Integrating Learning and Geometry for Robotics

Christian Wolf
Dezember 15th, 2020

AFIA

INSA LIRIS CNRS

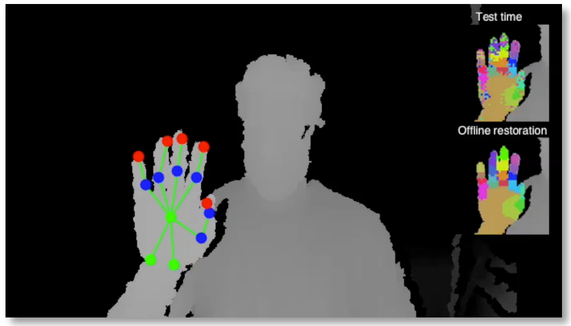
Who am I?

Christian Wolf, Associate Professor, HDR
Chair in Research and Teaching in Artificial Intelligence at INSA-Lyon,
LIRIS UMR CNRS 5205
liris.cnrs.fr/christian.wolf

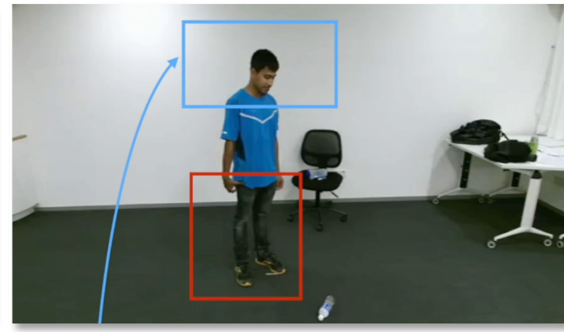
The group in Feb. 2020: Corentin Kervadec, Steeven Janny, Edward Beeching, Fabien Baradel, Théo Jaunet, Quentin Possamaï.

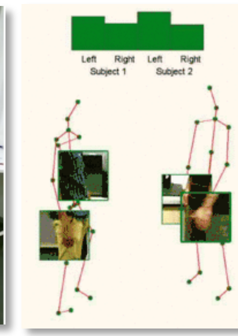
Learning vision & robotics

Gesture recognition

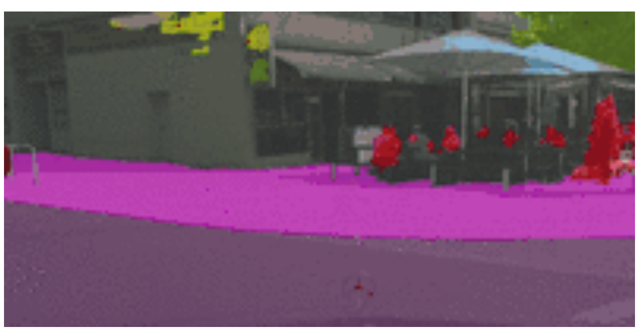


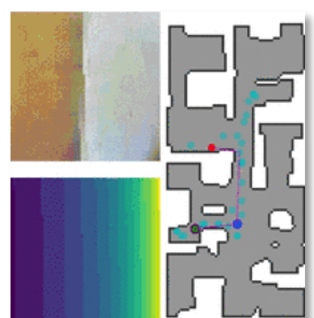
Pose estimation





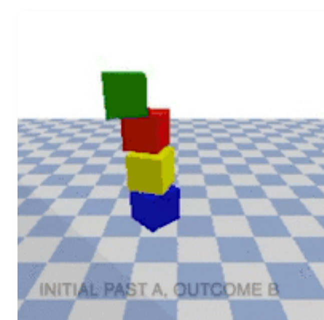
Activity Recognition



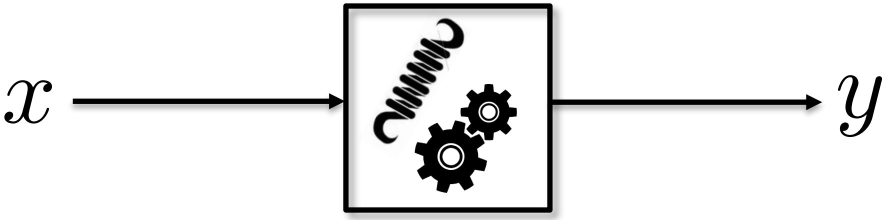


Robot Perception and Navigation

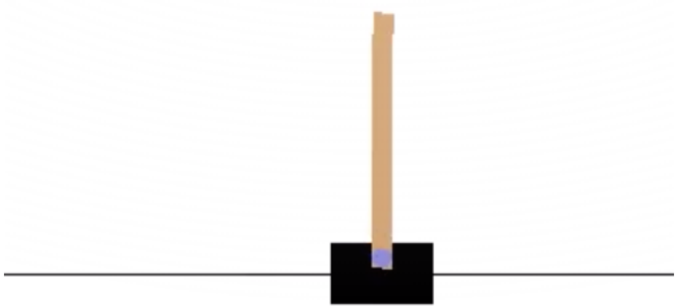
H-C Interaction



Physics

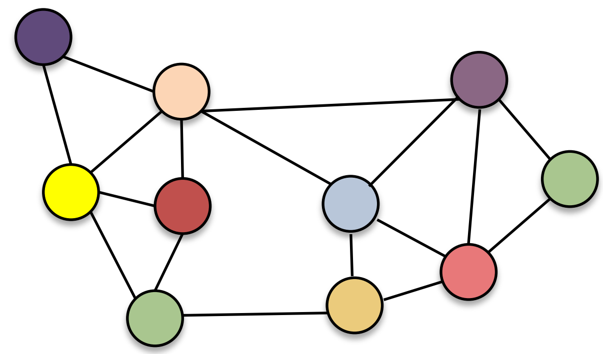


(Physical) model

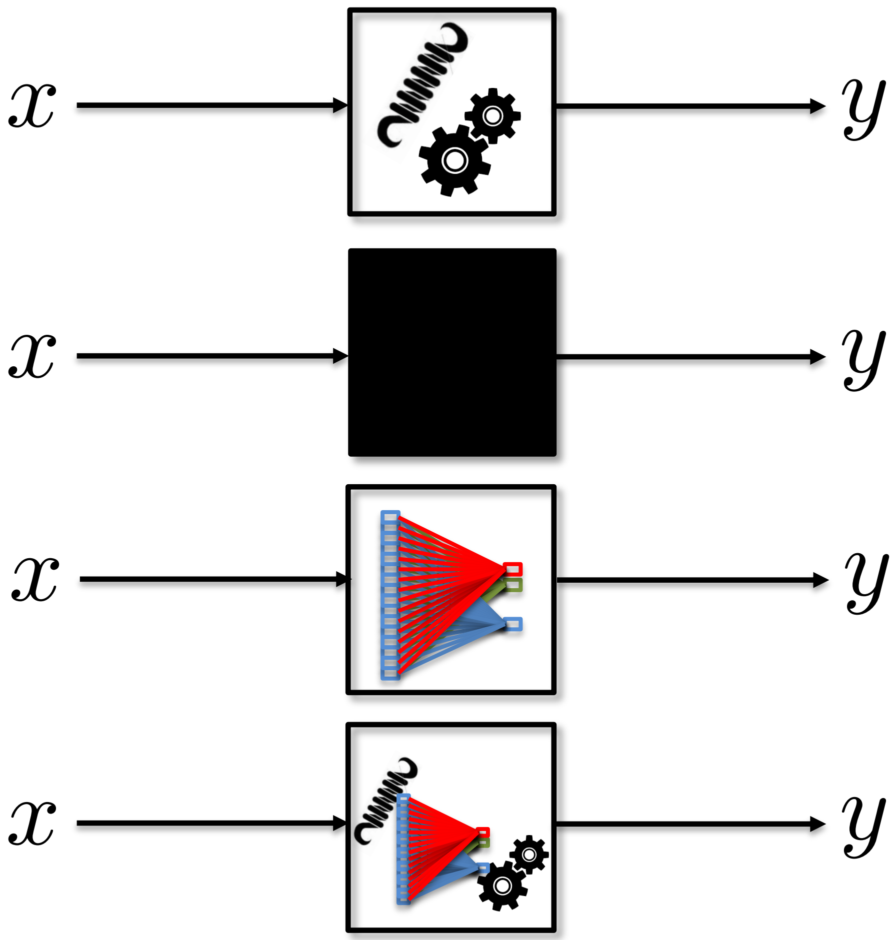


$$\ddot{\theta}_t = \frac{g\sin\theta_t + \cos\theta_t\left[\frac{-F_t - ml\dot{\theta}_t^2\sin\theta_t + \mu_c \operatorname{sgn}(\dot{x}_t)}{m_c+m}\right] - \frac{\mu_p\dot{\theta}_t}{ml}}{l\left[\frac{4}{3} - \frac{m\cos^2\theta_t}{m_c+m}\right]}$$

$$\ddot{x}_t = \frac{F_t + ml\left[\dot{\theta}_t^2\sin\theta_t - \ddot{\theta}_t\cos\theta_t\right] - \mu_c\operatorname{sgn}(\dot{x}_t)}{m_c+m}$$



Planing/shortest path
- **Dijkstra**
- **A***
- **Front Propagation**



(Physical) model

Black box
(unknown function)

White box
(learned but known
complex function)

Hybrid model

Learning to navigate in 3D environments

Office space

Homes

Exposition centers

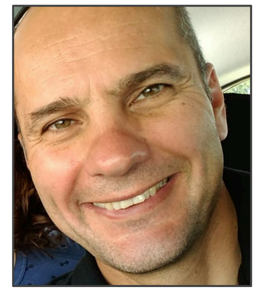
Hospitals

This work

Edward
Beeching

Jilles
Dibangoye

Olivier
Simonin



Christian
Wolf

Inria Chroma

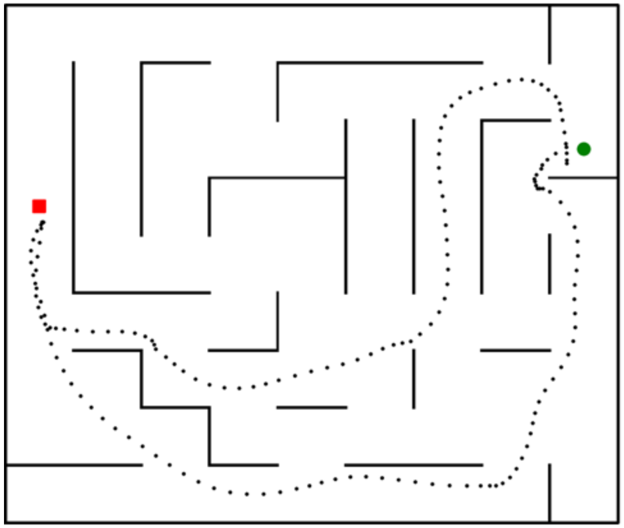
LIRIS

Find my keys!

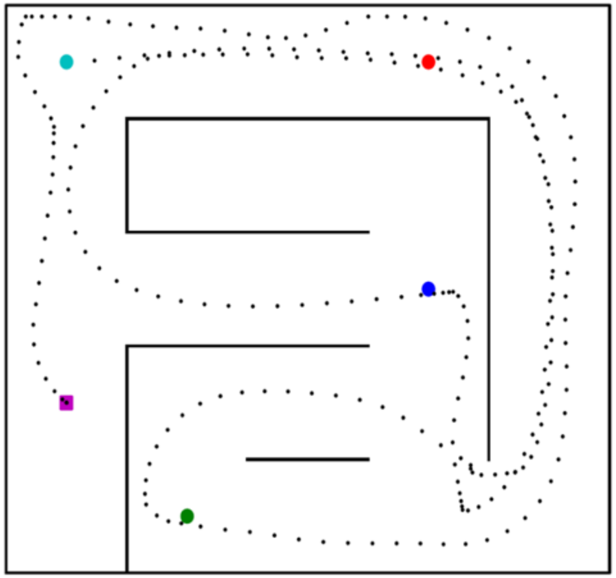
Can we learn a spatial representation which is richer than navigational space?

Can we learn it from reward alone?

Learning high-level reasoning



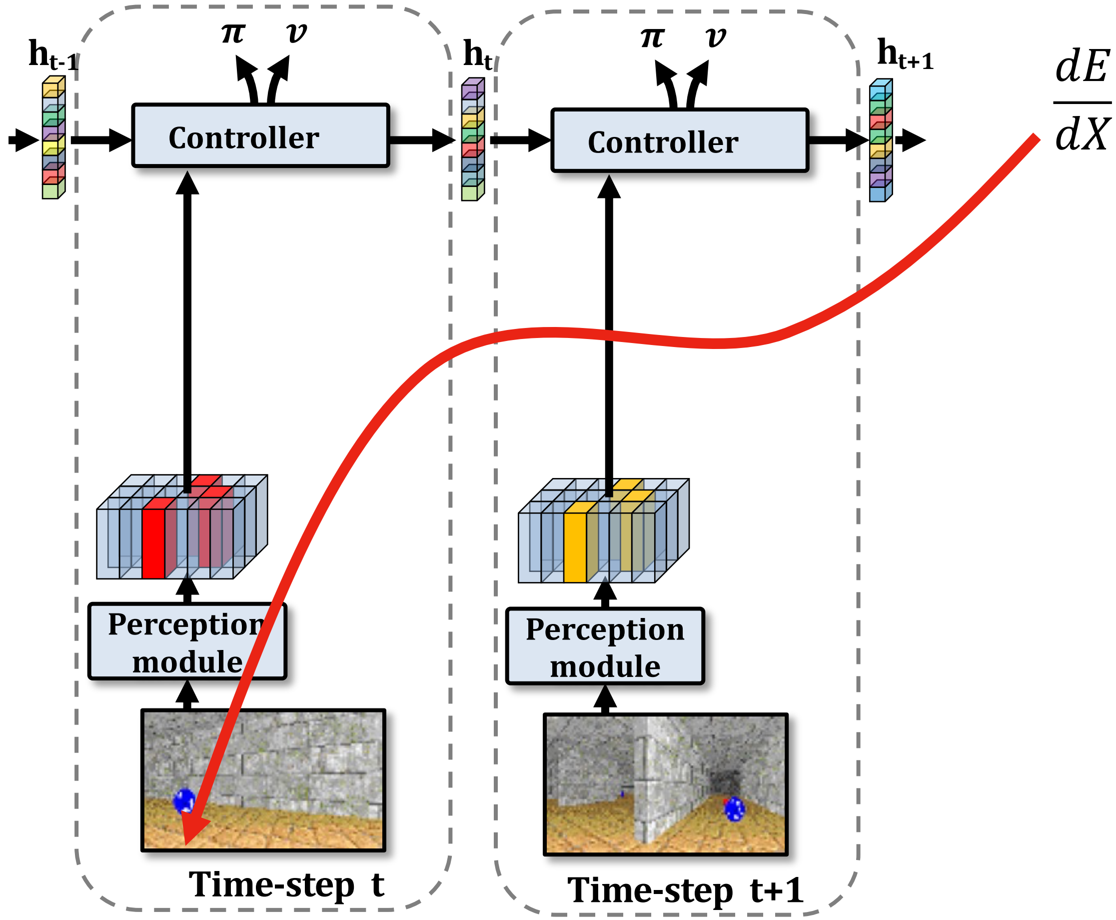
Find and return: find an object in a maze, return to starting point



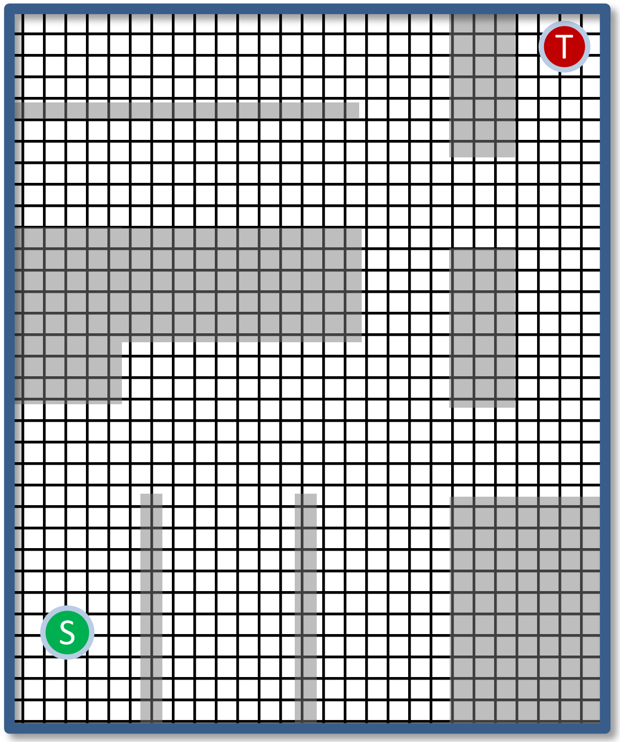
K-item: Collect k items in a fixed order

[Beeching et al., ICPR 2020]

The unstructured recurrent baseline

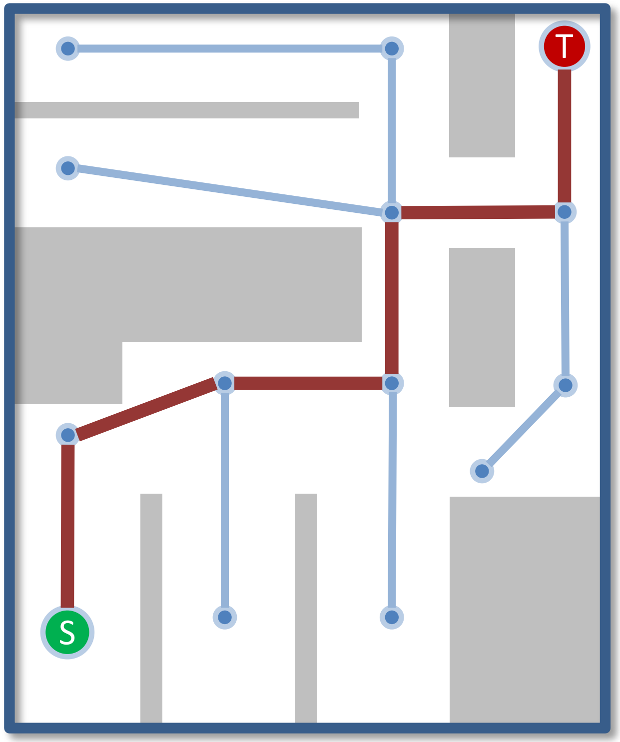


Spatial maps in robotics



Metric map
(=2D or 3D Grid)

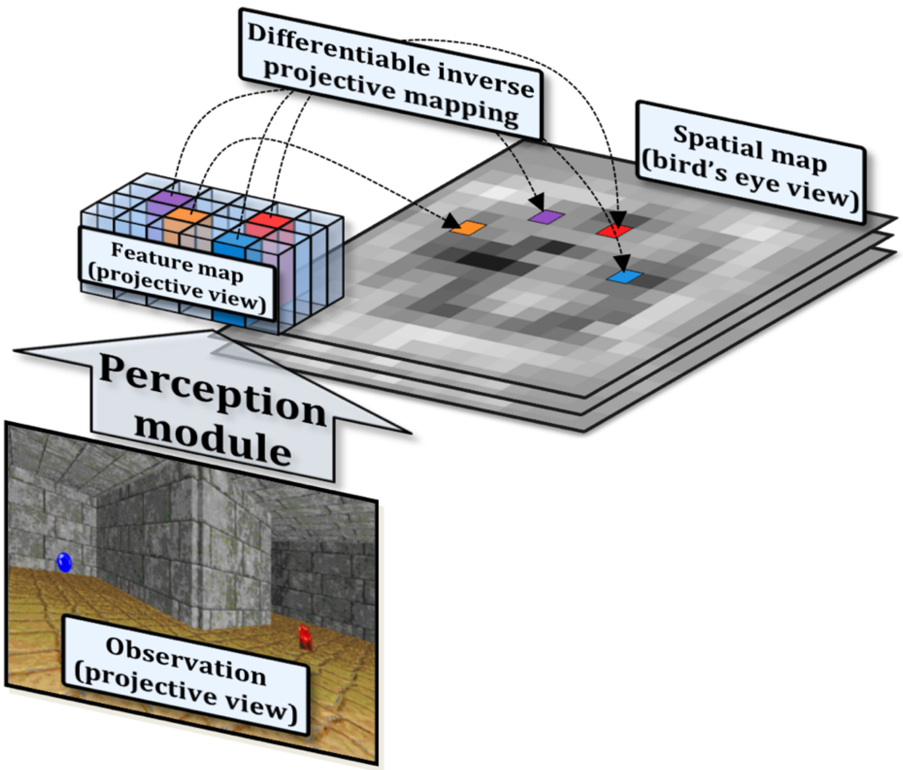
Beeching, Dibangoye, Simonin, Wolf,
EgoMap: Projective mapping and structured egocentric memory for Deep RL,
ECML-PKKD 2020



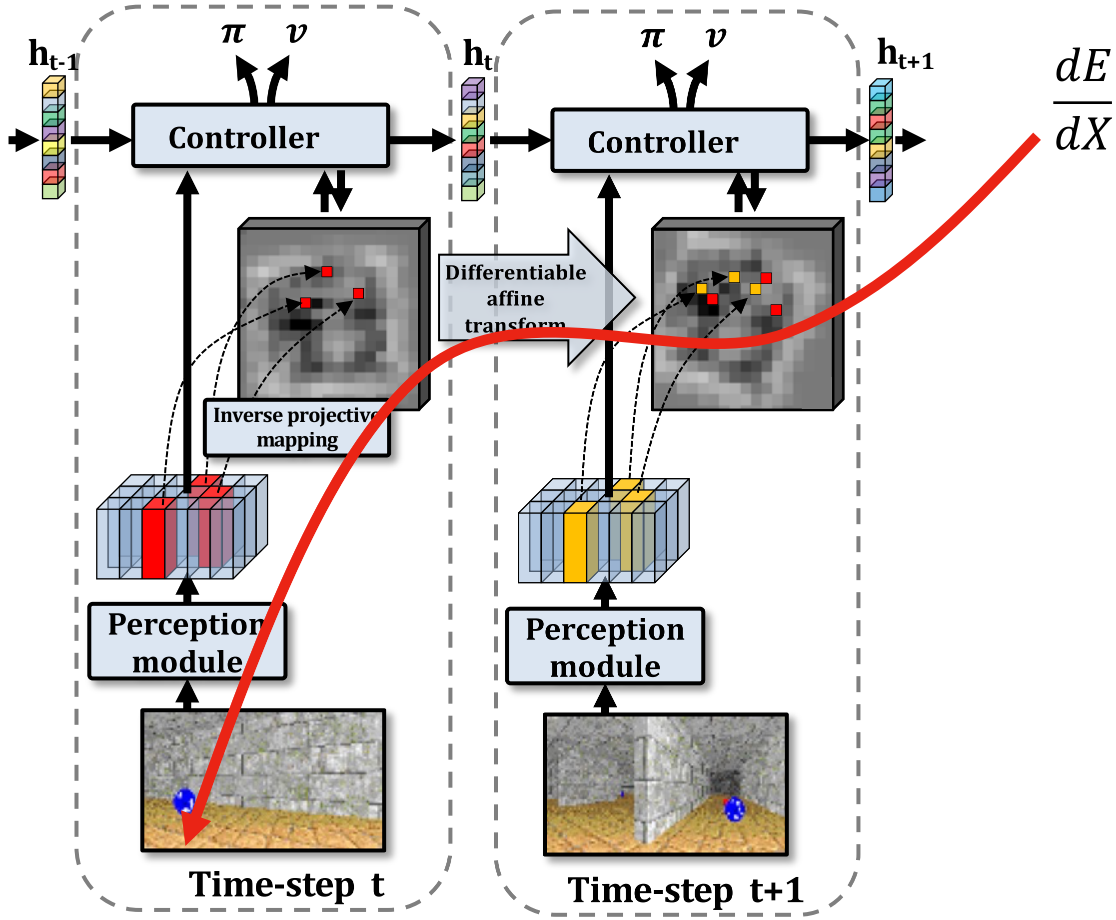
Topological map
(=Graph)

Beeching, Dibangoye, Simonin, Wolf,
Learning to plan with uncertain topological maps,
ECCV 2020

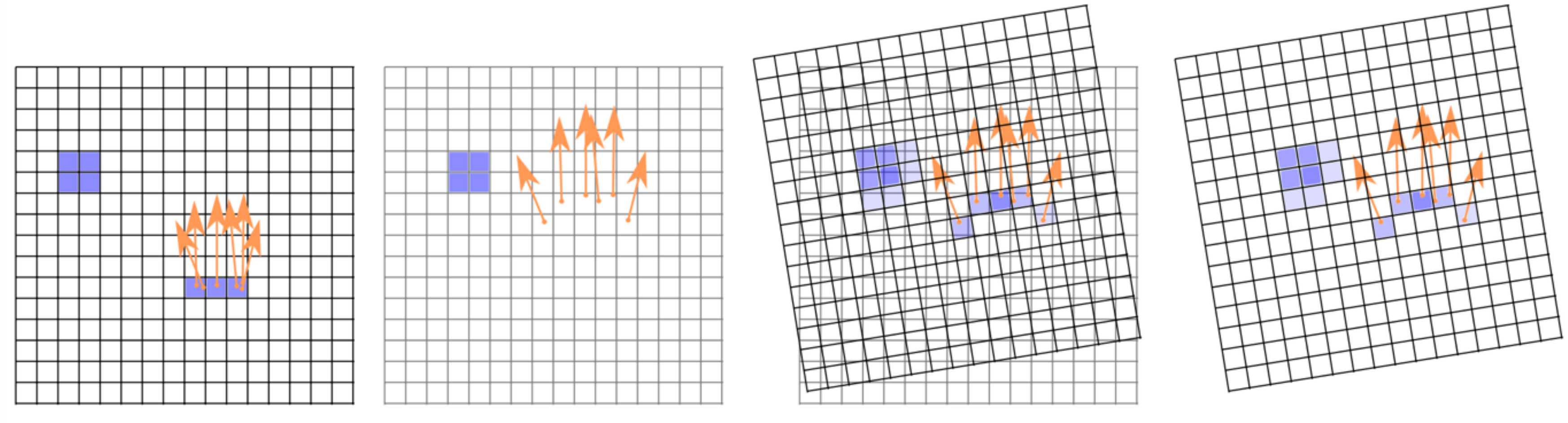
Projective mapping



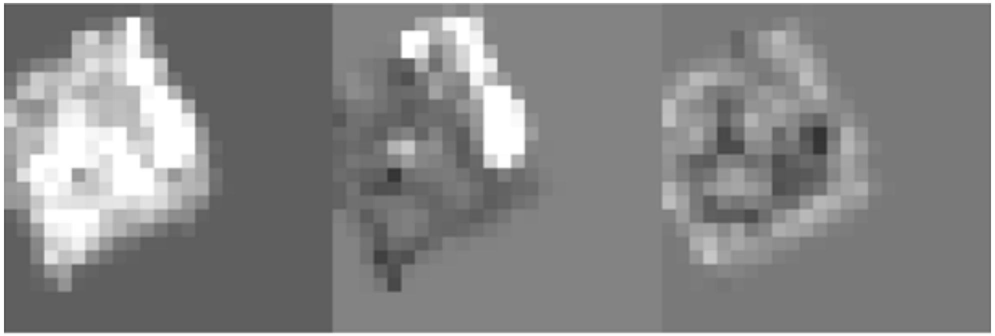
Learning agent control/behavior



Occupancy Grid vs. Egomap

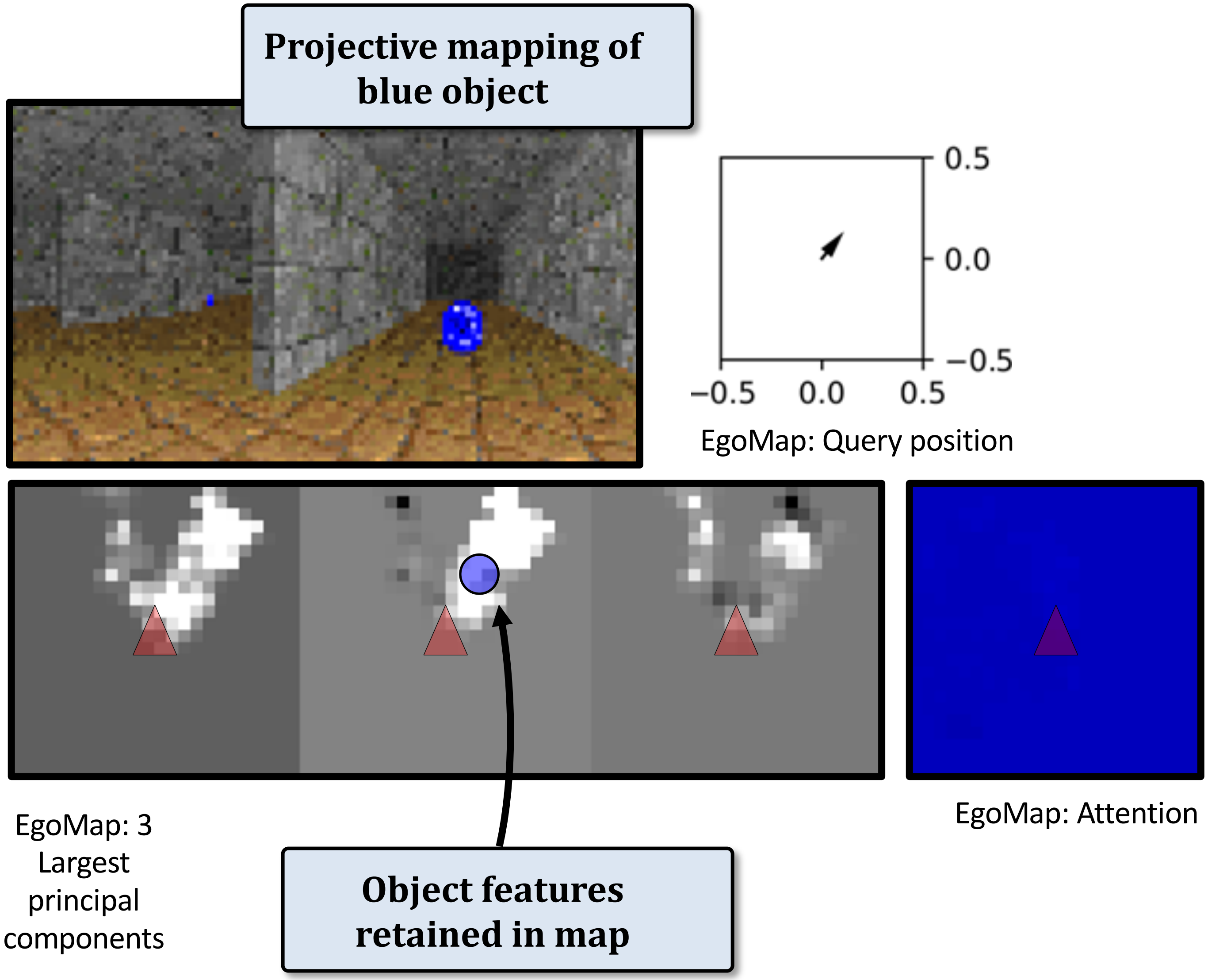


[Rummelhard, Negre, Laugier, 2015]

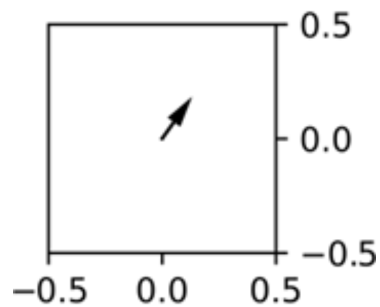


[Beeching, Dibangoye, Simonin, Wolf, ECML-PKDD 2020]

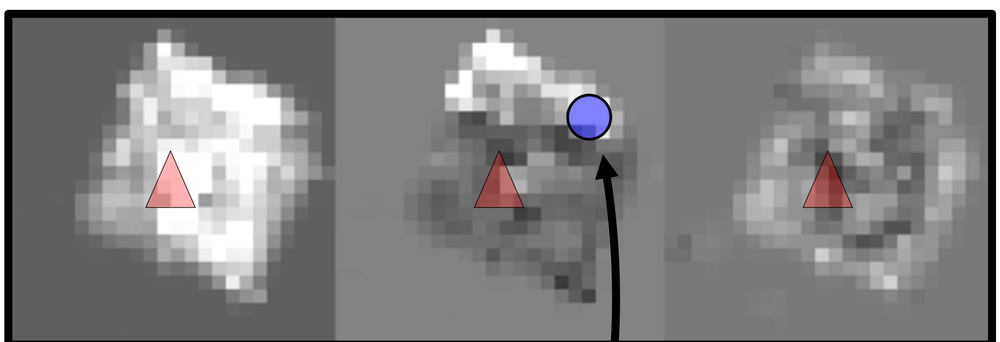
6 item scenario: time-step 005

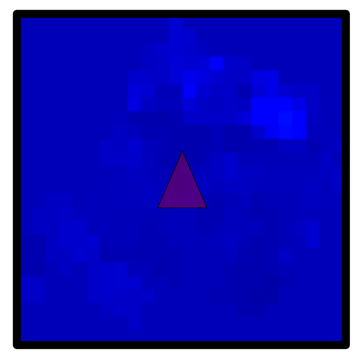


6 item scenario: time-step 105



EgoMap: Query position



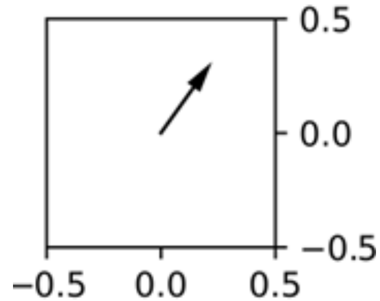


EgoMap: 3 Largest principal components

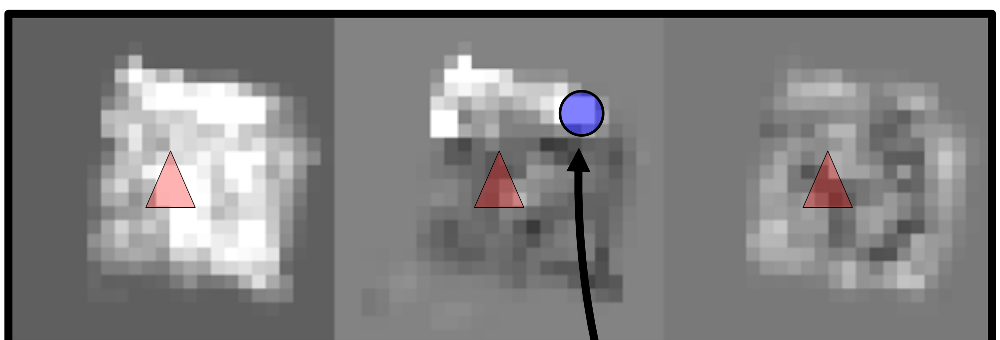
Object features retained in map

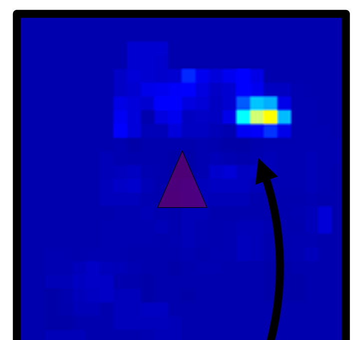
EgoMap: Attention

6 item scenario: time-step 108



EgoMap: Query position



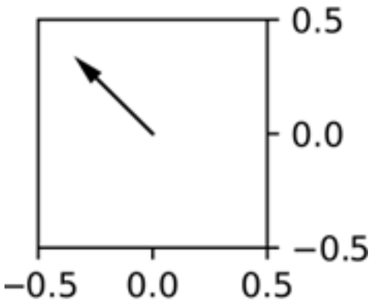


EgoMap: 3 Largest principal components

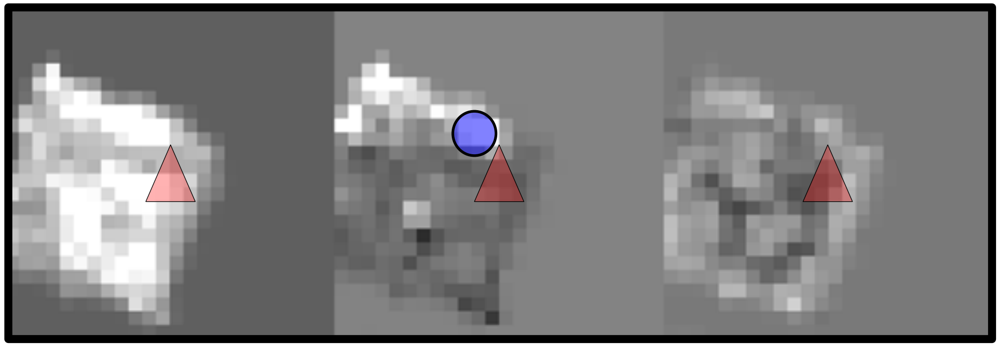
EgoMap: Attention

Collection of object n-1 triggers attention to object n

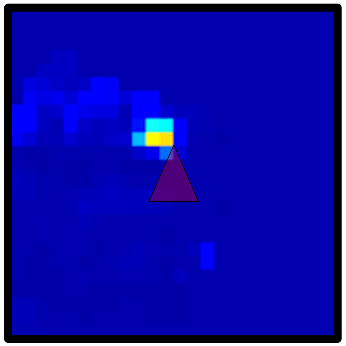
6 item scenario: time-step 134



EgoMap: Query position

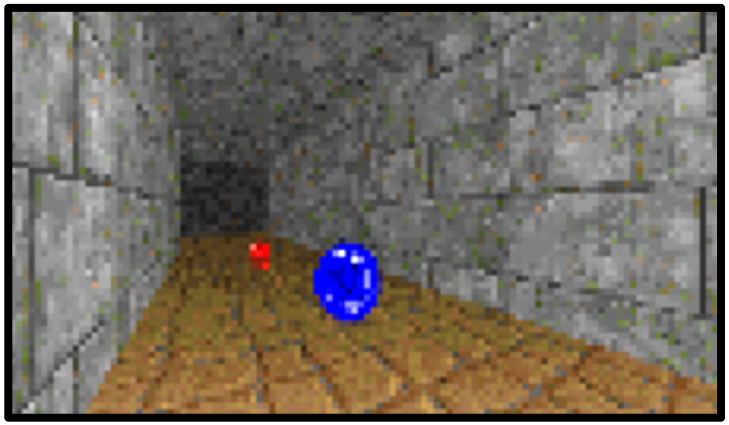


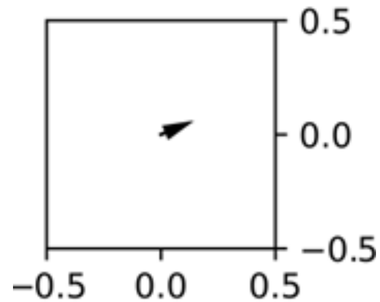
EgoMap: 3 Largest principal components



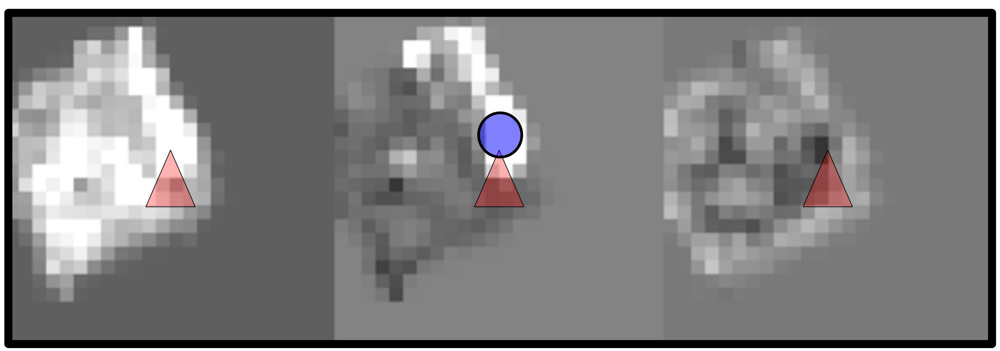
EgoMap: Attention

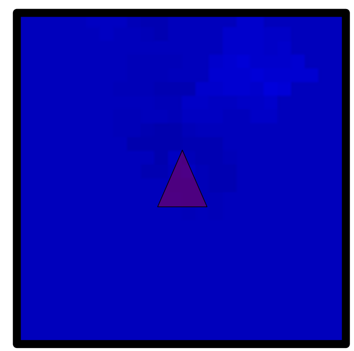
6 item scenario: time-step 140





EgoMap: Query position



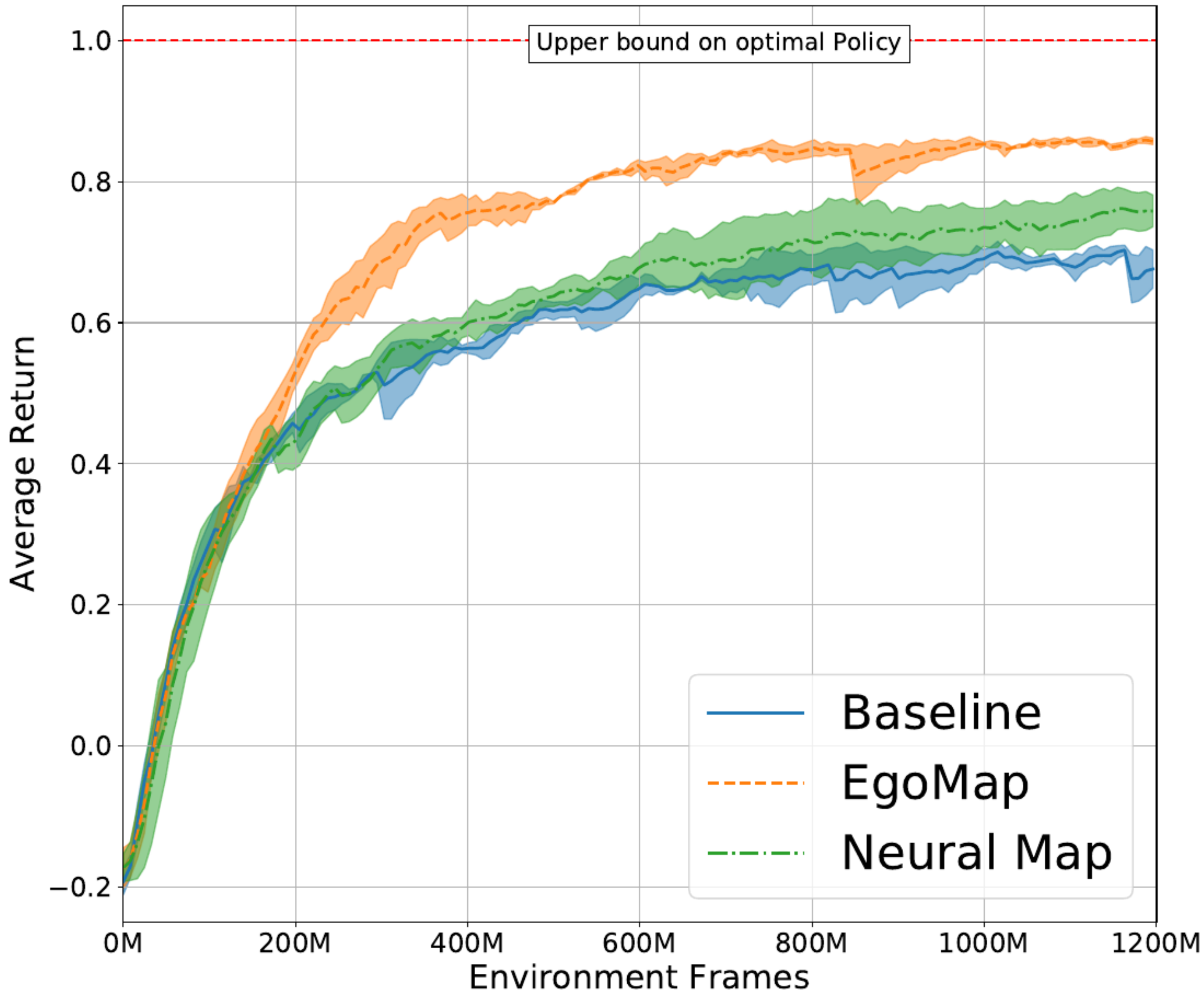


EgoMap: 3 Largest principal components

EgoMap: Attention

When the object is not occluded, the agent does not attend to it

Results



Upper bound on optimal Policy
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Average Return
0M
200M
400M
600M
800M
1000M
1200M
Environment Frames
Baseline
EgoMap
Neural Map

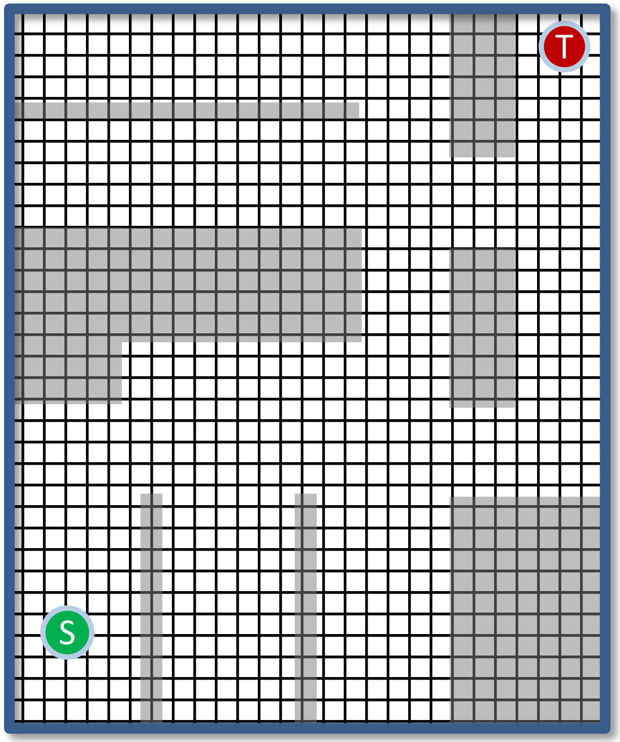
Quantitative results

	Scenario							
	4 item		**6 item**		**Find and Return**		**Labyrinth**	
Agent	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Random	-0.179	-0.206	-0.21	-0.21	-0.21	-0.21	-0.115	-0.086
Baseline	2.341 ± 0.026	2.266 ± 0.035	2.855 ± 0.164	2.545 ± 0.226	0.661 ± 0.003	0.633 ± 0.027	0.73 ± 0.02	0.694 ± 0.009
Neural Map	2.339 ± 0.038	2.223 ± 0.040	2.750 ± 0.062	2.465 ± 0.034	0.825 ± 0.070	0.723 ± 0.026	$\mathbf{0.769 \pm 0.042}$	0.706 ± 0.018
EgoMap	$\mathbf{2.398 \pm 0.014}$	$\mathbf{2.291 \pm 0.021}$	$\mathbf{3.214 \pm 0.007}$	$\mathbf{2.801 \pm 0.048}$	$\mathbf{0.893 \pm 0.007}$	$\mathbf{0.848 \pm 0.017}$	0.753 ± 0.002	$\mathbf{0.732 \pm 0.016}$
Optimum	2.5	2.5	3.5	3.5	1	1	1	1

Mapping objects required

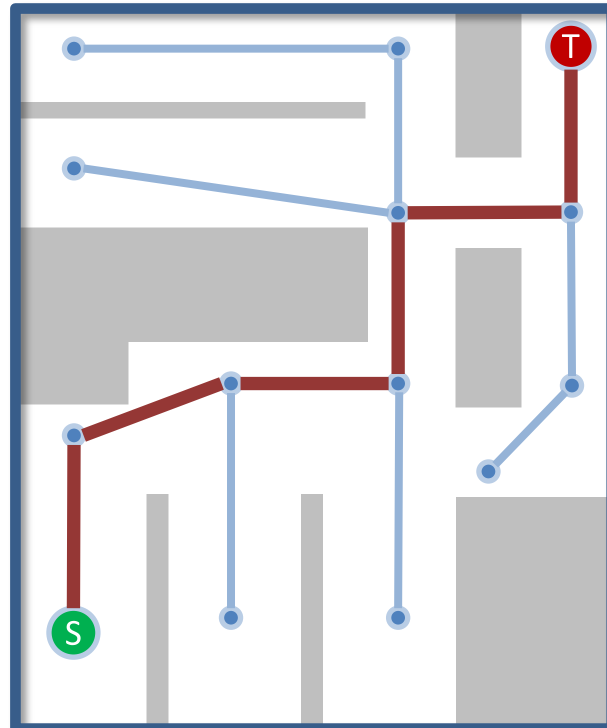
Not required

Spatial maps in robotics



Metric map
(=2D or 3D Grid)

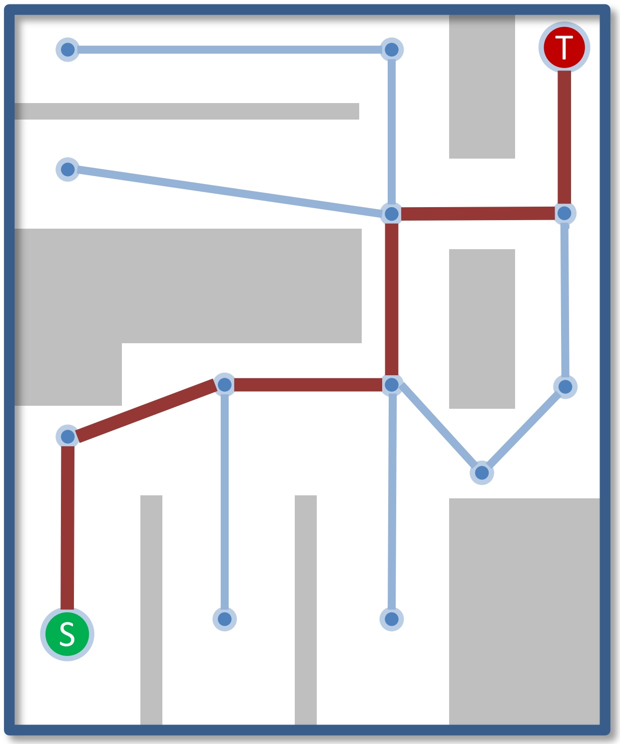
Beeching, Dibangoye, Simonin, Wolf,
EgoMap: Projective mapping and structured egocentric memory for Deep RL,
ECML-PKKD 2020



Topological map
(=Graph)

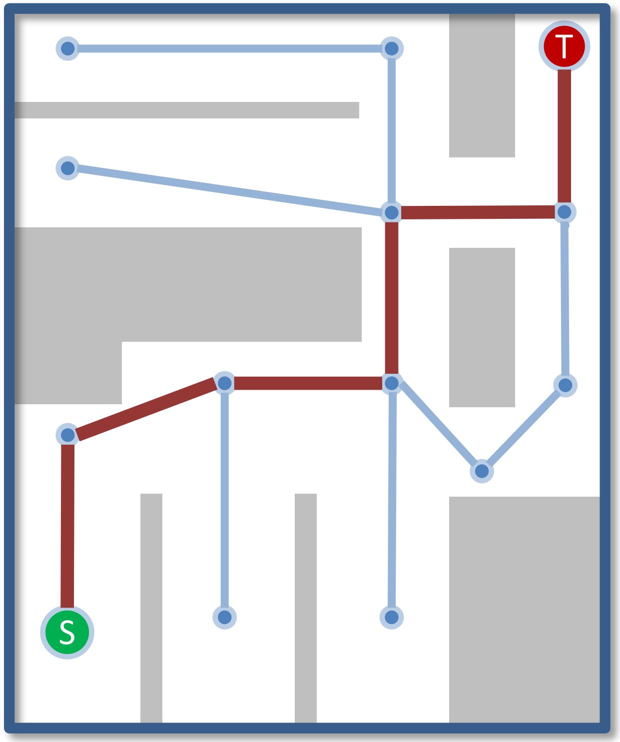
Beeching, Dibangoye, Simonin, Wolf,
Learning to plan with uncertain topological maps,
ECCV 2020

Uncertain topological maps

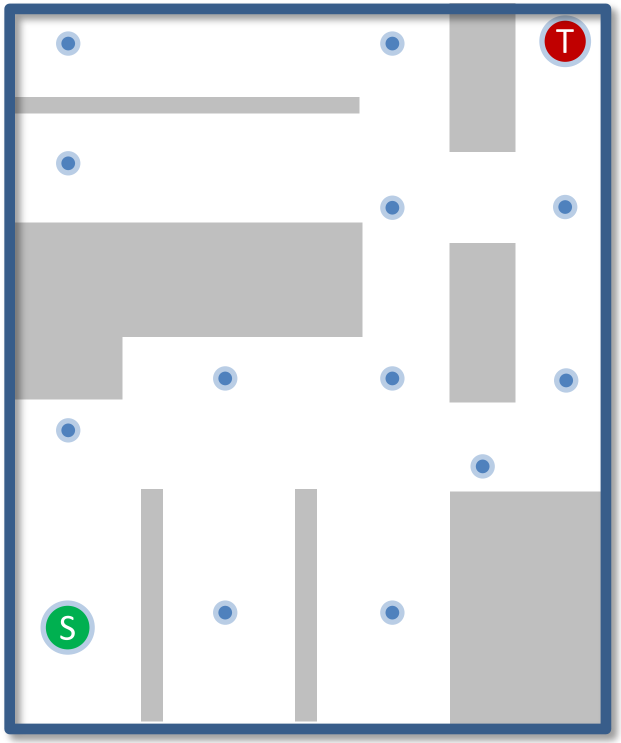


Ground truth

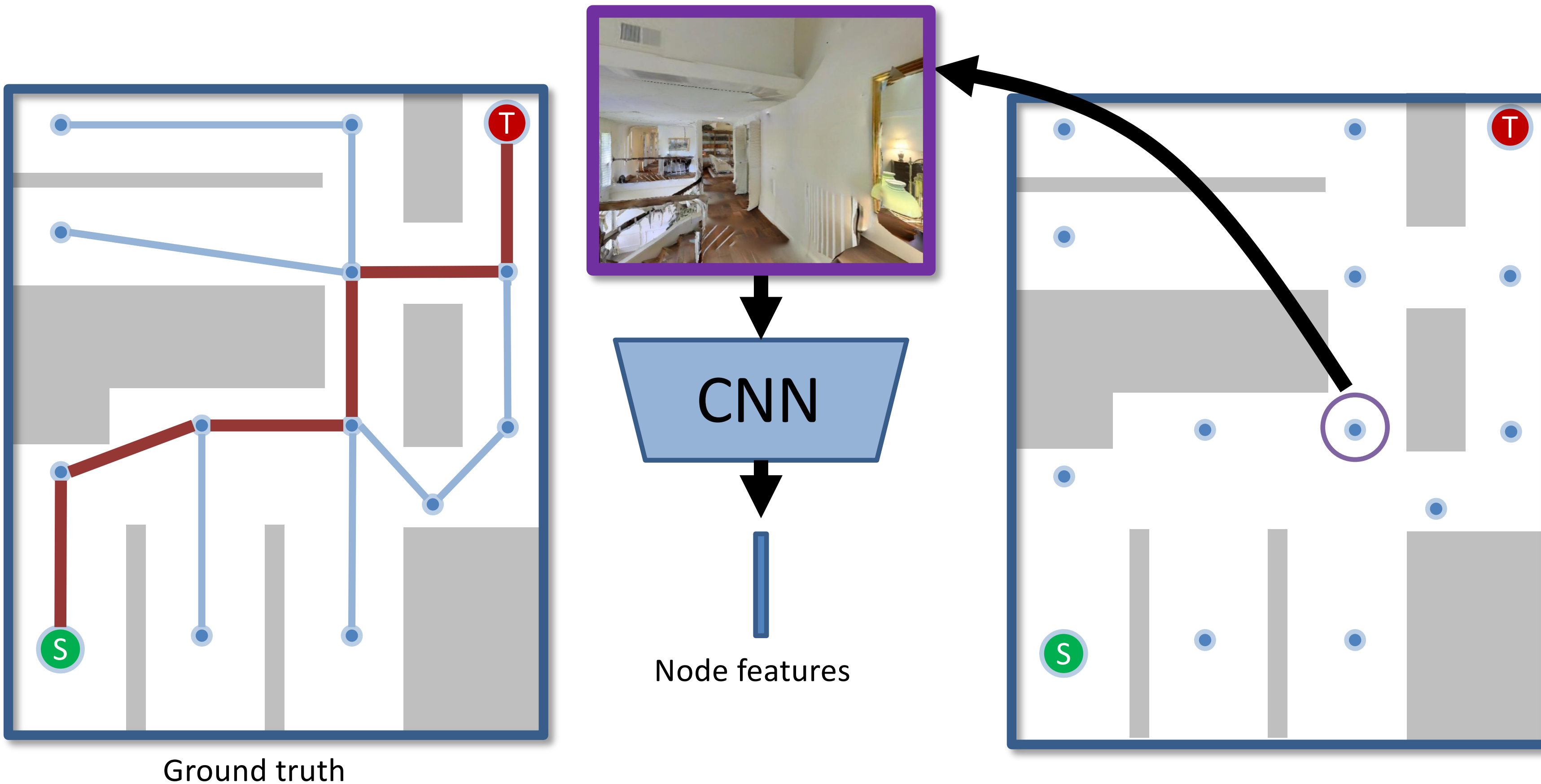
Uncertain topological maps



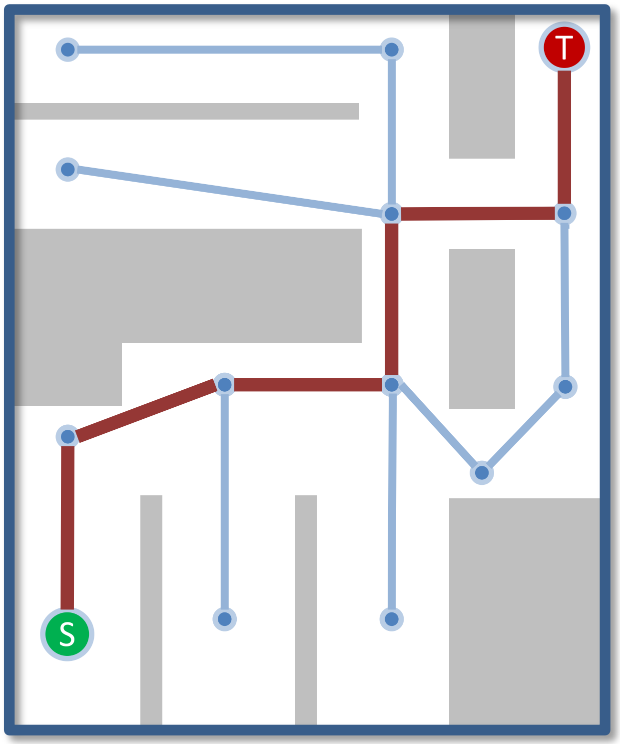
Ground truth



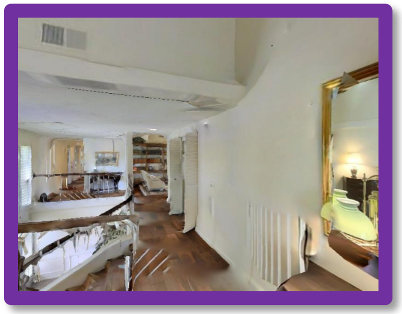
Uncertain topological maps

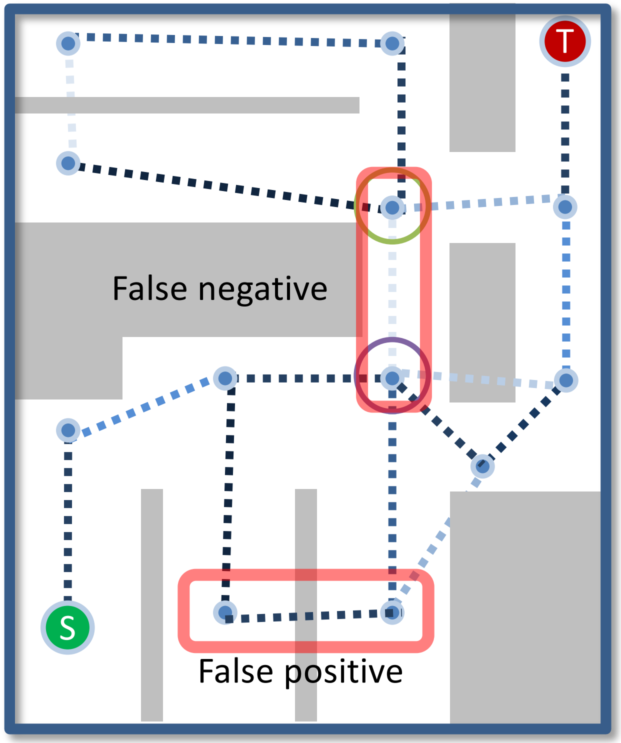


Uncertain topological maps



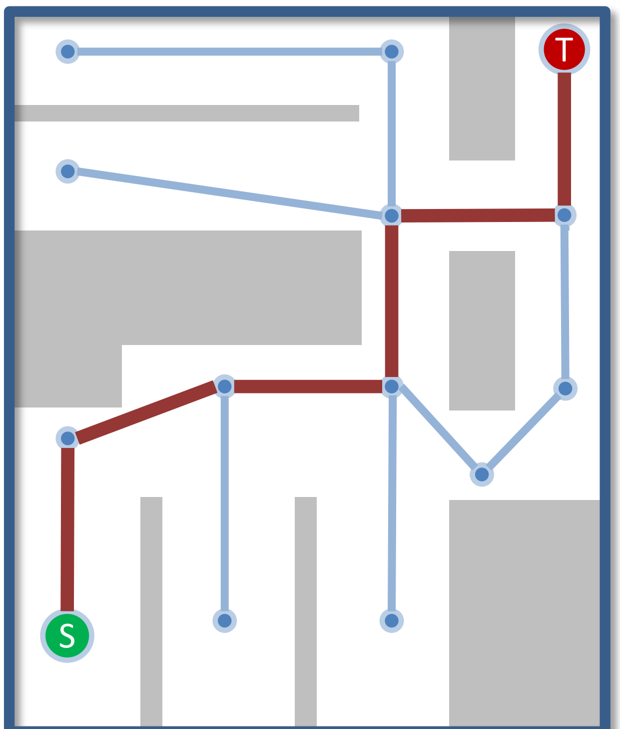
Ground truth



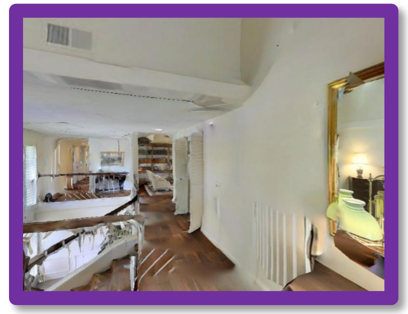


Uncertain graph

Uncertain topological maps

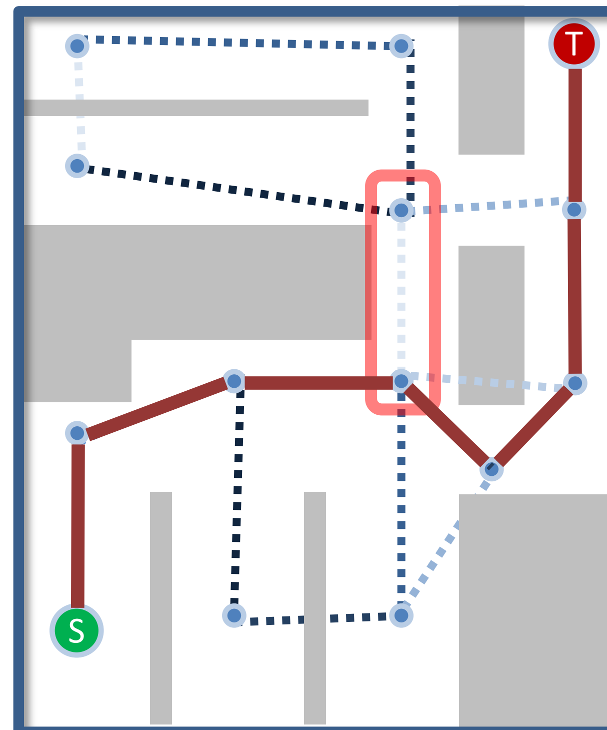


Ground truth



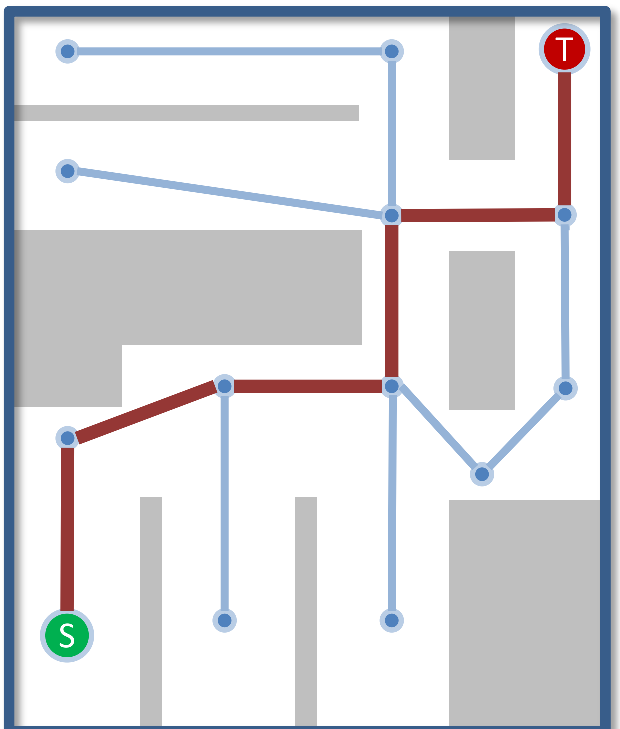
Classical planning:

- **Binary connectivity**
- **Cannot exploit visual information**

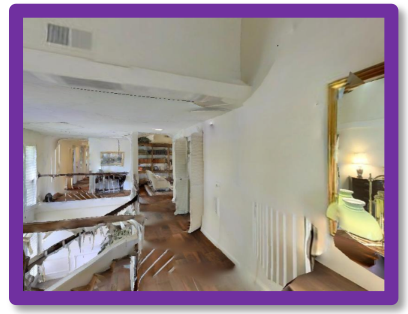


Uncertain graph

Uncertain topological maps

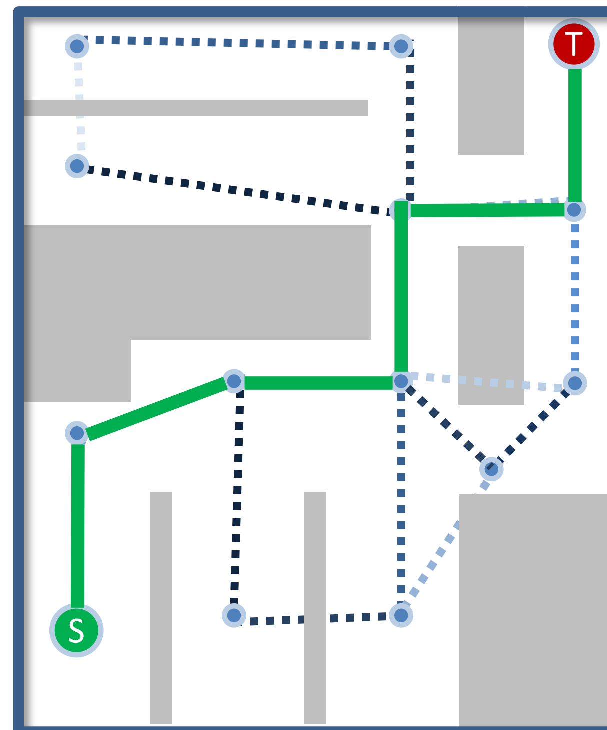


Ground truth



Classical planning:
- **Binary connectivity**
- **Cannot exploit visual information**

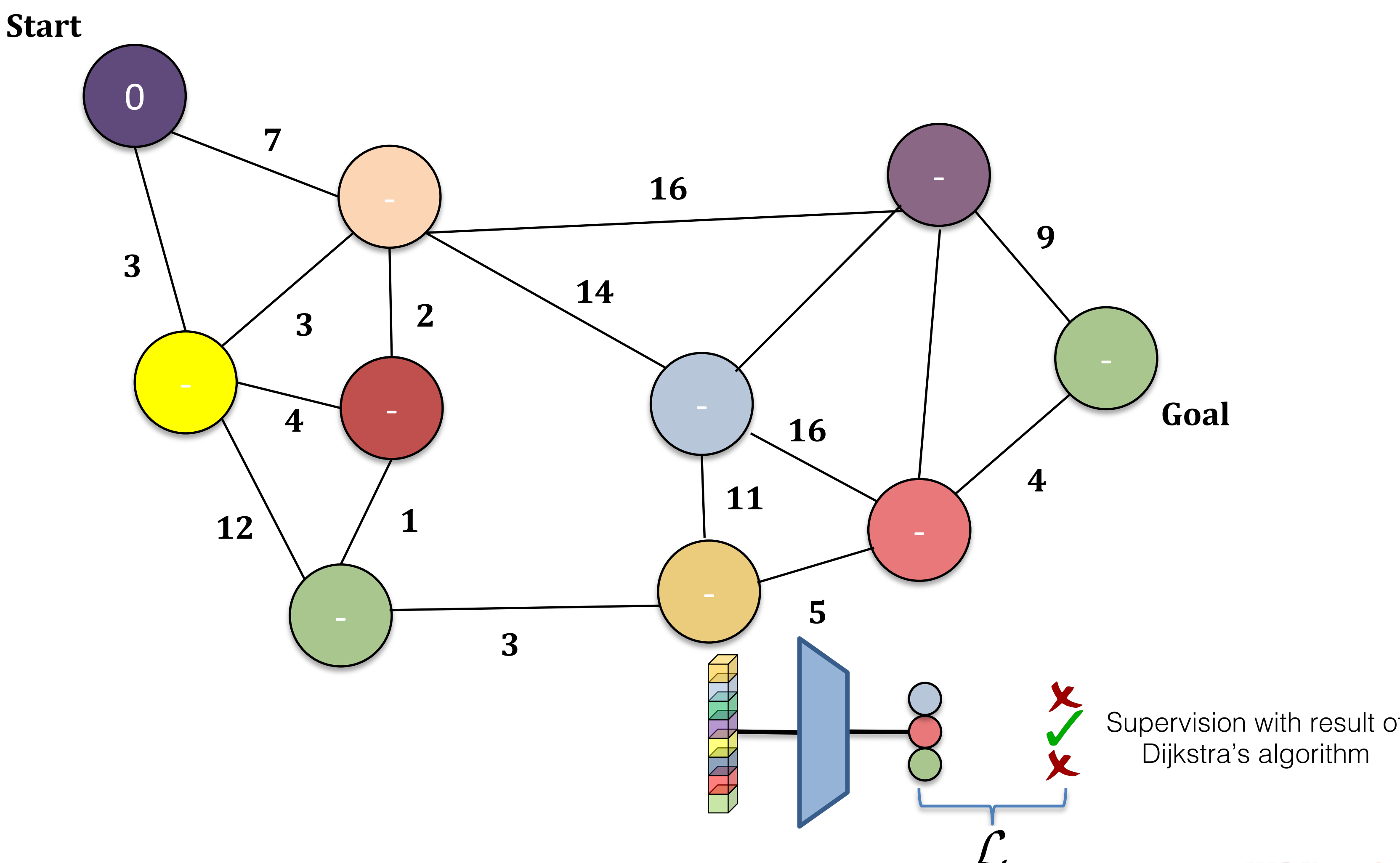
Neural planning:
- **Exploits uncertain connectivity**
- **Uses visual features**
- **Uses neighbor information**

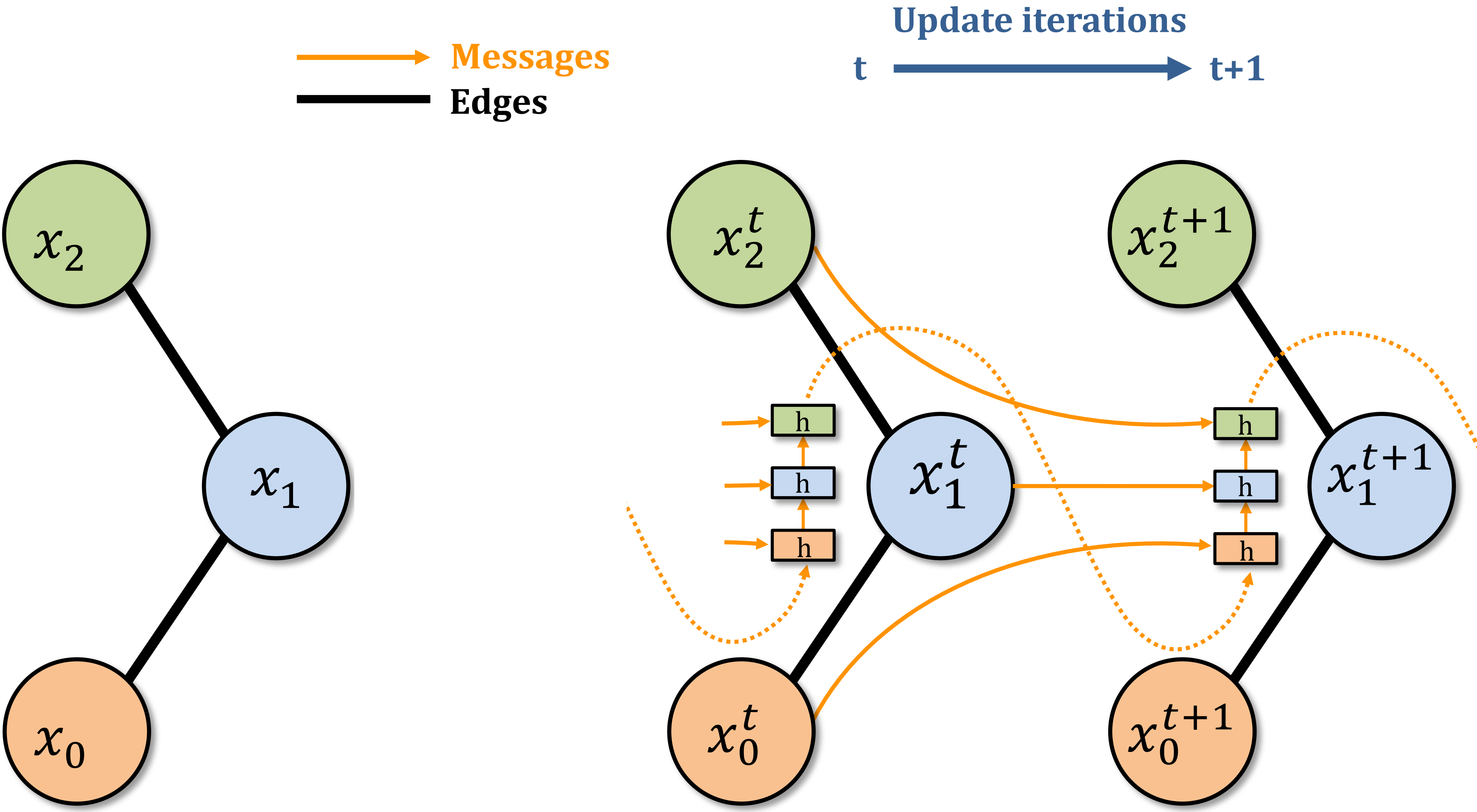


Uncertain graph

Planning as classification

with Graph Neural Networks





Messages
Edges
Update iterations
t
t+1
x_2
x_1
x_0
x_2^t
x_1^t
x_0^t
x_2^{t+1}
x_1^{t+1}
x_0^{t+1}
h
h
h
h
h
h

Results: Neural planner

(a) Uncertain graphs

Method	Acc	H-SPL
Symbolic (threshold)	0.114	0.184
Symbolic (custom cost)	0.115	0.269
Neural (w/o visual)	0.251	0.468
Neural (w visual)	**0.262**	**0.501**

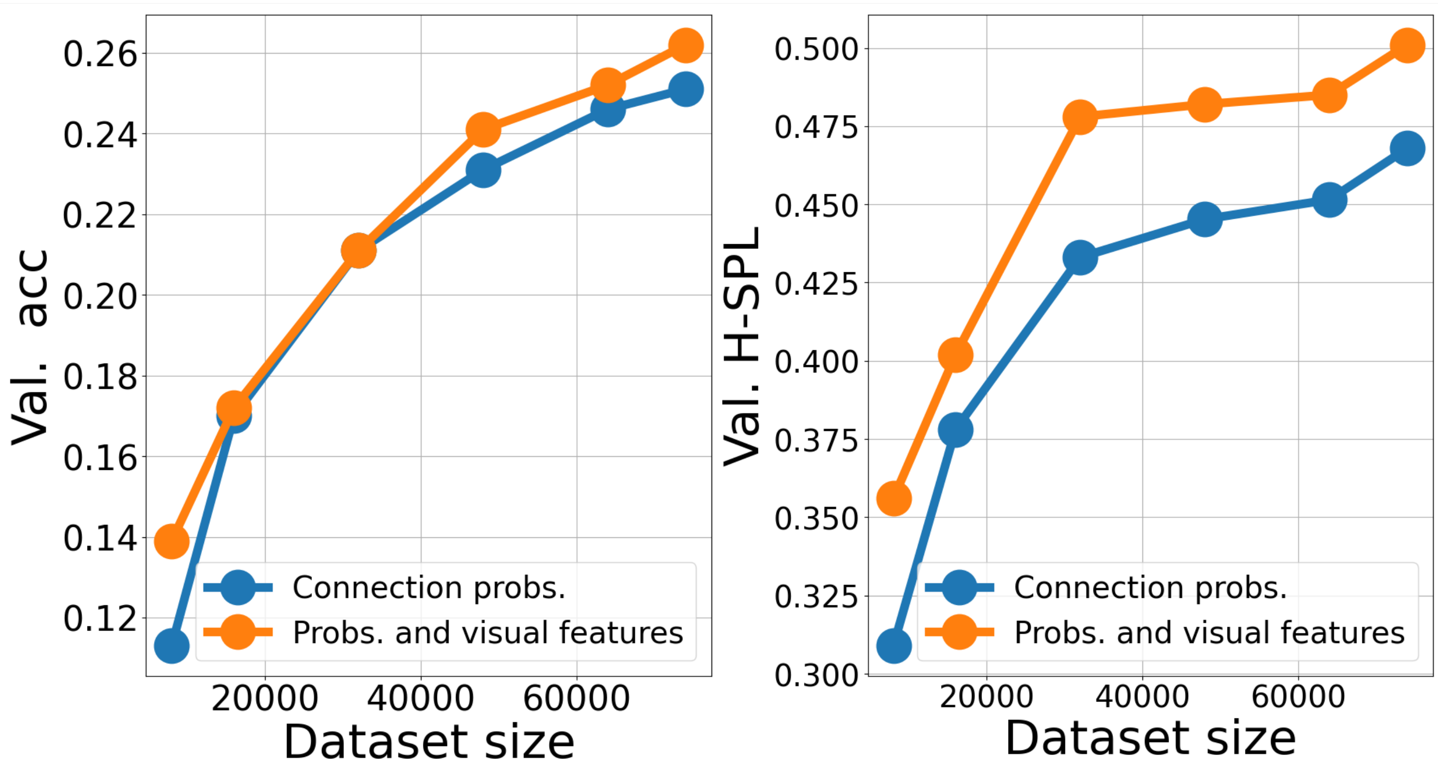
We can beat optimal algorithms ...

... data is King!

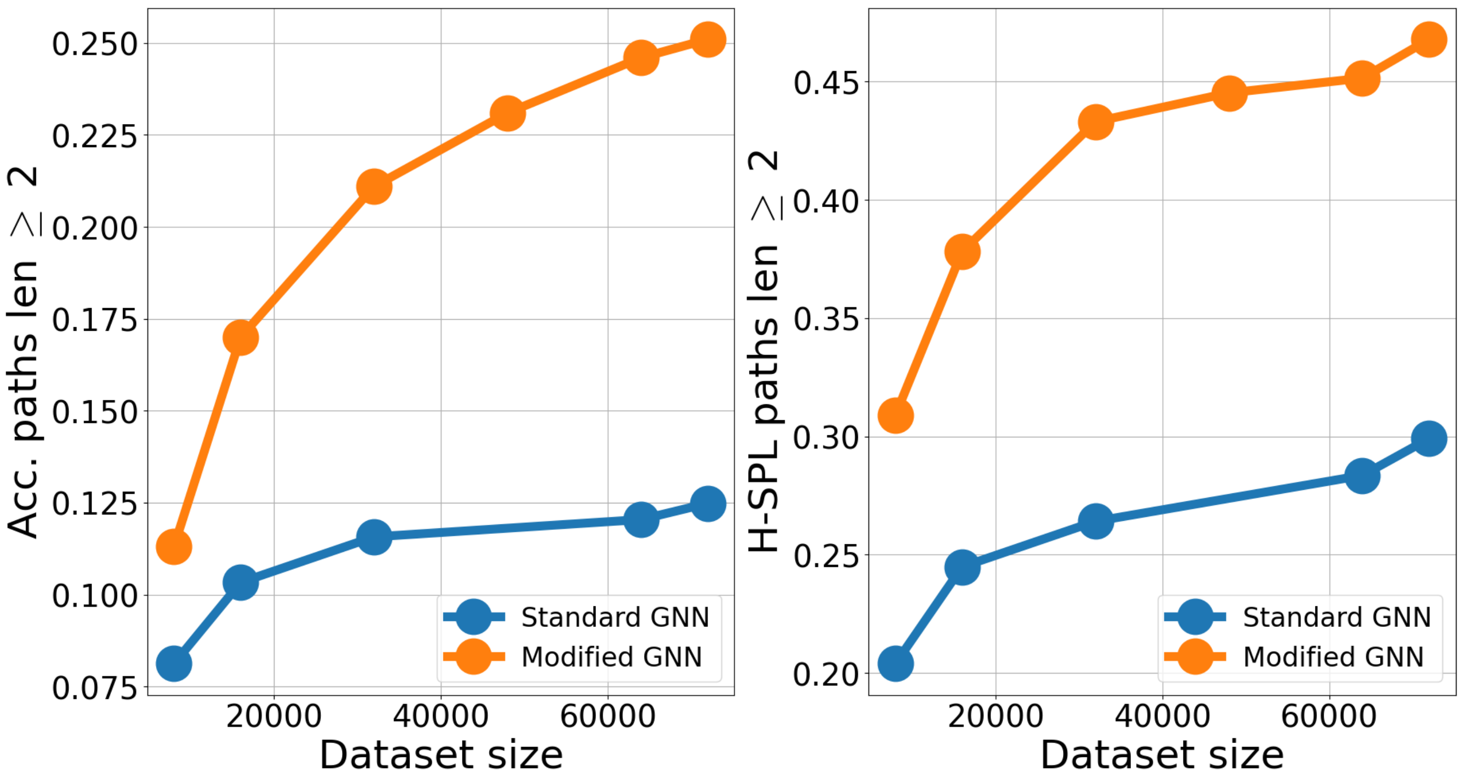
(b) Ground truth graphs

Method	Acc	H-SPL
Symbolic (GT)	**1.00**	**1.00**
Neural planner (GT)	0.921	0.983

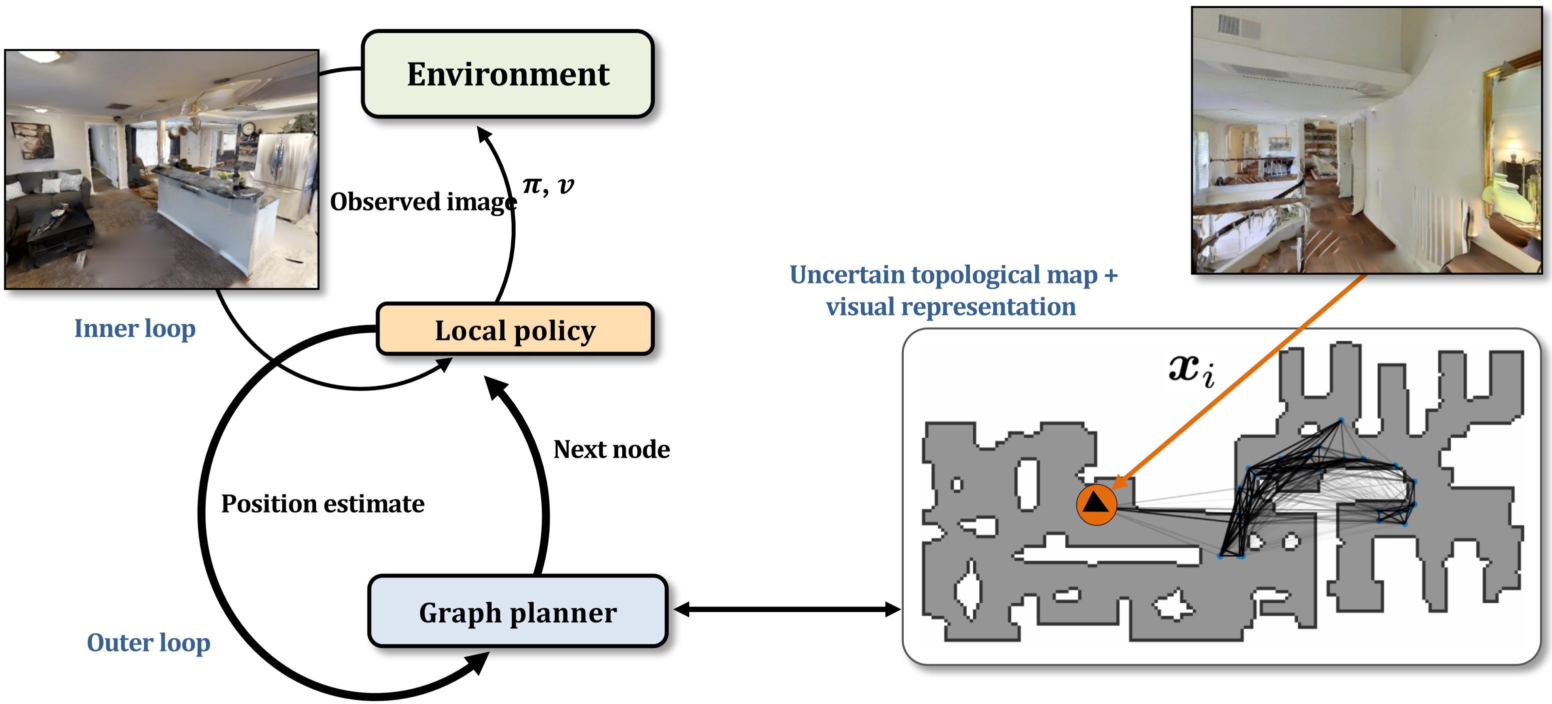
Ablation: Visual features



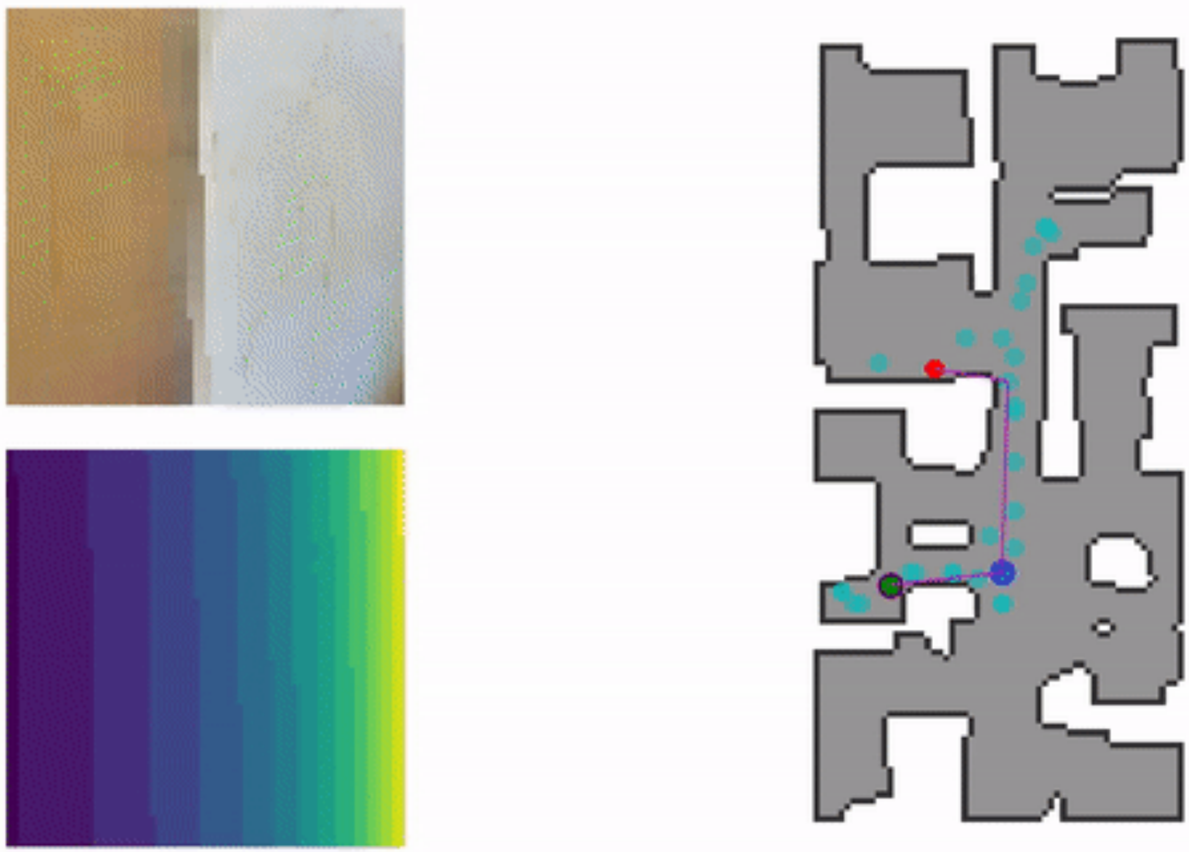
Ablation: GRU for min operation



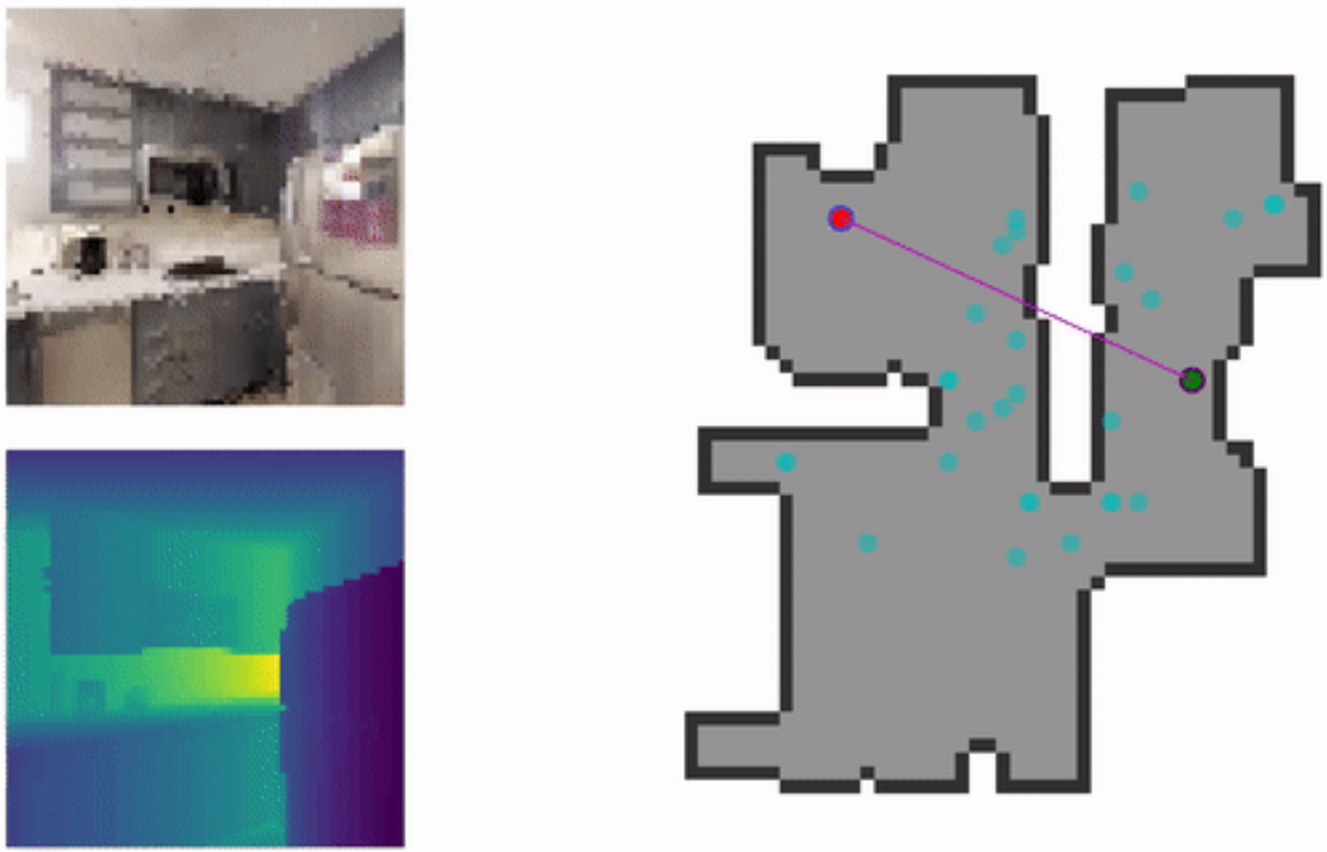
Hierarchical planning



Hierarchical planning and control



Failure Case



Results: Hierarchical planning and control

Method: Planner + Local policy	Success rate	SPL
Graph oracle (optimal point-goals, not comparable)	0.963	0.882
Random	0.152	0.111
Recurrent Image-goal agent	0.548	0.248
Symbolic (threshold)	0.621	0.527
Symbolic (custom cost)	0.707	0.585
Neural planner (sampling)	0.966	0.796
Neural planner (deterministic)	**0.983**	**0.877**

Conclusion

- We aim address the problem of planning and control in photorealistic 3D environments
- We imbue neural networks with inductive bias for planning:
  - Projective geometry (metric maps)
  - Graph based planning
- We learn to plan in uncertain environments
- Future work:
  - Dynamic graph creation
  - Sim2real transfer