Learning
in
Heuristic Search-based Planning

Maxim Likhachev

Associate Professor

Robotics Institute/NREC

Carnegie Mellon University

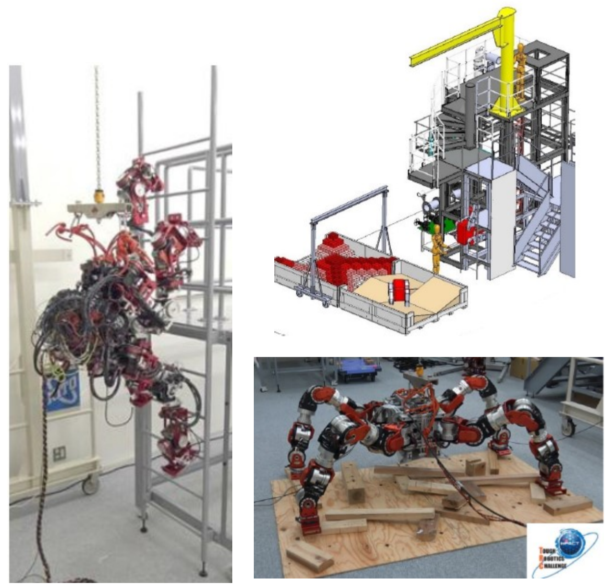
Search-based Planning Lab (SBPL)

Joint work with

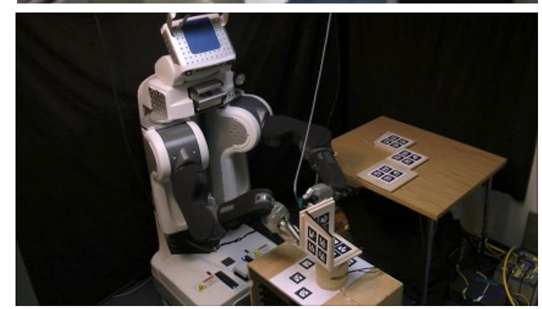
Ishani Chatterjee, Ben Cohen, Andrew Dornbush, Victor Hwang, Venkatraman Narayanan, Michael Phillips, Kalyan Vasudev

Going into the Real-world

- Robot models and simple world interactions can be pre-encoded
- Planning on those models enables the robots to operate under benign/narrow conditions right away



Waseda/Mitsubishi robot



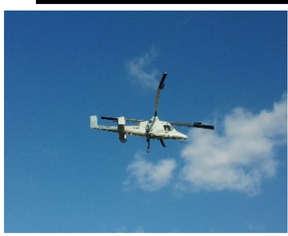
- **Real-world: real-time + going beyond what's given**

Learning in Search-based Planning

Speeding up planning

Learning cost function

Going beyond the prior model



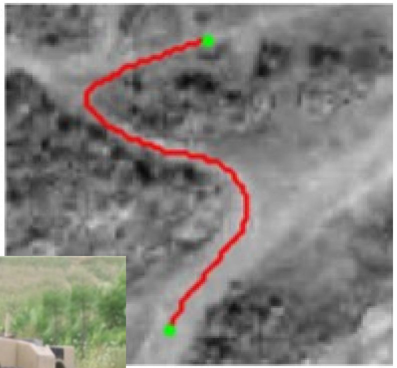
Waseda/ Mitsubishi

Re-use of previous results within search (Phillips et al.,'12; Islam et al.,'18)
Learning heuristic functions (Bhardwaj et al.,'17; Paden & Frazzoli,'17; Thayer et al.,'11)
Learning order of expansions (Choudhary et al.,'17)

Speeding up planning

Learning cost function

Going beyond the prior model



Crusher (from Ratliff et a., '09 paper)

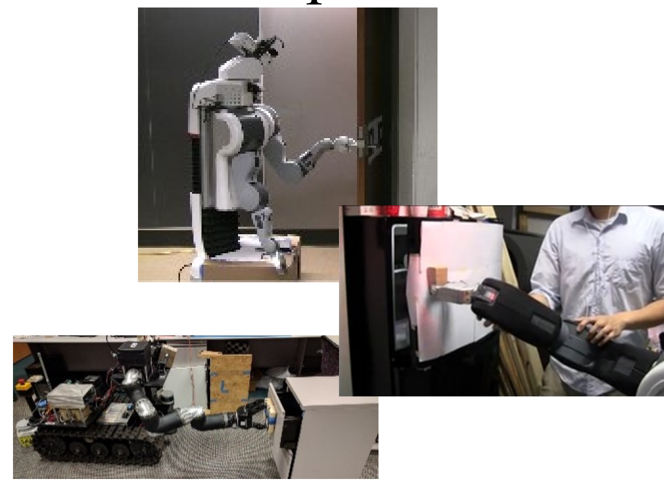
Learning a cost function from demonstrations (Ratliff et al.,'09; Wulfmeier et al.,'17)

Learning in Search-based Planning

Speeding up planning

Learning cost function

Going beyond the prior model

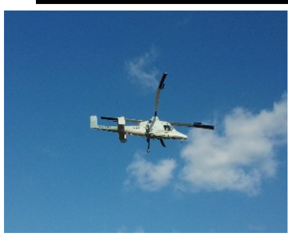


Learning additional dimensions to reason over (Phillips et al.,'13)
Combining learned skills and prior model (Vasudev et al., ongoing)

Speeding up planning

Learning cost function

Going beyond the prior model



Waseda/
Mitsubishi

Re-use of previous results within search (Phillips et al.,'12; Islam et al.,'18)
Learning heuristic functions (Bhardwaj et al.,'17; Paden & Frazzoli,'17; Thayer et al.,'11)
Learning order of expansions (Choudhary et al.,'17)

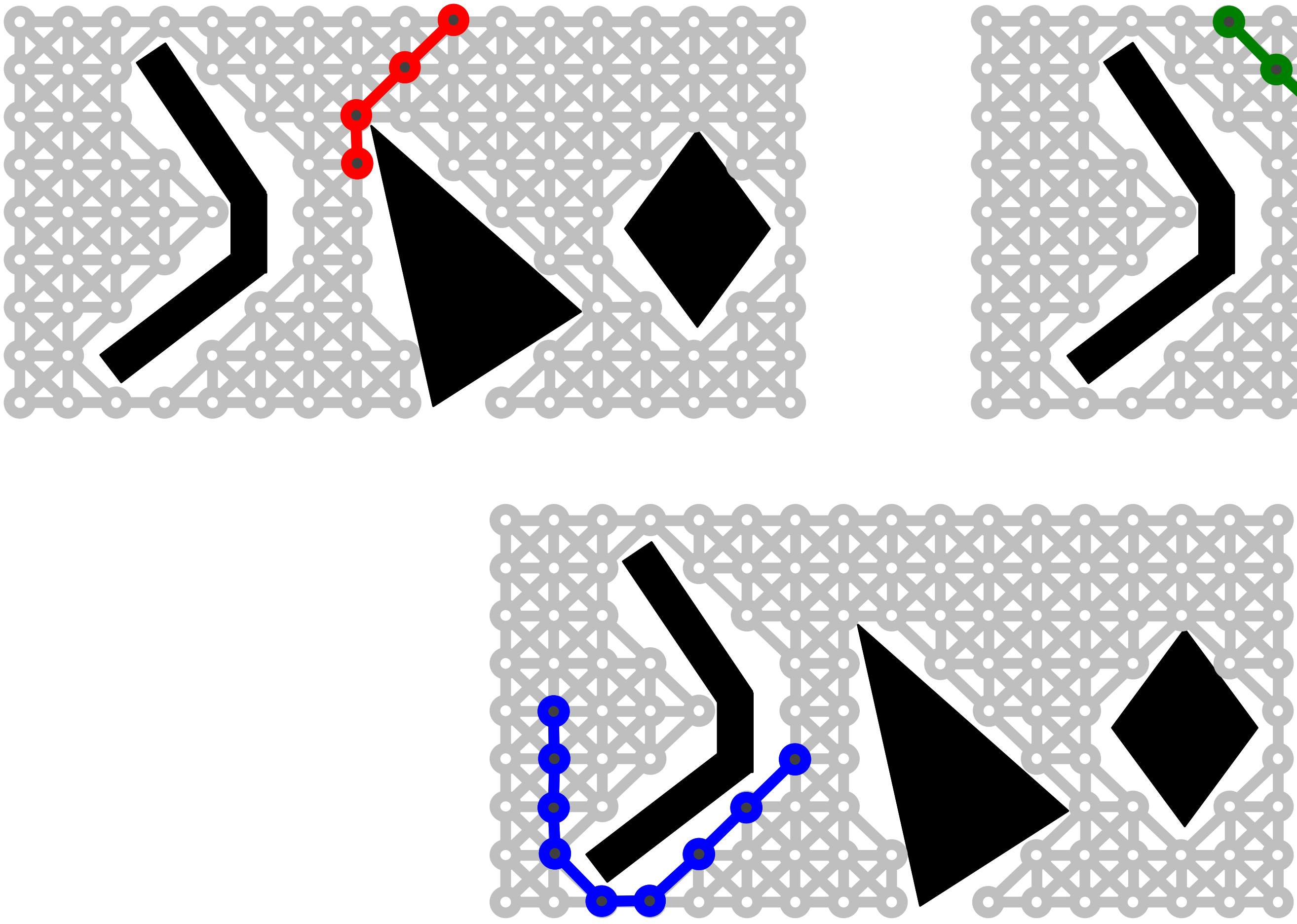
Experience Graphs [Phillips et al., RSS'12]

- Many planning tasks are repetitive
  - loading a dishwasher
  - opening doors
  - moving objects around a warehouse
  - …

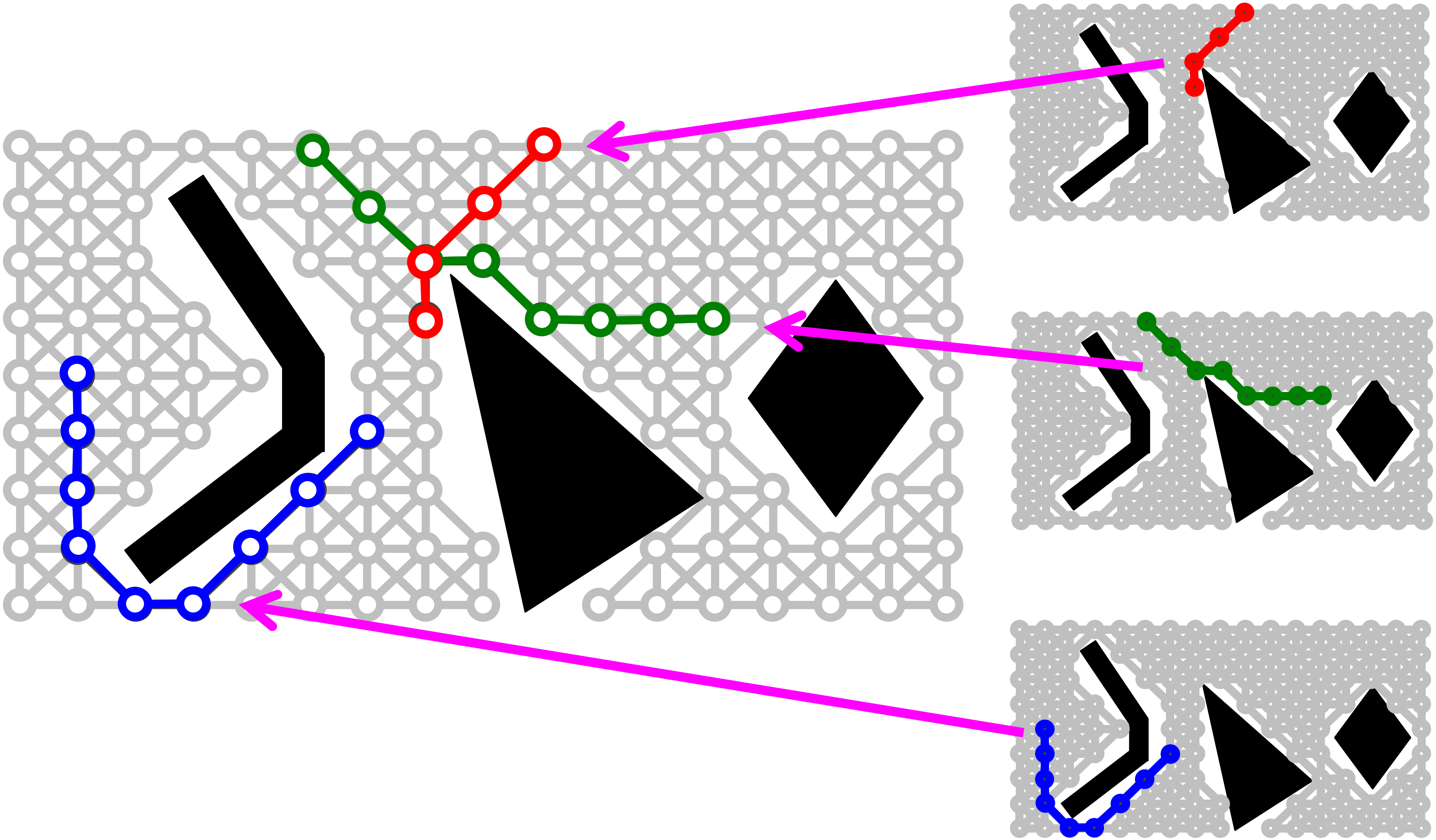
- Can we re-use prior experience to accelerate planning, in the context of search-based planning?

- Especially useful for high-dimensional problems such as mobile manipulation!

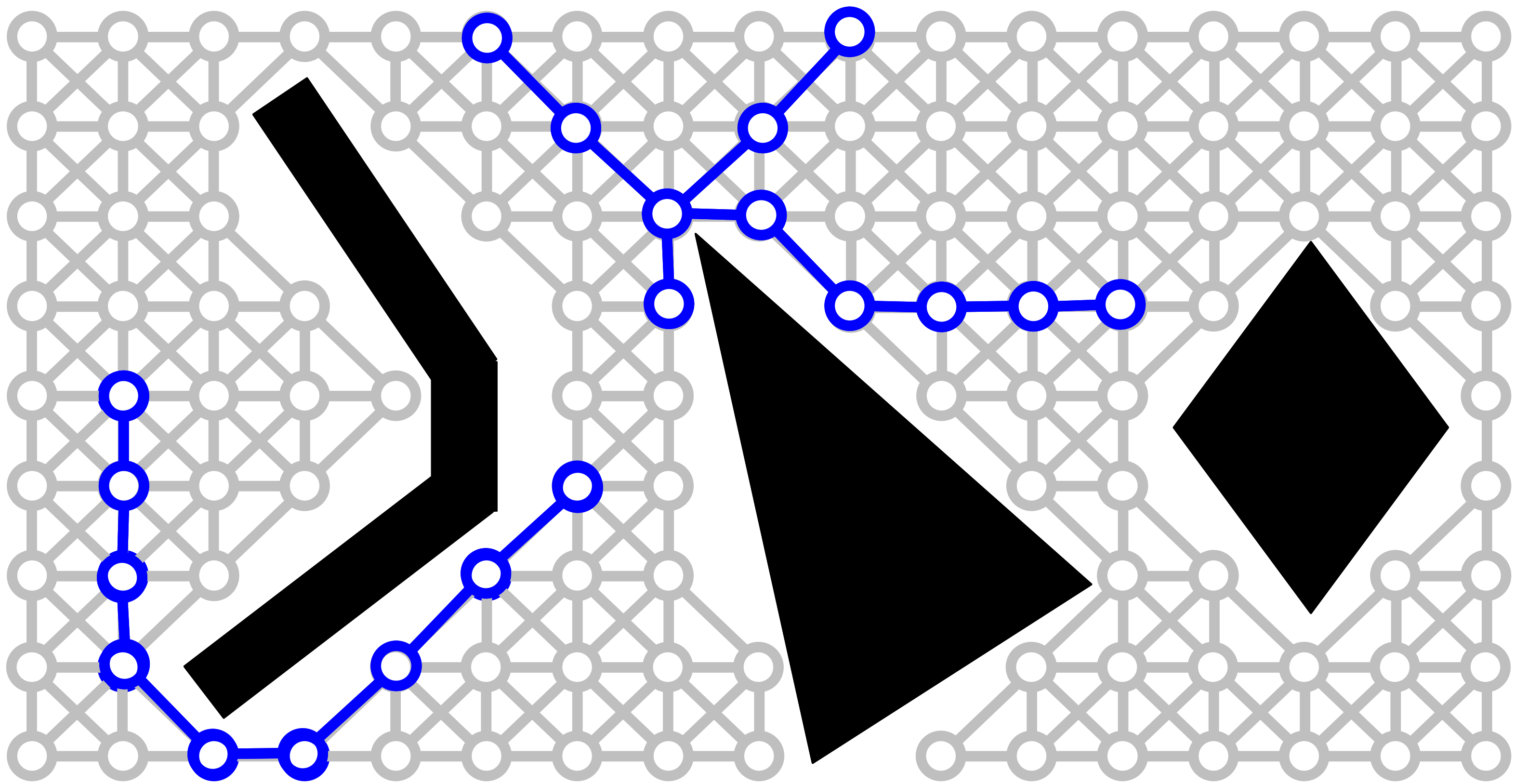
Given a set of previous paths (experiences)…



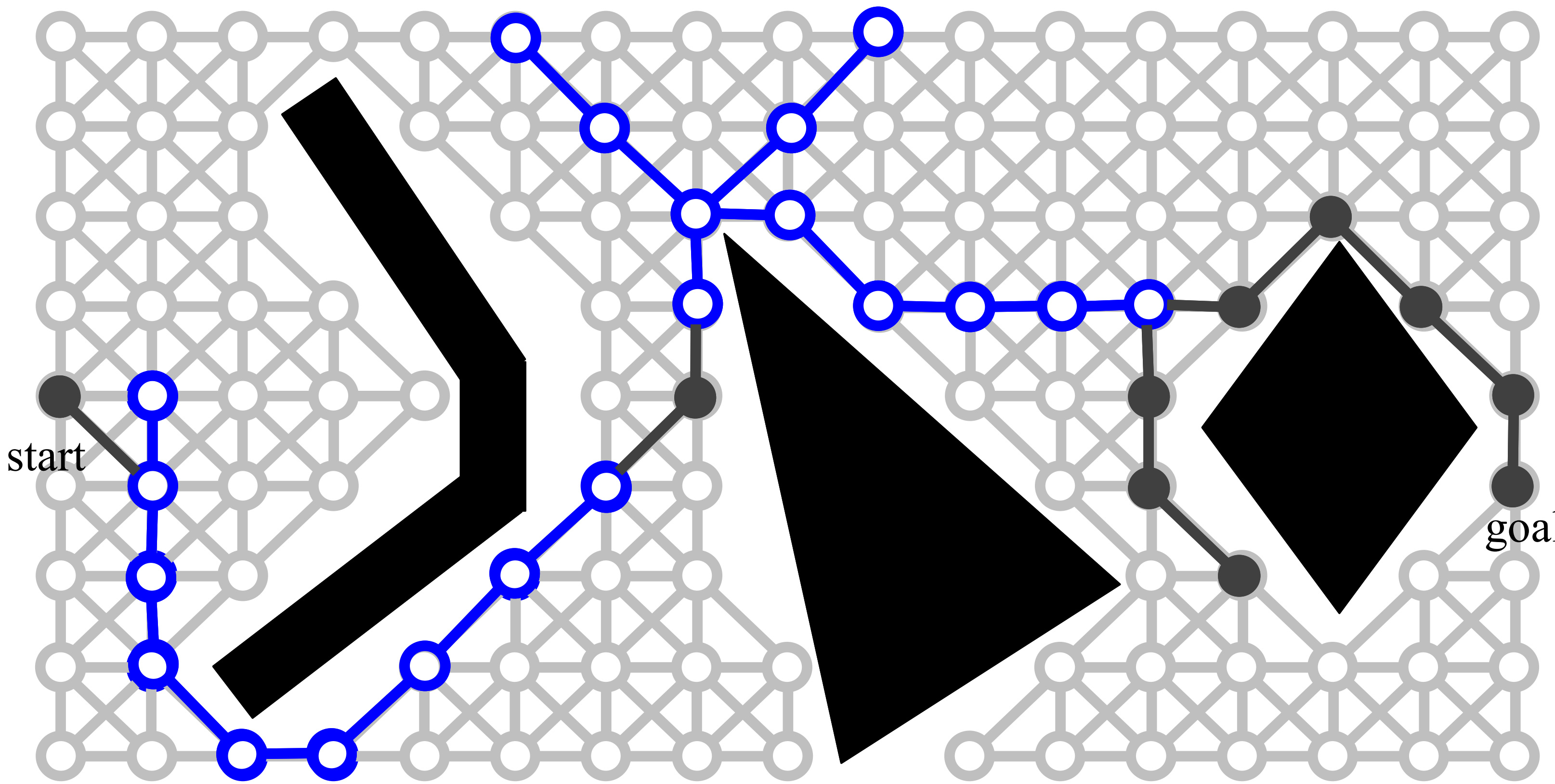
Put them together into an *E*-graph (Experience graph)



Given a new planning query…



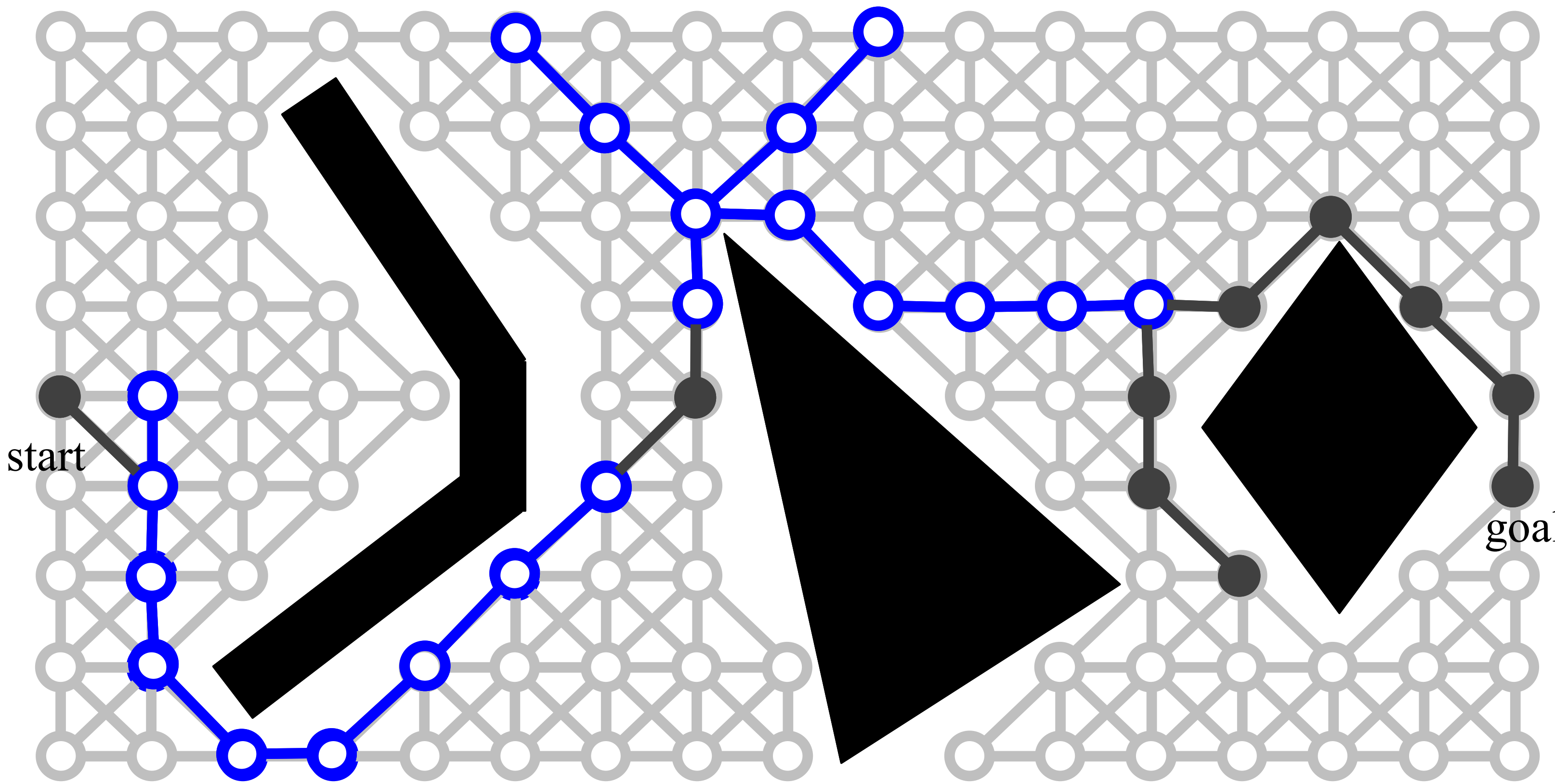
…would like to re-use E-graph to speed up planning in similar situations



…would like to re-use E-graph to speed up planning in similar situations

Re-use is via focusing search with a recomputed *hᵋ()* heuristic function:

$$h^{\mathcal{E}}(s_0) = \min_{\pi} \sum_{i=0}^{N-1} \min\{\varepsilon^{\mathcal{E}} h^G(s_i, s_{i+1}), c^{\mathcal{E}}(s_i, s_{i+1})\}$$



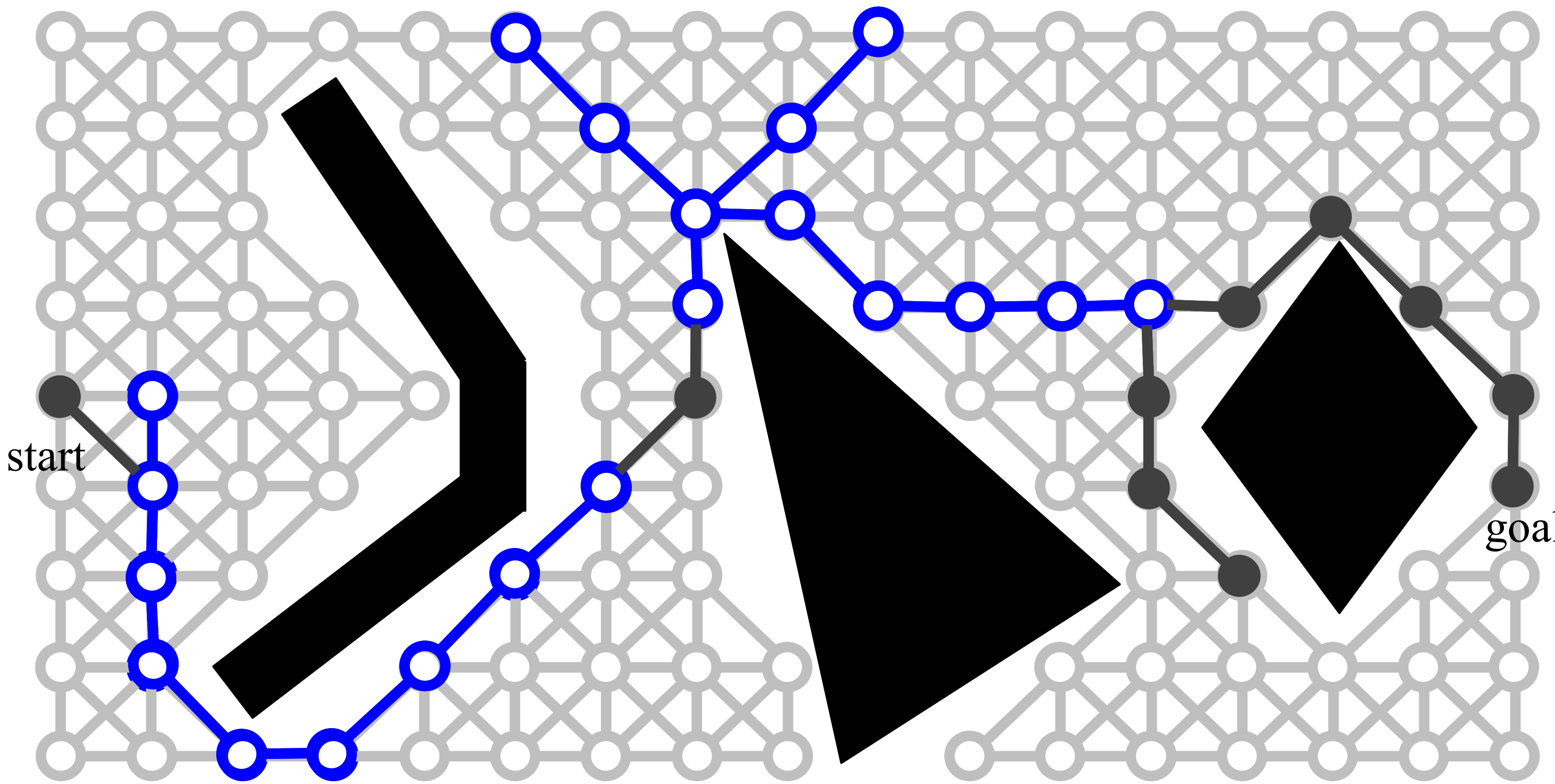
Experience Graphs [Phillips et al., RSS'12]

General idea:

Instead of biasing the search towards the goal, heuristics $h^{\varepsilon}(s)$ biases it towards a set of paths in Experience Graph

Re-use is via … function:

$$h^{\mathcal{E}}(s_0) = \min_{\pi} \sum_{i=0}^{N-1} \min\{\varepsilon^{\mathcal{E}} h^{G}(s_i, s_{i+1}), c^{\mathcal{E}}(s_i, s_{i+1})\}$$

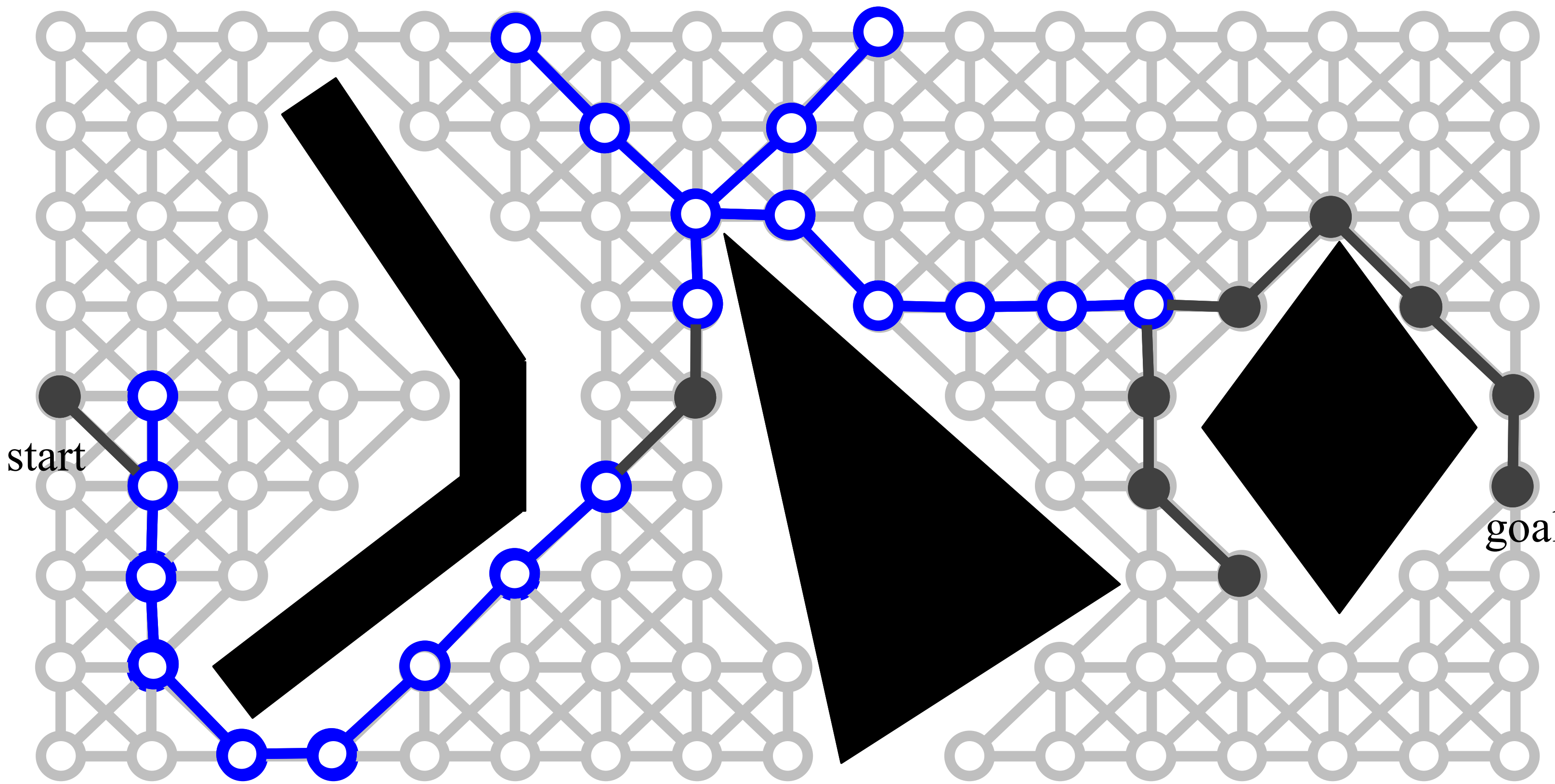


…would li… …ituations

Can be computed via a single Dijkstra's search on the Experience Graph

Re-use … …unction:

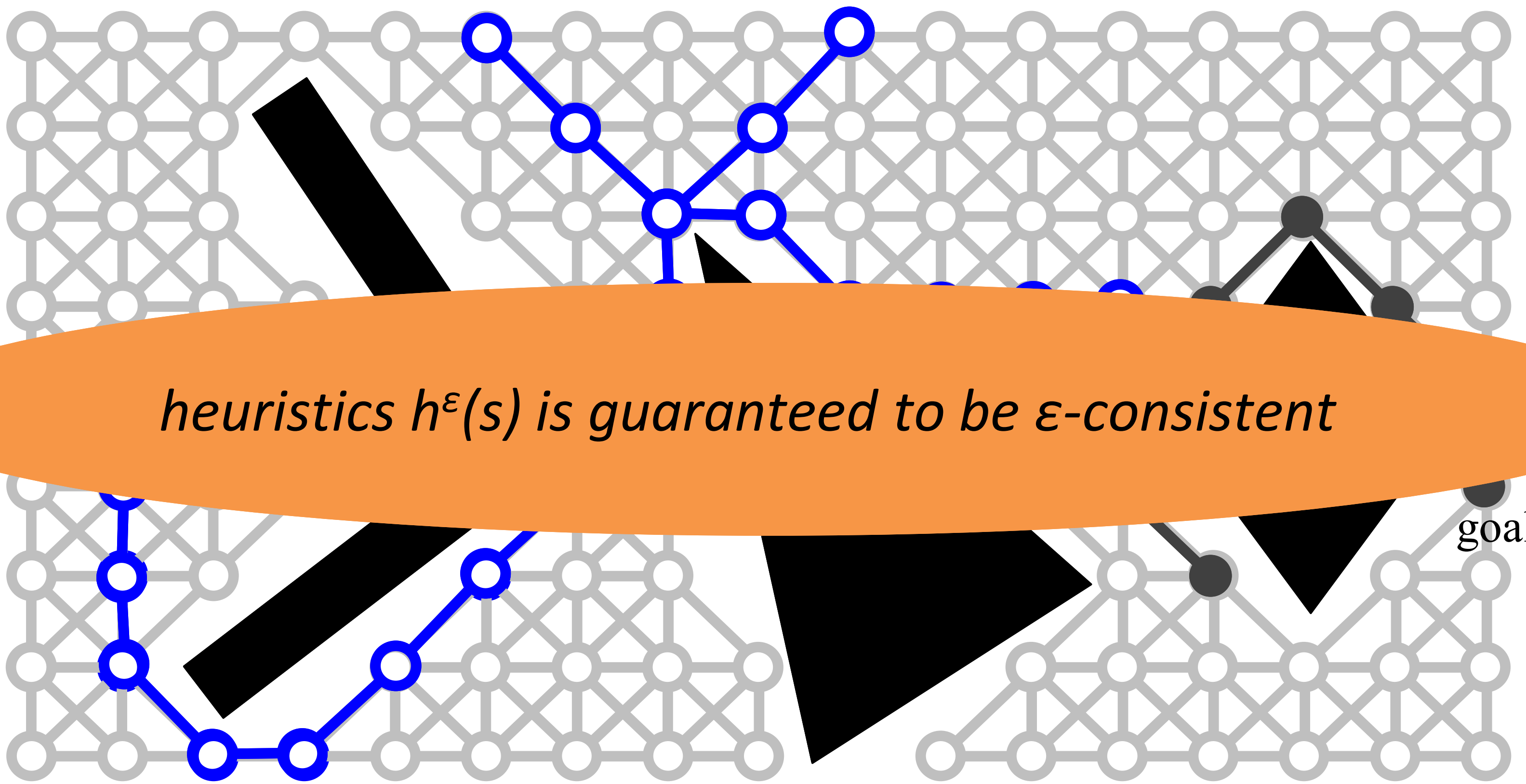
$$h^{\mathcal{E}}(s_0) = \min_{\pi} \sum_{i=0}^{N-1} \min\{\varepsilon^{\mathcal{E}} h^{G}(s_i, s_{i+1}), c^{\mathcal{E}}(s_i, s_{i+1})\}$$



…would like to re-use E-graph to speed up planning in similar situations

Re-use is via focusing search with a recomputed *hᵋ()* heuristic function:

$$h^{\mathcal{E}}(s_0) = \min_{\pi} \sum_{i=0}^{N-1} \min\{\varepsilon^{\mathcal{E}} h^G(s_i, s_{i+1}), c^{\mathcal{E}}(s_i, s_{i+1})\}$$



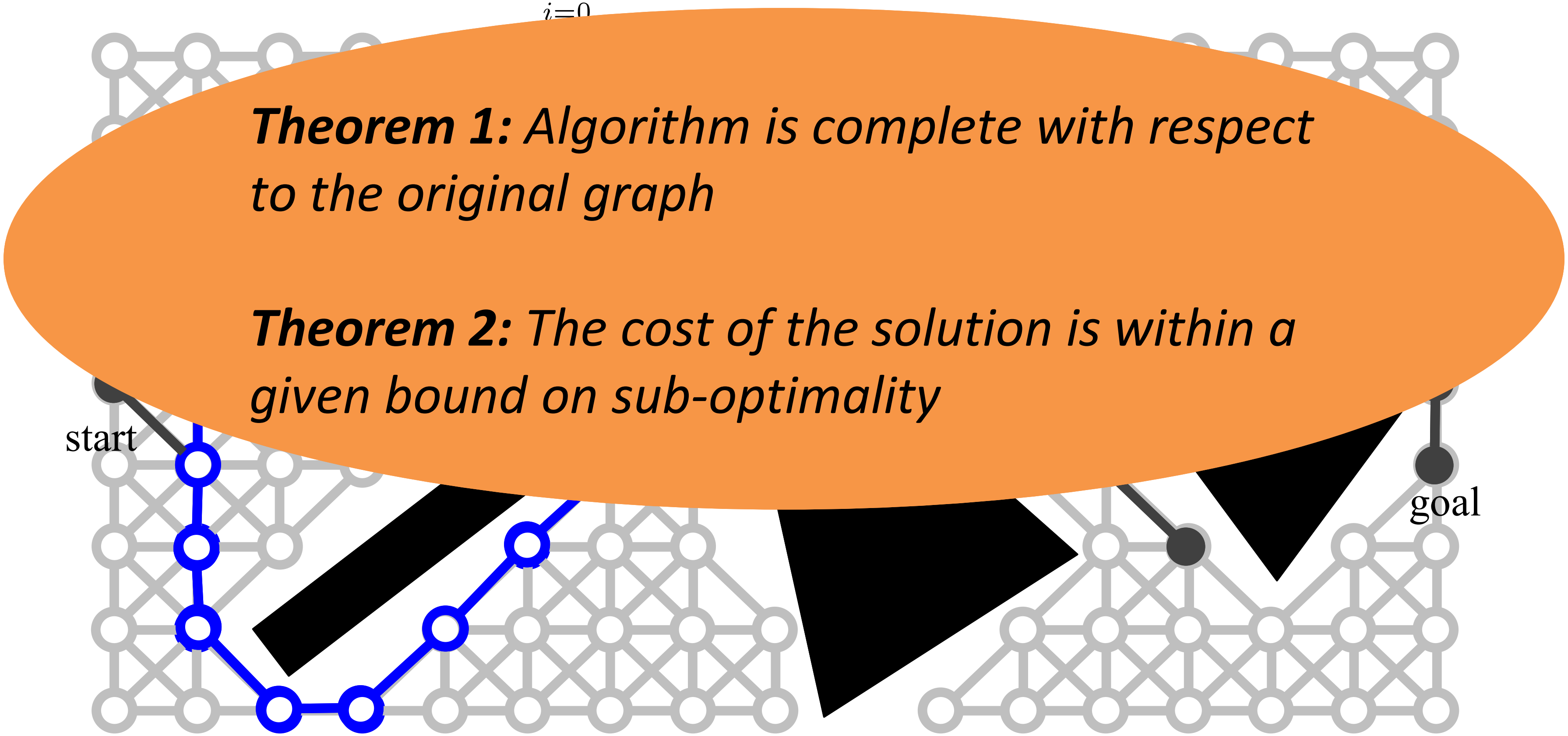
heuristics $h^{\varepsilon}(s)$ is guaranteed to be ε-consistent

Experience Graphs [Phillips et al., RSS'12]

…would like to re-use E-graph to speed up planning in similar situations

Re-use is via focusing search with a recomputed *h^ε()* heuristic function:

$$h^{\mathcal{E}}(s_0) = \min_{\pi} \sum_{i=0}^{N-1} \min\{\varepsilon^{\mathcal{E}} h^G(s_i, s_{i+1}), c^{\mathcal{E}}(s_i, s_{i+1})\}$$

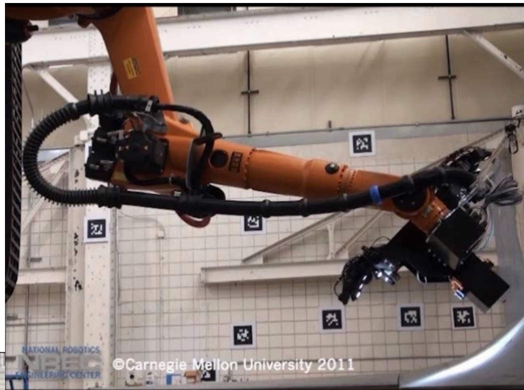


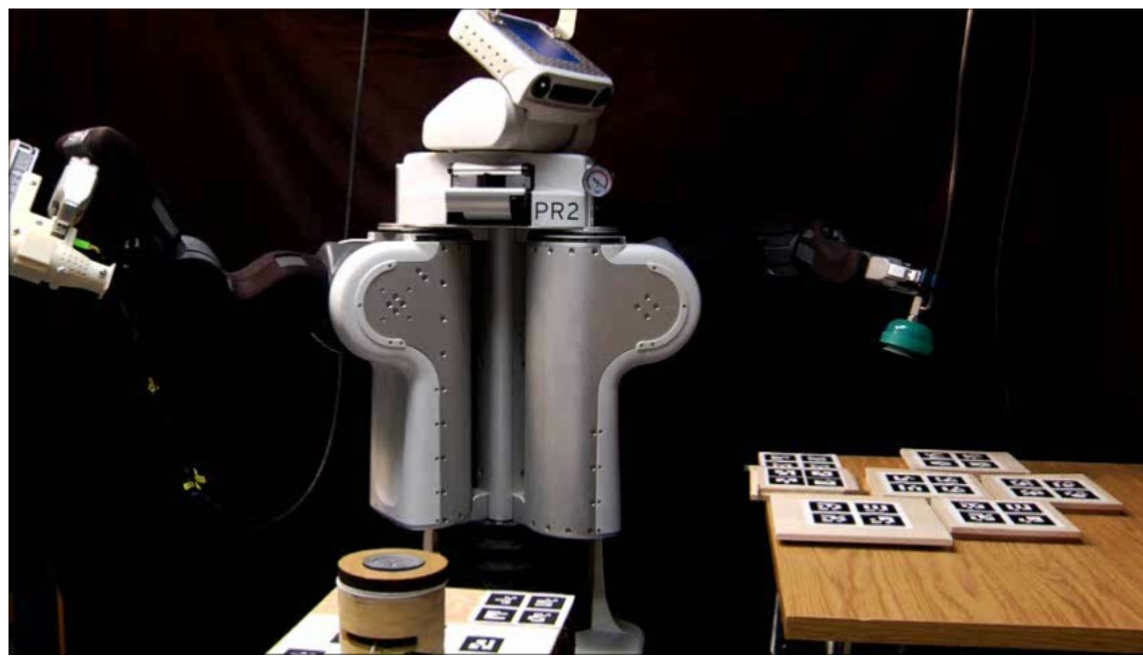
Theorem 1: *Algorithm is complete with respect to the original graph*

Theorem 2: *The cost of the solution is within a given bound on sub-optimality*

Application of Experience Graphs

- Learning to plan faster from experience and demonstrations



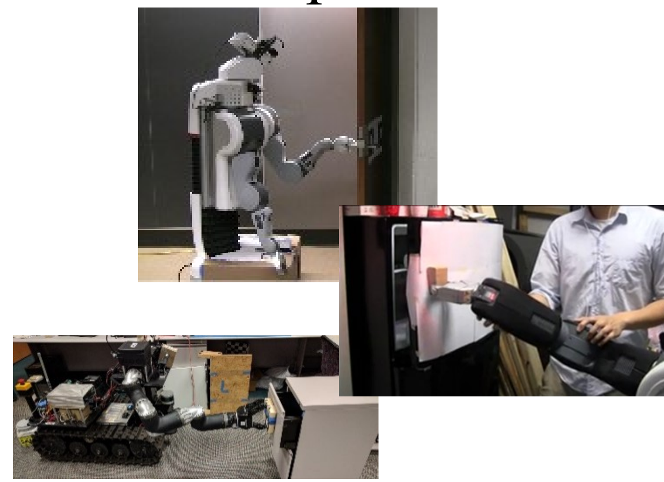


Learning in Search-based Planning

Speeding up planning

Learning cost function

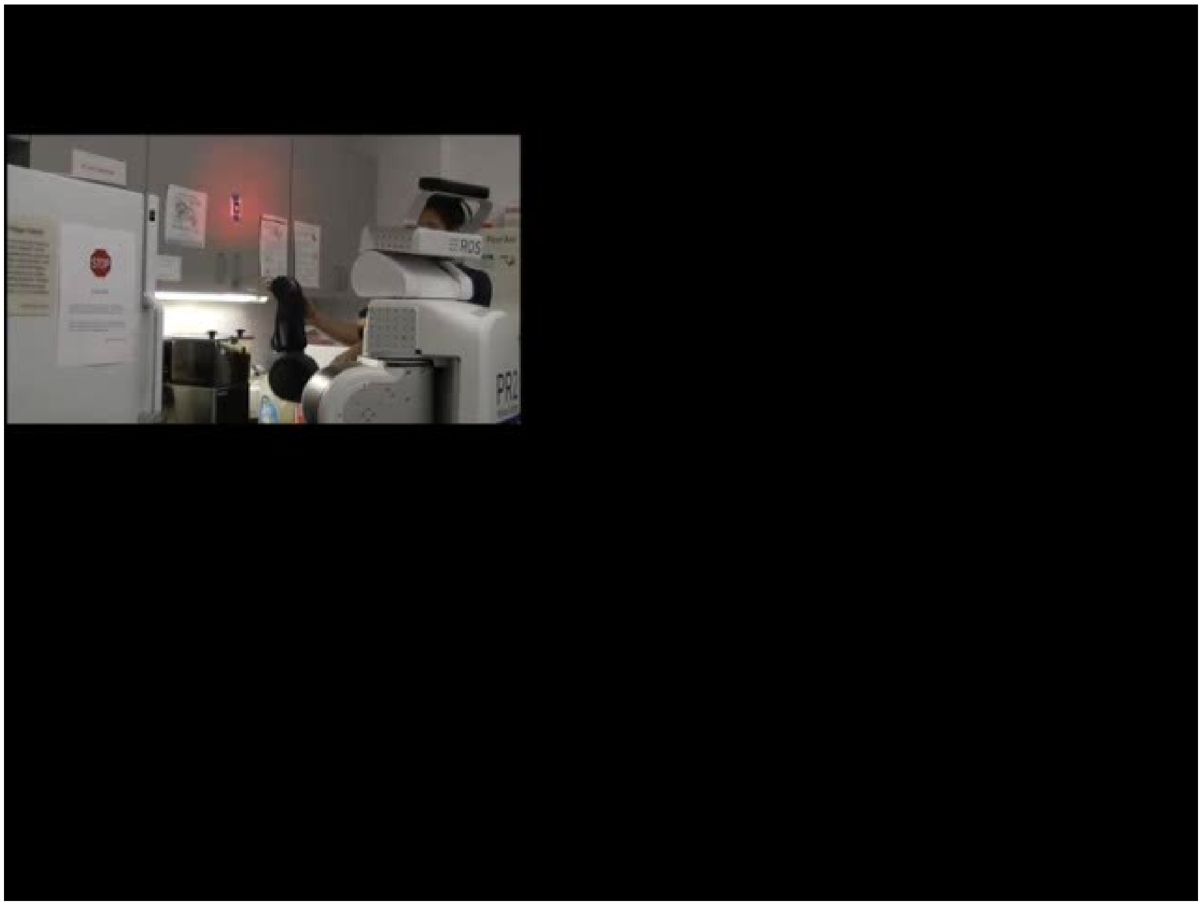
Going beyond the prior model

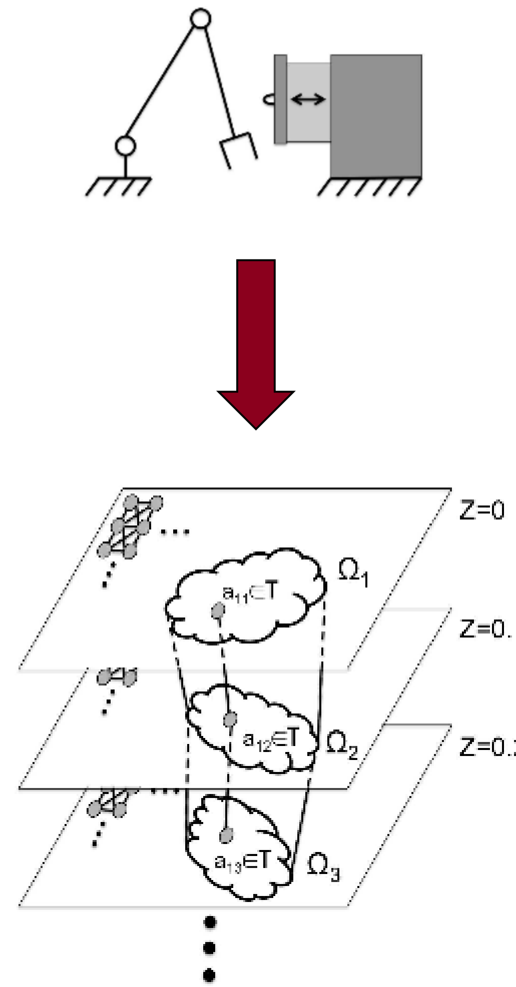


Learning additional dimensions to reason over (Phillips et al.,'13)
Combining learned skills and prior model (Vasudev et al., ongoing)

Learning Additional Dimensions

- Learning Additional Dimensions in the Graph from Demonstrations [Phillips et al., RSS’13]

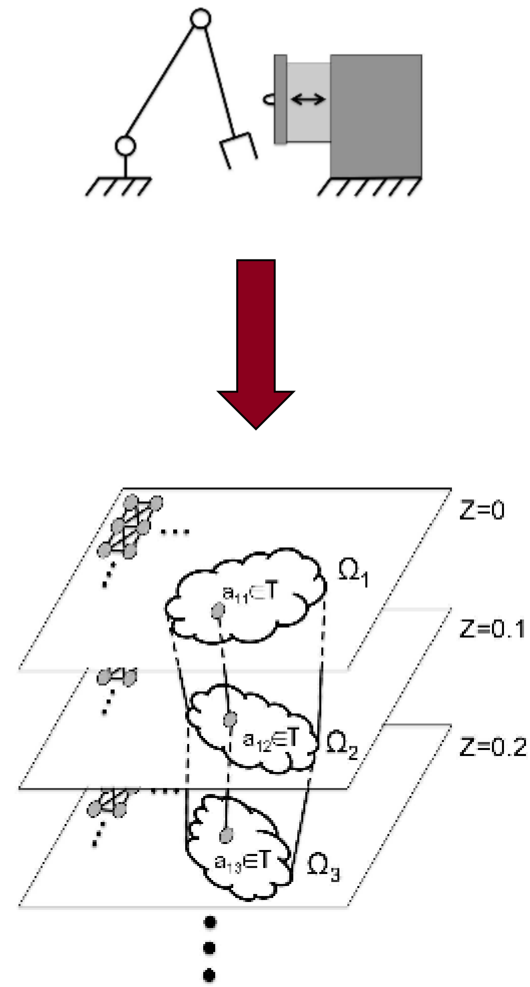




Learning Additional Dimensions

- Learning Additional Dimensions in the Graph from Demonstrations [Phillips et al., RSS'13]

Demonstrations provided in simulation; work by A. Dornbush

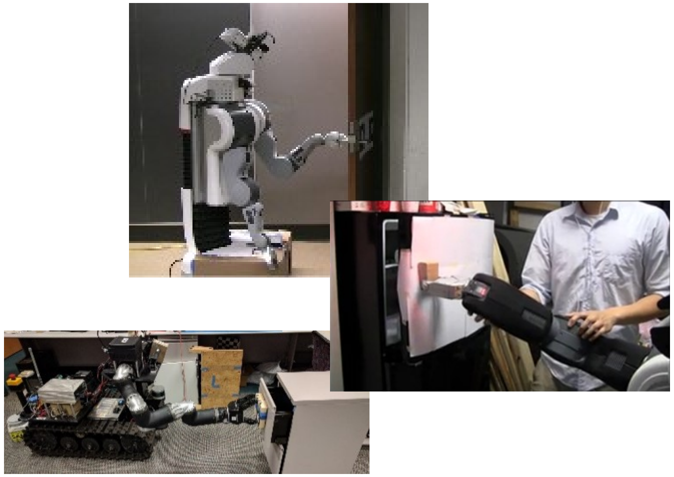


Learning in Search-based Planning

Speeding up planning

Learning cost function

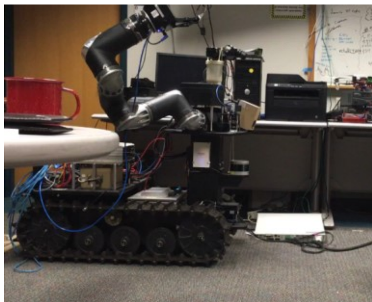
Going beyond the prior model

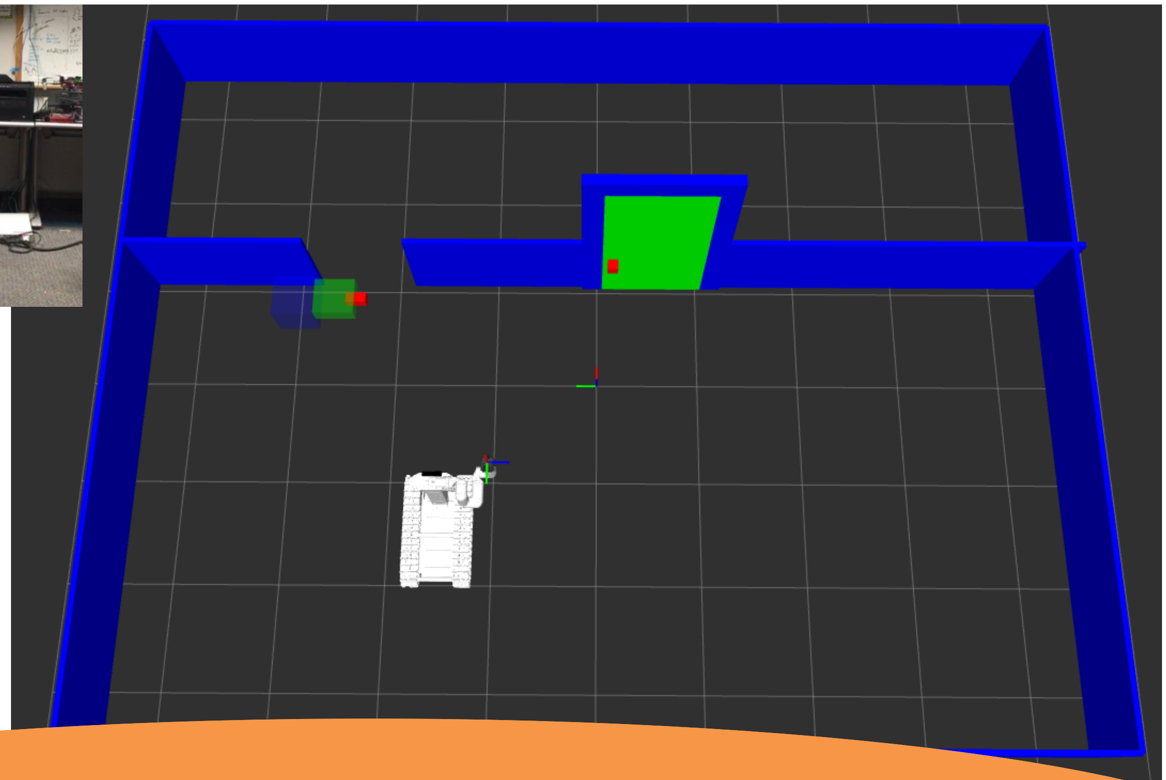


Learning additional dimensions to reason over (Phillips et al.,’13)

Combining learned skills and prior model (Vasudev et al., ongoing)

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i\ldots k}$ that include skills for pushing/pulling doors/drawer

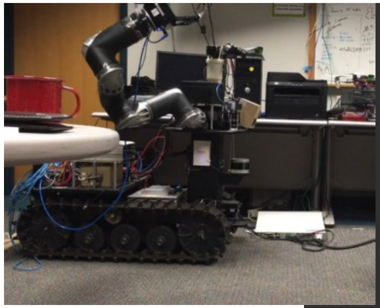


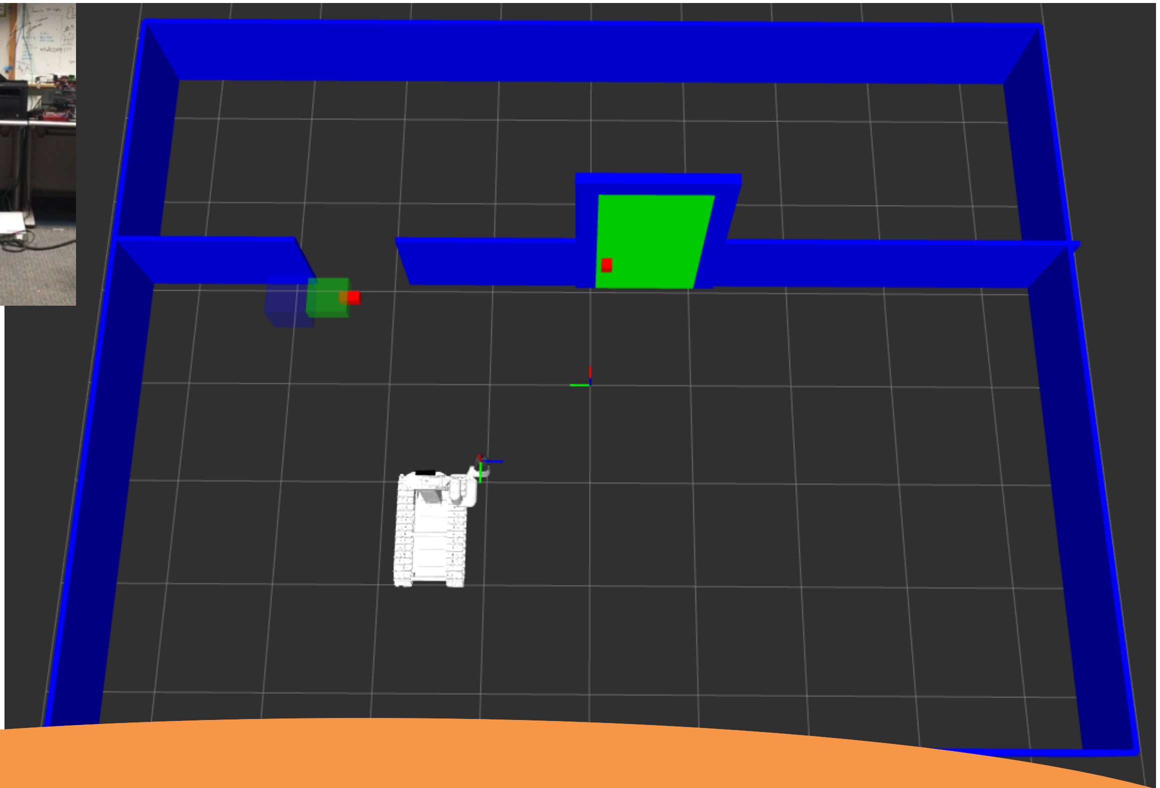


How skills $\psi^{i\ldots k}$ should be integrated with G, so that a planner can generate an overall plan?

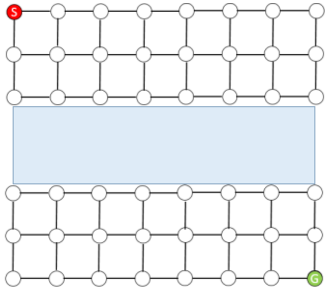
Integrating Learned Skills

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i \ldots k}$ that include skills for pushing/pulling doors/drawer





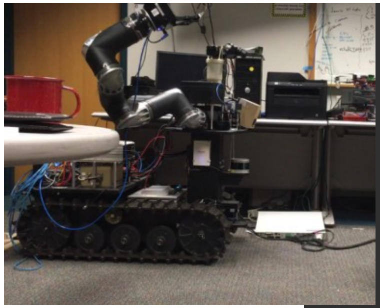
skills connect disconnected components in G

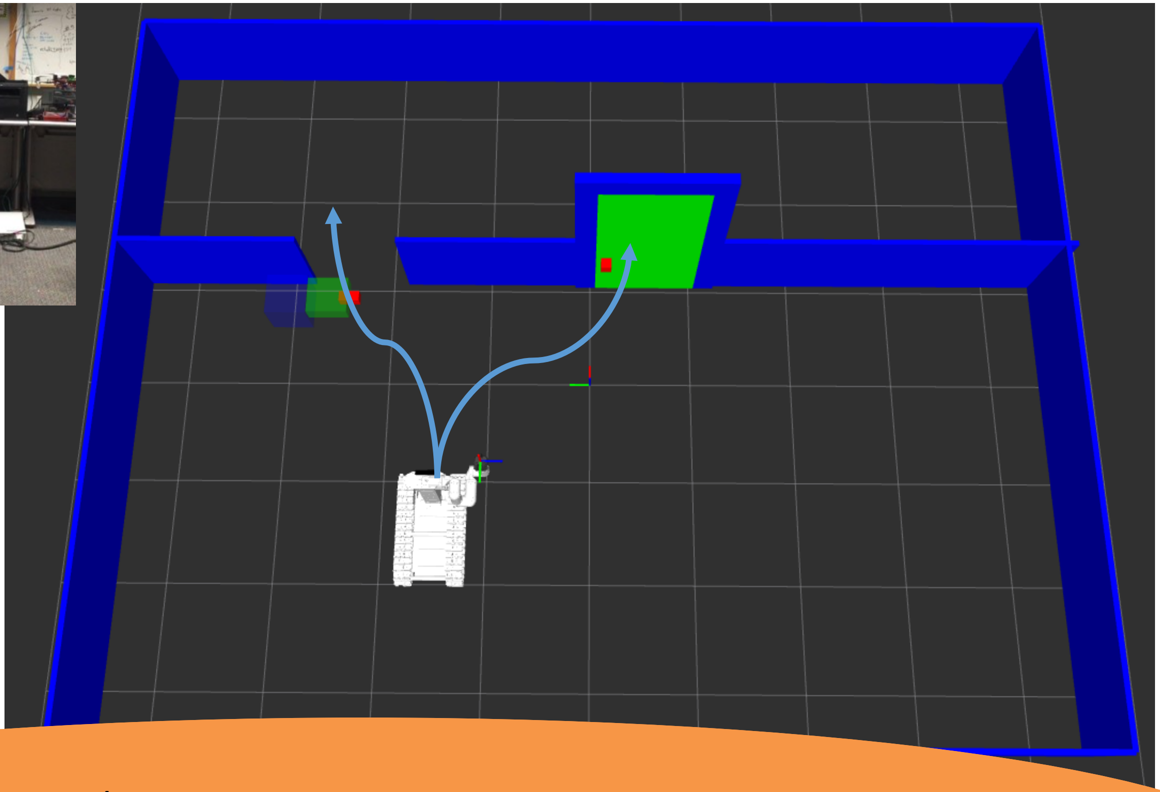


How skills $\psi^{i \ldots k}$ should be integrated with G, so that a planner can generate an overall plan?

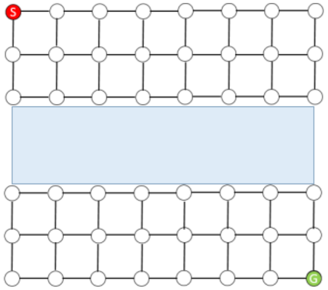
Integrating Learned Skills

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i \dots k}$ that include skills for pushing/pulling doors/drawer





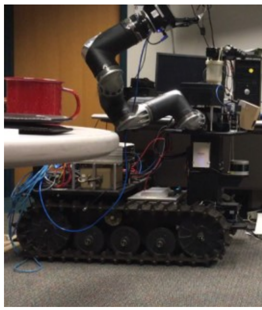
skills connect disconnected components in G



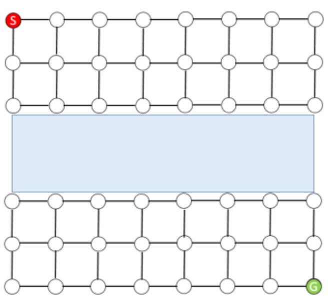
How skills $\psi^{i \dots k}$ should be integrated with G, so that a planner can generate an overall plan?

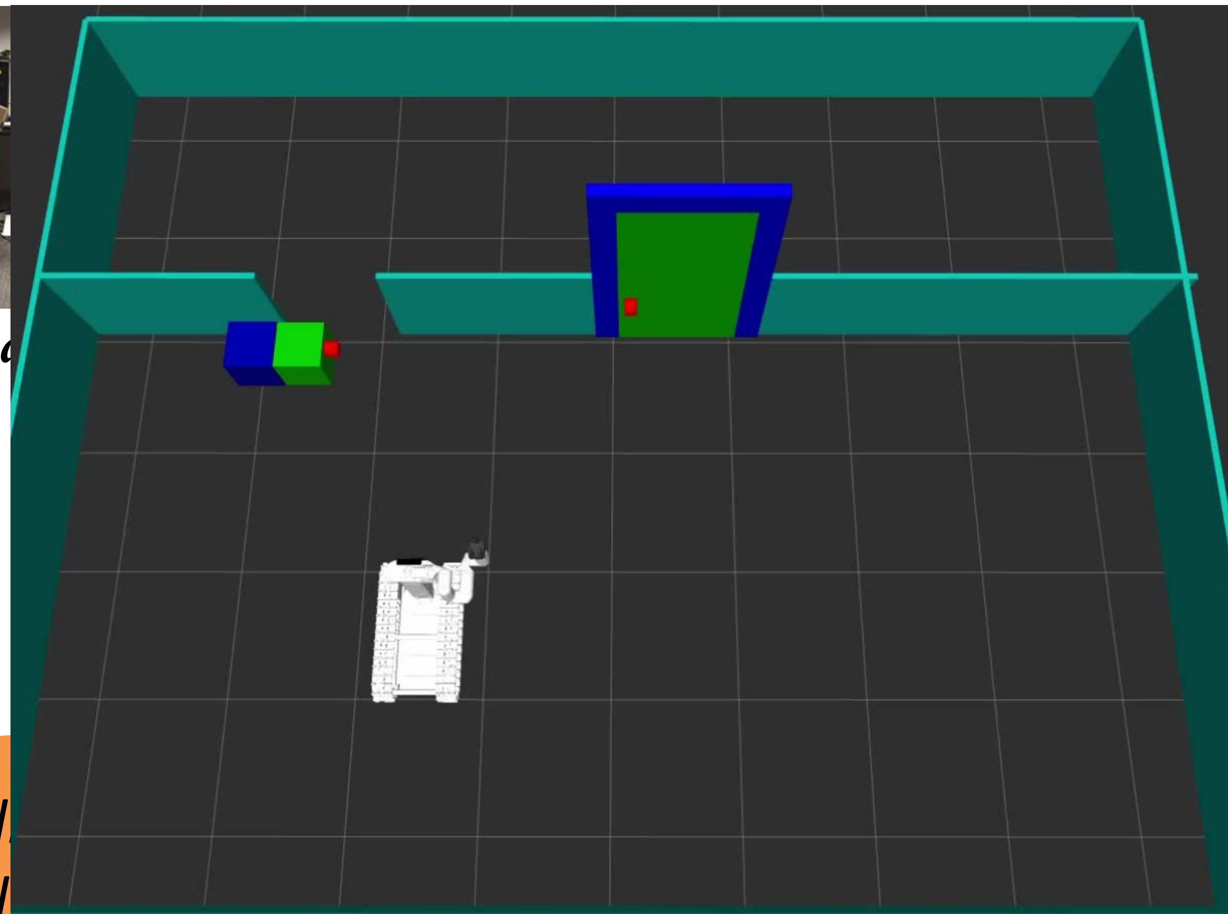
Integrating Learned Skills

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i \ldots k}$ that include skills for pushing/pulling doors/drawer



skills connect disconnected components in G





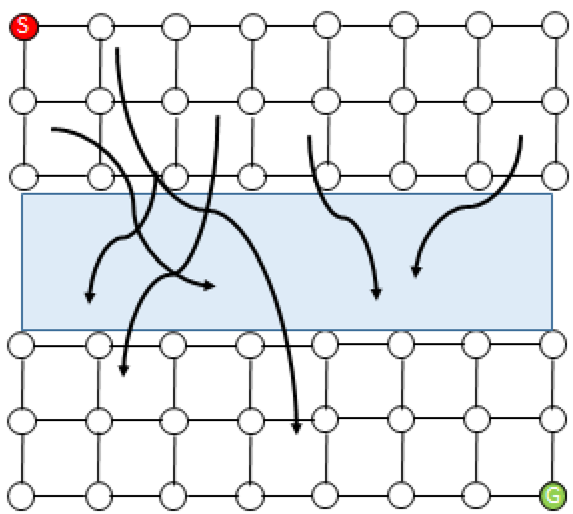
How skil...
that a pl...

Integrating Learned Skills

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i \ldots k}$ that include skills for pushing/pulling doors/drawer

We assume ψ^i: $X \rightarrow \{a, X'\}$,
and each X maps onto unique S

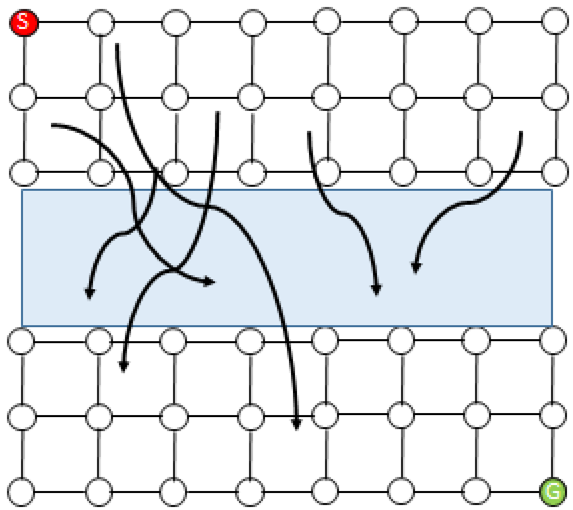
A skill could potentially be available at each state S,
but depending on data, at some states there is higher
confidence in its success than at others



Integrating Learned Skills

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i \ldots k}$ that include skills for pushing/pulling doors/drawer

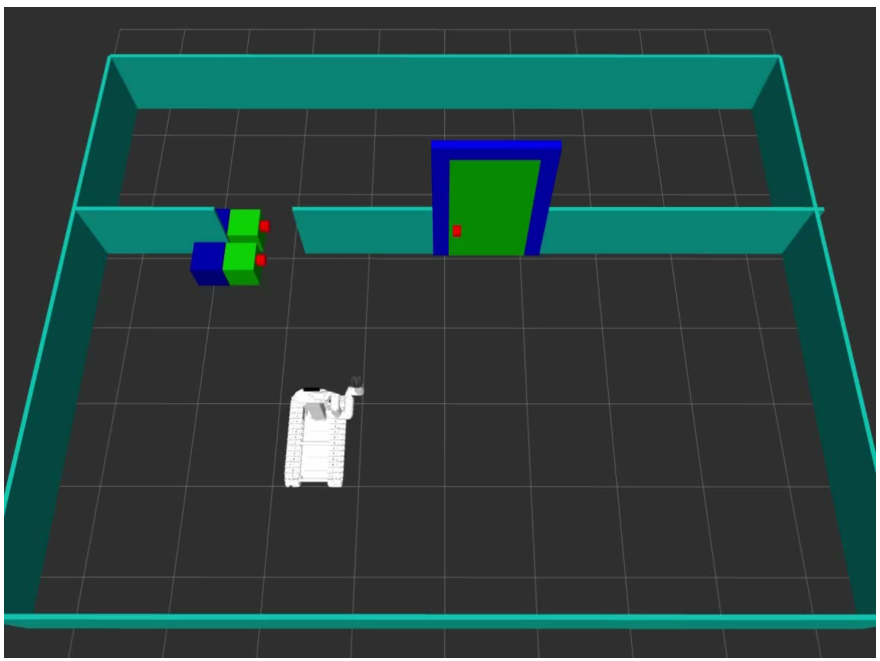
- If confidence is estimated (e.g., via Dropouts [Gal & Ghahramani]), then:
  - Option 1: cost(s,a',s') is inflated proportionally to the estimated confidence



Integrating Learned Skills

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i \ldots k}$ that include skills for pushing/pulling doors/drawer

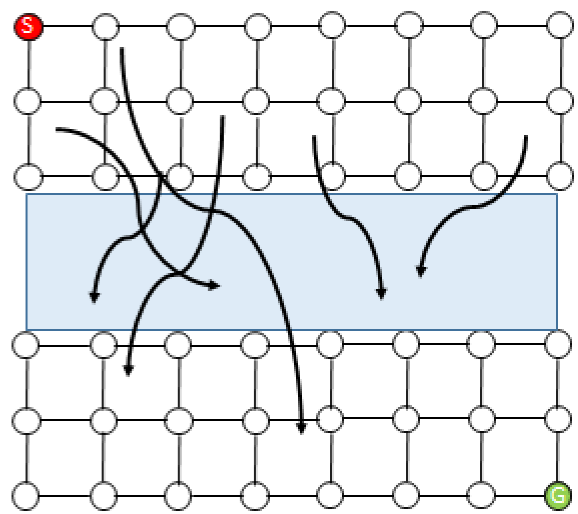
- If confidence is estimated (e.g., via Dropouts [Gal & Ghahramani]), then:
  - Option 1: cost(s,a',s') is inflated proportionally to the estimated confidence



Integrating Learned Skills

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i \ldots k}$ that include skills for pushing/pulling doors/drawer

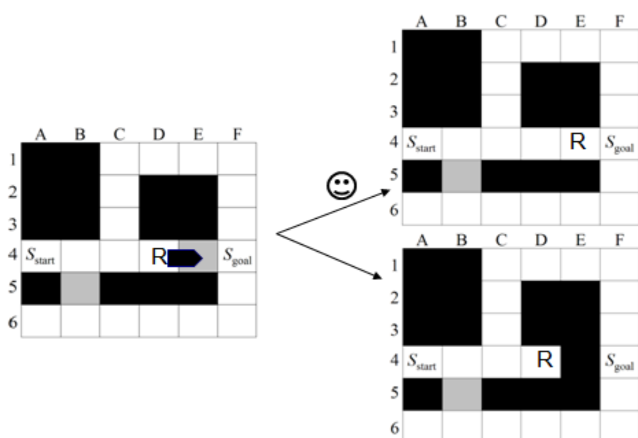
- If confidence is estimated (e.g., via Dropouts [Gal & Ghahramani]), then:
  - Option 1: cost(s,a',s') is inflated proportionally to the estimated confidence
  - **Option 2: represent the planning problem as POMDP ☹**



Integrating Learned Skills

- Suppose:
  - We have a graph $G = \{S, E\}$ that describes how the robot can move its base/arms
  - We have a set of k skills $\psi^{i \ldots k}$ that include skills for pushing/pulling doors/drawer

- If confidence is estimated (e.g., via Dropouts [Gal & Ghahramani]), then:
  - Option 1: cost(s,a',s') is inflated proportionally to the estimated confidence
  - **Option 2: represent the planning problem as POMDP ☹**
    - planning is exponential in *(S, ψ^i)* pairs
    - however, there exists a **clear preference** on the outcomes: *it is always preferred for a skill to be successful at a given S*



Formally, clear preference:

$arg\ min_{x'}(c(x,a,x') + v^*(x'))$,

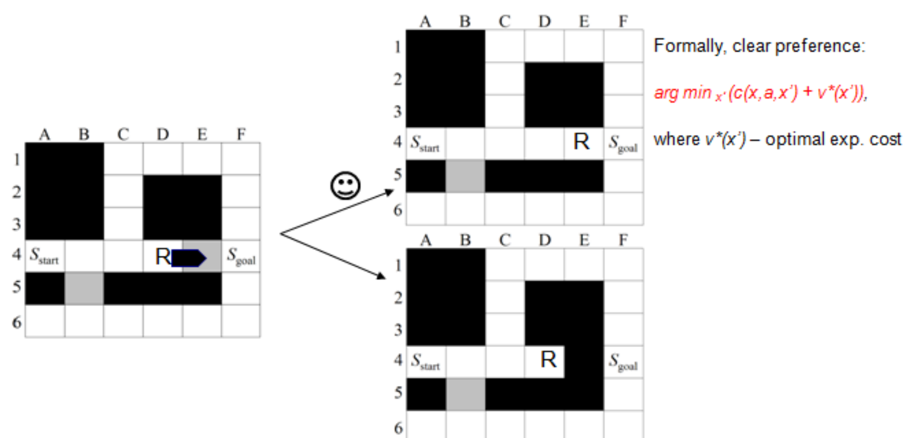
where $v^*(x')$ – optimal exp. cost

Integrating Learned Skills

- Suppos

Planning problem can be decomposed into a series of graph searches using PPCP (Likhachev & Stentz,'09):
- *avoids planning in a belief state-space*
- *scales to large-scale problems in real-time*
- *provides rigorous theoretical guarantees*

- If co]), then:
  - Option 1: cost(s,a ,s) ... estimated confidence
  - **Option 2: represent the planning problem as POMDP ☹**
    - planning is exponential in *(S, ψ^i)* pairs
    - however, there exists a **clear preference** on the outcomes: *it is always preferred for a skill to be successful at a given S*



Going Forward

- Explore option 2 (POMDP planning with uncertainty due to skills)
- Relax the assumption that each X maps onto unique S
- Apply the framework to few domains including navigation through crowded areas

Thanks!

- Students & Staff:
  - Ishani Chatterjee
  - Ben Cohen
  - Andrew Dornbush
  - Victor Hwang
  - Venkatraman Narayanan
  - Michael Phillips
  - Kalyan Vasudev
- Funding:
  - ARL
  - ONR
  - Mitsubishi