Optimal Control and Machine Learning in Robotics

Mo Chen | mochen@cs.sfu.ca

Multi-Agent Robotic Systems Lab | https://sfumars.com

CIFAR AI Chair, Amii Fellow, NCRN Distal Fellow

School of Computing Science, Simon Fraser University

Optimal Control and Machine Learning

Machine learning challenges

- Data efficiency and exploration
- Generalization

Solutions aided by optimal control

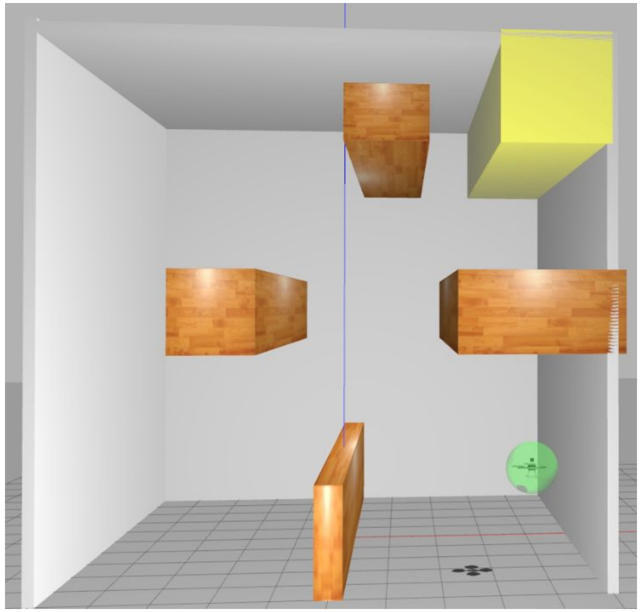
- Global guidance in exploration
- Low-level execution

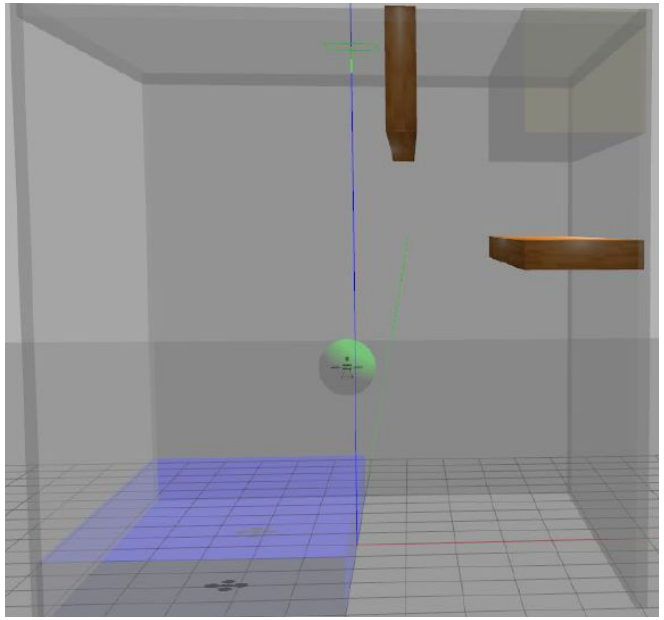
Outline

- Using control to improve data efficiency and exploration in RL
  - Global guidance via value functions from control
- Using control to improve generalization in learning
  - Follow-ahead robot using RL and control

Guiding Reinforcement Learning

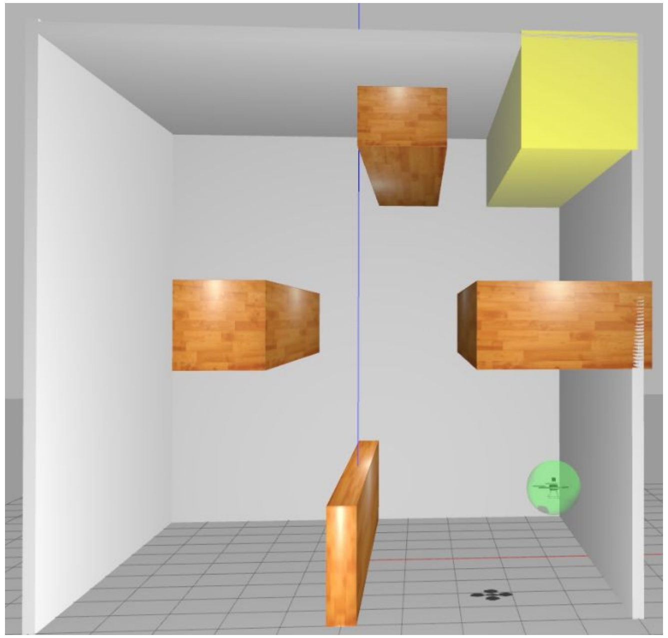
- Reinforcement learning:
  - Formalizes the idea of trial and error to train agents to behave in ways that are increasingly rewarding
  - Can require **a lot** of trial and error due to the challenge of exploration





Reinforcement Learning (Policy Gradients)

1. Try to perform a given task using the current policy many times
2. Adjust policy based on how well the task is performed
   - Goodness of performance measured by accumulated reward
   - But how good is good?



- Performance is compared to baselines
- Baselines are also learned from experience
  - Often, estimated by the value network / critic
  - But exploration is hard…

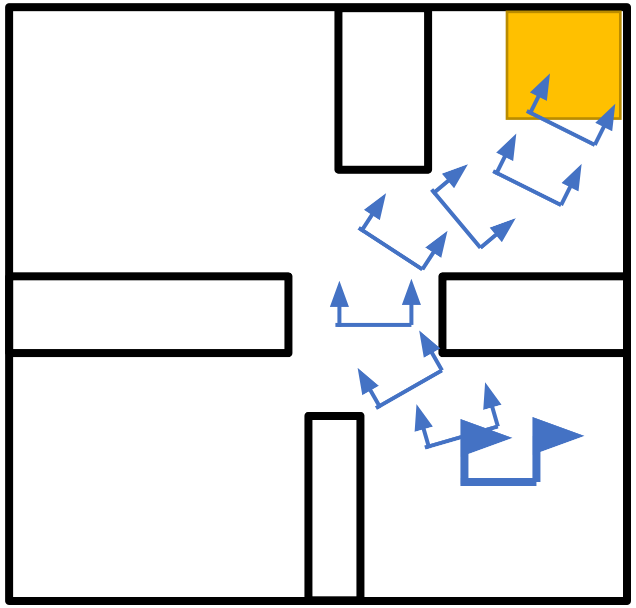
Baselines of Comparison

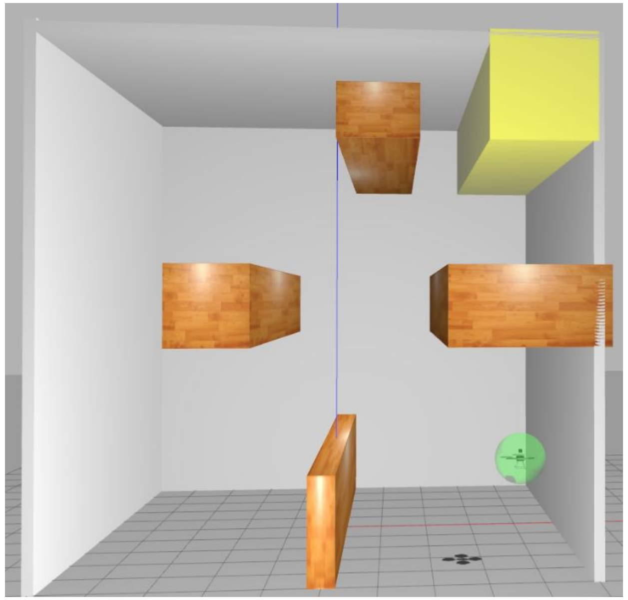
- Ideally, the baseline should be the optimal behaviour
  - But if we have the optimal behaviour, we’ve already solved the problem

- Optimal control-based approach:
  - Solve a **simplified version of the problem** using optimal control
  - Use the solution to the simplified problem as the baseline
  - Compatible and modular with respect to RL algorithms

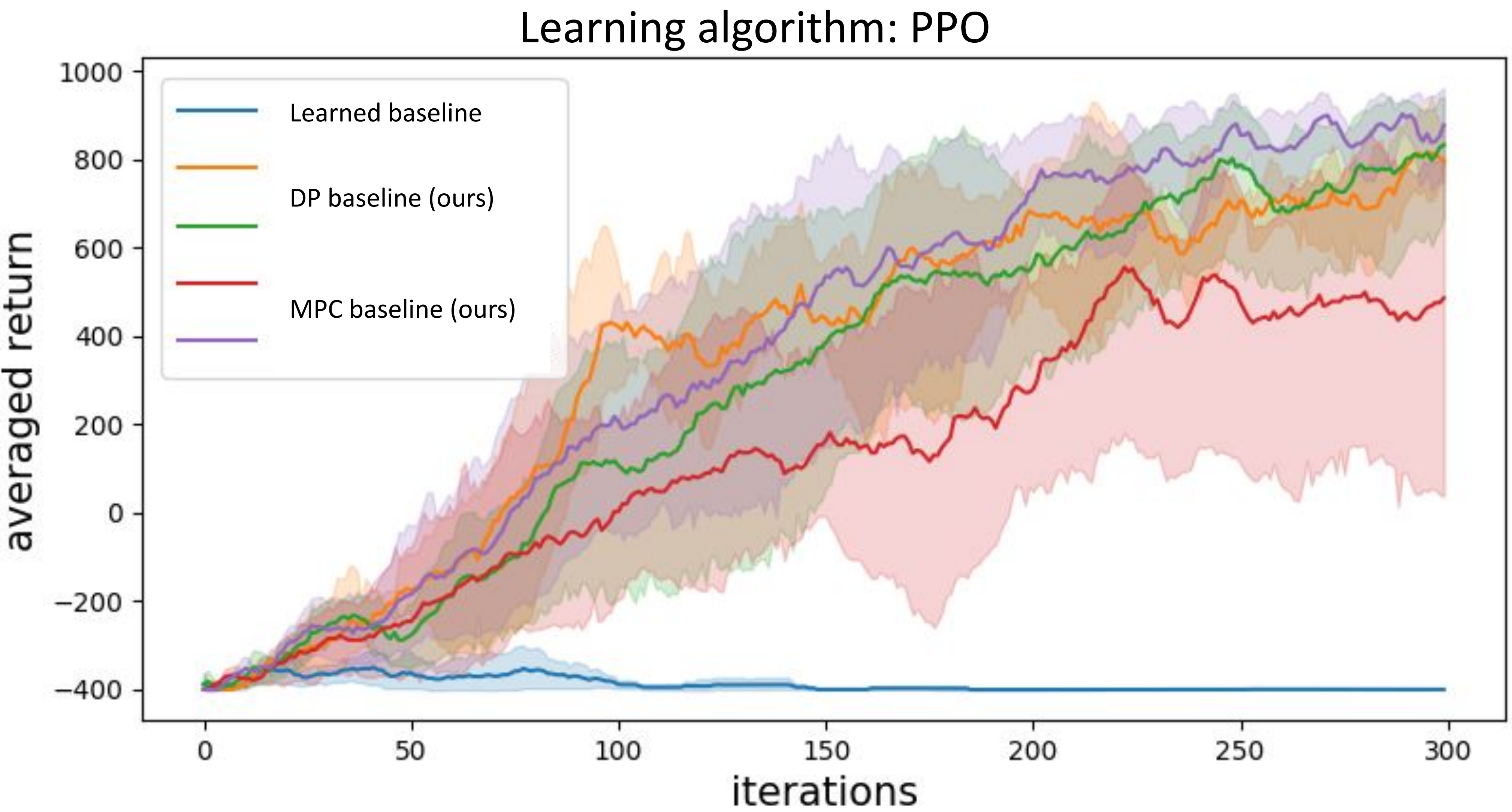
Baseline From Control

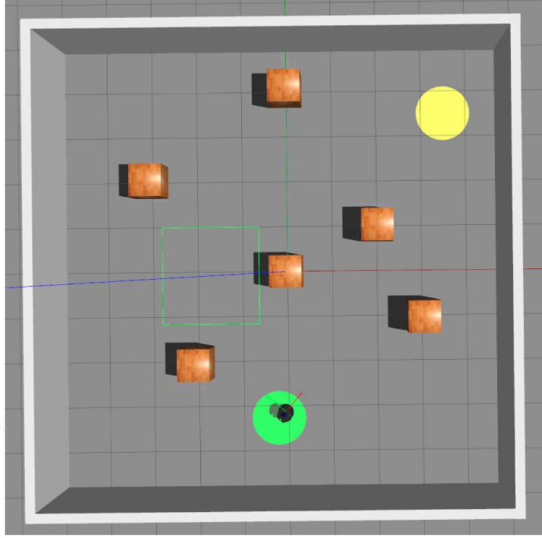
- Consider a simplified version of the problem
  - Approximate dynamics: $\dot{\hat{s}} = \hat{f}(\hat{s}, \hat{u})$
  - Solution via dynamic programming → value function $\hat{V}(\hat{s})$
  - Solution via MPC → $\tau_i = \left(s_{0,i}, a_{0,i}, s_{1,i}, a_{1,i}, \ldots\right)$ → value function $\hat{V}(\hat{s})$



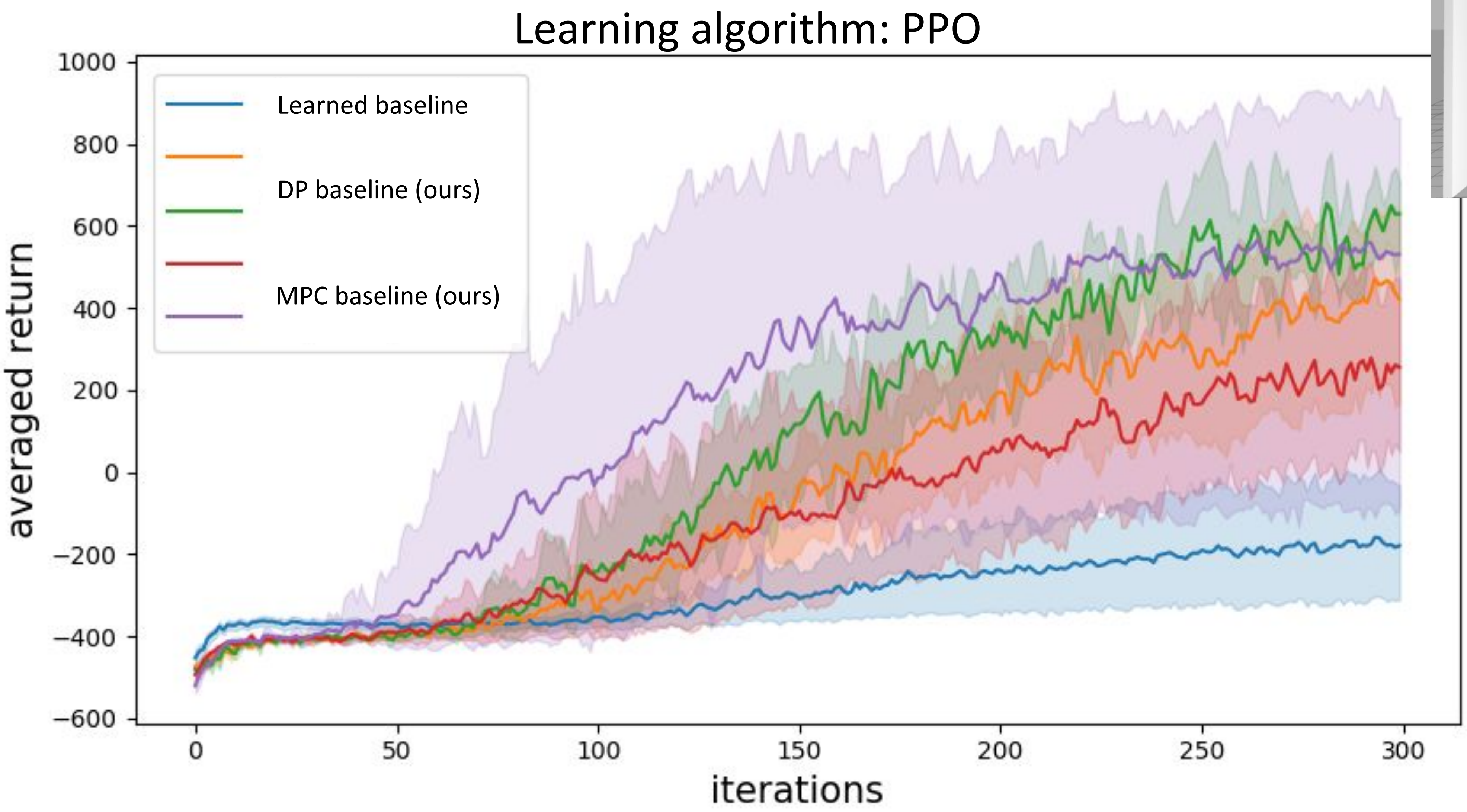


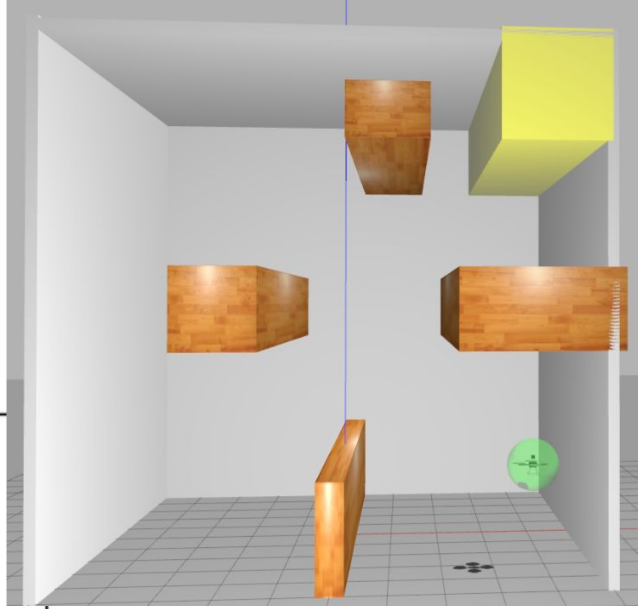
TurtleBot Environment



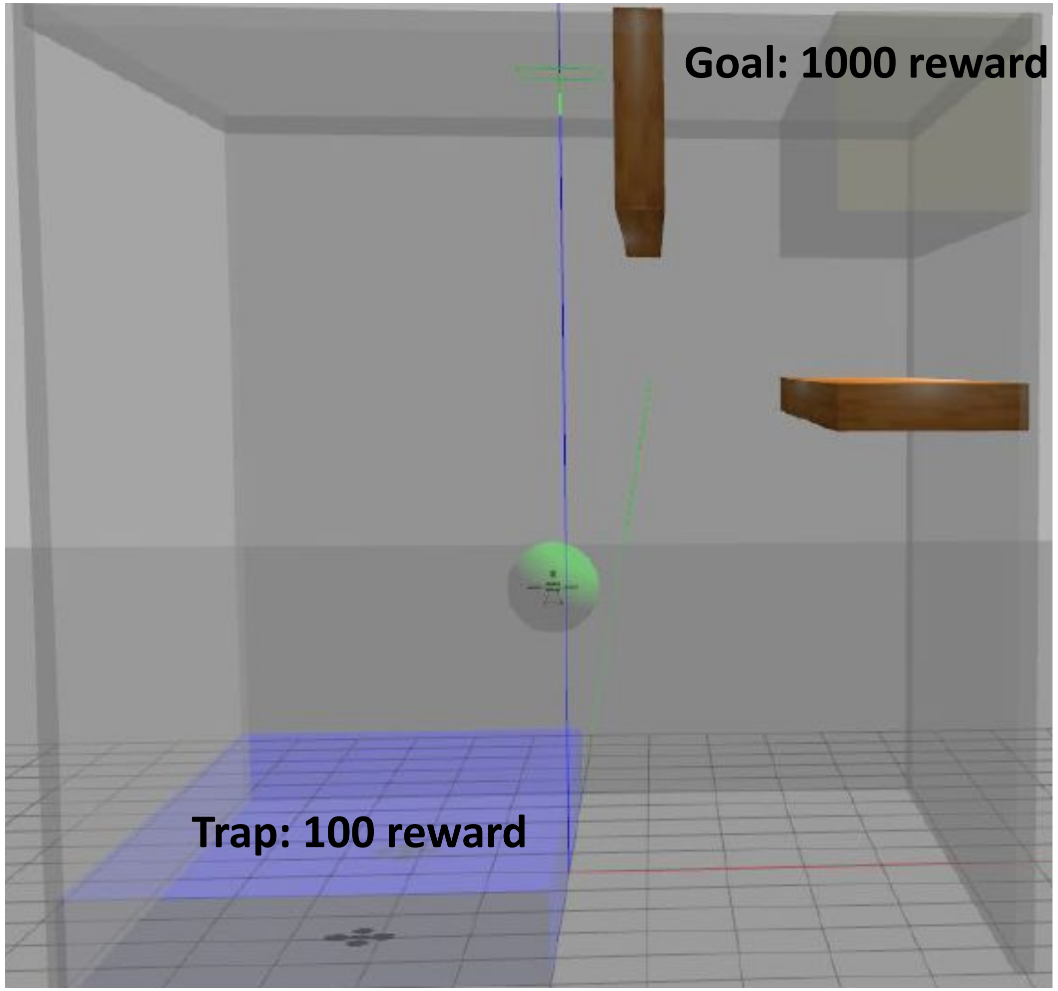


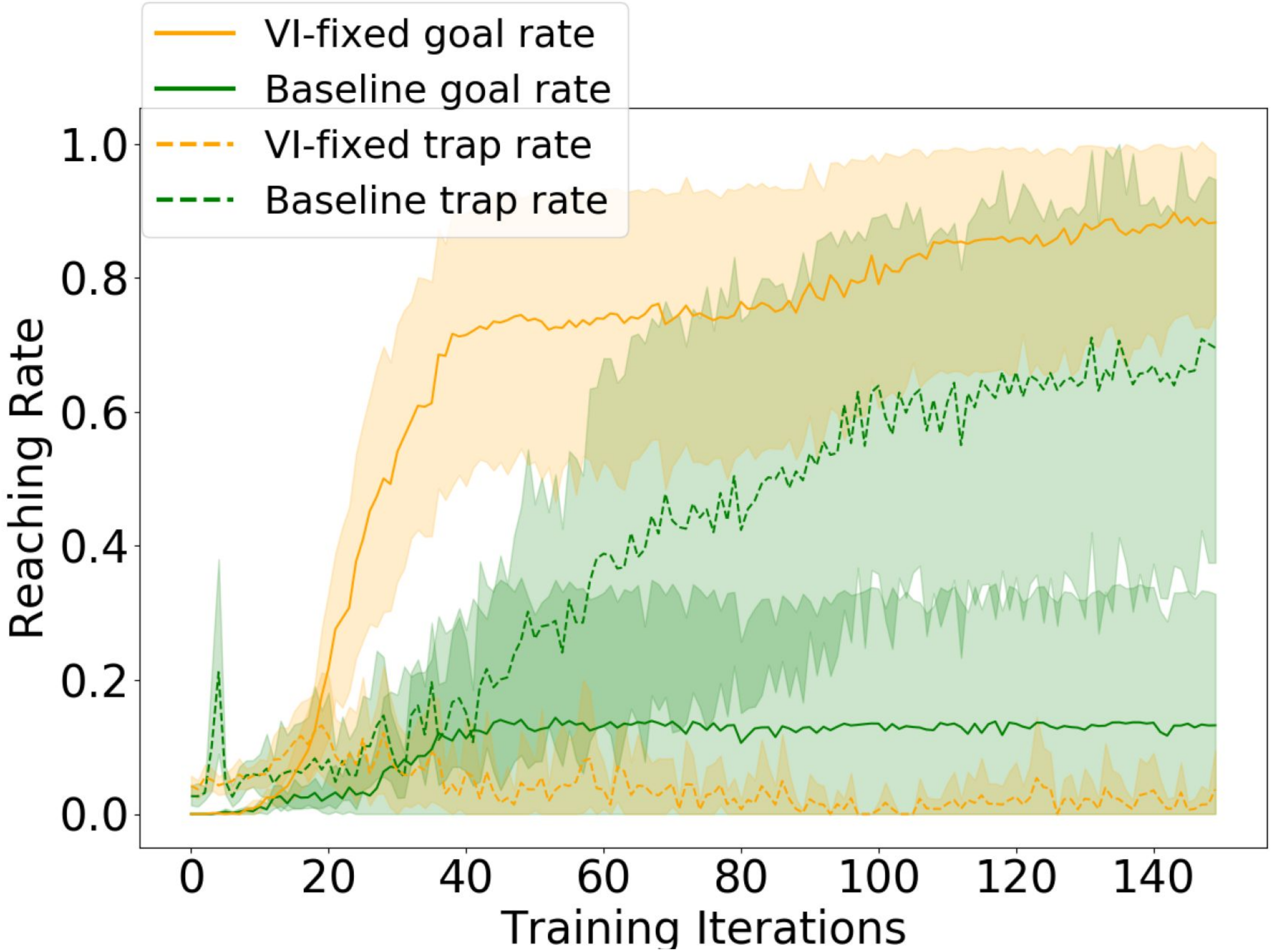
Quadrotor Environment





Trap Environment





Outline

- Using control to improve data efficiency and exploration in RL
  - Global guidance via value functions from control
- Using control to improve generalization in learning
  - Follow-ahead robot using RL and control

People Following Robots

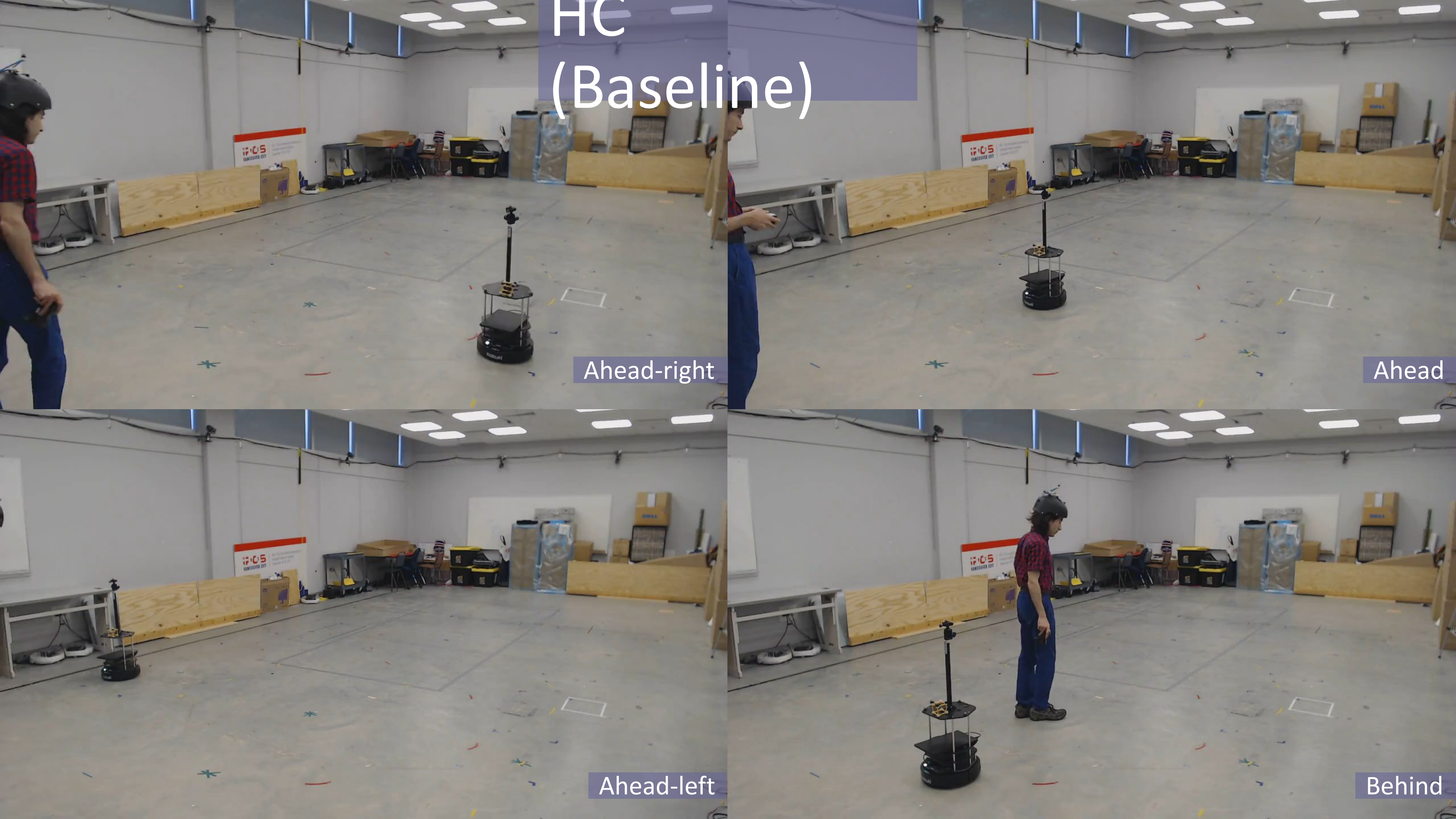
Autonomous suitcase (Travelmate)

Companion dog

Following Ahead

- Currently, robots can follow behind and follow beside a person
- “Following” ahead or in front of a person is much harder
- Challenge:
  - Predict the future trajectory of the person
  - Find a feasible and safe way to navigate ahead

HC
(Baseline)



High Level Idea

1. Predict human navigational intent
   - Difficult for model-based methods
   - Use data-driven method

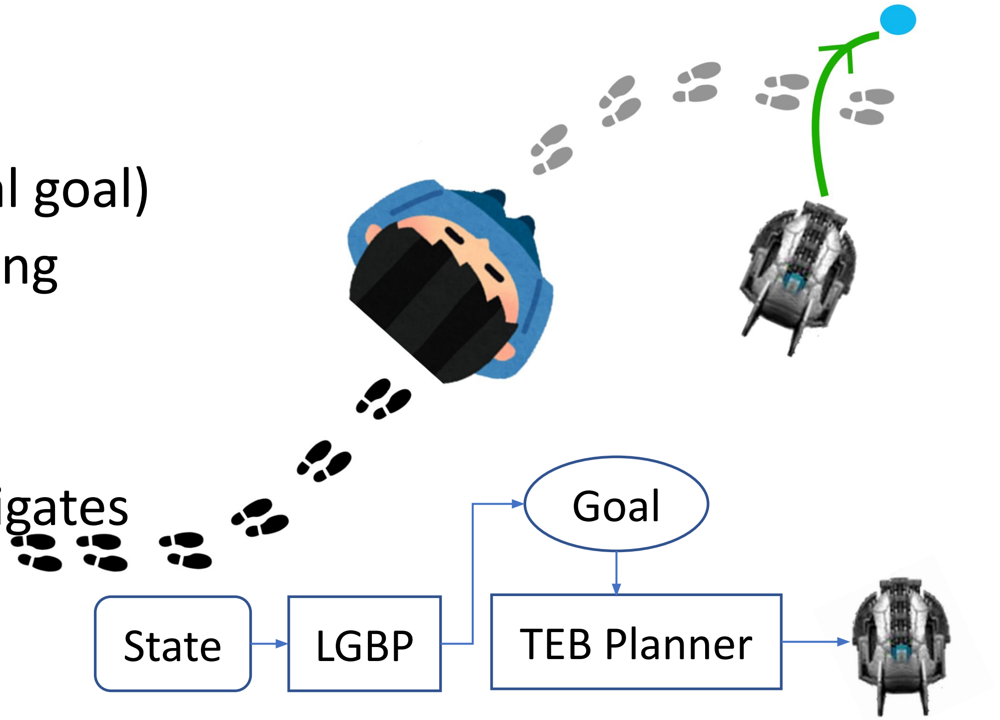
2. Move in front of the human
   - Easy if we know where the robot should go
   - Use control-based methods
   - No need to learn what we already know

Nikdel et al., “LBGP: Learning Based Goal Planning for Autonomous Following in Front,” ICRA 2021.

Task Decomposition

1. High-level decision-making
   - Choose where to go (output robot navigational goal)
   - Reinforcement learning with curriculum learning

2. Low-level execution
   - Classical timed-elastic band (TEB) planner navigates the robot toward the goal
   - No need to learn what we already know



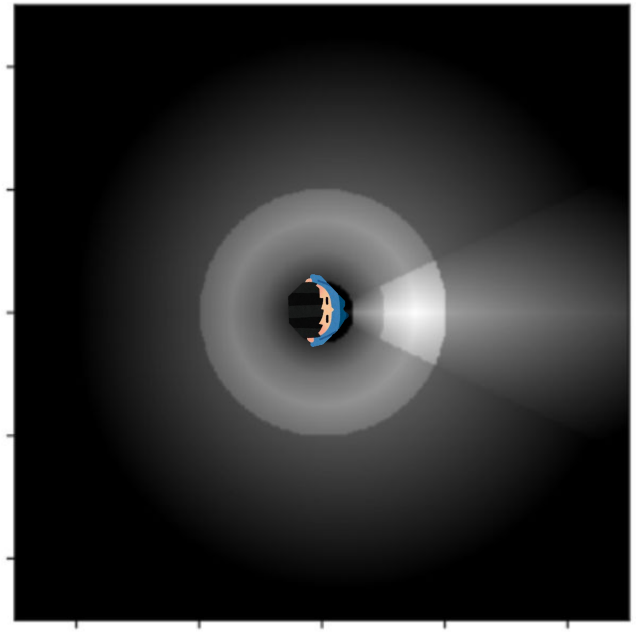
Nikdel et al., “LBGP: Learning Based Goal Planning for Autonomous Following in Front,” ICRA 2021.

Curriculum Reinforcement Learning

- Simulate increasingly difficult human trajectories

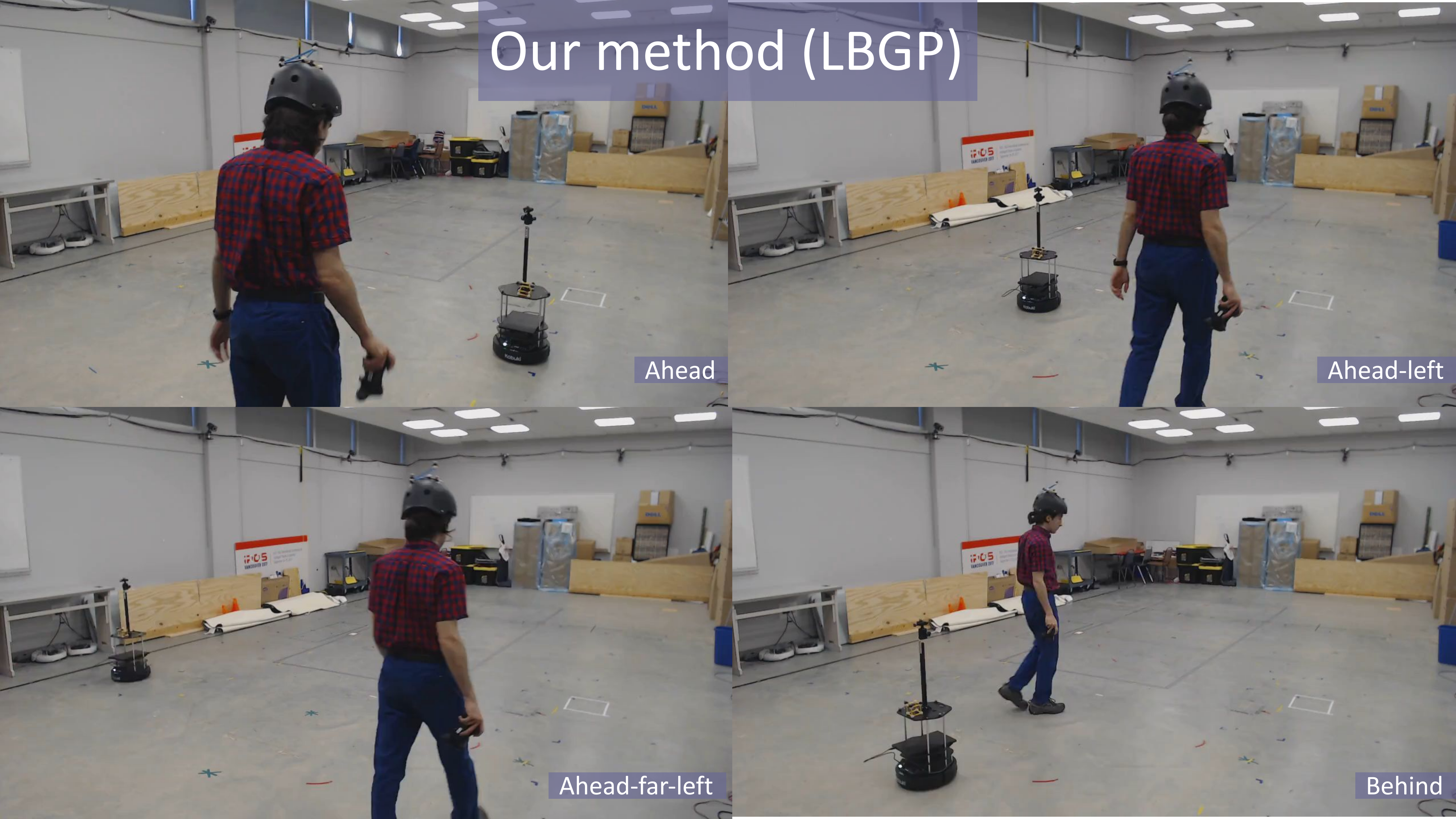
- Training done in simulation
    - Using Distributed Distributional Deterministic Policy Gradients
    - Observation space: history of relative pose of robot wrt human
    - Action space: short term robot navigational goal (waypoint)
    - Reward: based on relative pose of robot wrt human

Reward function



- Zero-shot simulation to reality transfer to unseen human trajectories


Our method (LBGP)
Ahead-right
Ahead
Ahead-left
Behind


Our method (LBGP)
Ahead
Ahead-left
Ahead-far-left
Behind

Summary

- Using control to improve data efficiency and exploration in RL
  - Global guidance via value functions from control
- Using control to improve generalization in learning
  - Follow-ahead robot using RL and control

Thank You!

Mo Chen | mochen@cs.sfu.ca

Multi-Agent Robotic Systems Lab | https://sfumars.com