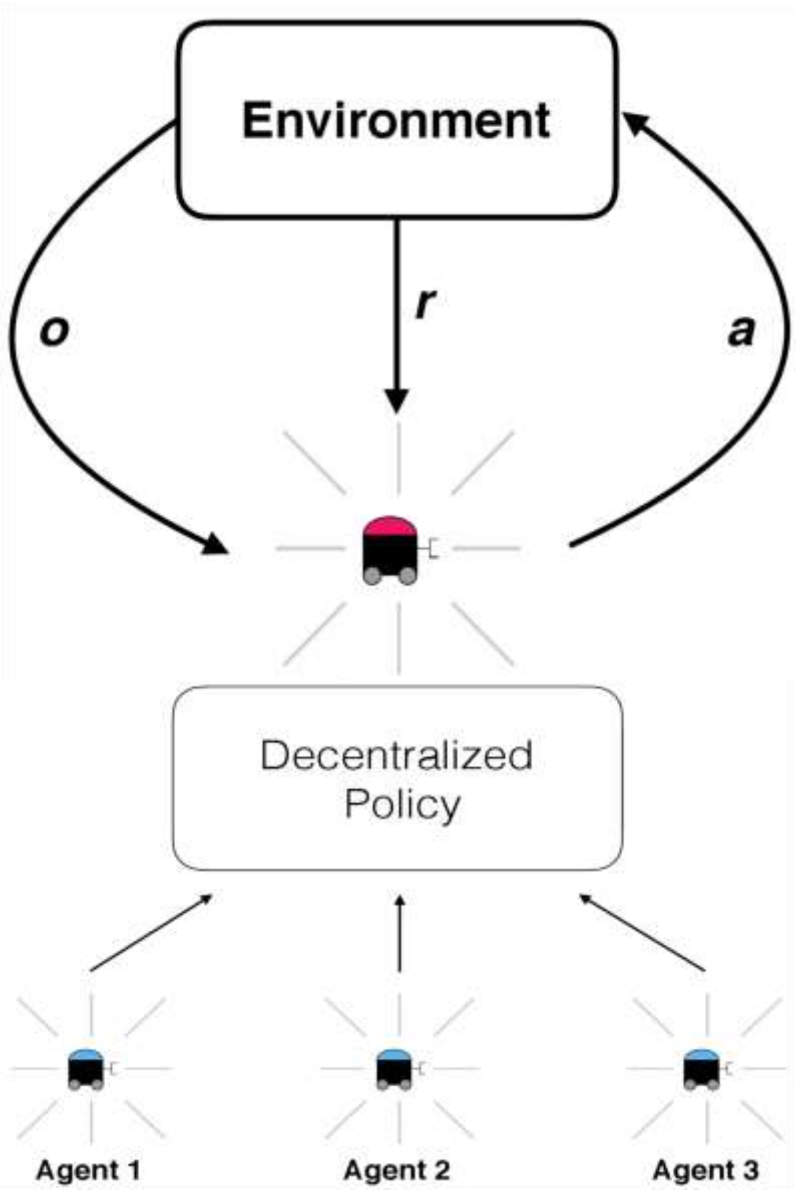
- Effective action space of a single agent
- Mitigate non-stationarity because all information is centralized



# Parameter Sharing Modules

- What if the agents have different roles?
  - Condition policy on the agent's identity

$$\pi(a \mid o, \mathbf{id})$$



# Model-free Reinforcement Learning via Policy Optimization

- Objective

$$J(\theta) = E_{\pi_\theta}\left[\underbrace{\sum_t \gamma^t r_t}_{\text{Return}}\right]$$

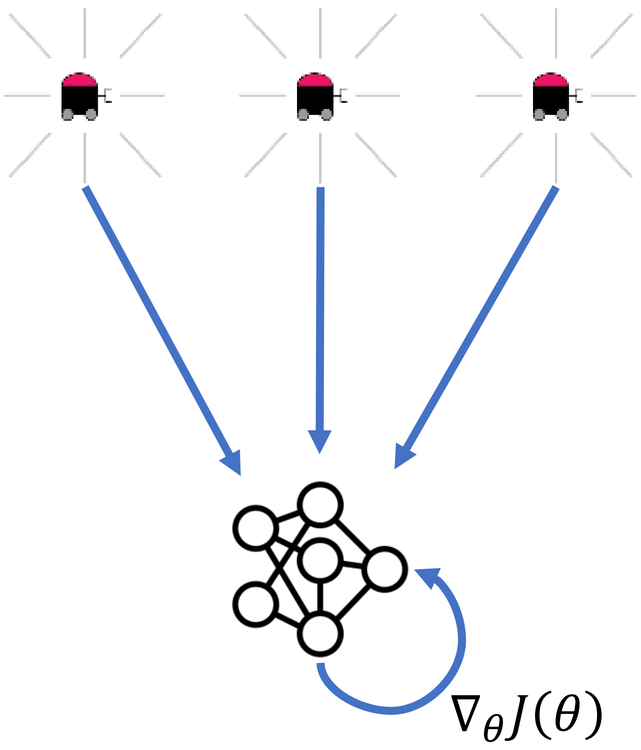
- Policy Gradient

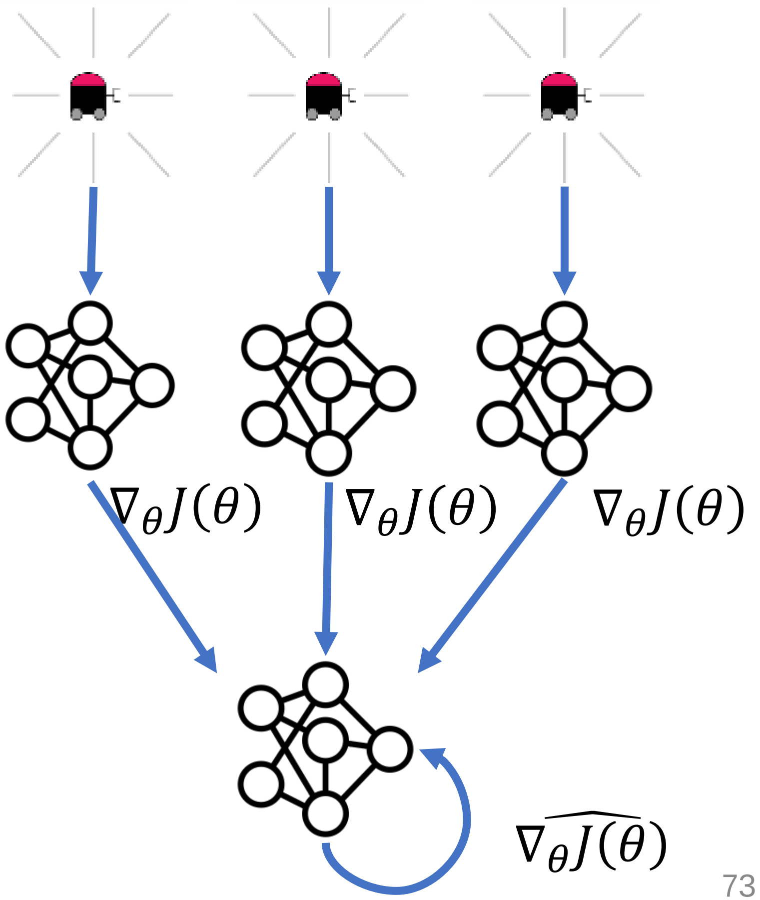
$$\nabla_\theta J(\theta) = E_{\pi_\theta}\left[\sum_t R_t \nabla_\theta \log \pi_\theta(u_t \mid s_t)\right]$$

**Various tricks to stabilize this non-linear stochastic optimization problem**

# Model-free Reinforcement Learning with Parameter Sharing Modules

- Implementation detail

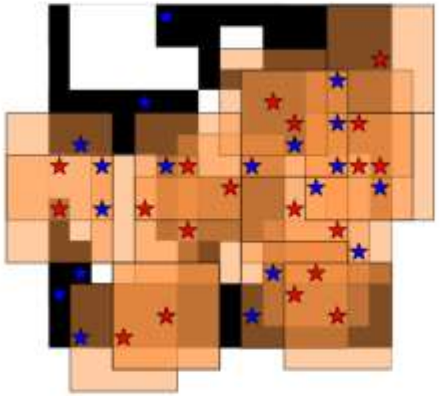


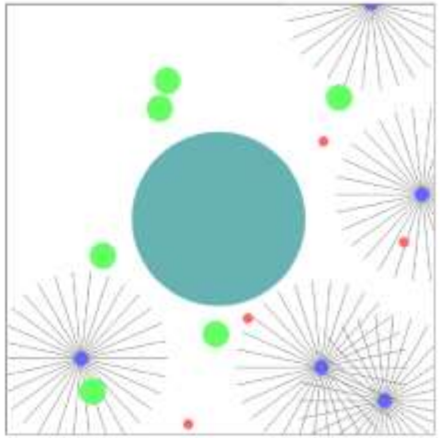
# Experiments

Comparison with fully centralized and decentralized approaches

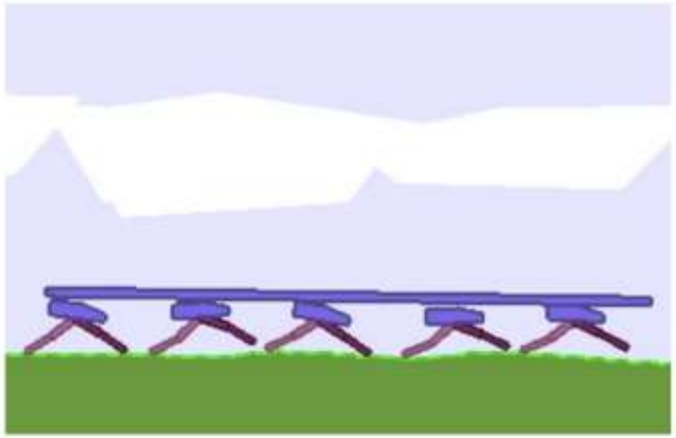
# Problem Domains



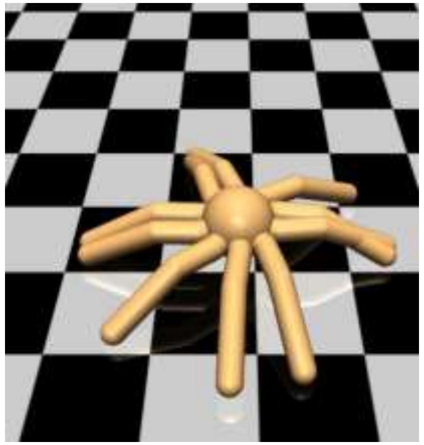
Pursuit



Waterworld

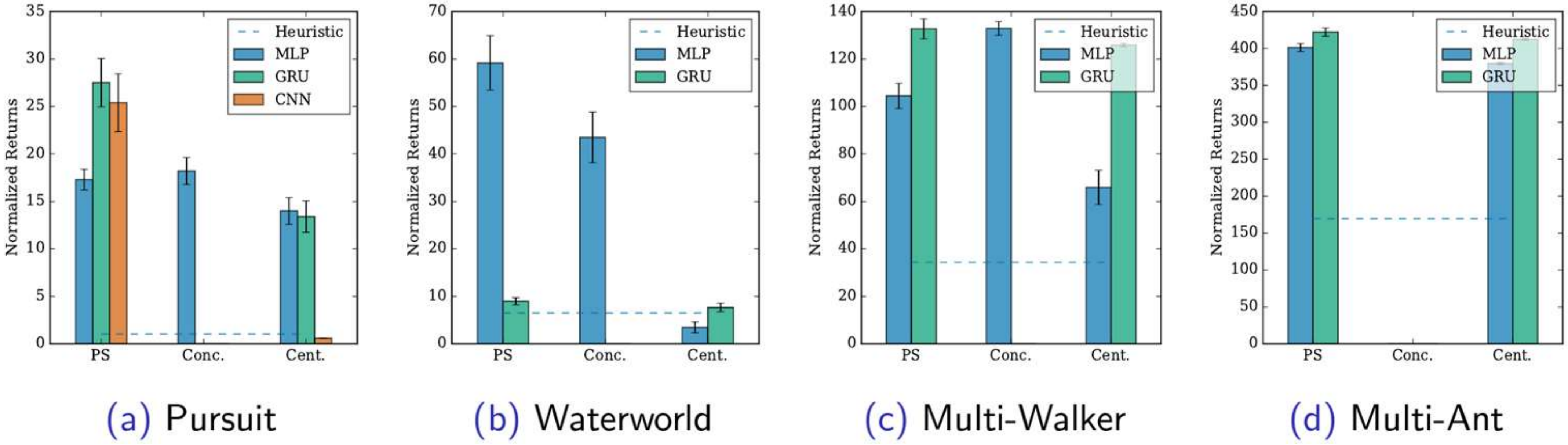


Multi-Walker



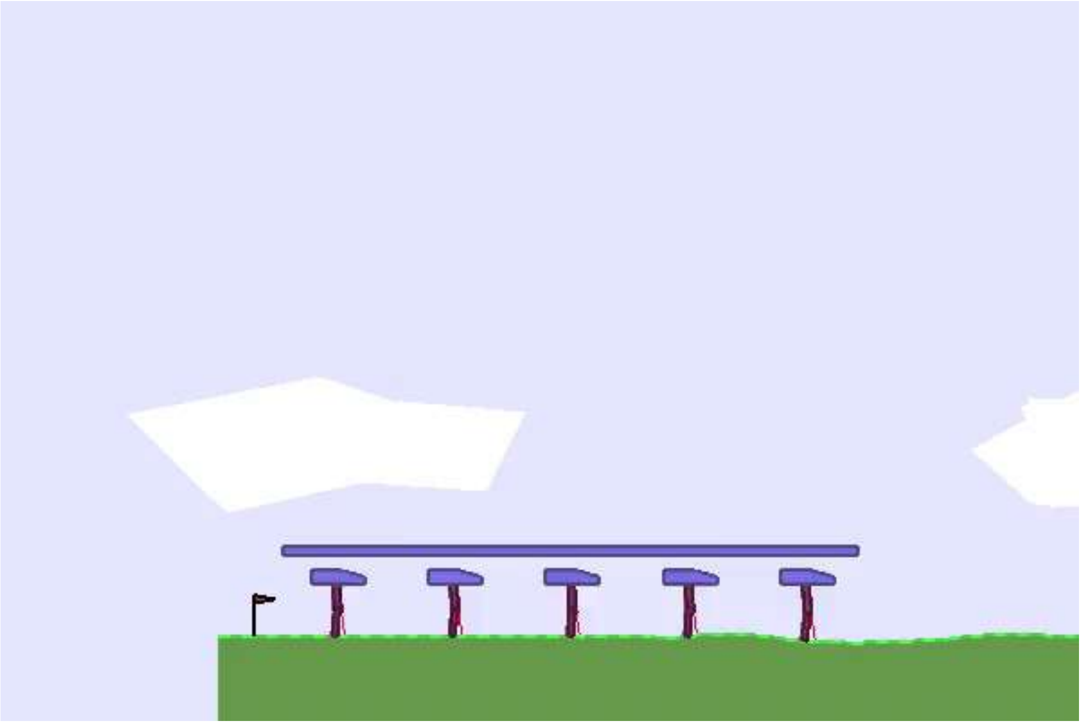
Multi-Ant

# Results




Parameter Sharing performs better or similar to fully decentralized or centralized methods

# Results

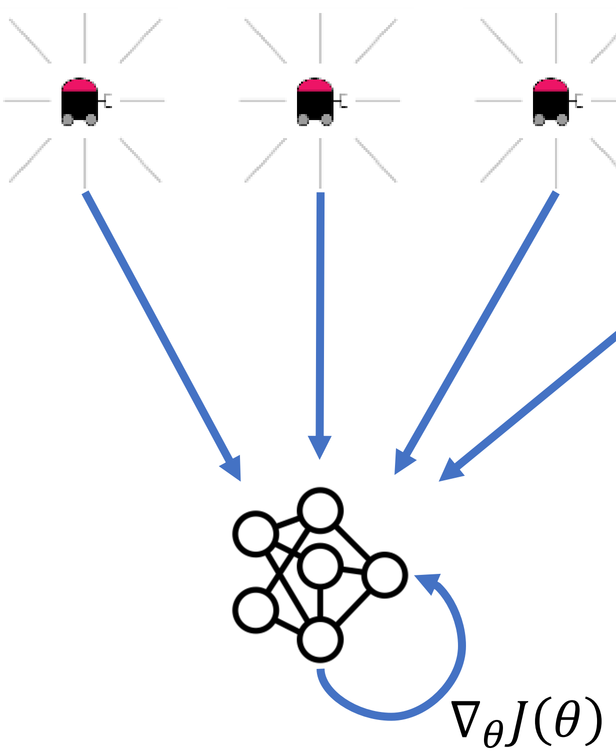


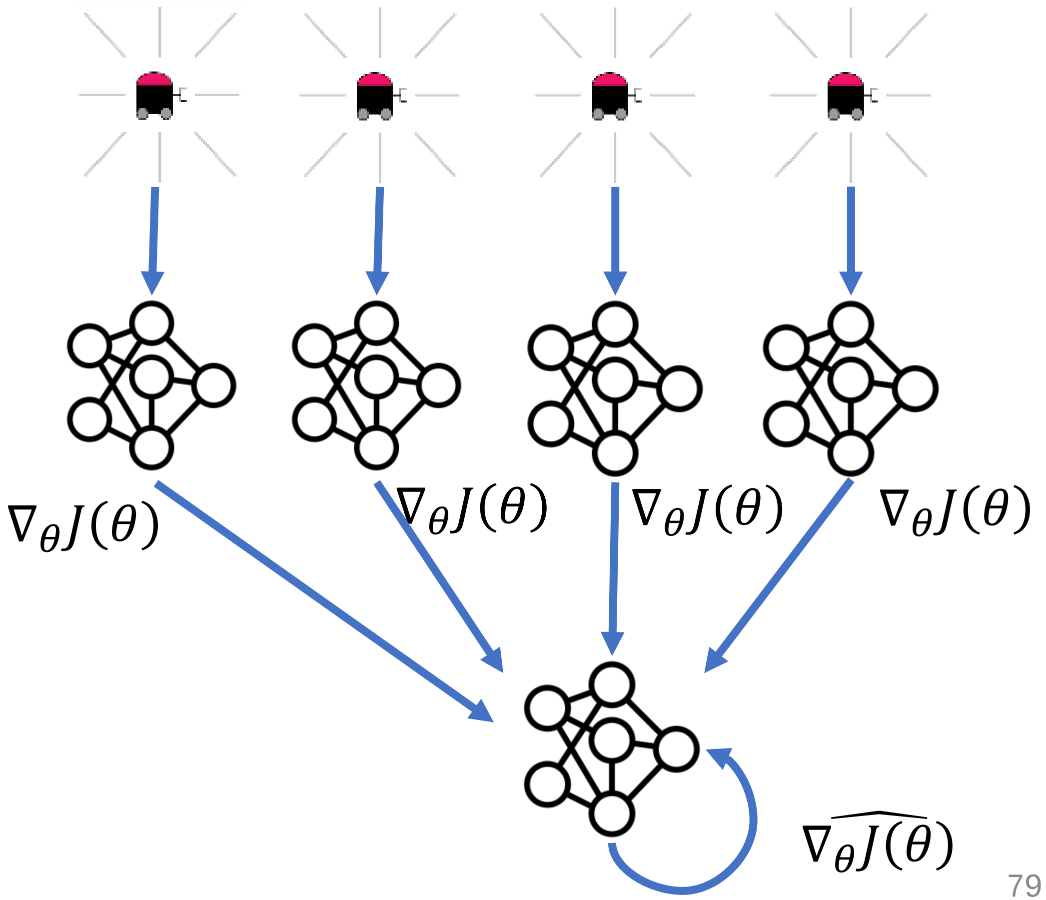
Agents are still not *that* effective in practice

# Scaling to Several Agents

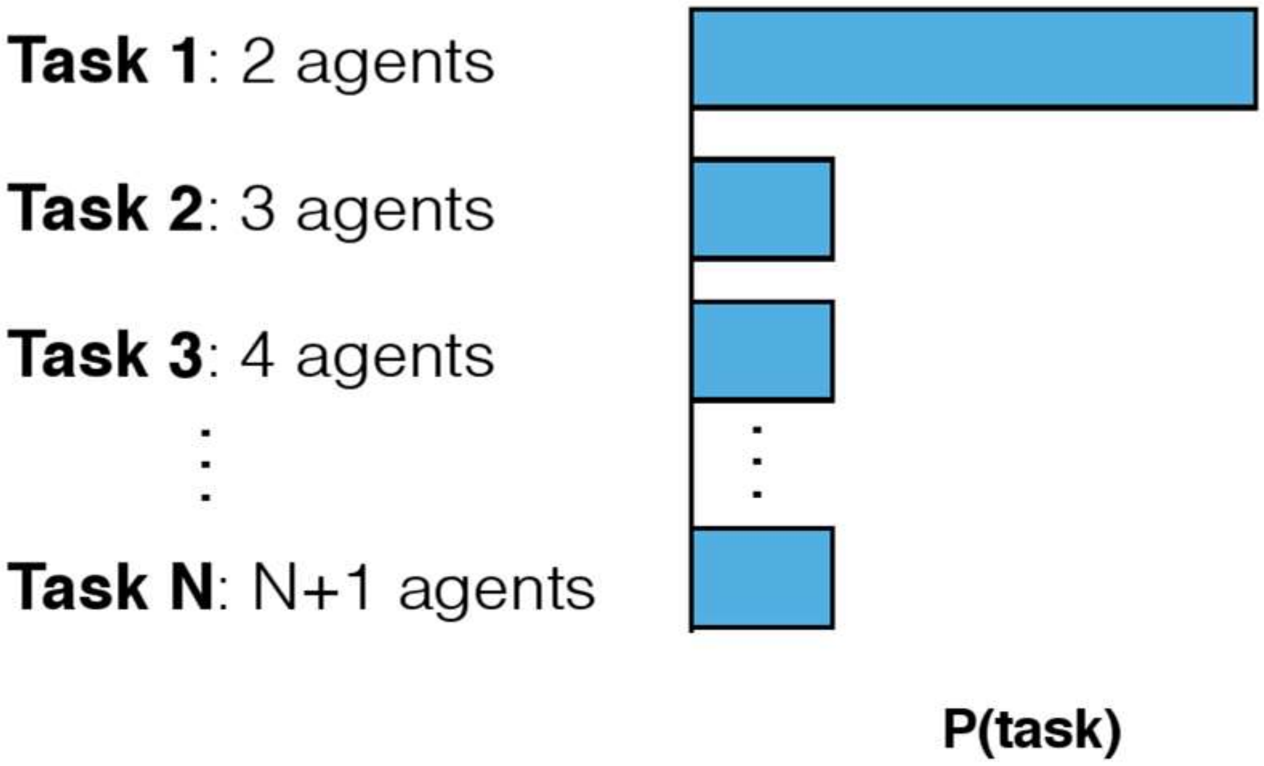
How modularity allows for curriculum learning

# Scaling to Several Agents



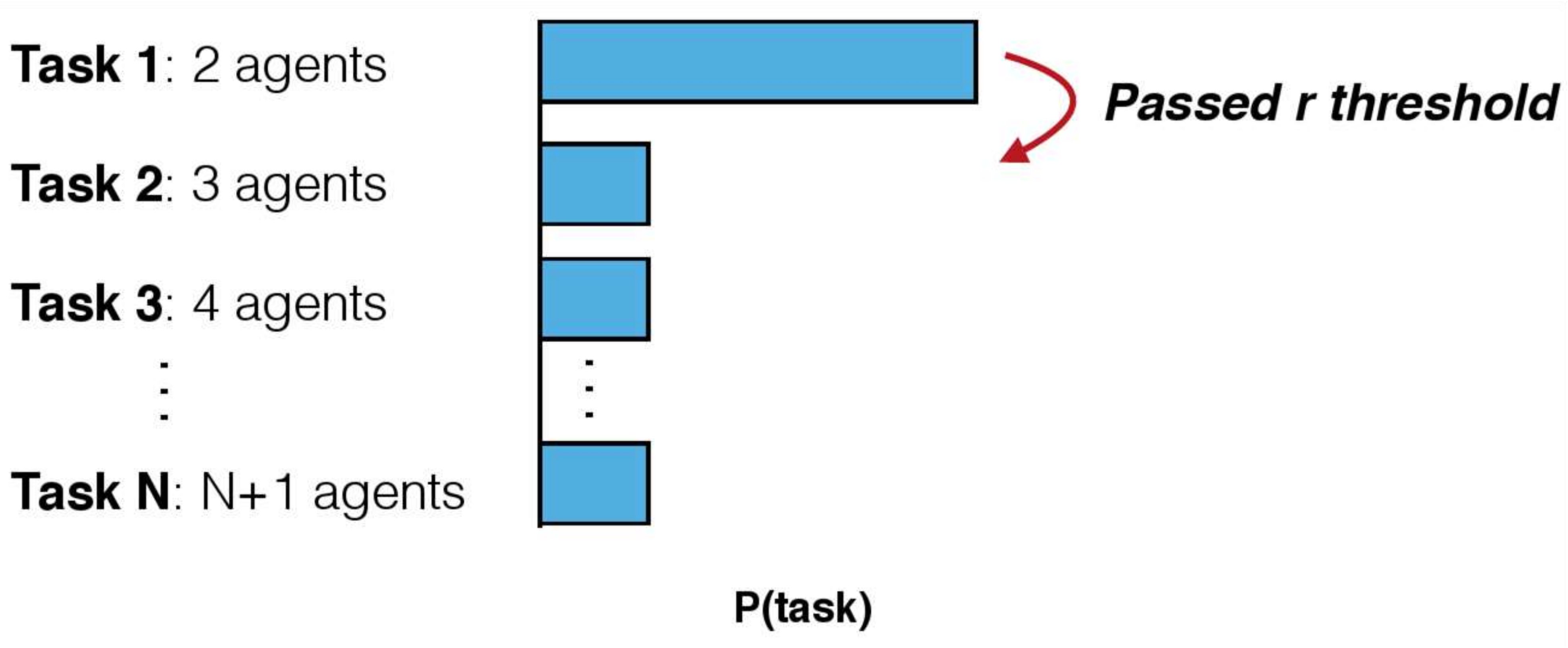



# Scaling to Several Agents

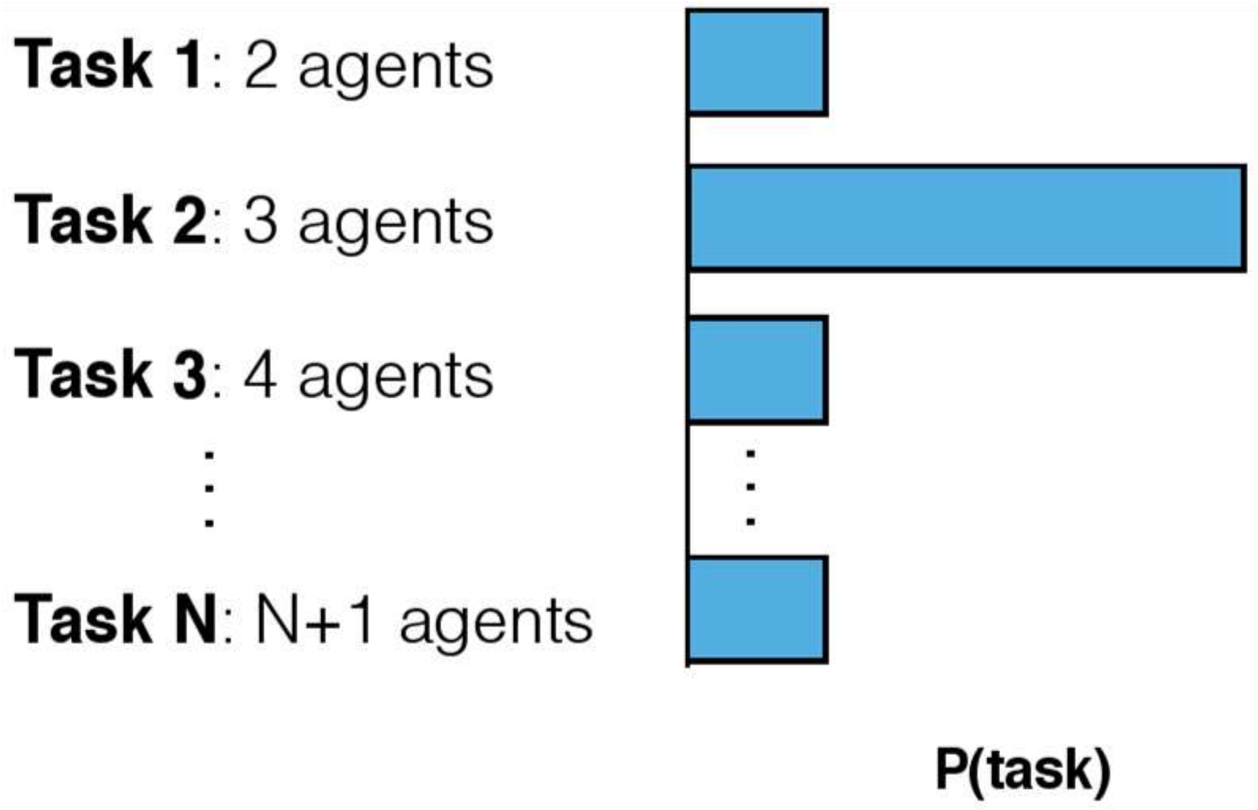



- Difficulty in tasks is defined in terms of number of cooperating agents
- Model the task distribution as a Dirichlet distribution with maximum weight assigned to the current task under consideration
- Use Parameter Sharing Decentralized Learning

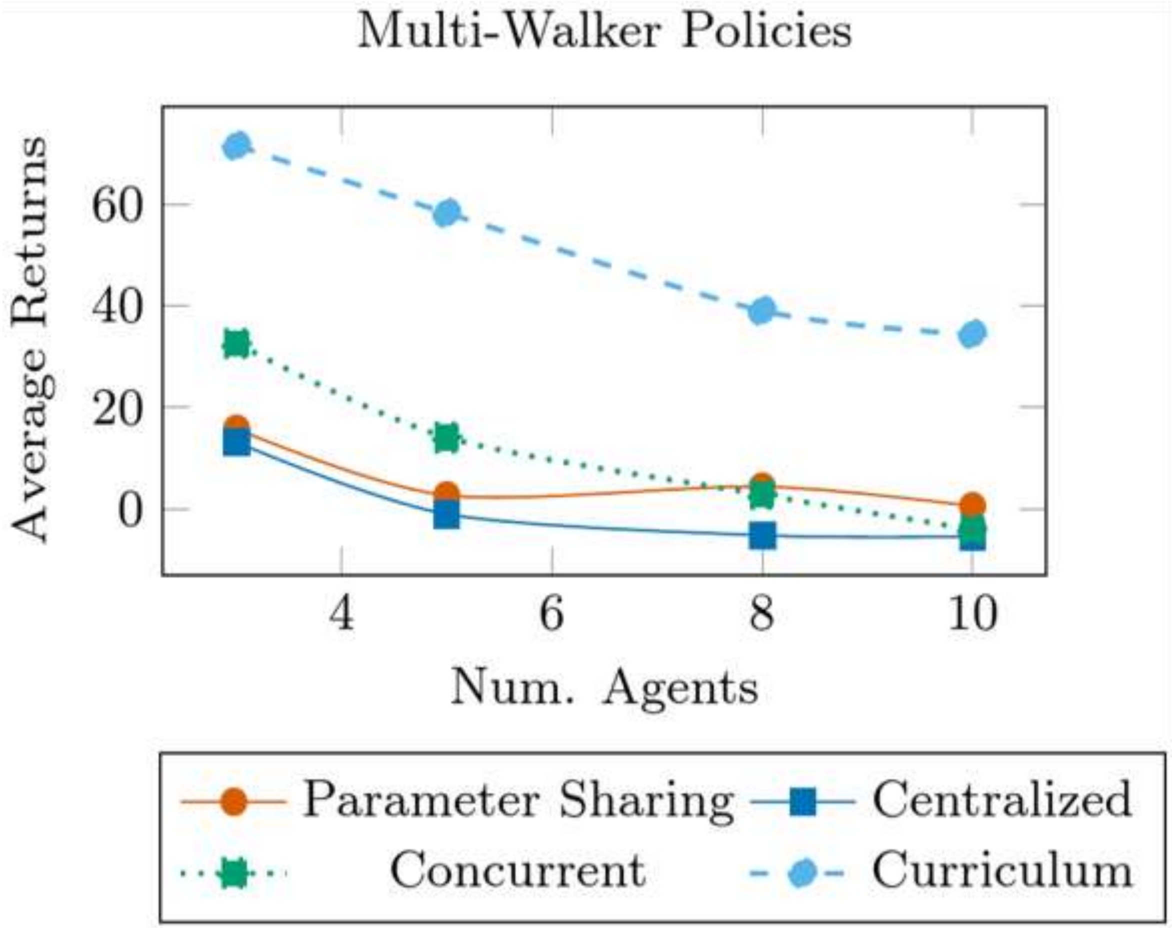
# Scaling to Several Agents

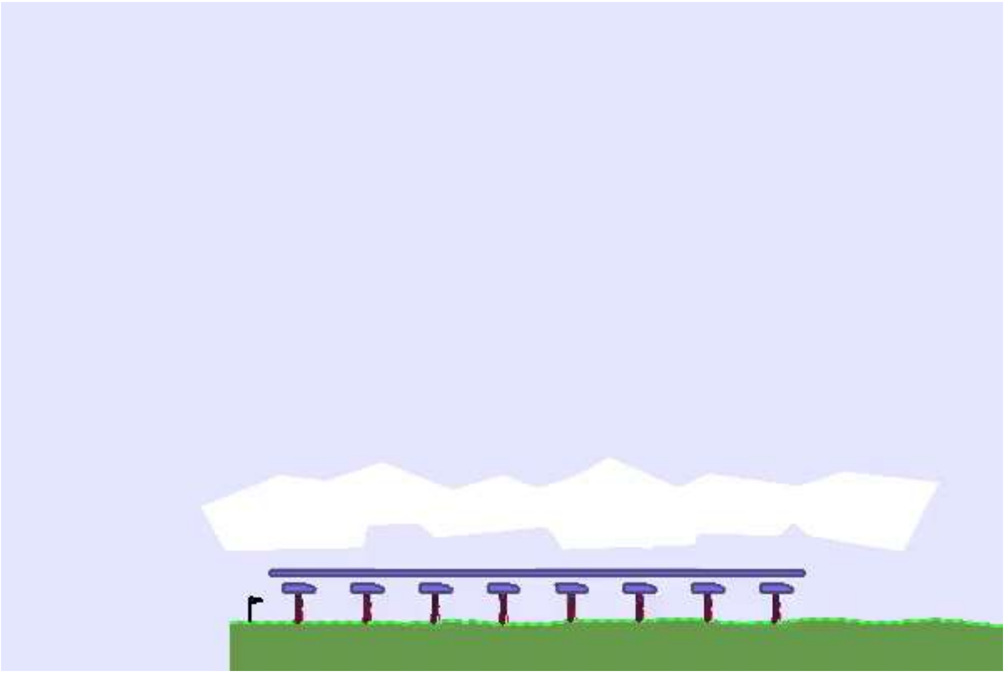


# Scaling to Several Agents



# Results



Parameter sharing allows an effective curriculum learning strategy that leads to policies that generalize a lot better

# MADDPG in Ray/RLlib

This implementation of MADDPG is recommended for research purposes only. If you want to actually learn something, use parameter sharing.

Ref: https://github.com/justinkterry/maddpg-rllib

arXiv.org > cs > arXiv:2005.13625

Search...

Help | Advance

Computer Science > Machine Learning

[Submitted on 27 May 2020 (v1), last revised 24 Jul 2020 (this version, v4)]

**Parameter Sharing is Surprisingly Useful for Multi-Agent Deep Reinforcement Learning**

Justin K Terry, Nathaniel Grammel, Ananth Hari, Luis Santos

# Take-aways

- Centralizing multi-agent systems as single controlling entity makes the problem intractable due to exploding action space
- Homogeneity allows sharing parameters between agent modules
- Parameter sharing mitigates non-stationarity during learning
- Combined with curriculum learning leads to better generalized policies that scale to 10s of agents

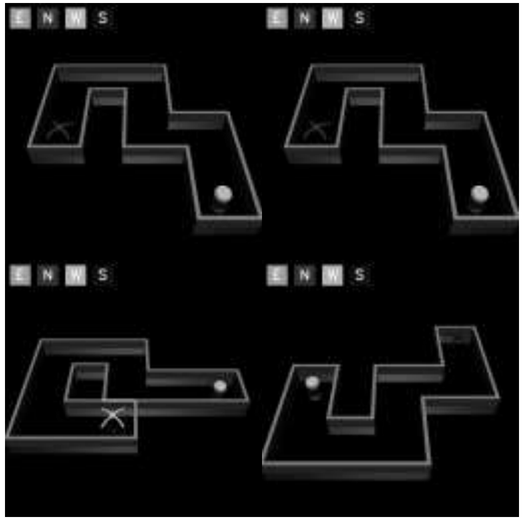
# Summary and Future Work

# Contributions

AAMAS 2017

**Functional Modularity**
Agent-informed modules

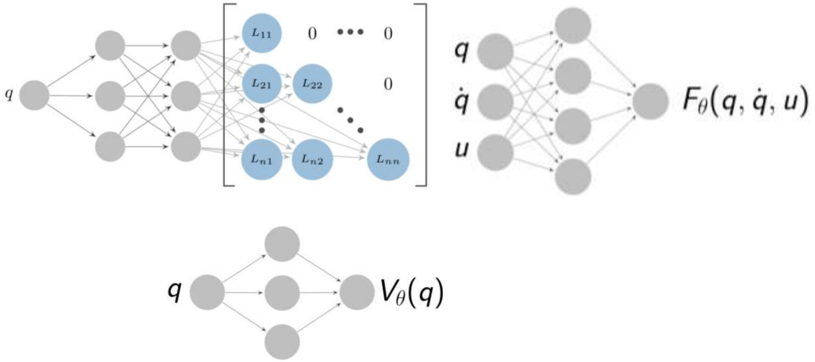
AAMAS 2019



**Temporal Modularity**
Task-informed modules

L4DC 2020

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = F(q, \dot{q}, u)$$



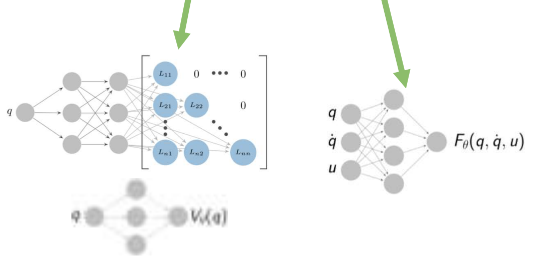

**Architectural Modularity**
Physics-informed modules

1. Identification of specific ways modularity comes into play during design of decision-making systems
2. Application of these modular design principles, reduces sample complexity and improves generalization

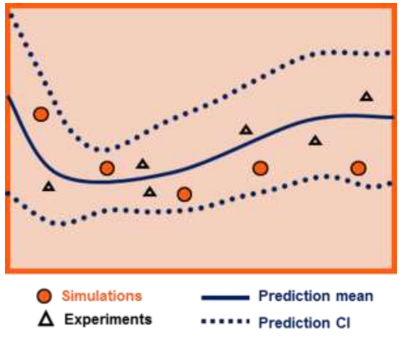
# Models that scale *and* illuminate

## Domain Awareness

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = F(q, \dot{q}, u)$$




## Quantified Uncertainty




## Interpretability

Captum
WHYNOT
Distill

## Accessibility

GitHub
Gym


**Domain Awareness**

1. Respect physical principles, constraints, symmetries
2. Reduce data requirements
3. Improve reliability

**Quantified Uncertainty**

1. Reliable predictions
2. Guide exploration
3. Experiment design

**Interpretability**

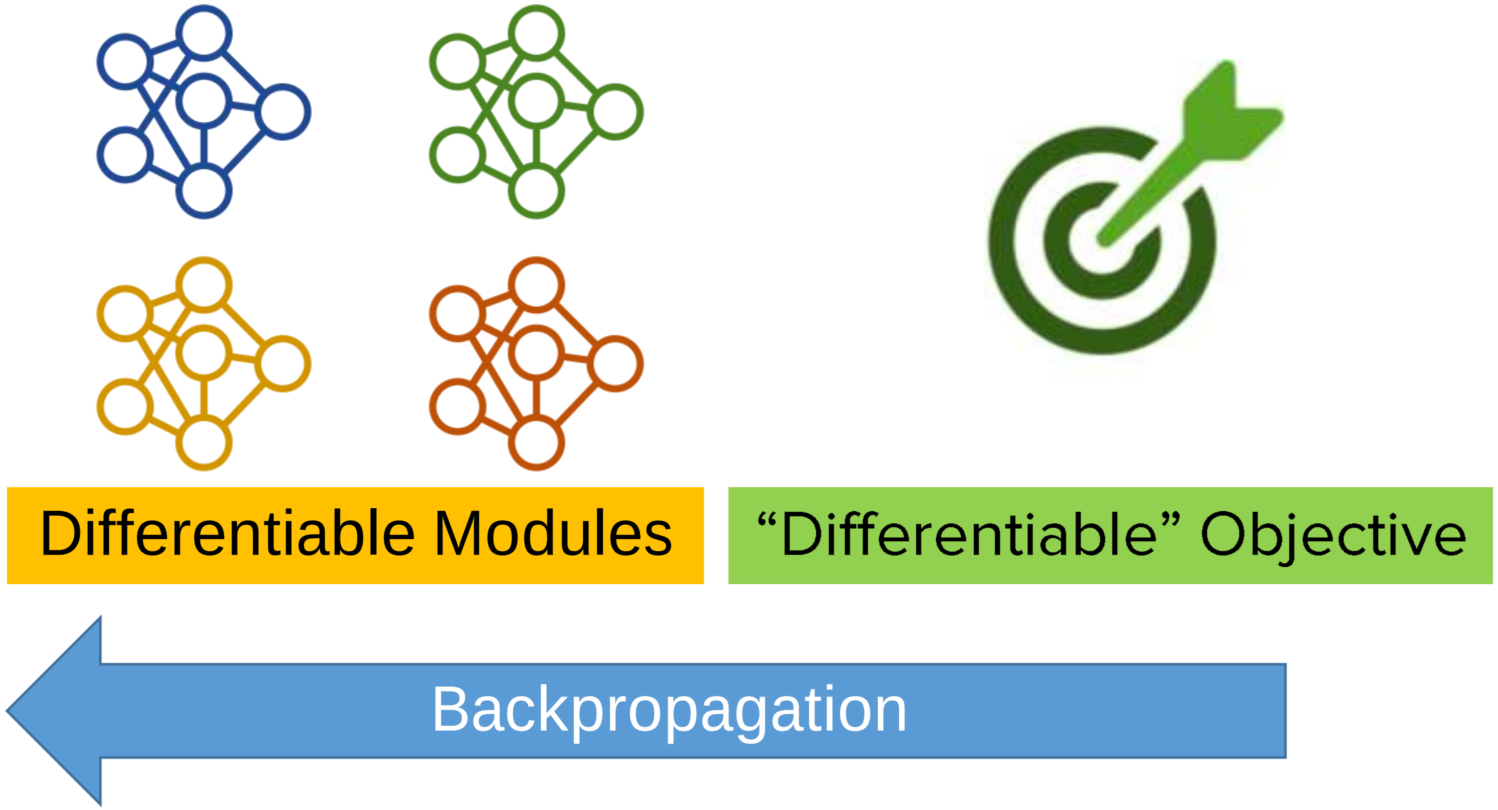
1. Sensitivity analysis
2. Causal analysis
3. Visualization

**Accessibility**

1. Community software
2. Benchmark problems
3. Extensible APIs

# Composing Modular Objectives

- Gradient descent and its discontents




Known: composing different modules to optimize a single, shared objective

# Composing Modular Objectives

- "Loss composition" in the wild
  - Generative Adversarial nets (GANs)
  - Adversarial training
  - Hyperparameter optimization by implicit function theorem
  - Intrinsic curiosity modules for RL


1. Goodfellow, I. "NIPS 2016 tutorial: Generative adversarial networks." arXiv preprint arXiv:1701.00160 (2016).
2. Madry, Al, et al. "Towards deep learning models resistant to adversarial attacks." *arXiv preprint arXiv:1706.06083* (2017).
3. Lorraine, J., Vicol, P. and Duvenaud, D., "Optimizing millions of hyperparameters by implicit differentiation." AISTATS, (2020)
4. Pathak, D., Agrawal, P., Efros, A.A. and Darrell, T., "Curiosity-driven exploration by self-supervised prediction." CVPR (2017)


Unknown: Can game theory and other ideas from multi-agent systems help make this process less ad-hoc?

# Thank You!