Learning to plan with uncertain topological maps

ECCV 2020

Webpage: https://edbeeching.github.io/papers/learning_to_plan




Edward Beeching

Jilles Dibangoye

Olivier Simonin

Christian Wolf

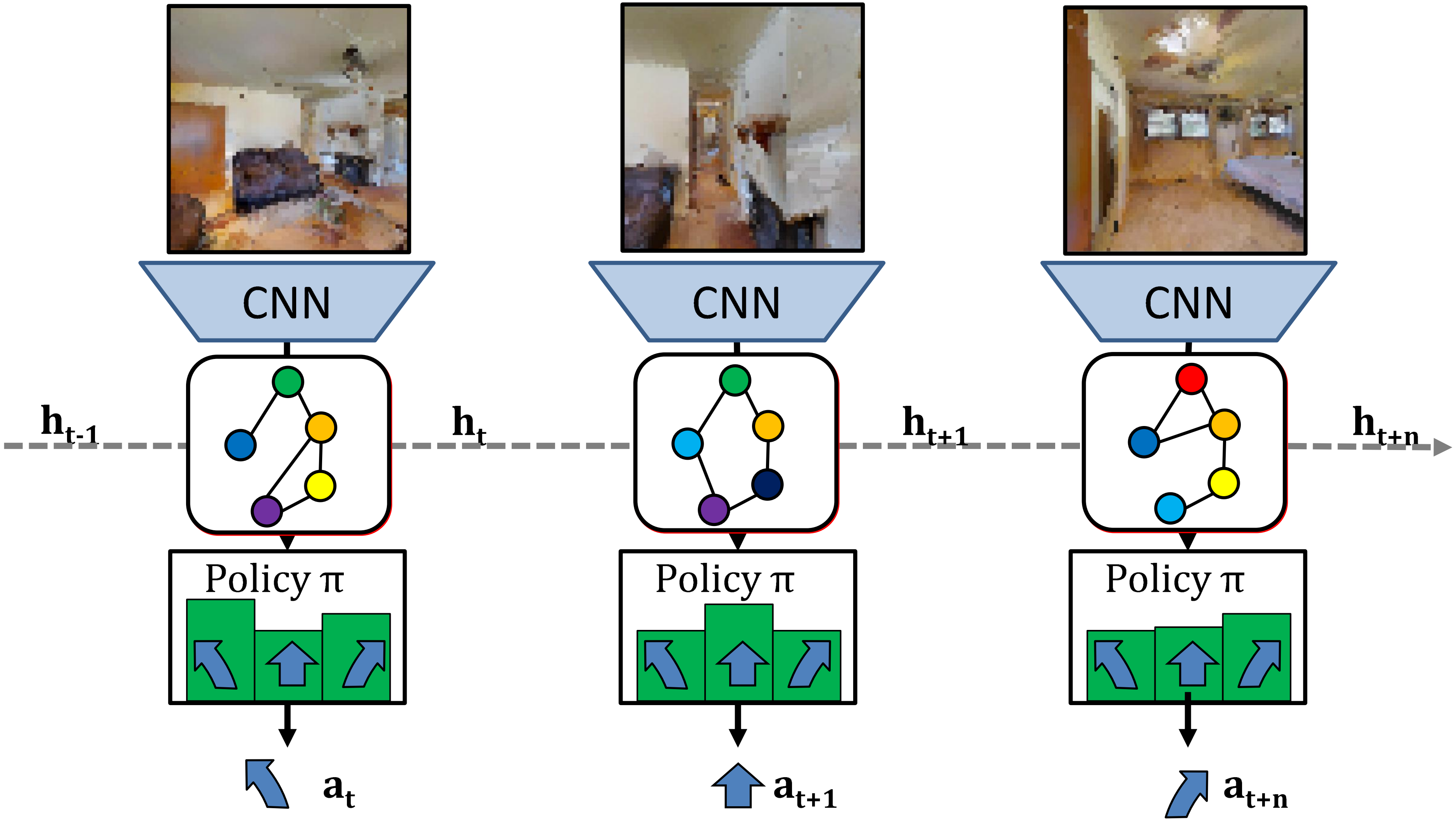
Introduction

- We aim address the problem of planning and control in photorealistic 3D environments Habitat (Savva et al., ICCV 2019)

- This work:
  - Explores the implementation of a neural path planning algorithm
  - To learn to plan in uncertain environments
  - Performs hierarchical planning and control by coupling a high level planner with a low level policy

- Our neural planner:
  - Incorporates visual features in the planning process
  - Exploits regularities in an uncertain graph adjacency matrix
  - Implements an augmented GNN architecture with recurrent updates and a novel gating mechanism

Introduction – Standard recurrent agent

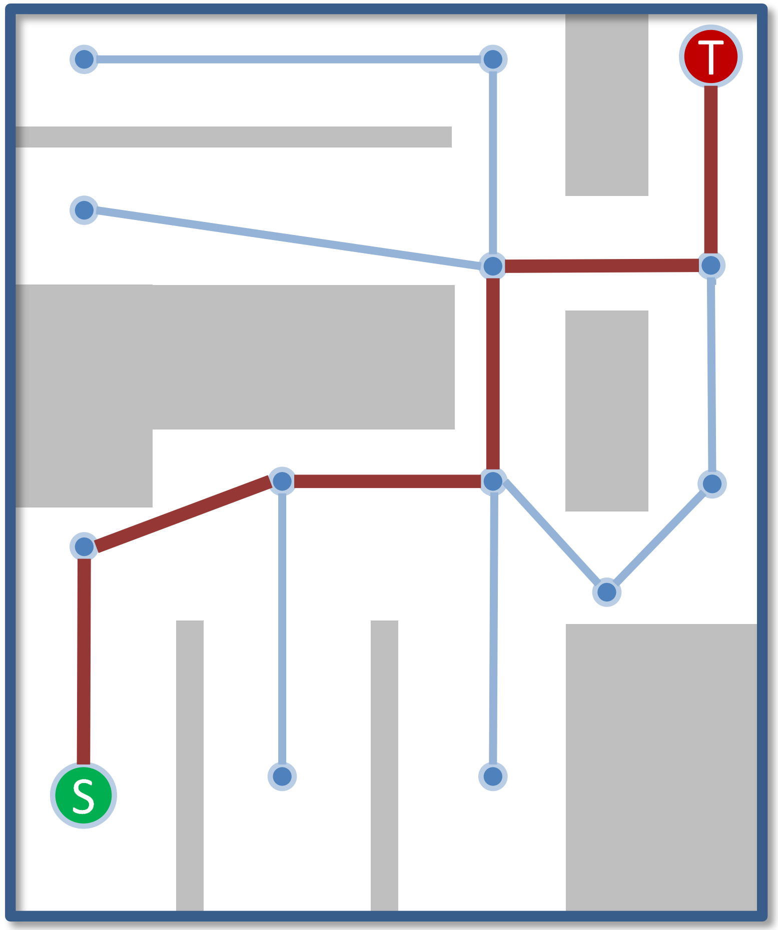


Mirowski et al. ICLR 2017, Jaderburg et al. ICLR 2017, Mirowski et al . ICLR 2018

Introduction – Related works

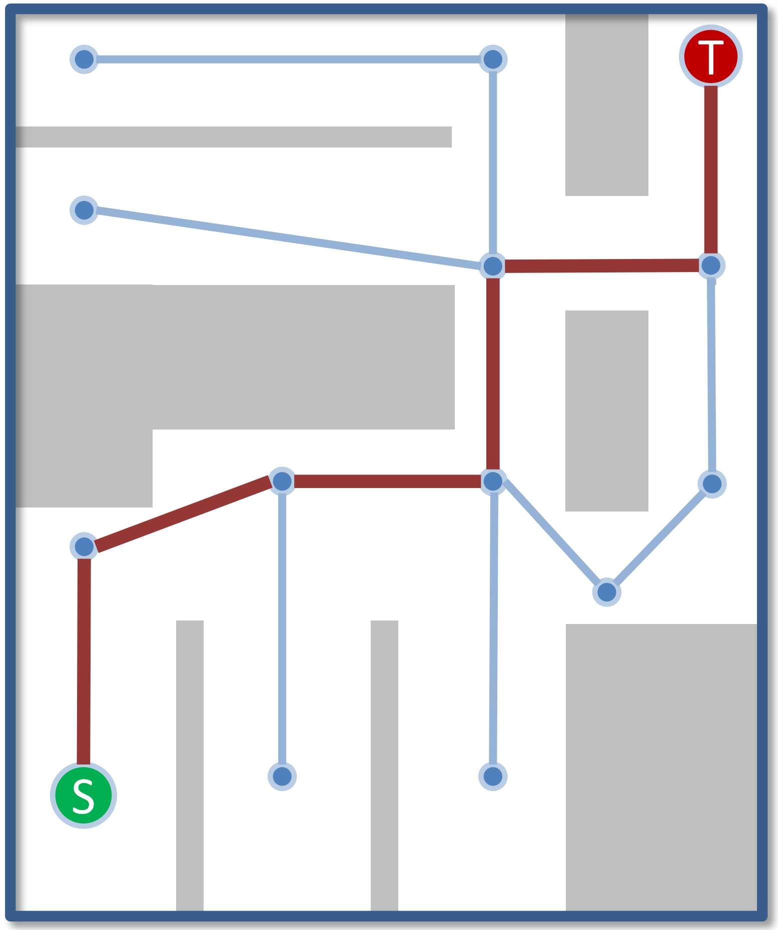
- Related works:
  - Semi-parametric Topological Memory for Navigation (Savinov et al. ICLR 2018)
    - Couples classical planning and learning in the ViZDoom (Kempka et al. CCIG 2016) environment
  - Search on the replay buffer (Eysenbach et al., NeurIPS 2019)
    - Uses values estimates to build a graph
    - Does not generalize to unseen environments
  - Neural Topological SLAM for Visual Navigation (Chaplot et al. CVPR 2020)
    - Combines SLAM with a novel graph based method
    - Relies on classical planning
- This work:
  - Learns a neural approximation of a classical planning algorithm (Bellman-Ford)
  - Demonstrates the method outperforms classical methods on uncertain graphs.

Learning to plan with uncertain topological maps

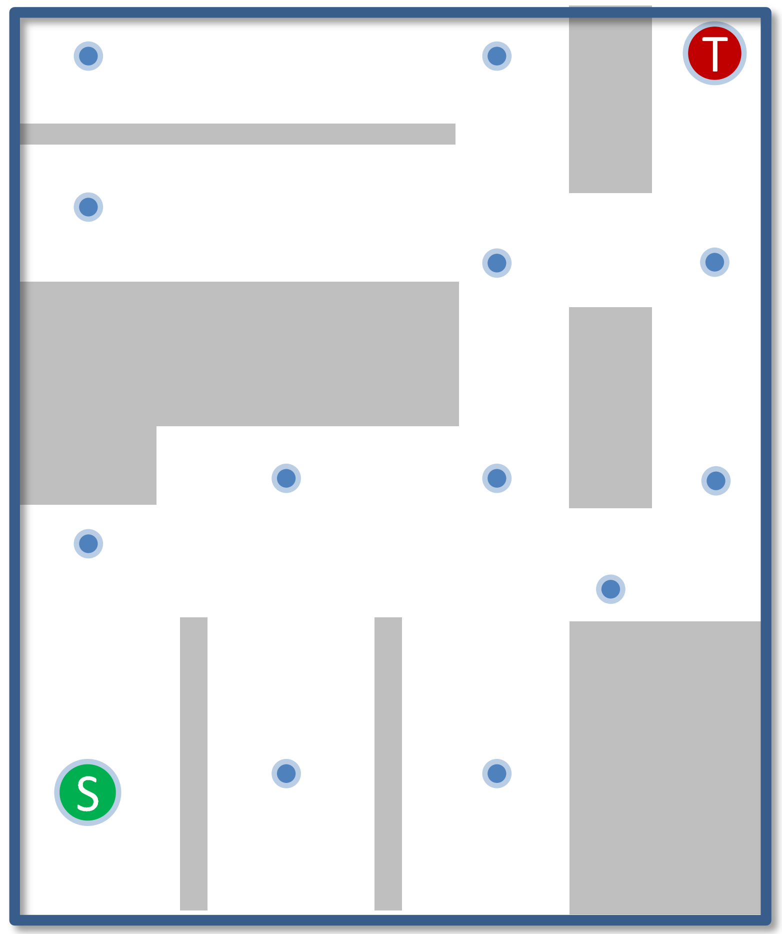


Ground truth

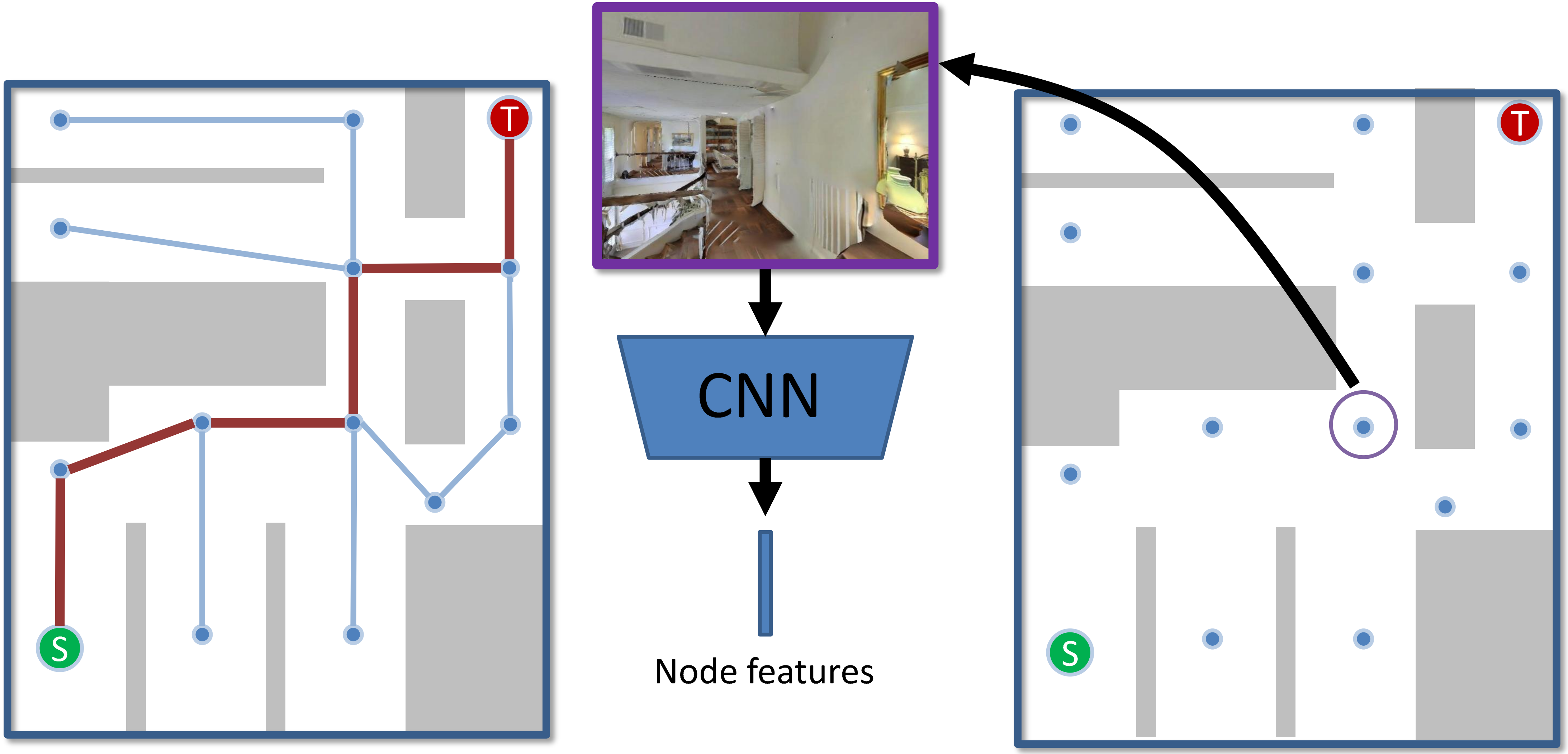
Learning to plan with uncertain topological maps



Ground truth

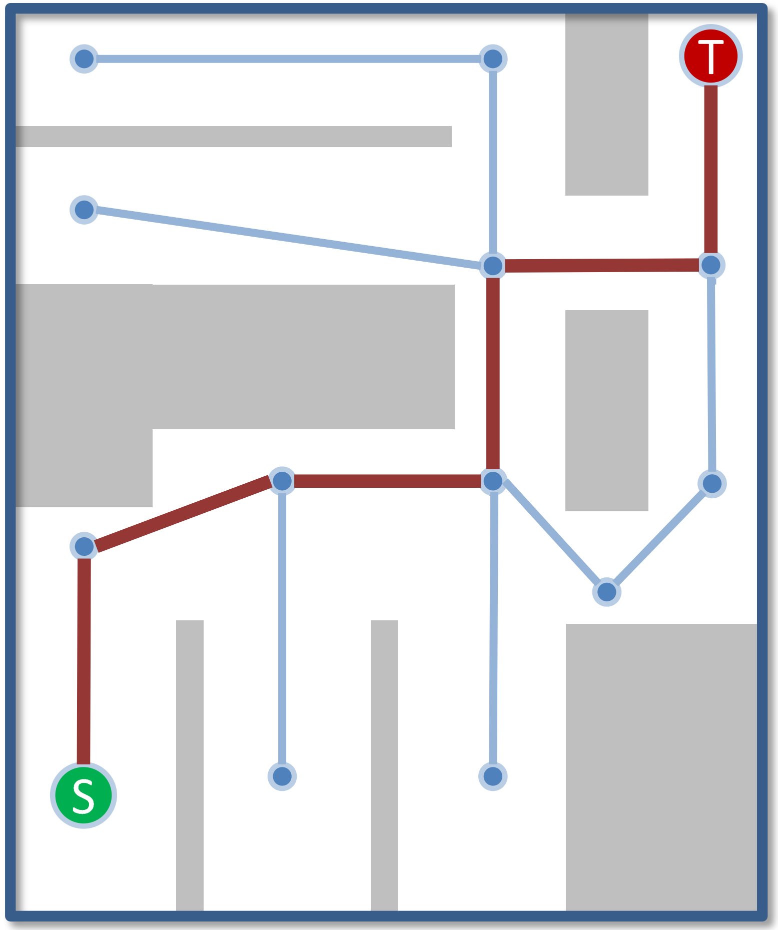


Learning to plan with uncertain topological maps

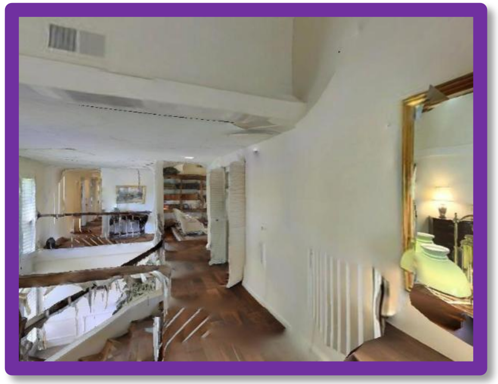


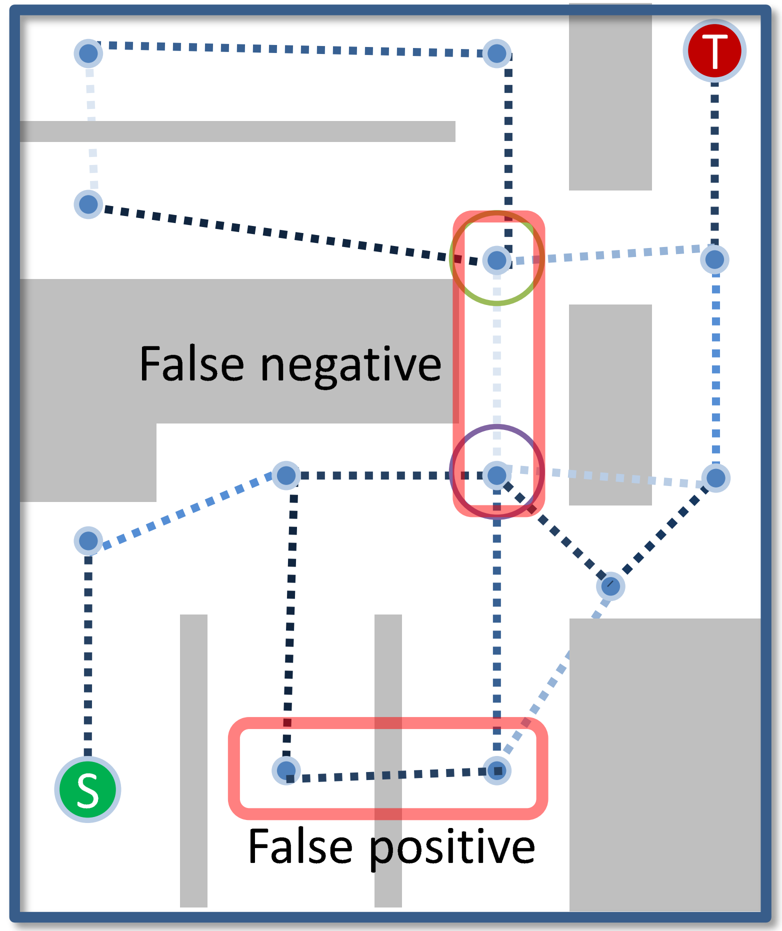
Ground truth

Learning to plan with uncertain topological maps



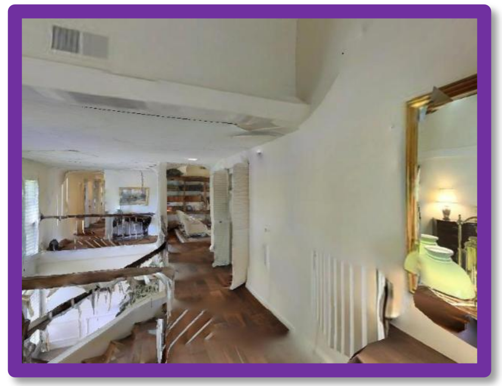
Ground truth

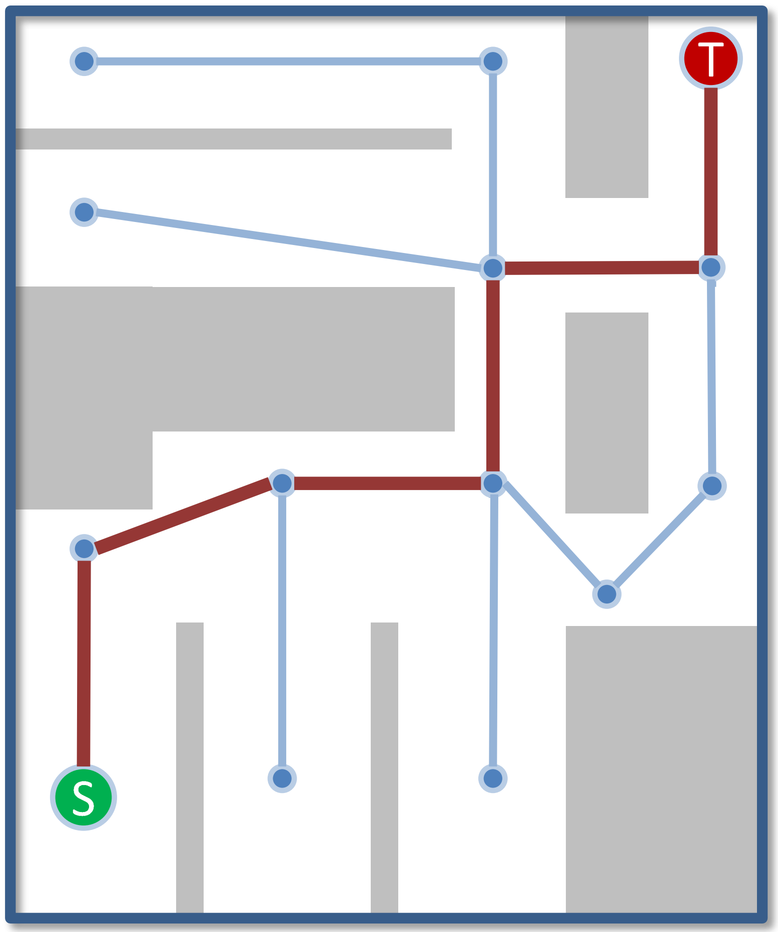




Uncertain graph

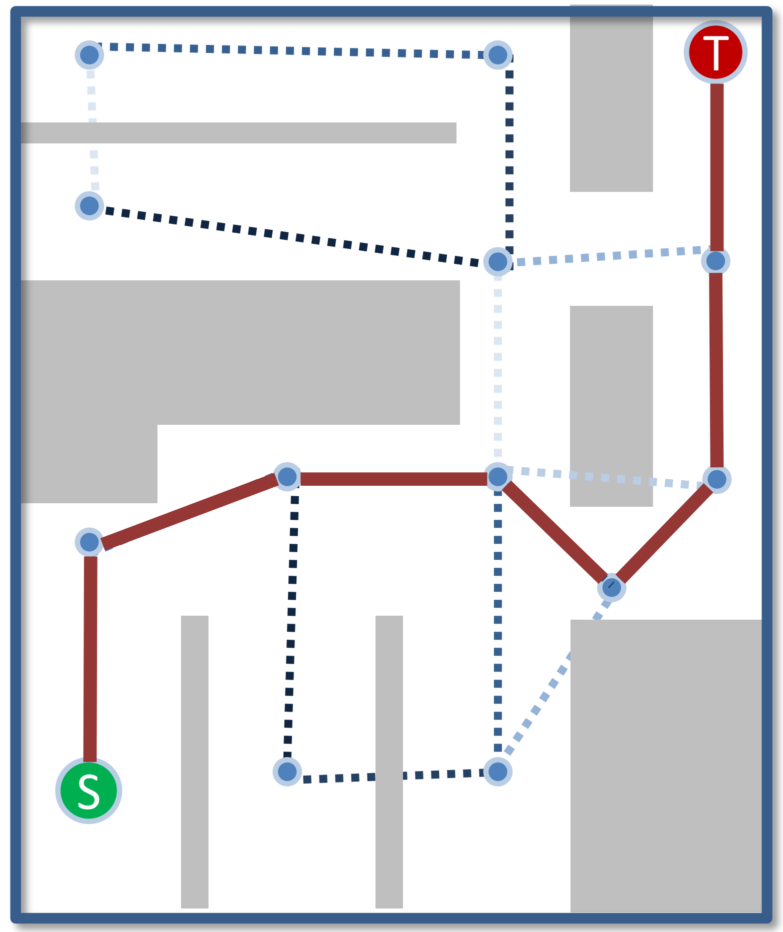
Learning to plan with uncertain topological maps





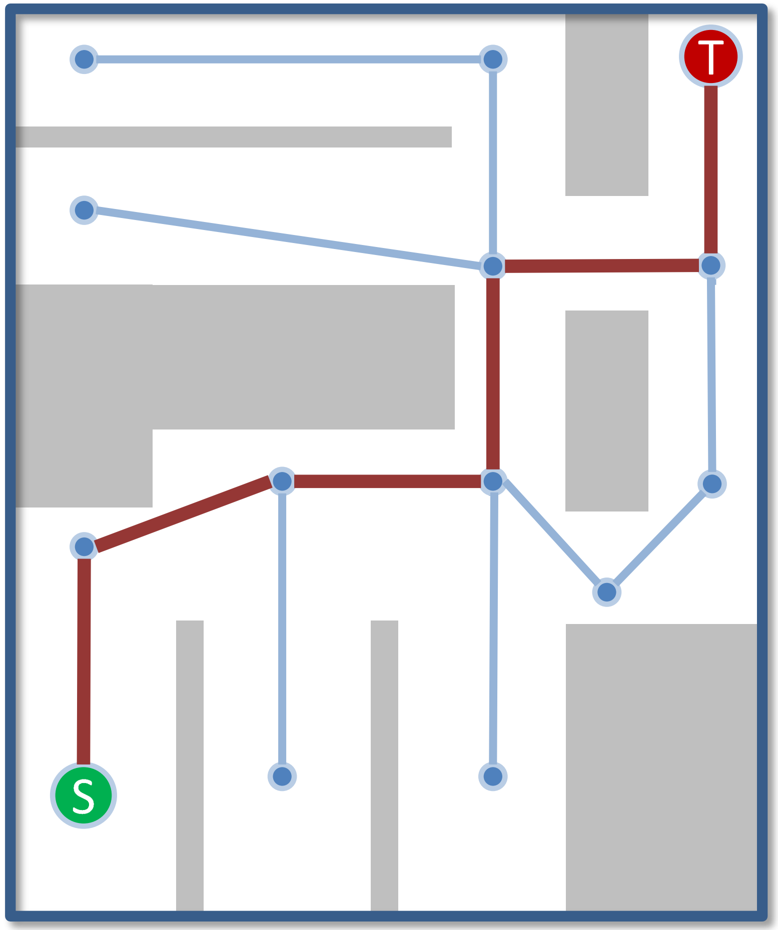
Ground truth

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

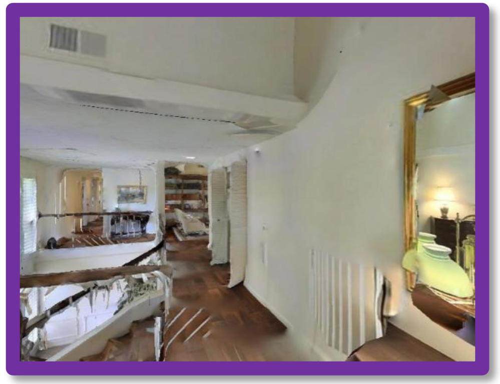


Uncertain graph

Learning to plan with uncertain topological maps

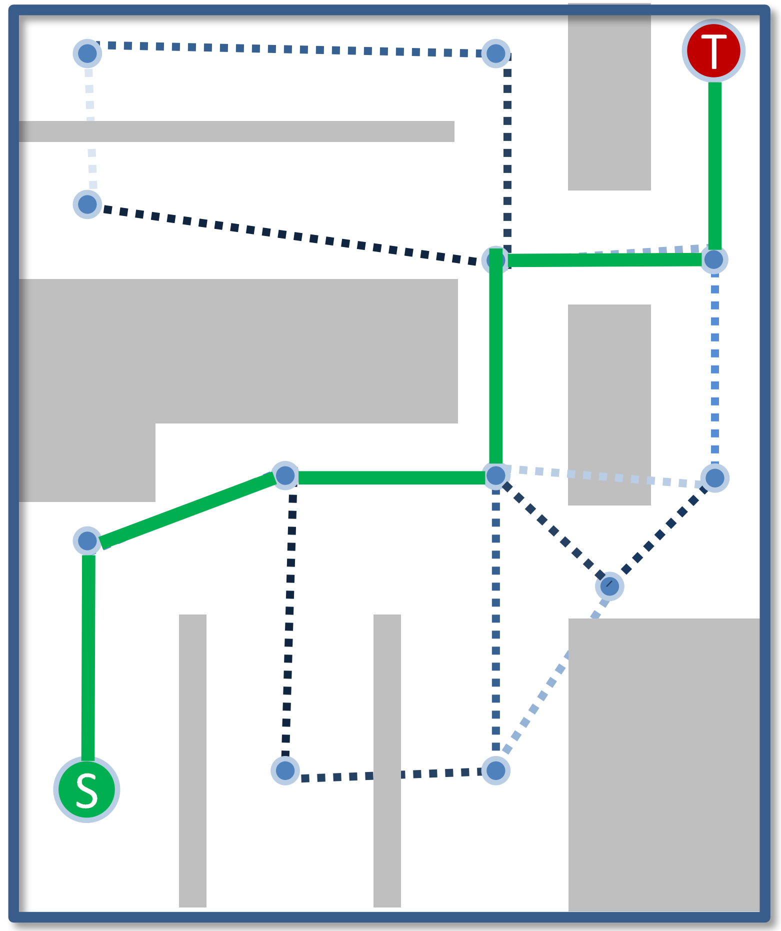


Ground truth



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

Neural planning:
- **Exploits uncertain connectivity**
- **Uses visual features**
- **Learn from regularities in the graph structure**



Uncertain graph

Graph generation

- Graphs were generated by training:
  - an exploratory policy trained to maximize coverage (Chen et al. ICLR 2019)
  - a parameterized function $f(\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{h}_t)$ that predicts whether there is line of sight connectivity between two observations.

- Dataset generation:
  - Each graph contains 32 nodes
  - 72,000,000 training examples
  - 16,000,000 test examples

Training

- Treat planning as a classification problem
- During training: ground truth optimal paths are available
- For each node, perform an n-class classification of the next node on the optimal path to the target
- Each node, i, contains a feature embedding $\boldsymbol{x}_i = [\boldsymbol{v}_i, \boldsymbol{e}_i, t_i, \boldsymbol{d}_i, \boldsymbol{s}_i]$

  Where:

  $\boldsymbol{v}_i$: visual features

  $\boldsymbol{e}_i$: edge connection probabilities

  t_i: boolean value indicating if the node is the target

  $\boldsymbol{d}_i$: distances from node i to all other nodes

  $\boldsymbol{s}_i$: one hot vector identifying the node

Inductive bias

- Inspiration – Classical path planning algorithms
  - Dijkstra’s algorithm:
    - Contains priority queue which is not differentiable
  - Bellman Ford:
    - Iterates over neighbors
    - Passes messages
    - Updates bounds
    - Potential for a neural approximation with an augmented GNN