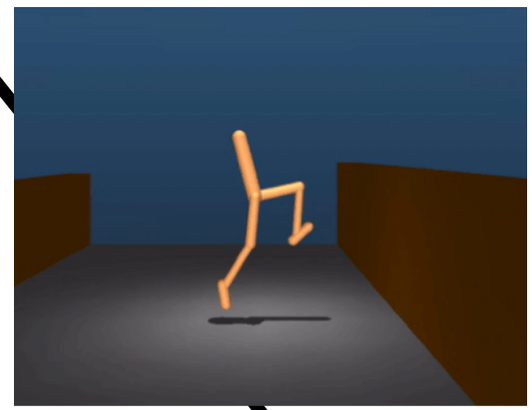
Machine Learning for Robot Planning and Control

Byron Boots
Georgia Tech Robot Learning Lab

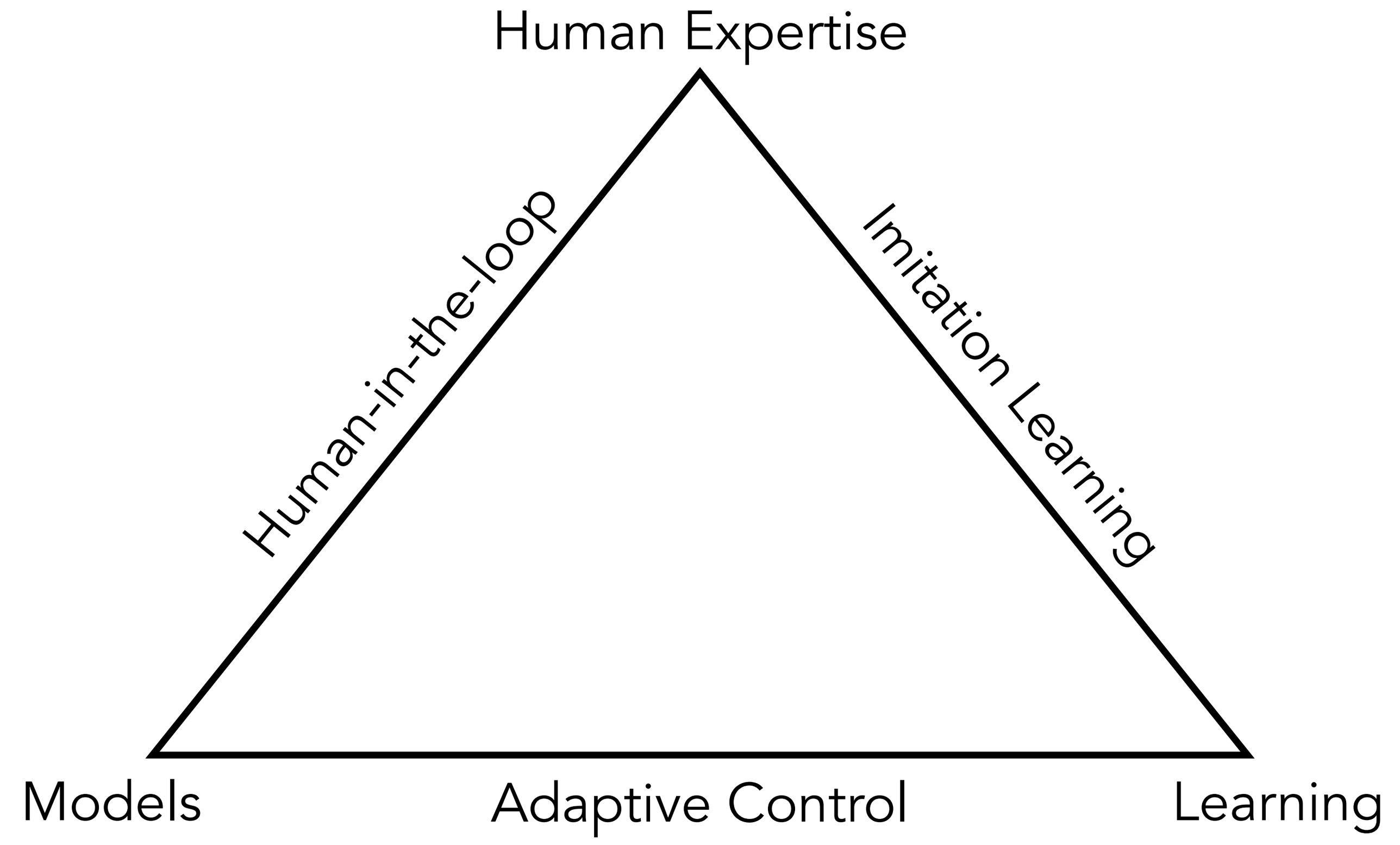
Georgia Tech Institute for Robotics and Intelligent Machines

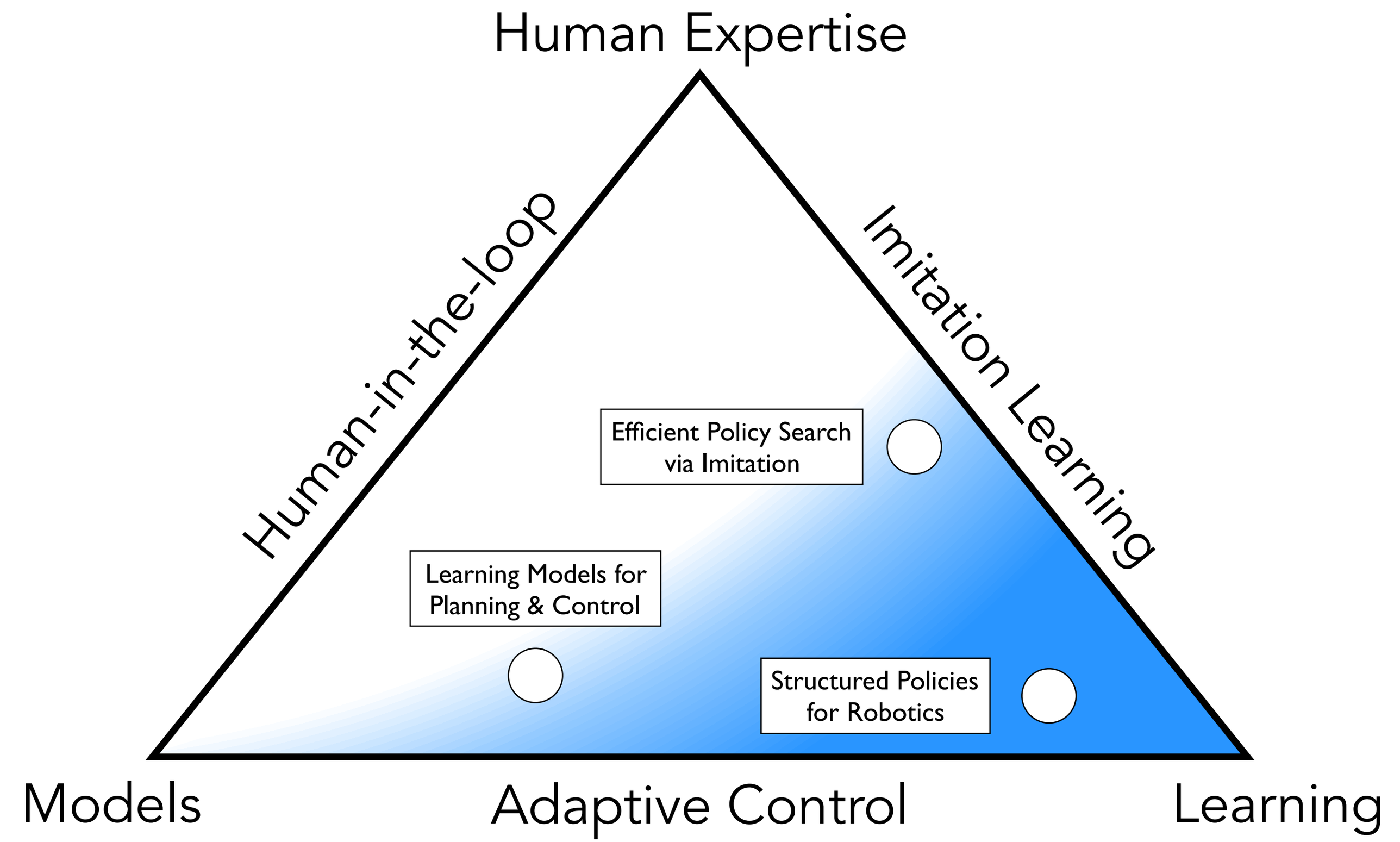
Human Expertise



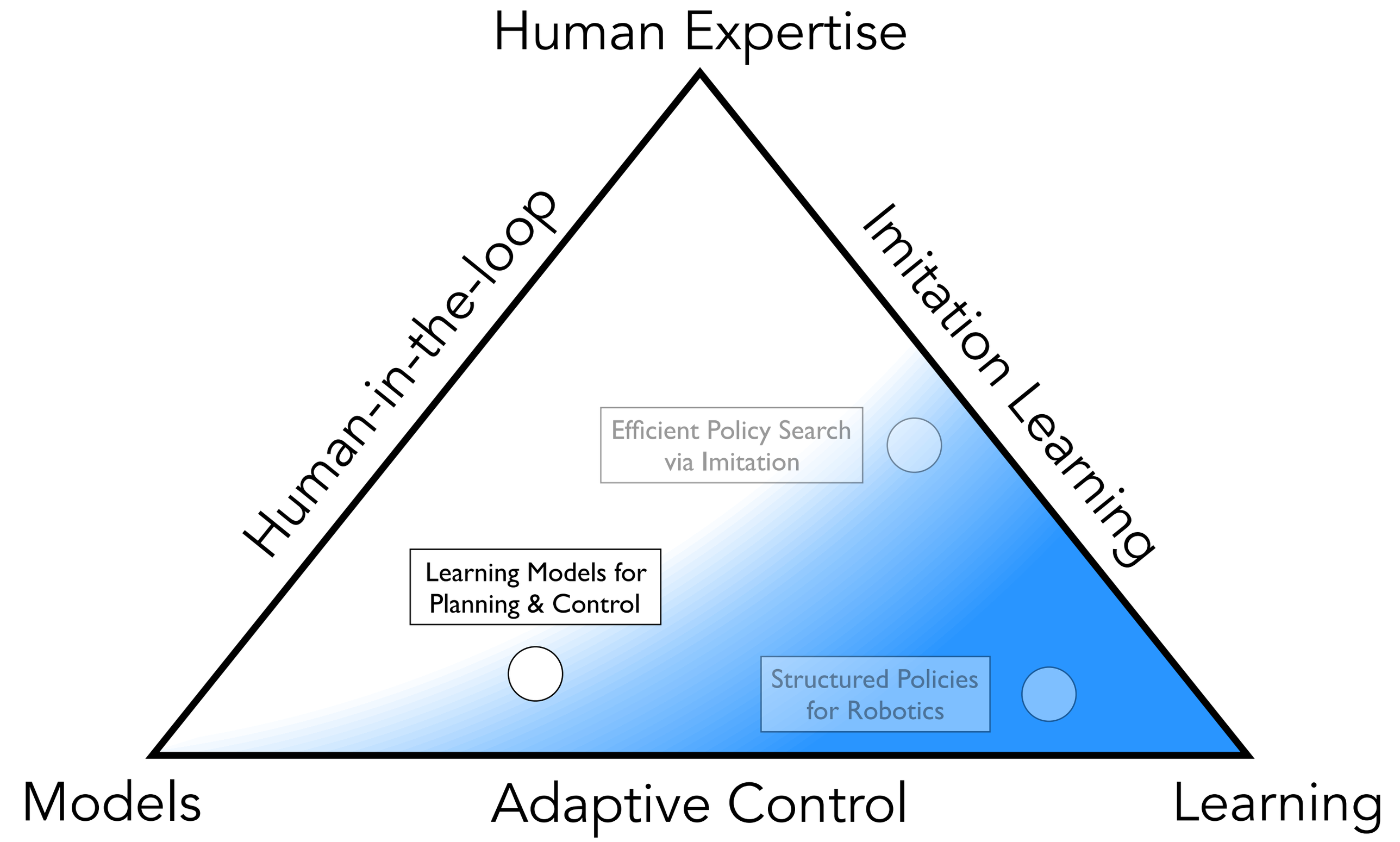
Models

Learning


Human Expertise
Human-in-the-loop
Imitation Learning
Models
Adaptive Control
Learning


Human Expertise
Human-in-the-loop
Imitation Learning
Efficient Policy Search via Imitation
Learning Models for Planning & Control
Structured Policies for Robotics
Models
Adaptive Control
Learning

Learning Models



Task: Aggressive Offroad Driving

Georgia Tech Institute for Robotics and Intelligent Machines

with Panos Tsiotras, Evangelos Theordou, Jim Rehg

Task: Aggressive Offroad Driving

Georgia Tech Institute for Robotics and Intelligent Machines

Terrestrial Agility: Drive Faster Than Human Pilots, Don’t Crash!

steering

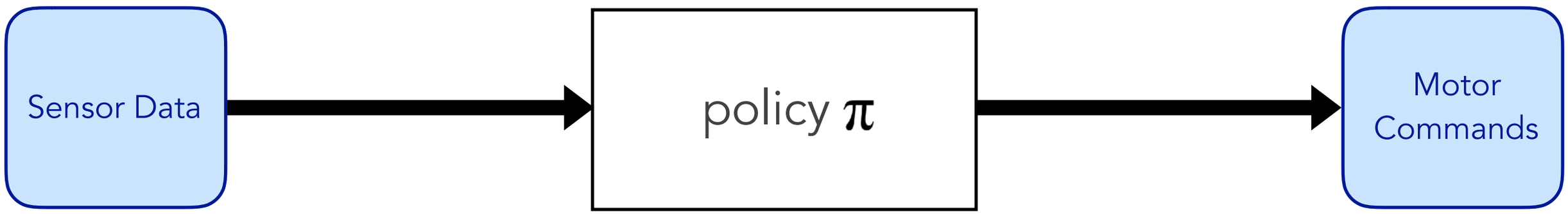
throttle

Task: Aggressive Offroad Driving

Georgia Tech Institute for Robotics and Intelligent Machines

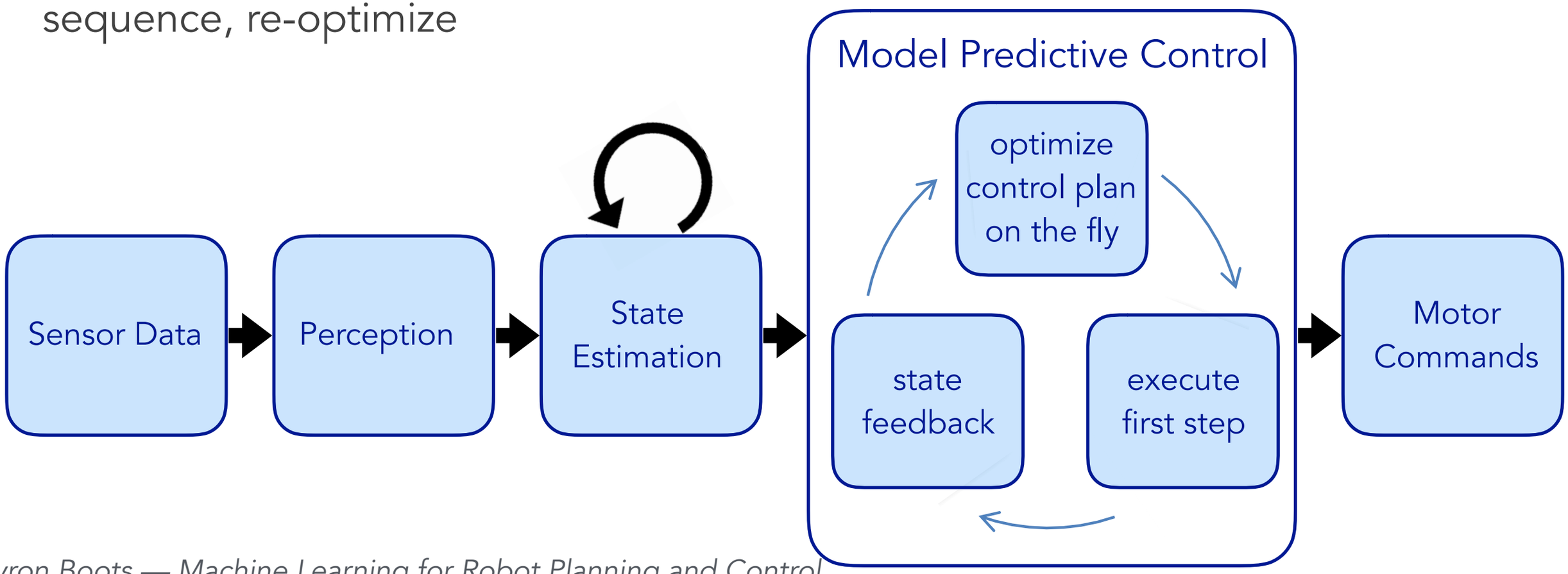
Terrestrial Agility: Drive Faster Than Human Pilots, Don't Crash!

The AI problem:



Engineering may be useful…

- **Perception**: Differential GPS and IMU give accurate outdoor position to within a few centimeters
- **State Estimation**: Map computed from a GPS survey of the environment
- **Assume a System Model:** use Model Predictive Control (MPC) to generate actions
- **MPC**: Optimize an open loop control sequence, execute a small portion of sequence, re-optimize



Why is this hard?

Georgia Tech Institute for Robotics and Intelligent Machines

If you don't get the dynamics right...

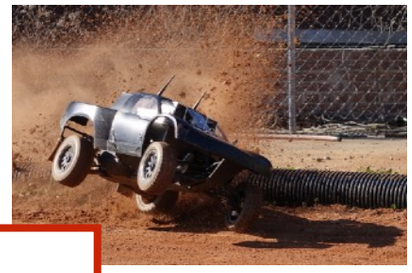
Georgia Tech Institute for Robotics and Intelligent Machines

Model Predictive Control

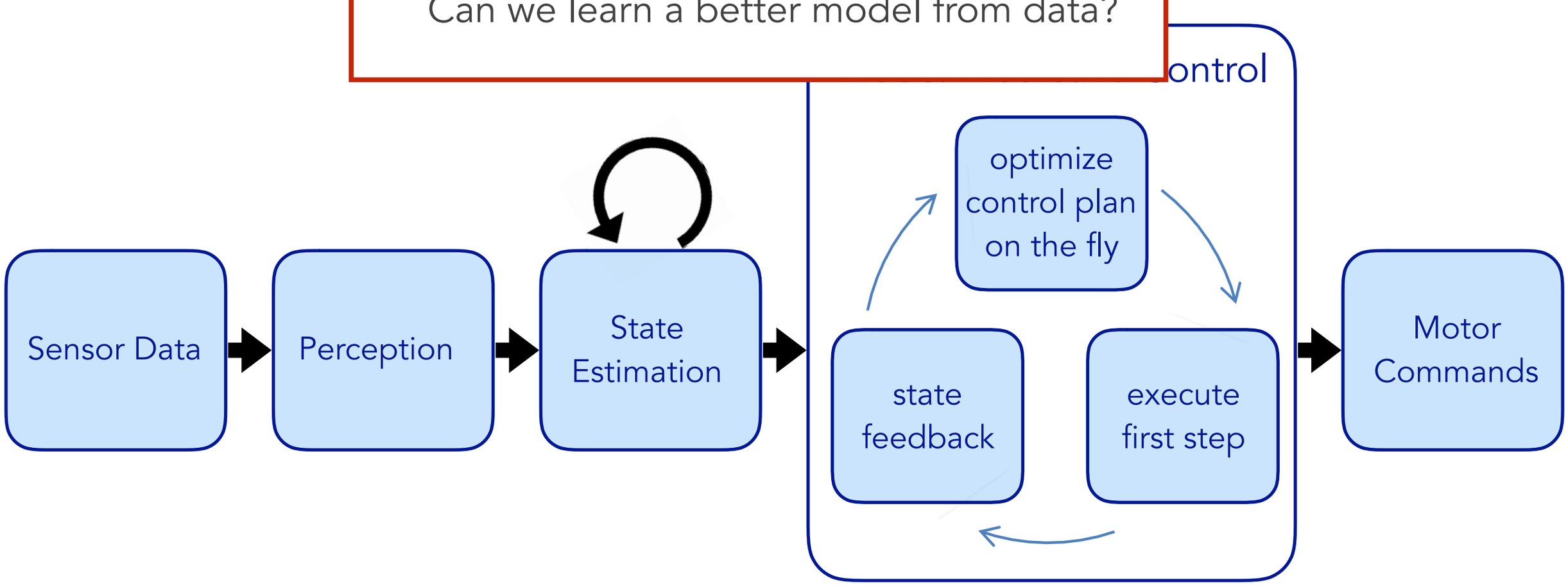
- Model Predictive Control has a long history of successful applications

- Most methods are geared towards linear-quadratic systems with convex constraints. We have **nonlinear dynamics** and **non-convex costs/constraints**

- Usually use a physics-based models, **makes simplifications and assumptions that may not be accurate**



Can we learn a better model from data?



Neural Network Dynamics

- We assume general discrete-time nonlinear state-space dynamics: $\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t, \mathbf{u}_t)$
- We assume that the state $\mathbf{x}$ is partitioned as $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$
- We only need to learn a function f so that the full state transition is:

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t, \mathbf{u}_t) = \begin{bmatrix} \mathbf{q}_t + \dot{\mathbf{q}}_t \Delta t \\ \dot{\mathbf{q}}_t + f(\mathbf{x}_t, \mathbf{u}_t) \Delta t \end{bmatrix}$$

- We use fully connected networks with two hidden layers

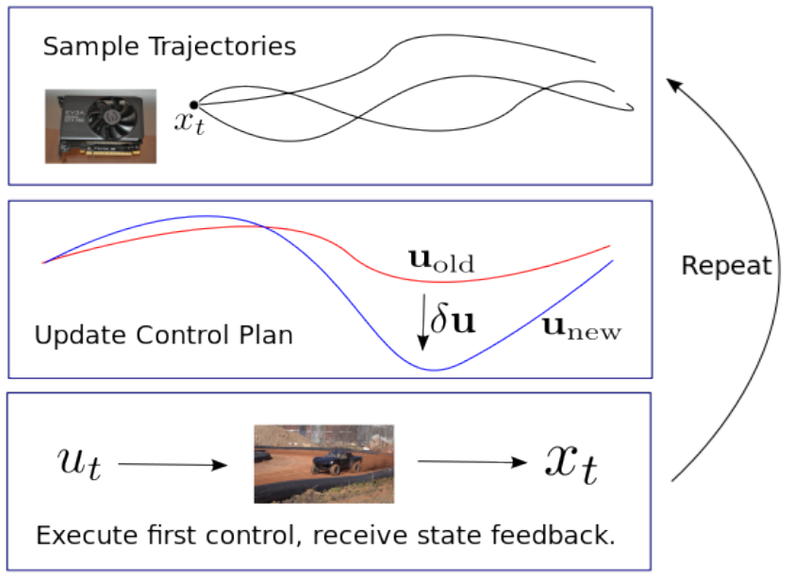
MPC is often formulated for control-affine dynamics….

Information Theoretic MPC for Model-based Reinforcement Learning
[Williams, Wagener, Goldfain, Drews, Rehg, Boots, Theodorou; ICRA 2017]

Model Predictive Path Integral Control

(sampling-based approach to optimal control)

1. Sample and evaluate trajectories
2. Compute control update
3. Execute first control in sequence, receive state feedback
4. Repeat, using the un-executed portion of the previous control sequence to warm-start the trajectory

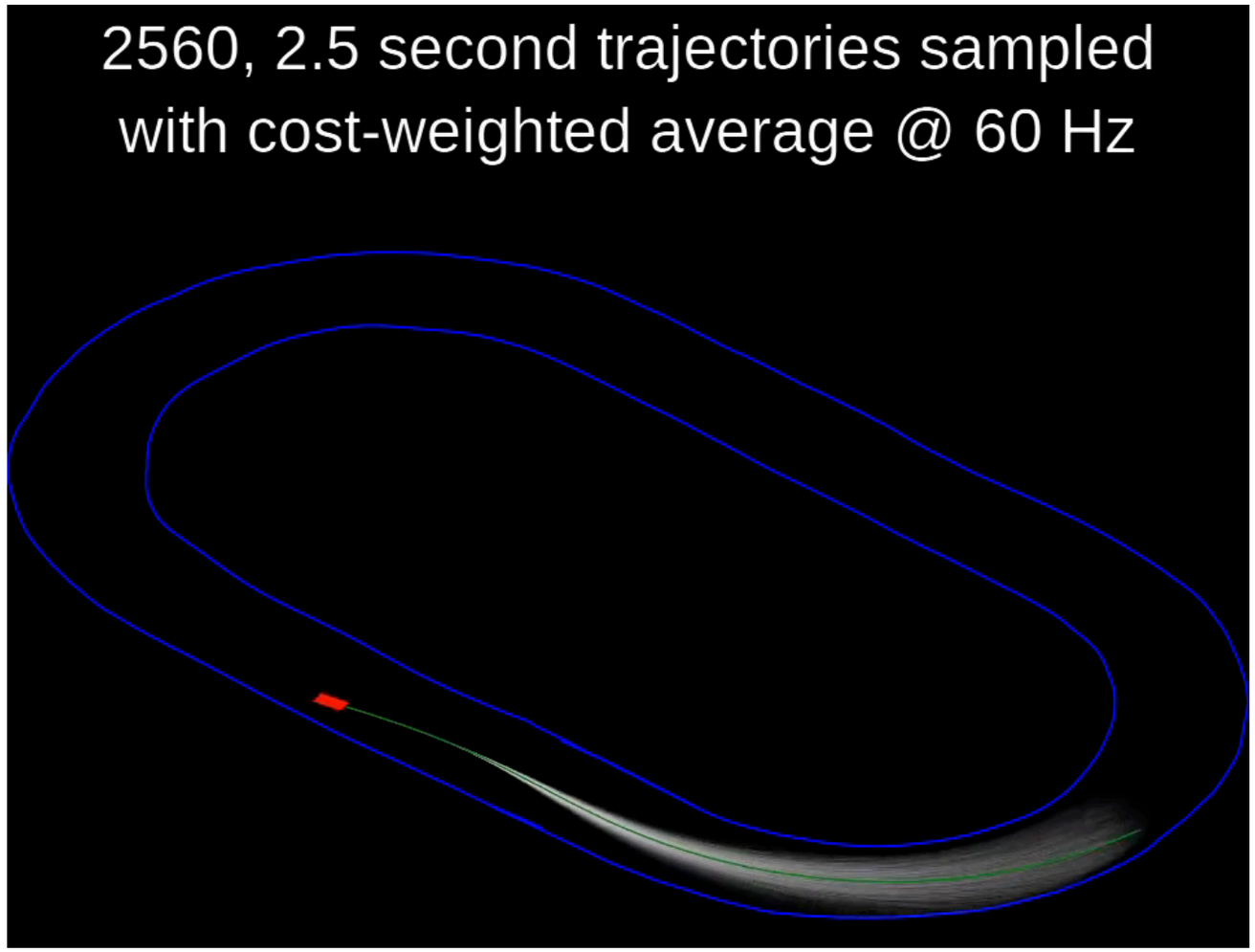


MPPI is naturally parallelizable, can use GPU to execute up to150,000 trajectories per second

Model Predictive Path Integral Control

Georgia Tech Institute for Robotics and Intelligent Machines

(sampling-based approach to optimal control)



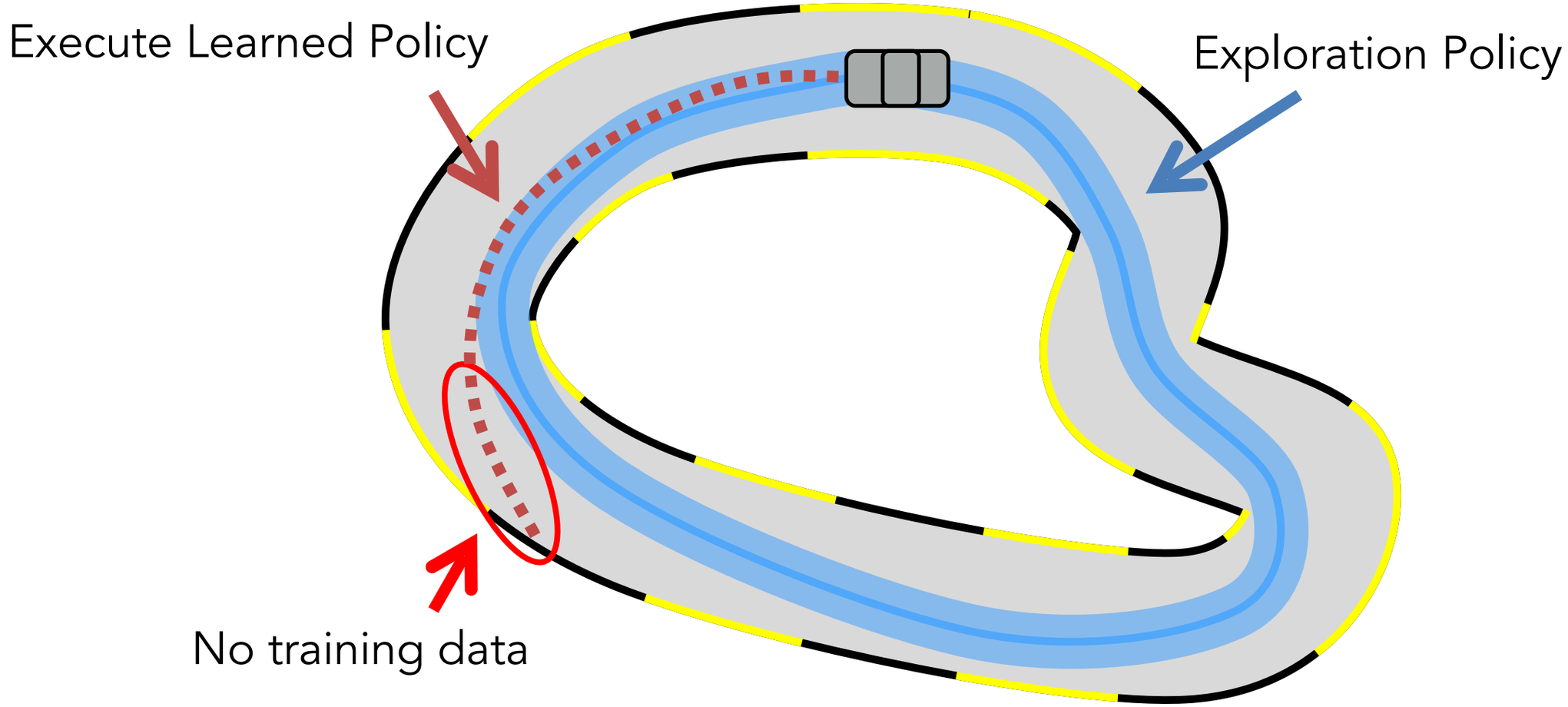
MPPI is naturally parallelizable, can use GPU to execute up to150,000 trajectories per second

Learning Neural Network Dynamics

$$\arg\min_{\theta} \mathbb{E}_{\rho_{\text{explore}}} \left[\sum_t ||\ddot{q}_t - f_\theta(x_t, u_t)||_2^2 \right]$$

Distribution of $\ddot{q}, x, u$ induced by the exploration policy

Minimize NN prediction error



covariate shift: mismatch between training and test distributions

$$\arg\min_{\theta} \mathbb{E}_{\rho_{\text{explore}}} \left[\sum_t ||\ddot{q}_t - f_\theta(x_t, u_t)||_2^2 \right]$$

learning

$$\arg\min_{\theta} \mathbb{E}_{\rho_\theta} \left[\sum_t ||\ddot{q}_t - f_\theta(x_t, u_t)||_2^2 \right]$$

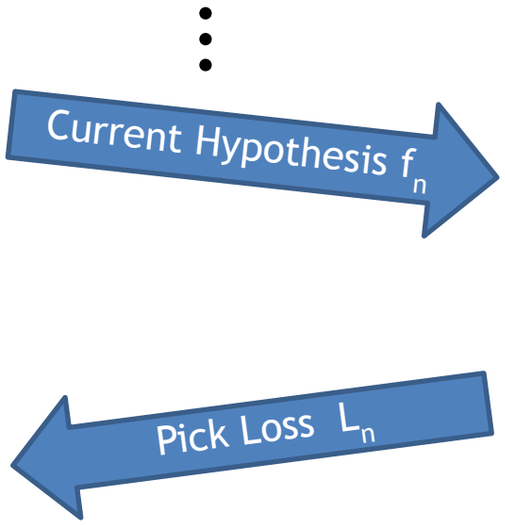
…much more complicated…

$$\arg\min_{\theta} \mathbb{E}_{\rho_\theta} \left[\sum_t ||\ddot{q}_t - f_\theta(x_t, u_t)||_2^2 \right]$$

…much more complicated…

Can view problem as a **Game,** reduce to **Online Learning**:

Player 1: Learner (dynamics)



Player 2: Optimal Controller

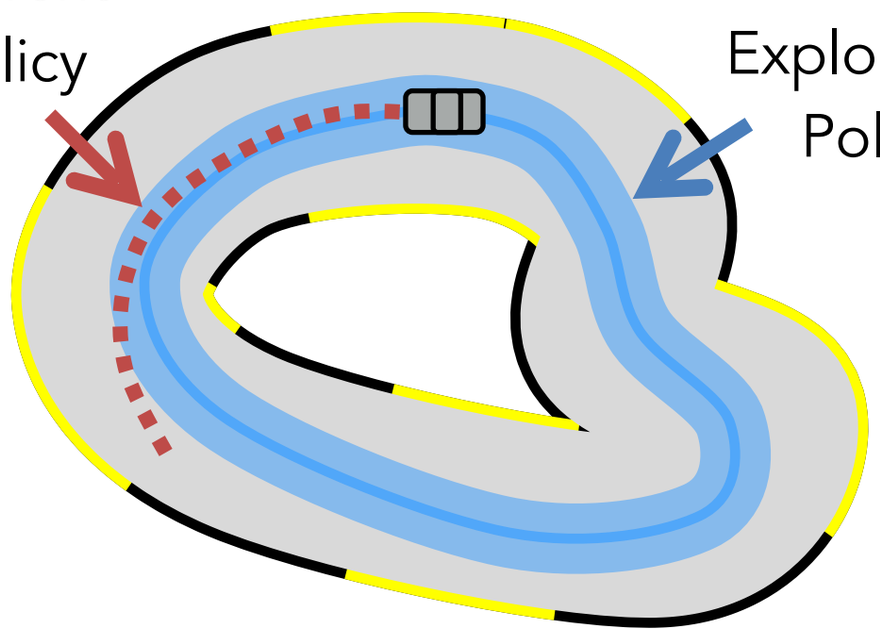
Online Learning: Dataset Aggregation

Georgia Tech Institute for Robotics and Intelligent Machines

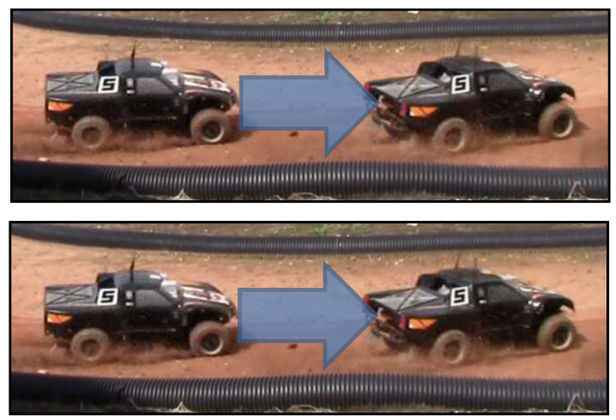
Collect Data

Current Policy

Exploration Policy



New Data

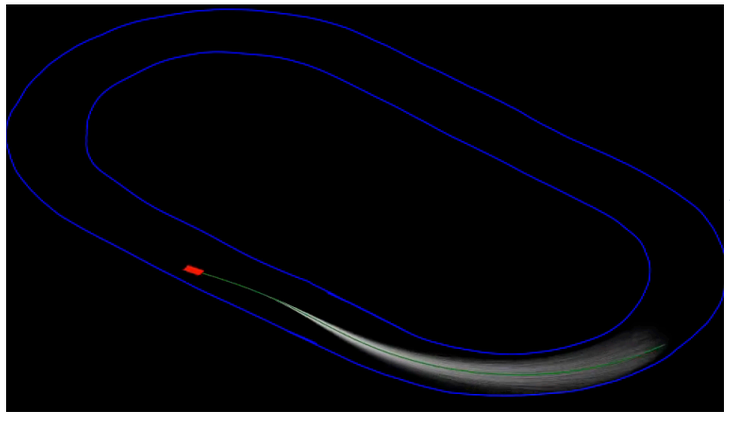


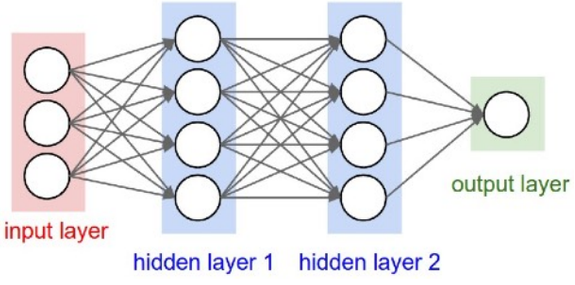
solve with DAgger

(Follow The Regularized Leader)

[Ross & Bagnell, 2012]

Add to Dataset





Attempt the Task With MPPI

Retrain Model

Example Training Run

The initial dynamics model is trained from human driving. After each training phase, we augment the dataset with the new data, and then retrain the model.

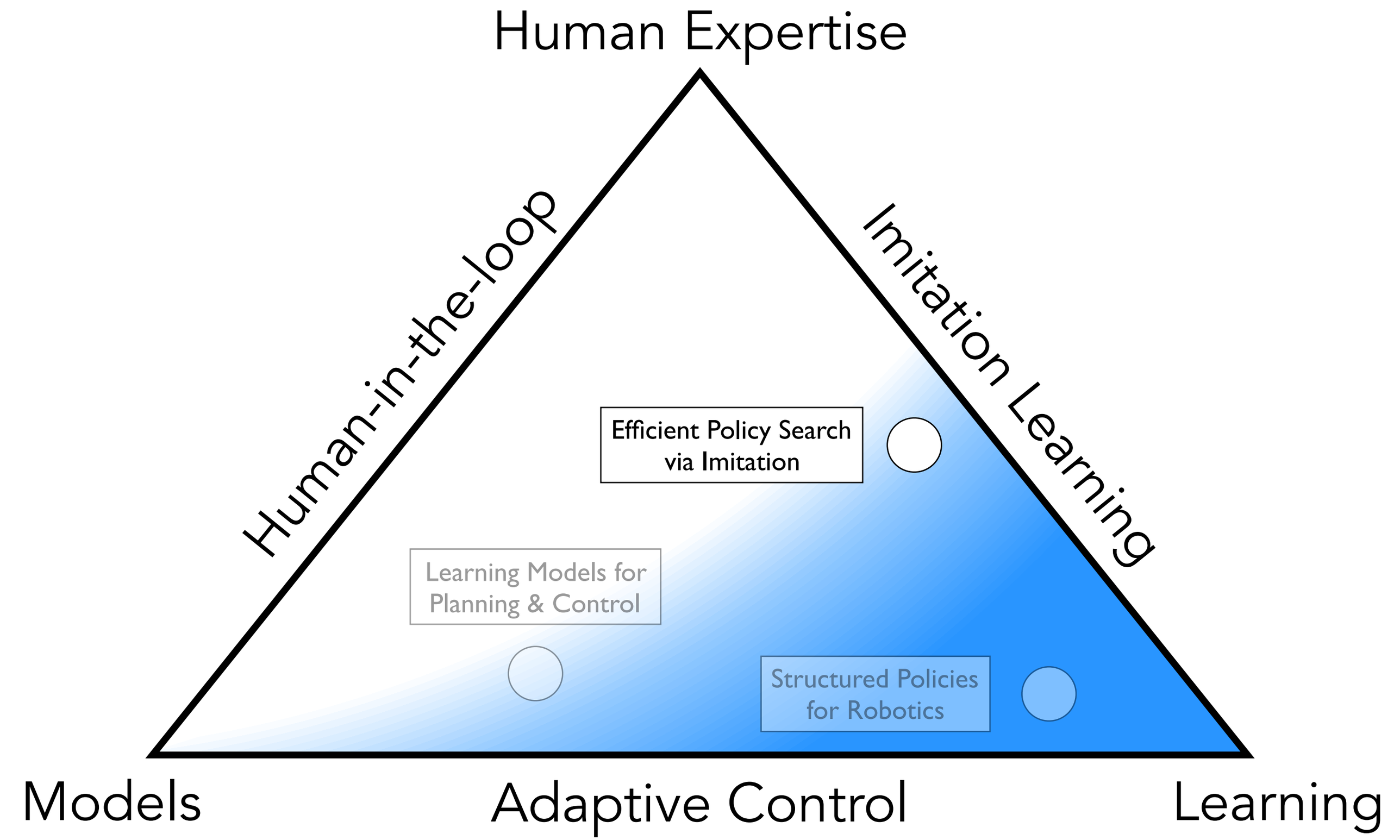
Neural network configuration: 6-32-32-4

Robustness

Georgia Tech Institute for Robotics and Intelligent Machines

Imitation Learning

Georgia Tech Institute for Robotics and Intelligent Machines

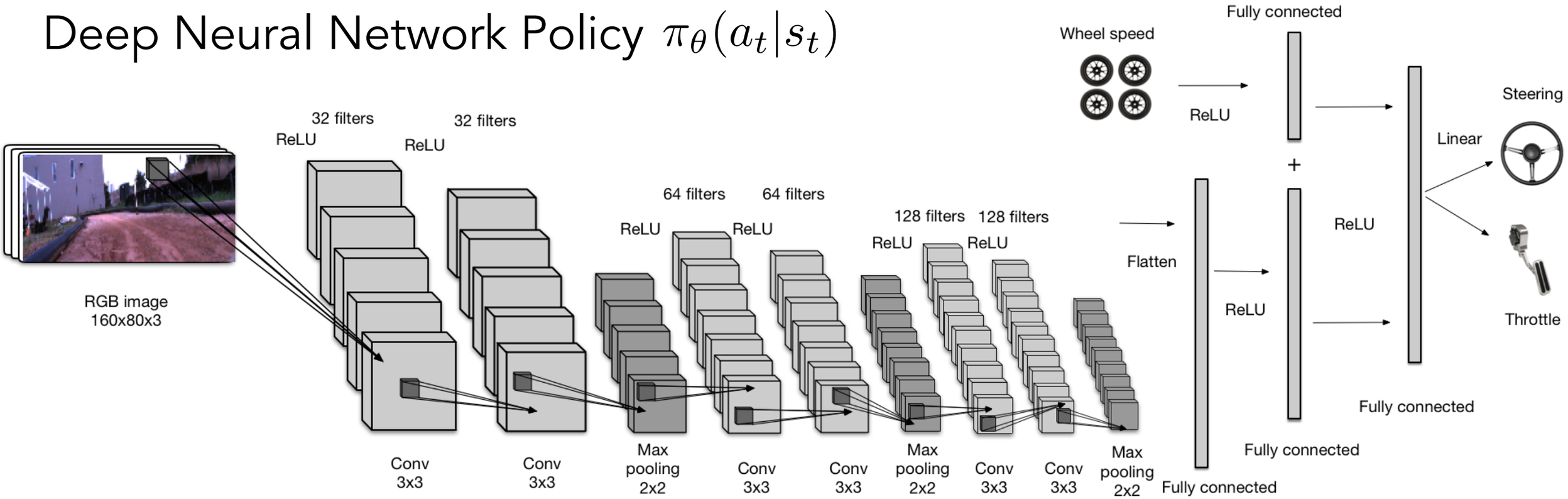


Task: Aggressive Offroad Driving

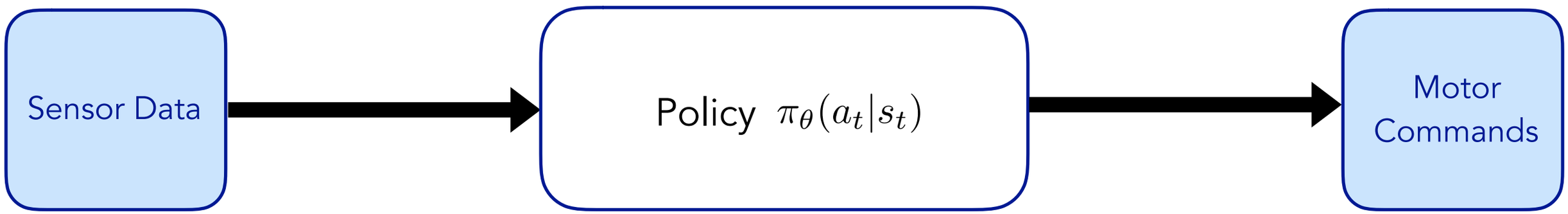
Georgia Tech Institute for Robotics and Intelligent Machines

With Vision!!

Deep Neural Network Policy $\pi_\theta(a_t|s_t)$



train end-to-end somehow



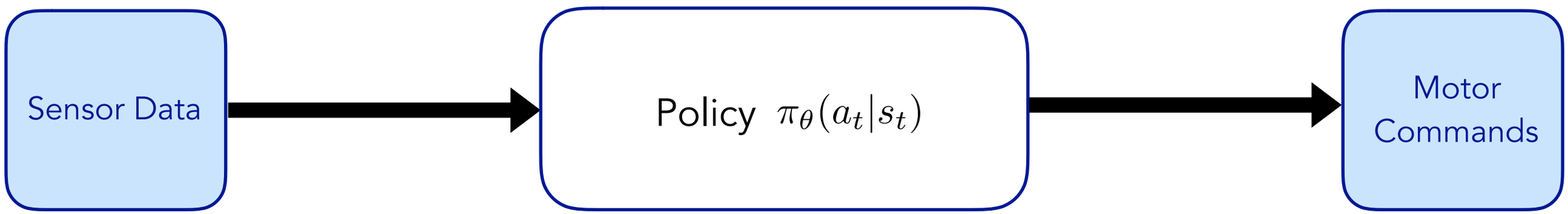
Reinforcement Learning

Find a Policy: $\pi_\theta(a_t|s_t)$

Objective: $\theta^* = \arg\min_\theta \mathbb{E}_{\rho_\theta}\left[\sum_t c(s_t, a_t)\right]$

Distribution of trajectories (sequences of states and actions) induced by $\pi_\theta(a_t|s_t)$.

instantaneous state-action cost (slow bad, crashing bad, lots of actuation bad)

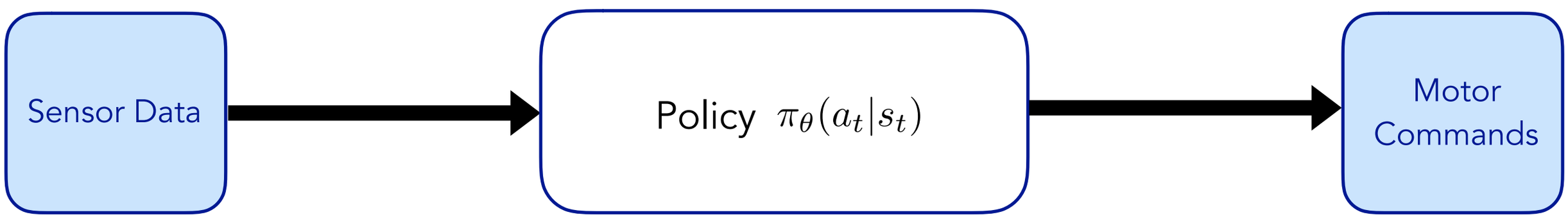


RL Policy Gradient Methods [Williams 92]

$$\nabla_\theta J(\theta) = \sum_t [\nabla_\theta \ln(\pi_\theta(a_t|s_t))Q^\pi(s_t, a_t)]$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta)$$

descending the gradient increases log-likelihood of actions that result in low accumulated cost with respect to the current policy

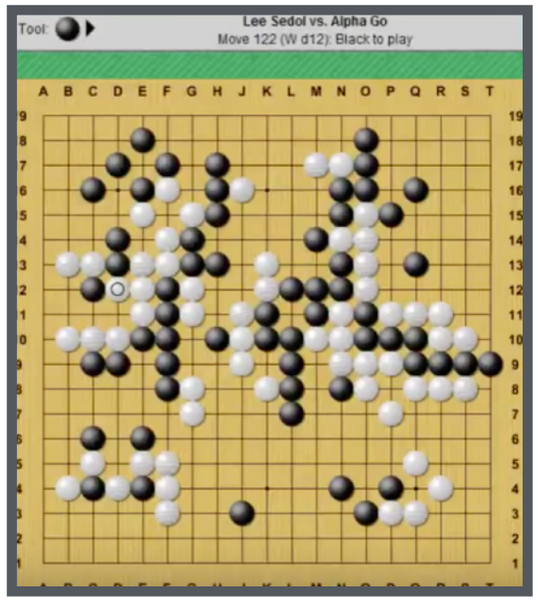


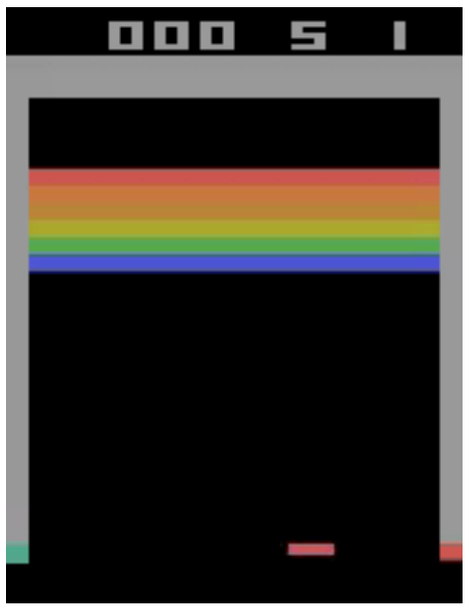
RL Policy Gradient Methods [Williams 92]

$$\nabla_\theta J(\theta) = \sum_t [\nabla_\theta \ln(\pi_\theta(a_t|s_t))Q^\pi(s_t, a_t)]$$

Trial and Error: exploration via (e.g.) a stochastic policy requires a huge number of interactions

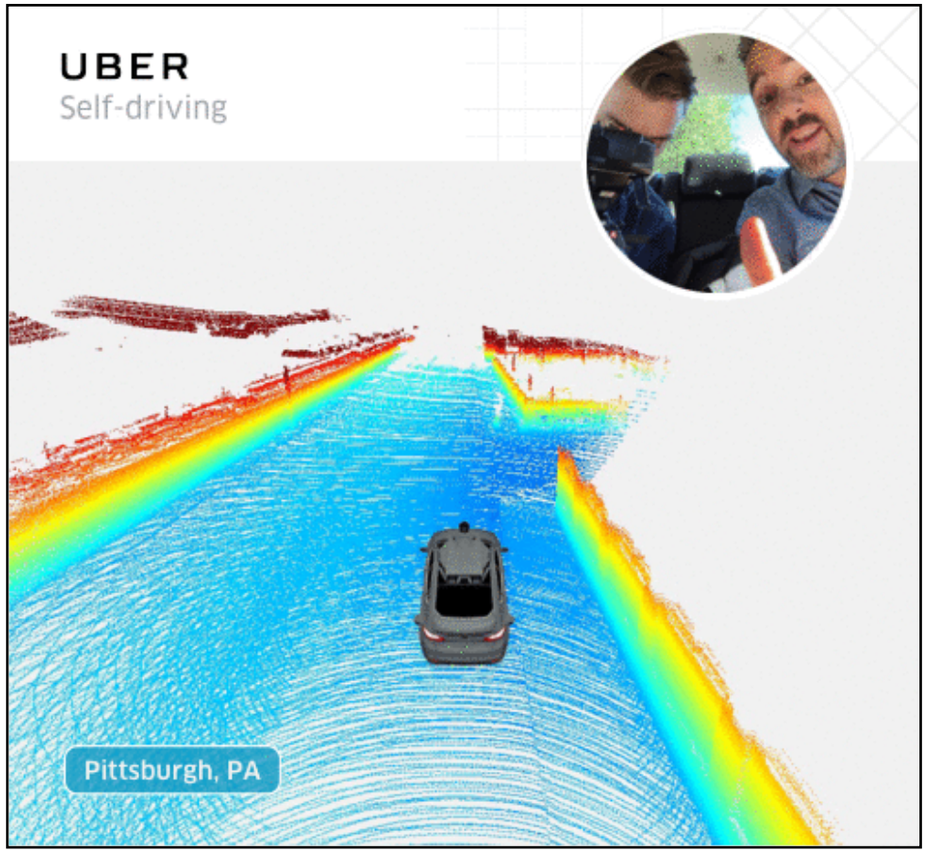
This works for games and simulated tasks



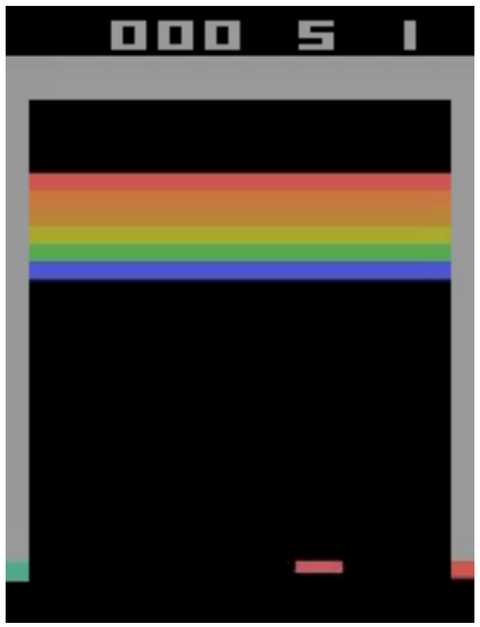


Reinforcement Learning (RL)

Georgia Tech Institute for Robotics and Intelligent Machines

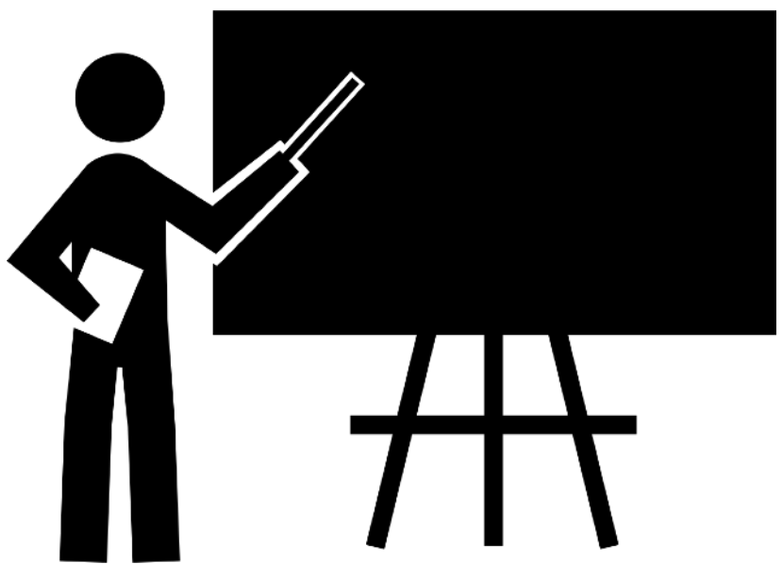


Imitation Learning



Random exploration
(e.g., epsilon greedy, stochastic policies)
Inefficient!

For robots: **Costly**! **Dangerous!**



Exploration under Expert Guidance
(e.g., Imitation Learning)
More Efficient!

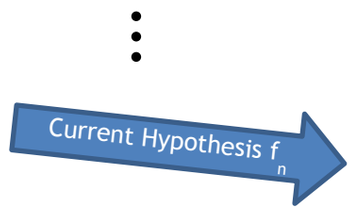
For robots: **Safer**!

Imitation Learning

We will view imitation learning as an **online learning** problem:

1. Initialize a policy
2. Execute the policy and evaluate feedback on performance w.r.t. expert
3. Update the policy

Player 1: Learner



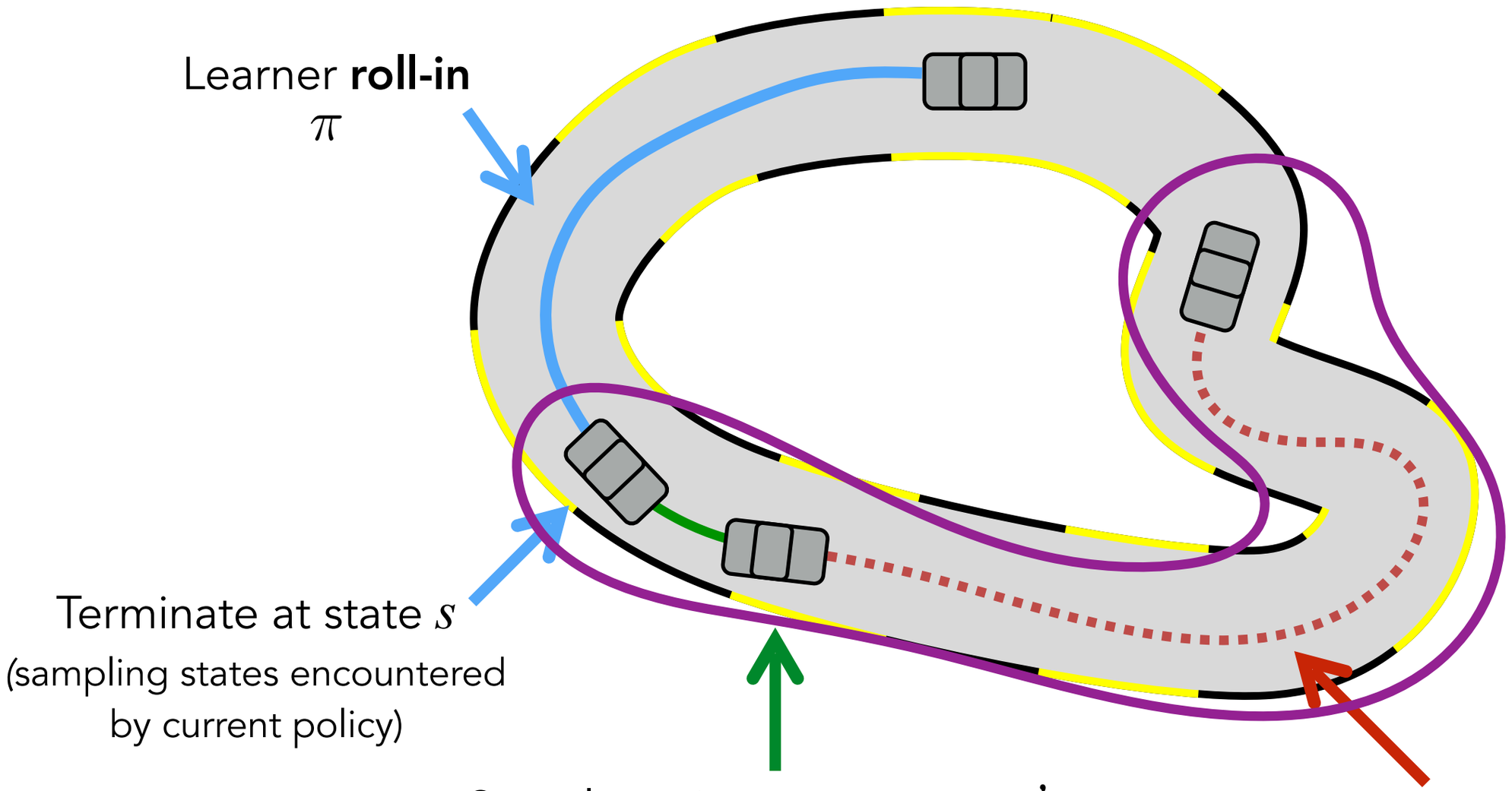
Player 2: Expert

AggreVaTeD (Aggregate Values to Imitate)

[Sun, Venkatraman, Gordon, Boots, Bagnell; ICML 2017]

AggreVaTeD: Roll-In, Roll-Out

Georgia Tech Institute for Robotics and Intelligent Machines



Want to adjust policy parameters to increase the (log) likelihood of actions that result in low cost-to-go **with respect to the expert.**

AggreVaTeD

RL Policy Gradient

$$\sum_t [\nabla_\theta \ln(\pi_\theta(a_t|s_t))Q^\pi(s_t,a_t)]$$

slowly bootstrap policy to improve over itself via trial-and-error (many interactions)

$$\sum \left[\nabla_\theta \ln(\pi_\theta(a_t|s_t))Q^{\pi^*}(s_t,a_t)\right]$$

AggreVaTeD
(Imitation Learning)

quickly learn to imitate expert policy (big step-sizes, few interactions)

Different implementation options for AggreVaTeD:

- Natural Policy Gradient
- TRPO
- PPO
- Actor-Critic
- GAE
- etc.

Imitation Learning (AggreVaTeD): $\sum_t \left[\nabla_\theta \ln(\pi_\theta(a_t|s_t))Q^{\pi^*}(s_t, a_t)\right]$

- **Is IL actually better than RL?**

  Deeply AggreVaTeD: Differentiable Imitation Learning for Sequential Prediction (ICML 2017)

  with *Wen Sun, Arun Venkatraman, Geoff Gordon, Drew Bagnell*

  - There exists an MDP for which IL requires **exponentially** fewer samples than RL
  - For general MDPs, IL requires **polynomially** fewer samples than RL

- **Does IL Converge?**

  Convergence of Value Aggregation for Imitation Learning (AISTATS 2018)

  with *Ching-An Cheng*

  - **Convergence requires stability:** small change in policy results in small change to state distribution. **We can enforce this**.

Imitation Learning (AggreVaTeD): $\sum_t \left[\nabla_\theta \ln(\pi_\theta(a_t|s_t))Q^{\pi^*}(s_t, a_t)\right]$

- **How can we estimate $Q^{\pi^*}(s_t, a_t)$ in practice?**

  Agile Off-Road Autonomous Driving Through End-to-End Deep Imitation Learning (RSS 2018)

  with *Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou*

  - If MPC expert, might **get Q-function for free**
  - If expert performance is stable, can **estimate an upper bound** on cost-to-go

- **Can we combine IL and RL to improve over the Expert?**

  Truncated Horizon Policy Search: Combining Reinforcement Learning and Imitation Learning (ICLR 2018)

  with *Wen Sun, Drew Bagnell*

  Fast Policy Learning using Imitation and Reinforcement (UAI 2018)

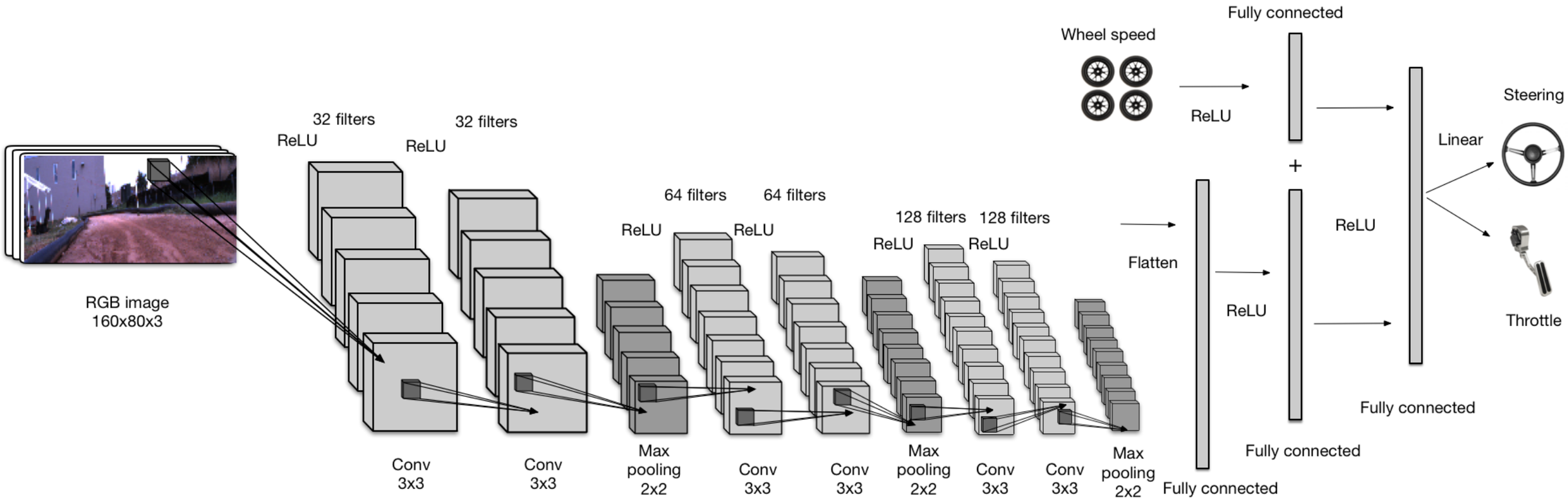
  with *Ching-An Cheng, Xinyan Yan, Nolan Wagener*

  - **Yes!**

Back to Racing!!

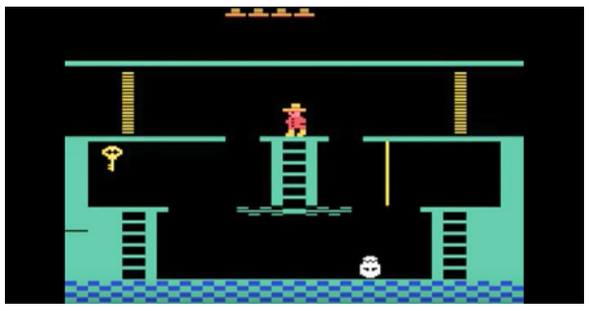
Georgia Tech Institute for Robotics and Intelligent Machines

Terrestrial Agility: Drive Faster Than Human Pilots, Don’t Crash!

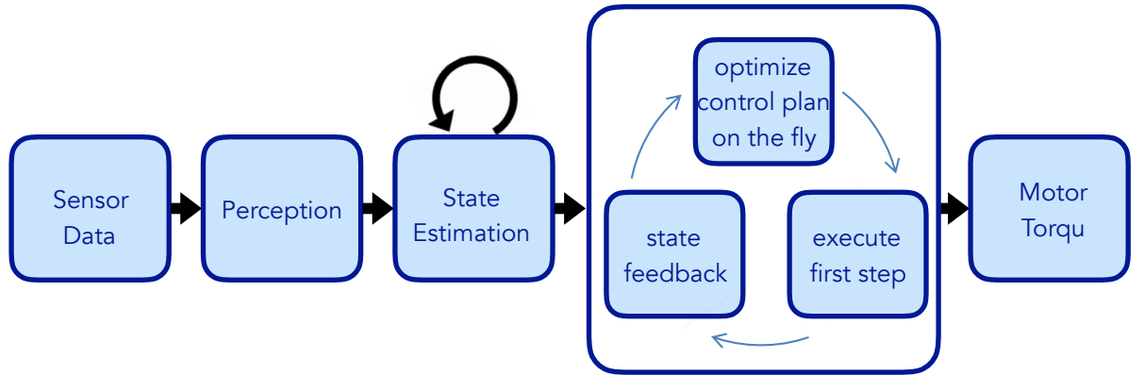


Imitation Learning: Experts

Human

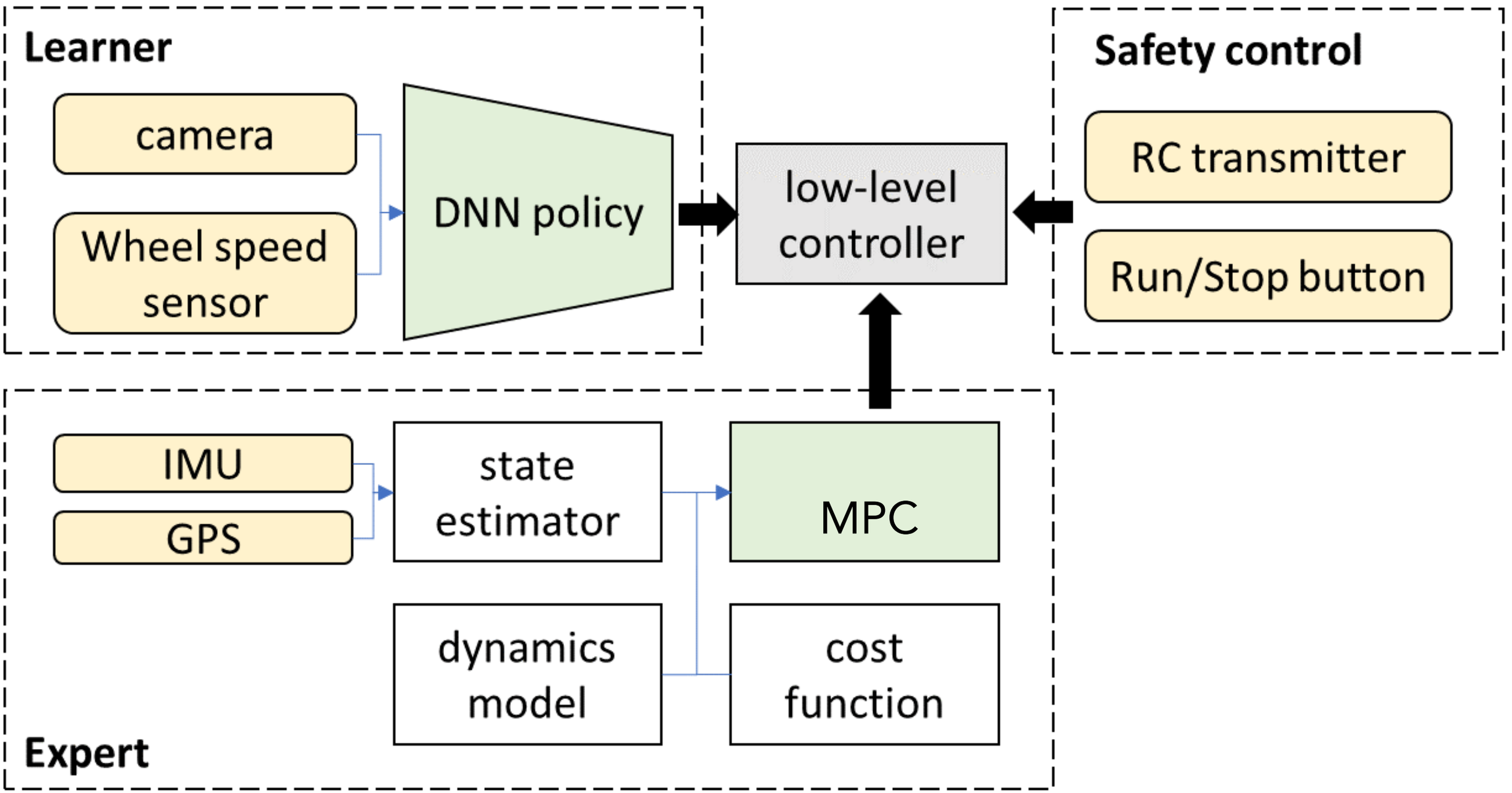


Algorithm or System
with **More Resources**
during training



Training the Policy

Georgia Tech Institute for Robotics and Intelligent Machines

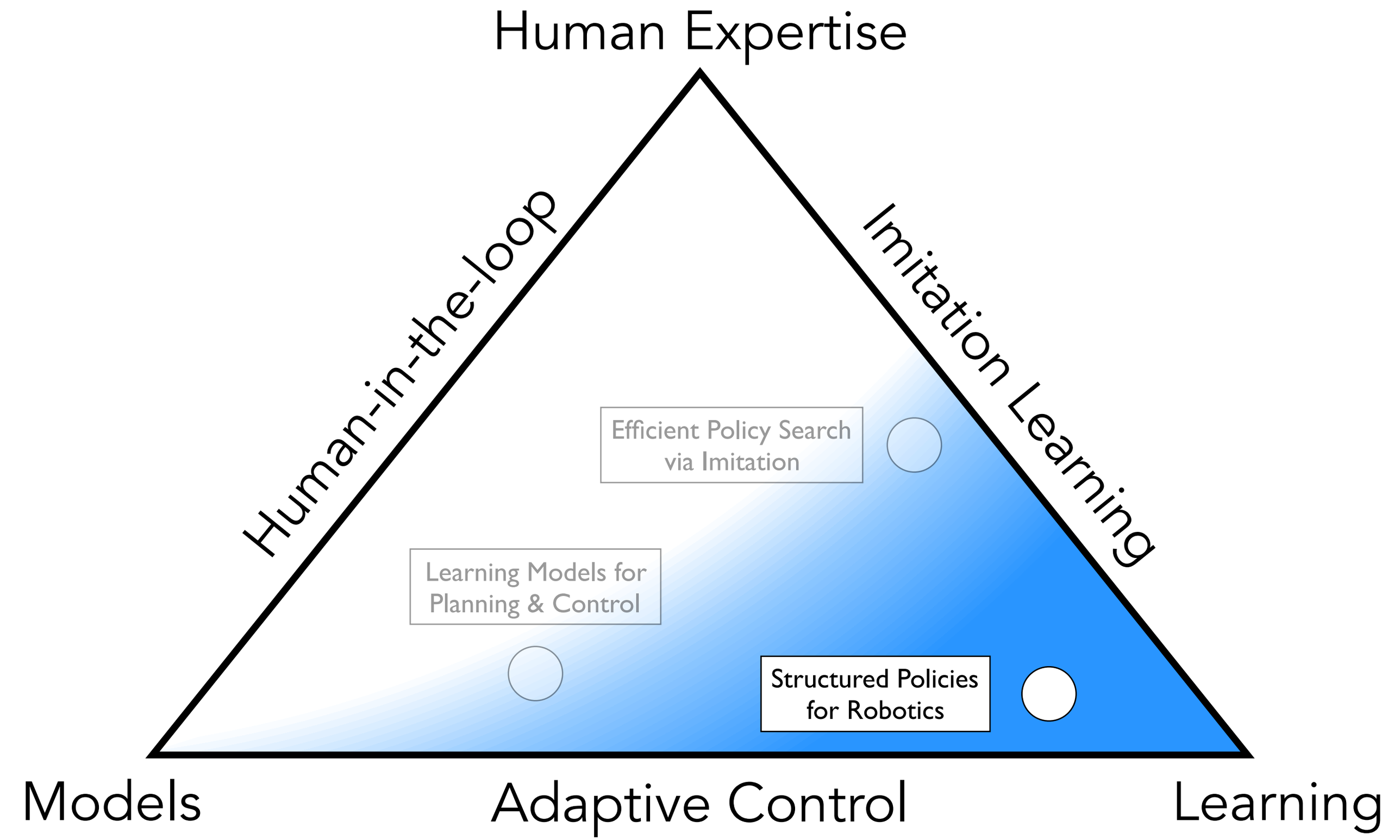


Agile Off-Road Autonomous Driving Through End-to-End Deep Imitation Learning

[Pan, Cheng, Saigol, Lee, Yan, Theodorou, Boots; RSS 2018]

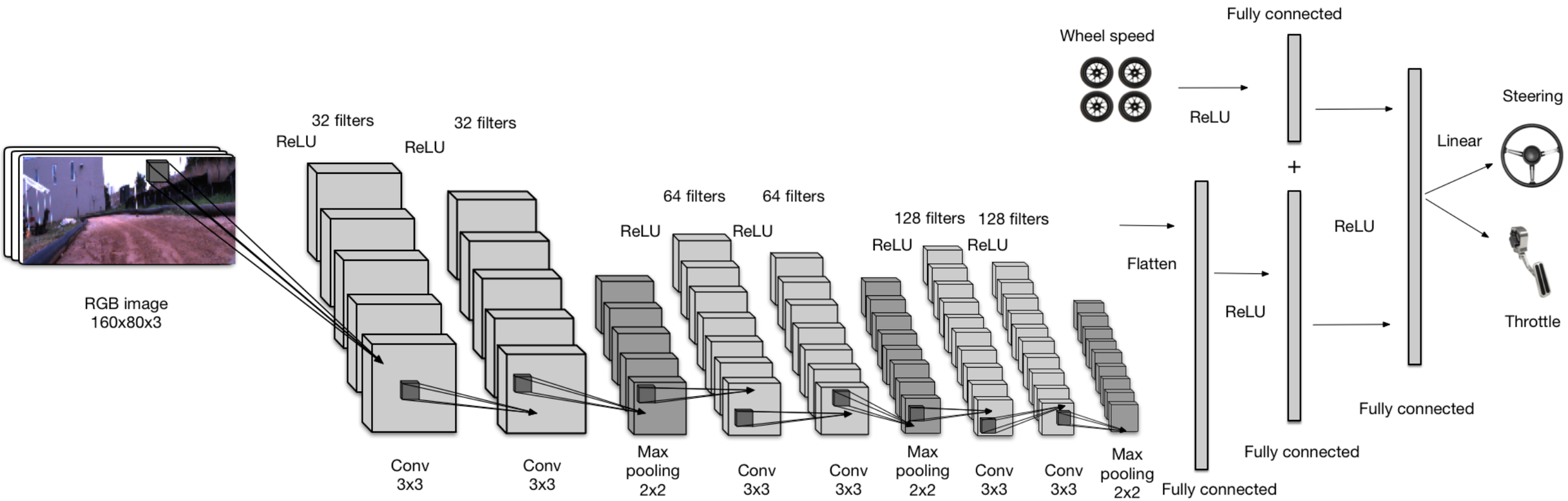
Results

Georgia Tech Institute for Robotics and Intelligent Machines

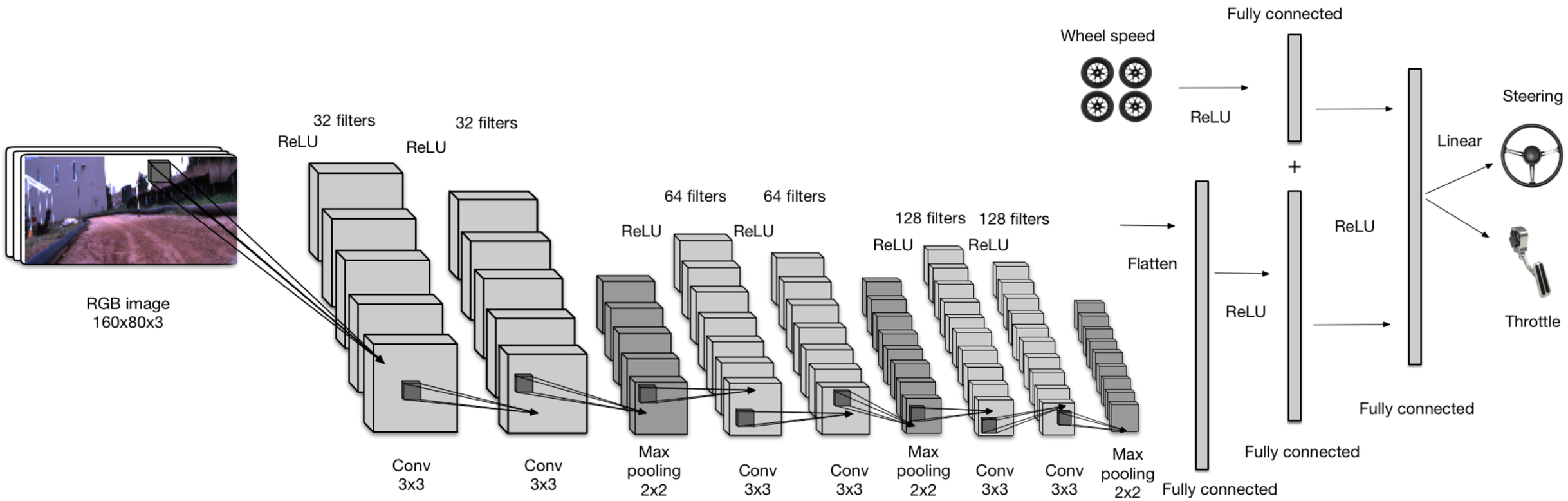

Human Expertise
Human-in-the-loop
Imitation Learning
Efficient Policy Search via Imitation
Learning Models for Planning & Control
Structured Policies for Robotics
Models
Adaptive Control
Learning

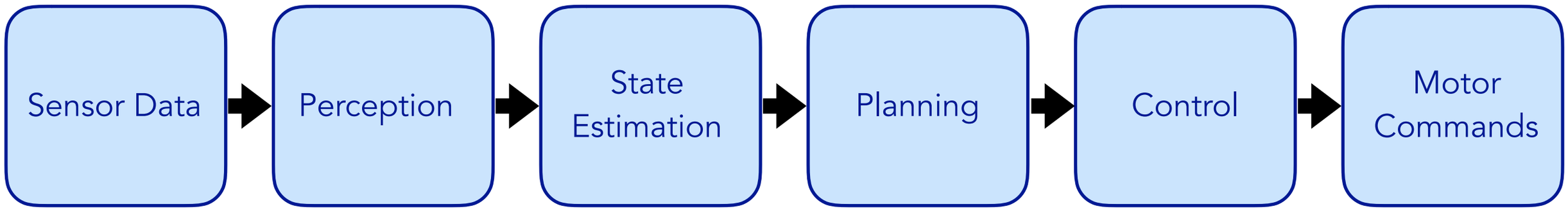
How should a robot be parameterized?

Georgia Tech Institute for Robotics and Intelligent Machines



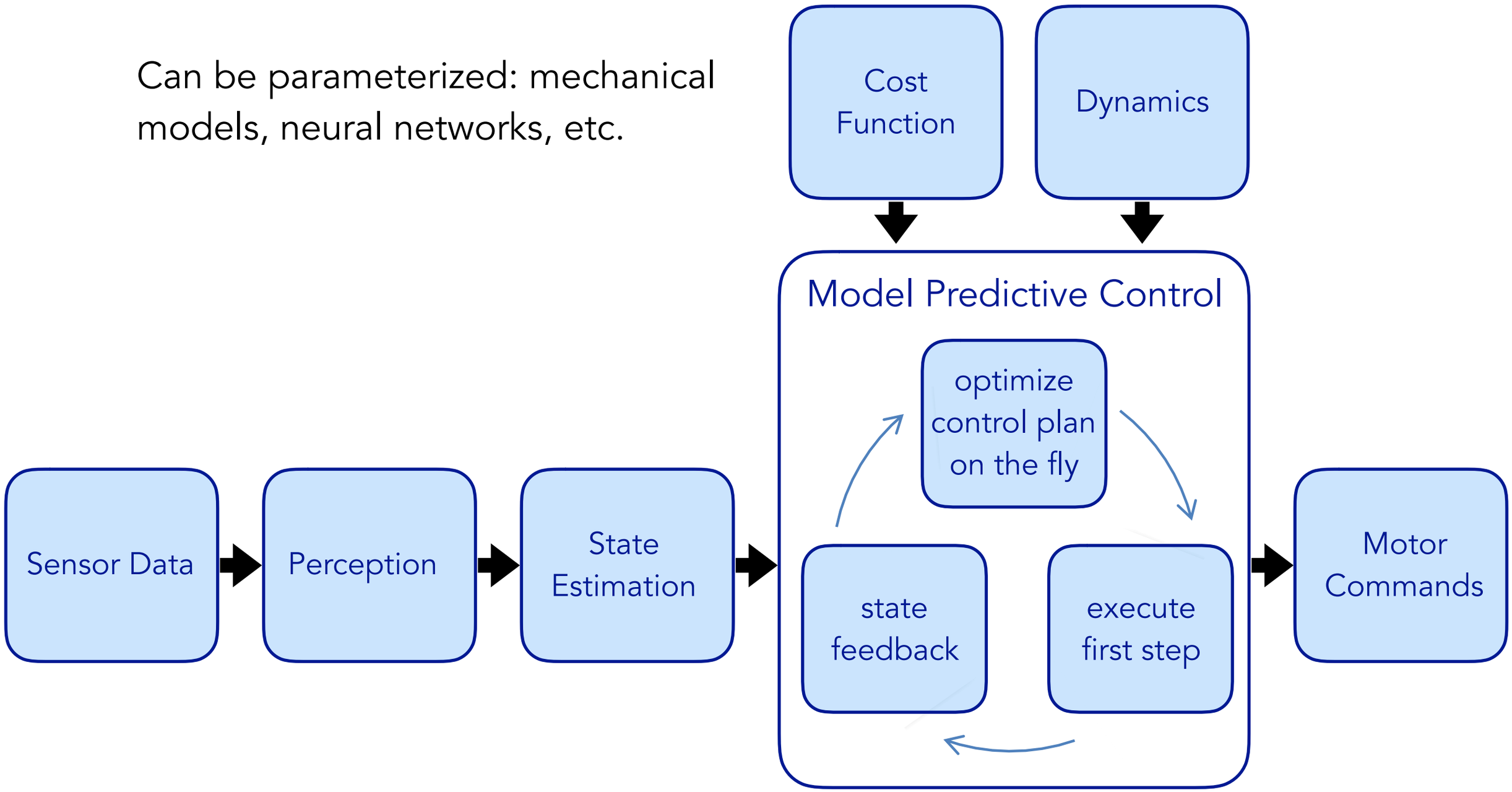
How should a robot be parameterized?





Model Predictive Control

Can be parameterized: mechanical models, neural networks, etc.



MPC as a Generic Policy Class

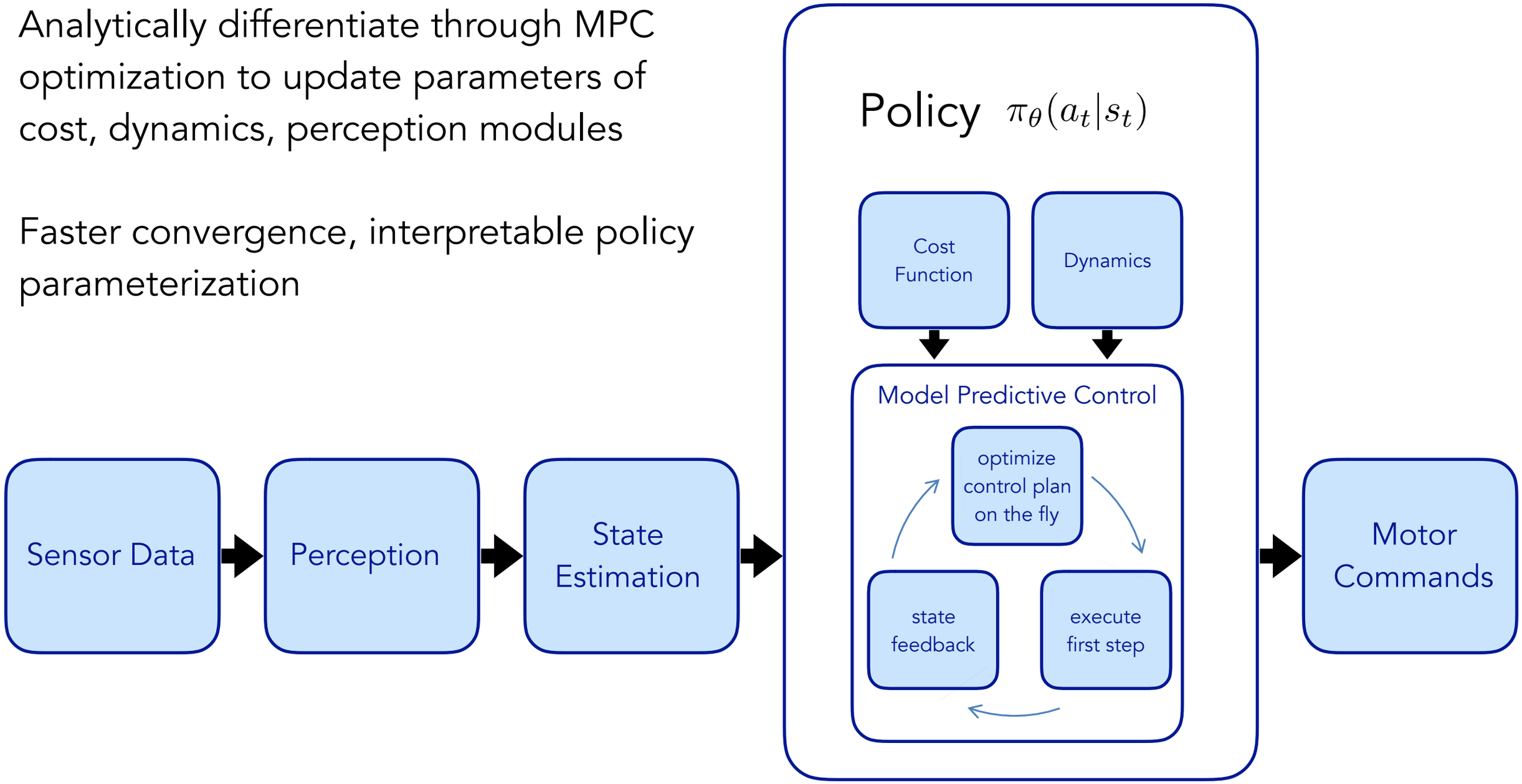
Georgia Tech Institute for Robotics and Intelligent Machines

Differentiable MPC for End-to-end Planning and Control

[Amos, Sacks, Jiminez, Boots, Kolter; NIPS 2018]

Analytically differentiate through MPC optimization to update parameters of cost, dynamics, perception modules

Faster convergence, interpretable policy parameterization



MPC as a Generic Policy Class

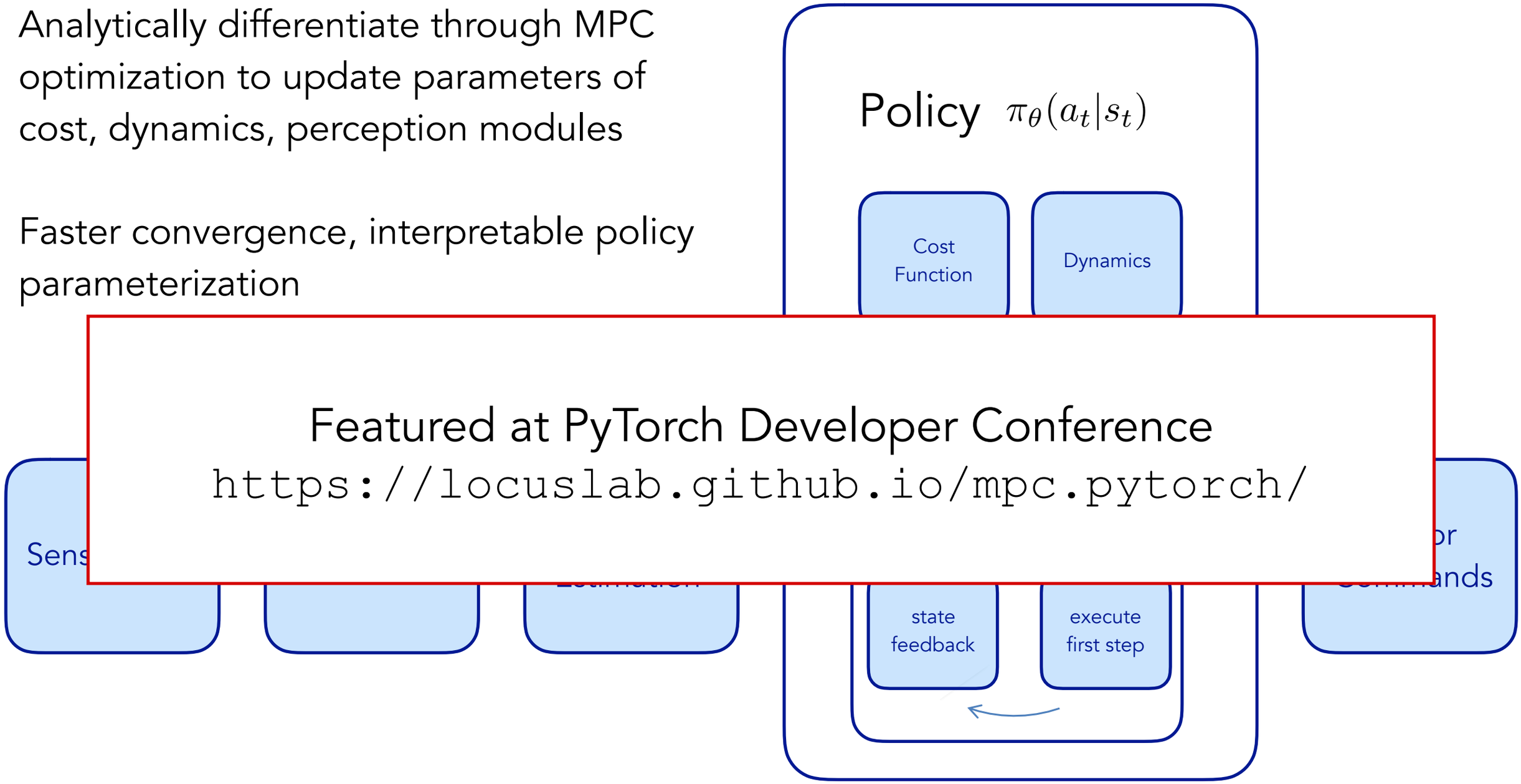
Georgia Tech Institute for Robotics and Intelligent Machines

Differentiable MPC for End-to-end Planning and Control

[Amos, Sacks, Jiminez, Boots, Kolter; NIPS 2018]

Analytically differentiate through MPC optimization to update parameters of cost, dynamics, perception modules

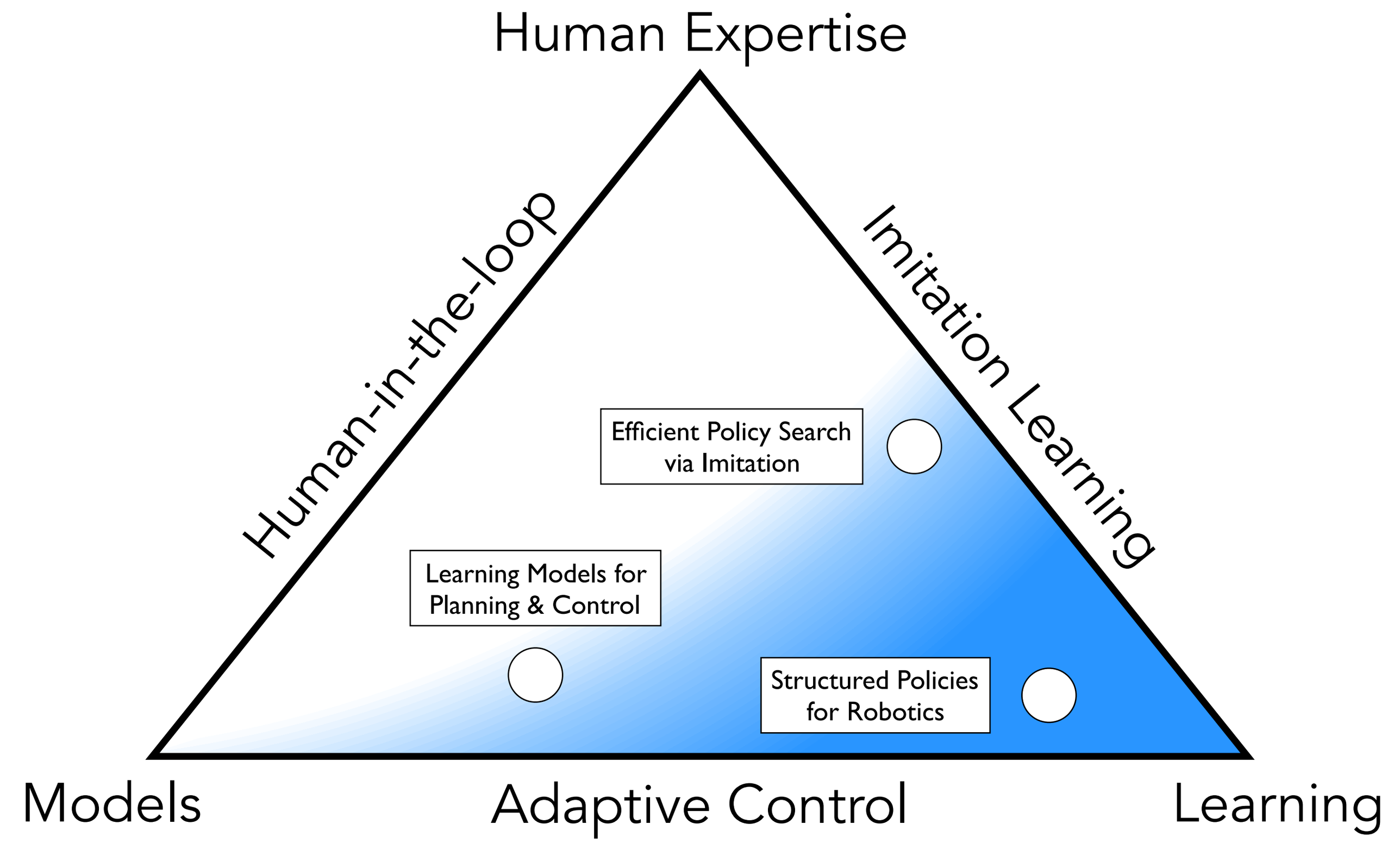
Faster convergence, interpretable policy parameterization



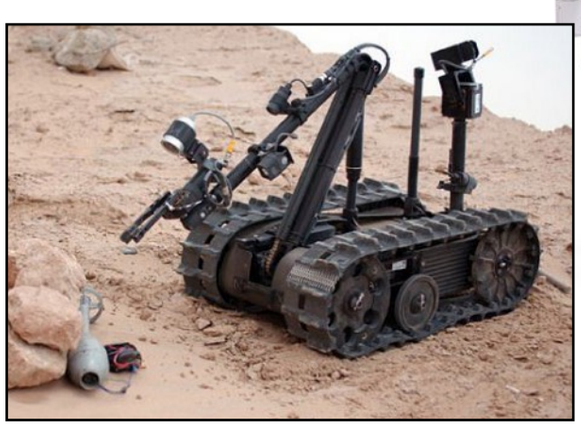
Featured at PyTorch Developer Conference

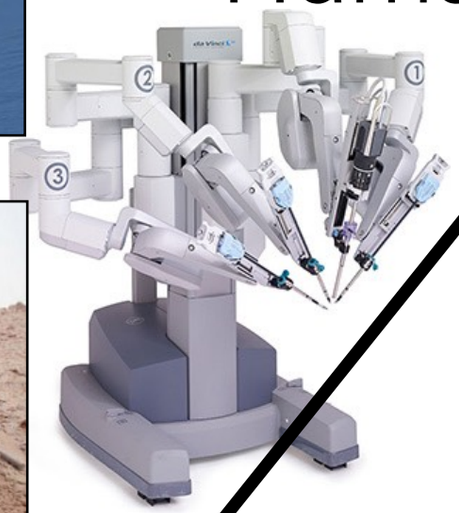
`https://locuslab.github.io/mpc.pytorch/`

Final Thoughts



Final Thoughts





Human Expertise

Models

Learning

Grady Williams

Nolan Wagener

Paul Drews

Brian Goldfain

Yunpeng Pan

Ching-An Cheng

Xinyan Yan

Kamil Saigol

Keuntaek Lee

Thanks!