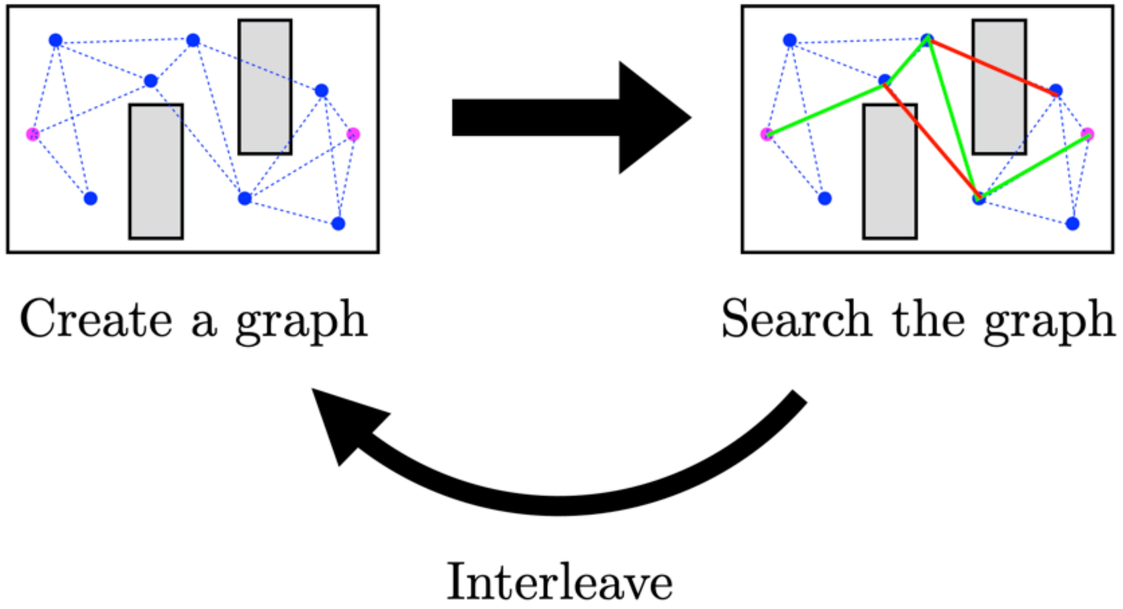
Learning to Search

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

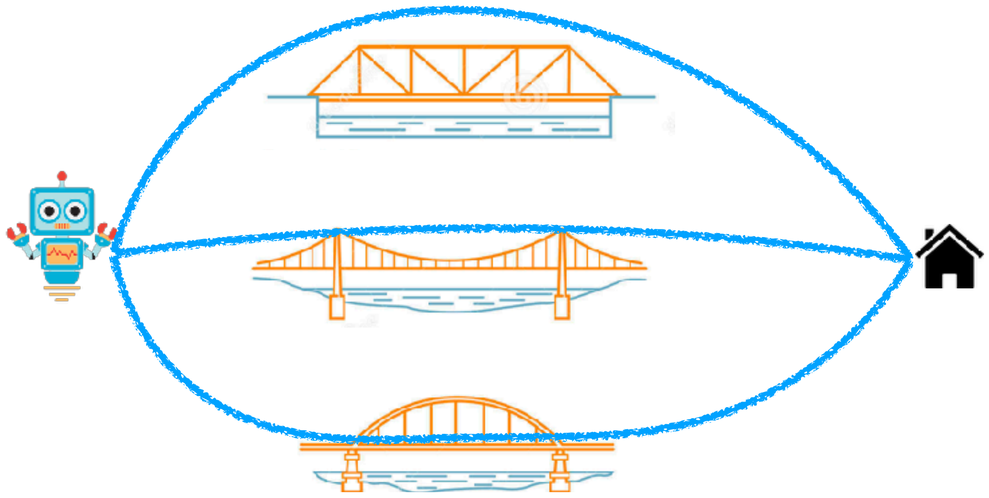
We can use learning for TWO things:

HOW to search?



Learn how to speed up search

WHAT to search?



Learn the underlying cost of a path

CRUSHER robot from CMU

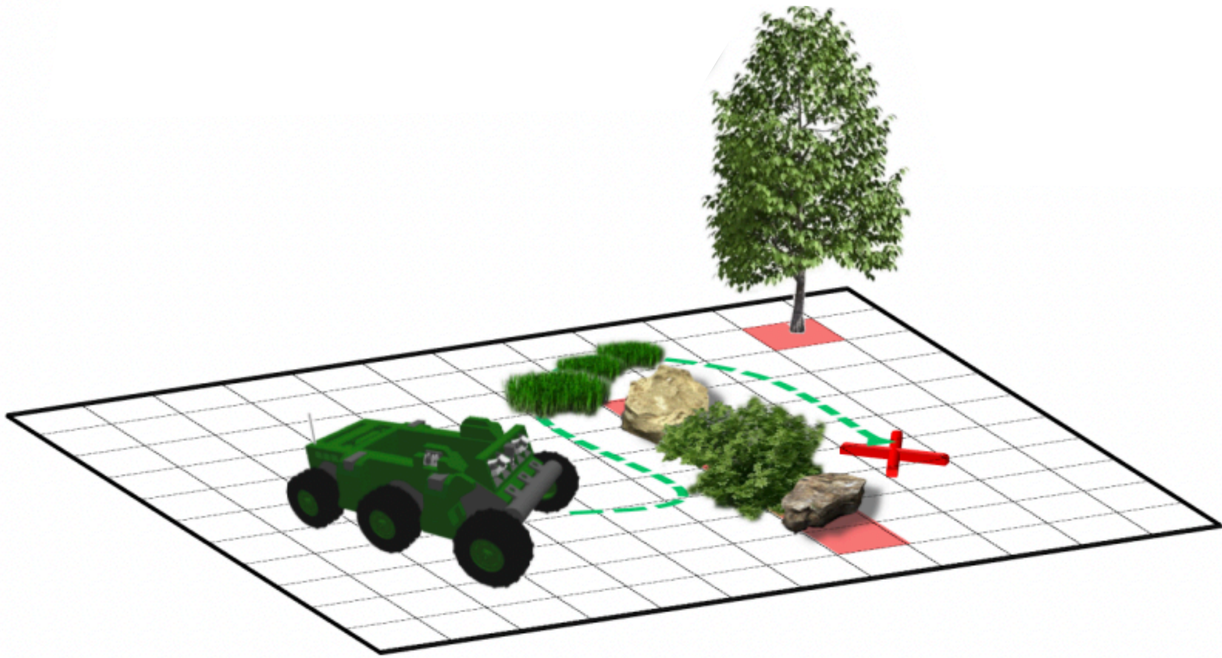
Activity!

Think-Pair-Share!

Think (30 sec): We want CRUSHER to go from A to B. What are some of the components for the cost function? How can we weigh these various components?

Pair: Find a partner

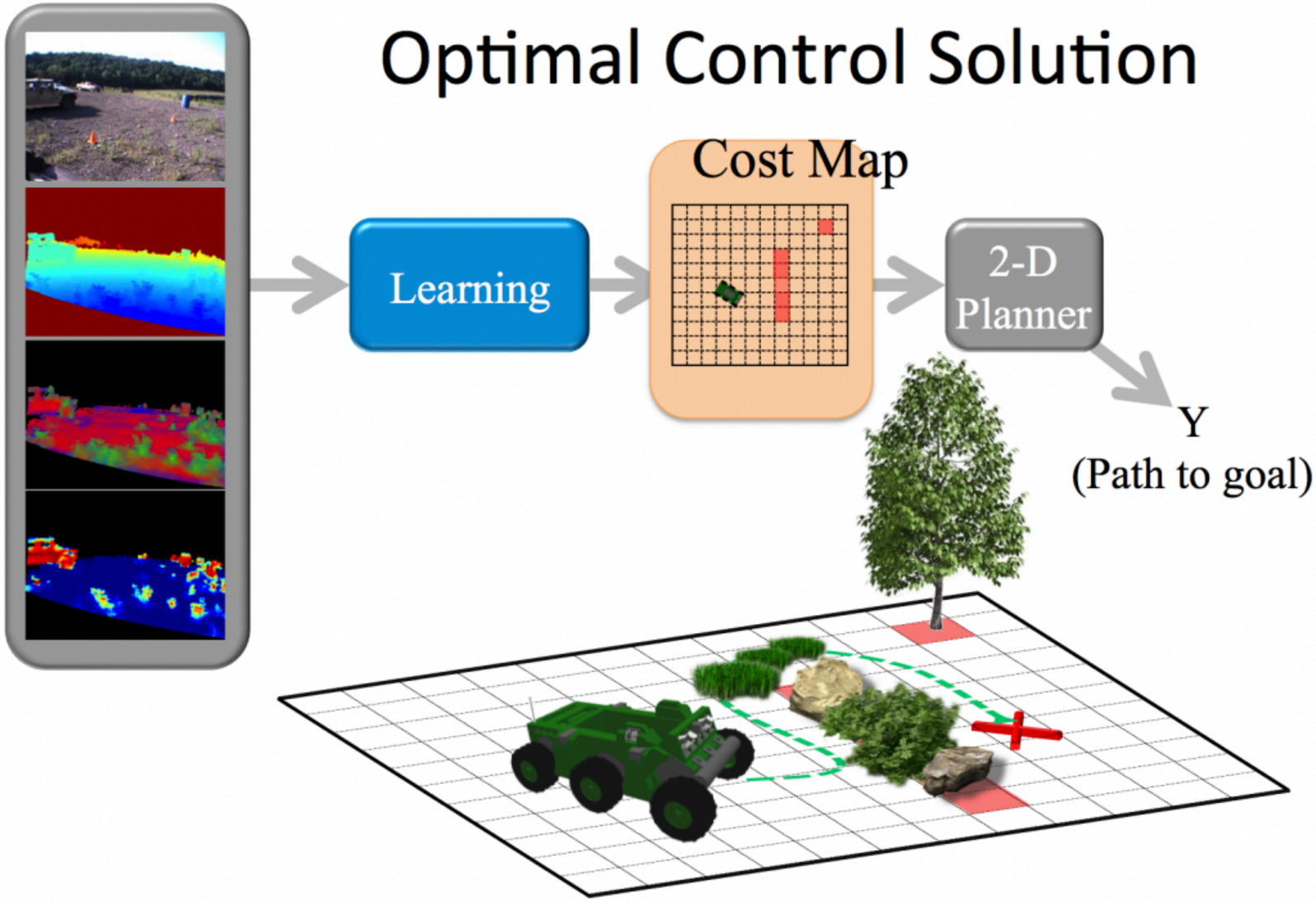
Share (45 sec): Partners exchange ideas



Wait ... why can't we use DAGGER?

Why learn cost functions vs learn the policy?

Can we learn a cost function for CRUSHER navigation?



Let’s formalize!

Learning to Search (LEARCH)

Min distance

Stay on roads

Stay near trees

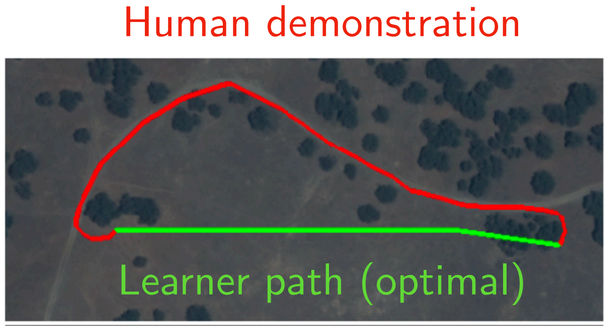
Learning to Search:
Functional Gradient Techniques for Imitation Learning

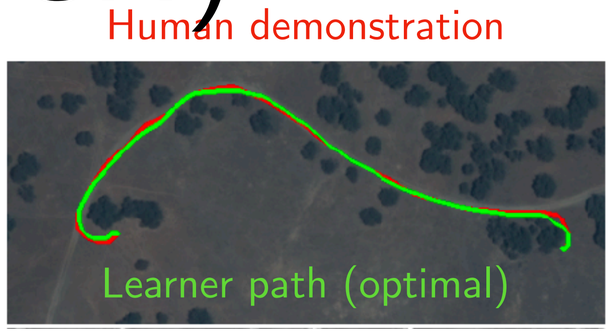
Nathan D. Ratliff
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ndr@ri.cmu.edu

David Silver
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
dsilver@ri.cmu.edu

J. Andrew Bagnell
Robotics Institute and Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213
dbagnell@ri.cmu.edu

Learning to Search (LEARCH)



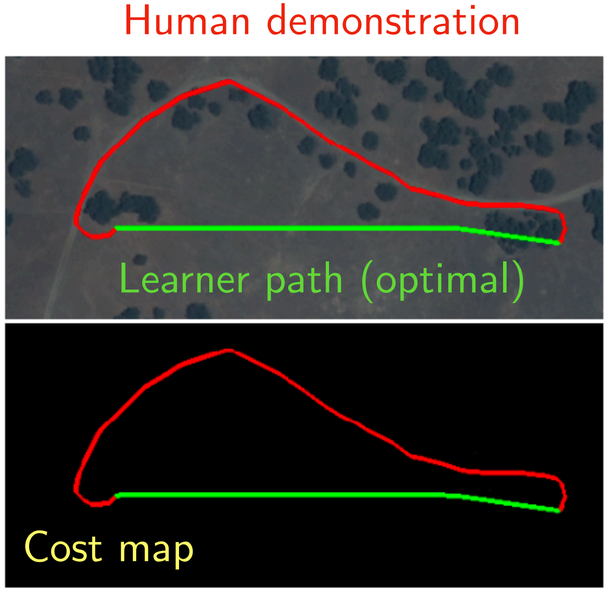


Given dataset: $\{\xi_i^h, \phi_i\}_{i=1}^N$

(Human demo) *(Map)*

Solve for cost $C_\theta(\xi)$

Learning to Search (LEARCH)



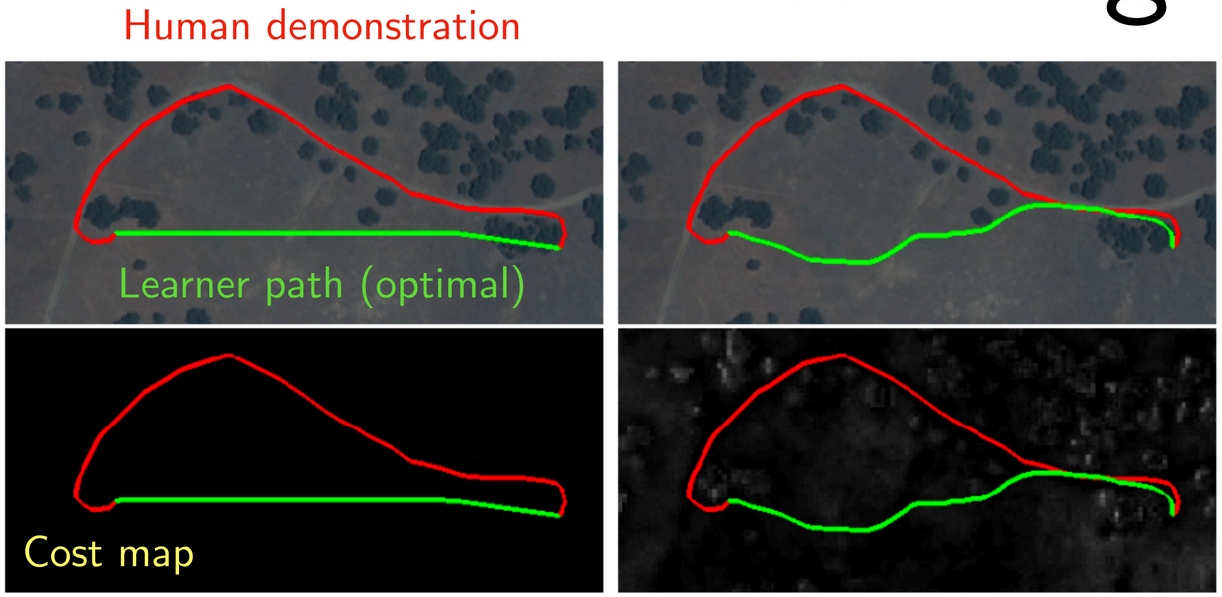
for $i = 1,\ldots,N$ # Loop over datapoints

$$\xi_i^* = \min_{\xi}[C_\theta(\xi, \phi_i) - \gamma(\xi, \xi^h)]$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i^*, \phi_i) + \nabla_\theta R(\theta)]$$

(Push down human cost) *(Push up planner cost)* # Update cost

Learning to Search



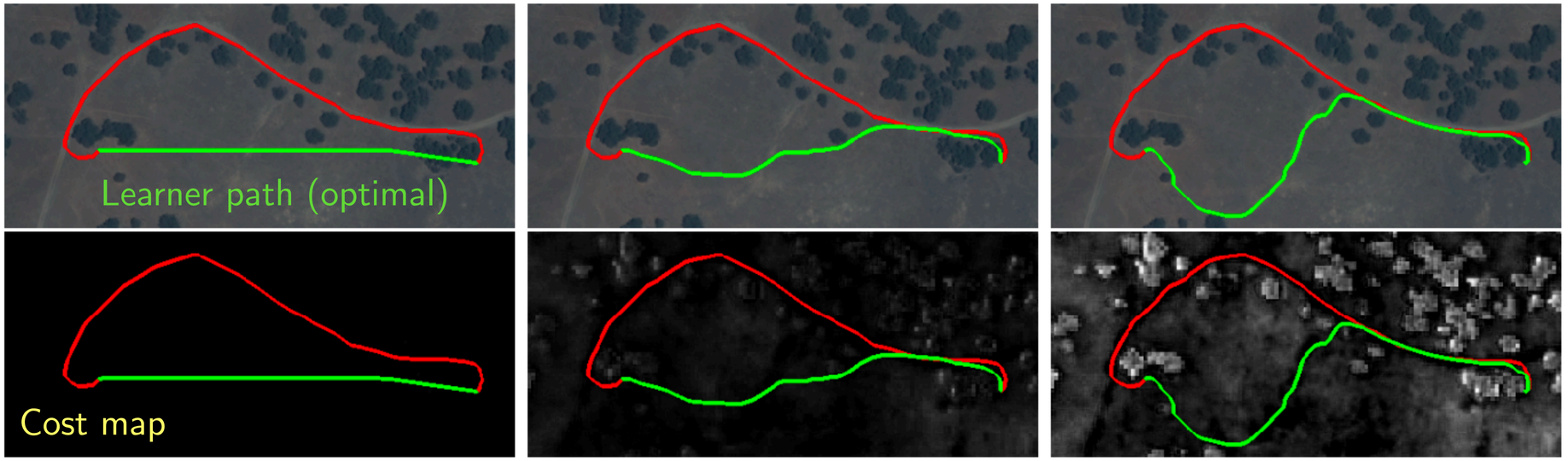
for $i = 1,\ldots,N$ # Loop over datapoints

$$\xi_i^* = \min_{\xi}[C_\theta(\xi, \phi_i) - \gamma(\xi, \xi^h)]$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i^*, \phi_i) + \nabla_\theta R(\theta)]$$ # Update cost

(Push down human cost) *(Push up planner cost)*

Learning to Search (LEARCH)



for $i = 1,\ldots,N$ # Loop over datapoints

$$\xi_i^* = \min_{\xi}[C_\theta(\xi, \phi_i) - \gamma(\xi, \xi^h)]$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i^*, \phi_i) + \nabla_\theta R(\theta)]$$

(Push down human cost) *(Push up planner cost)* # Update cost