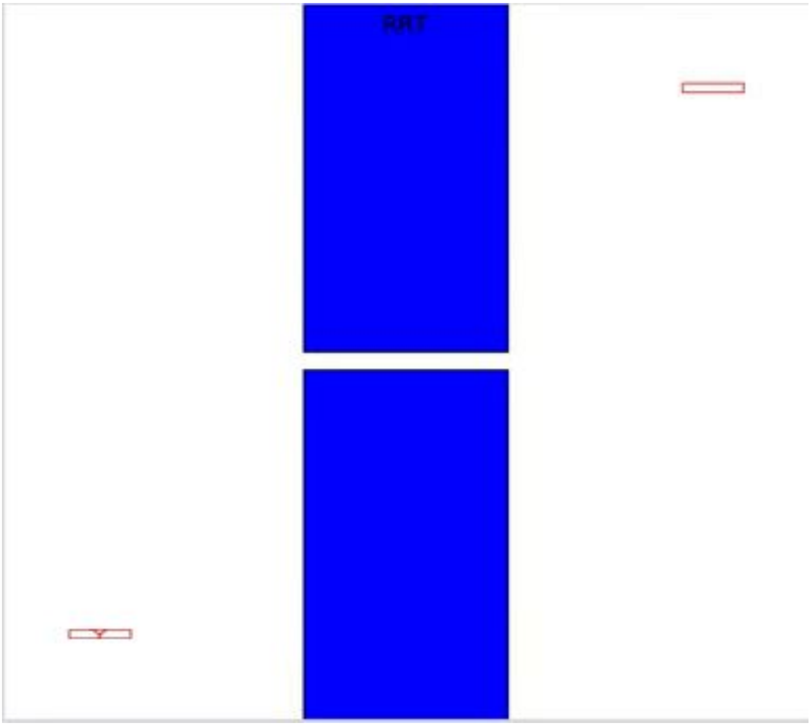
Learning to Sample in an Online Fashion for Robot Motion Planning

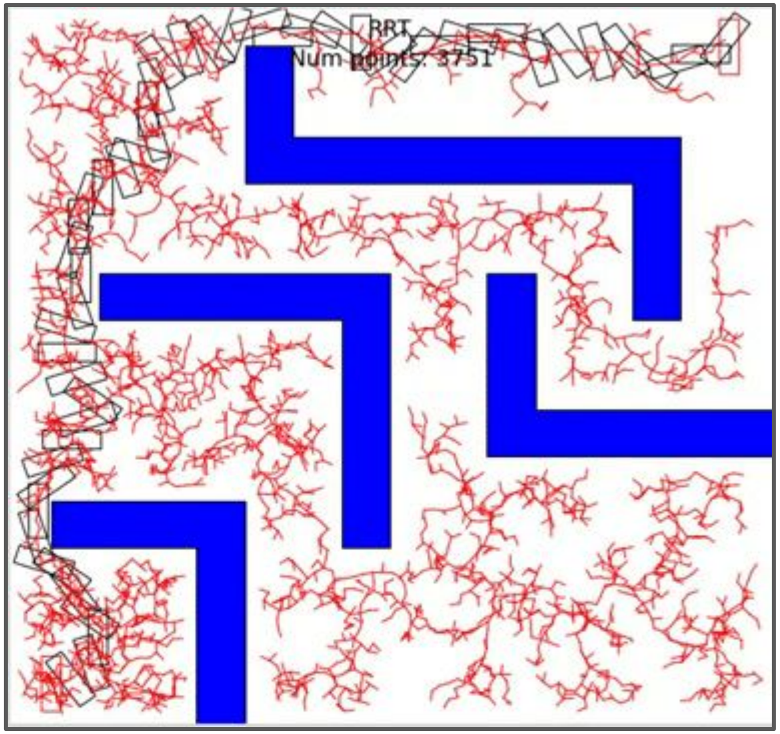
Rogerio Bonatti, Ratnesh Madaan,
Brian Okorn, Sam Zeng

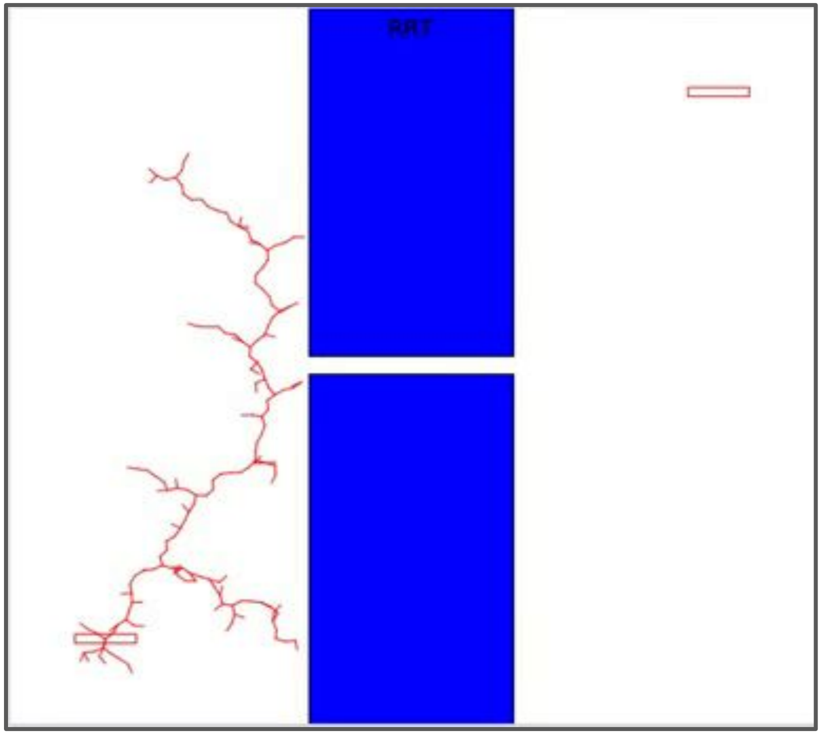
What are Rapidly-Exploring Random Trees (RRT)?



RRT: The Piano-Movers Problem
https://www.youtube.com/watch?v=rPgZyq15Z-Q&

What are Rapidly-Exploring Random Trees?

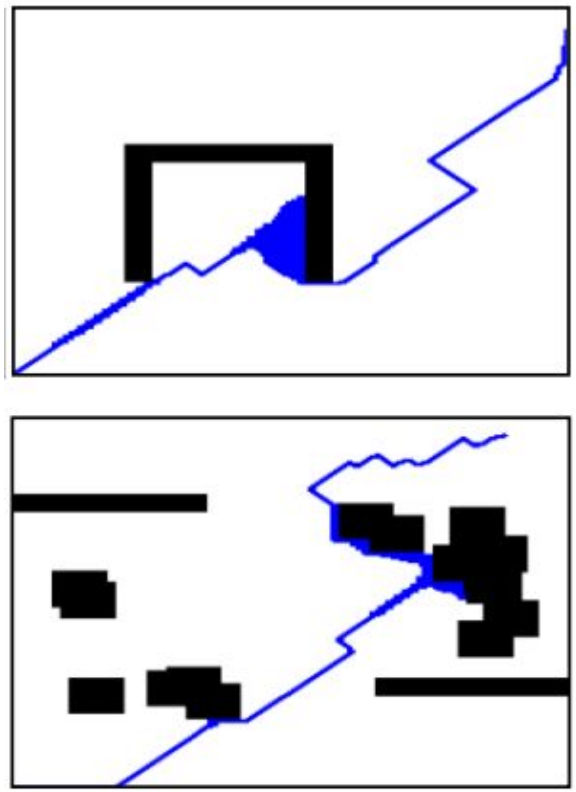




RRT: The Piano-Movers Problem
https://www.youtube.com/watch?v=rPgZyq15Z-Q&

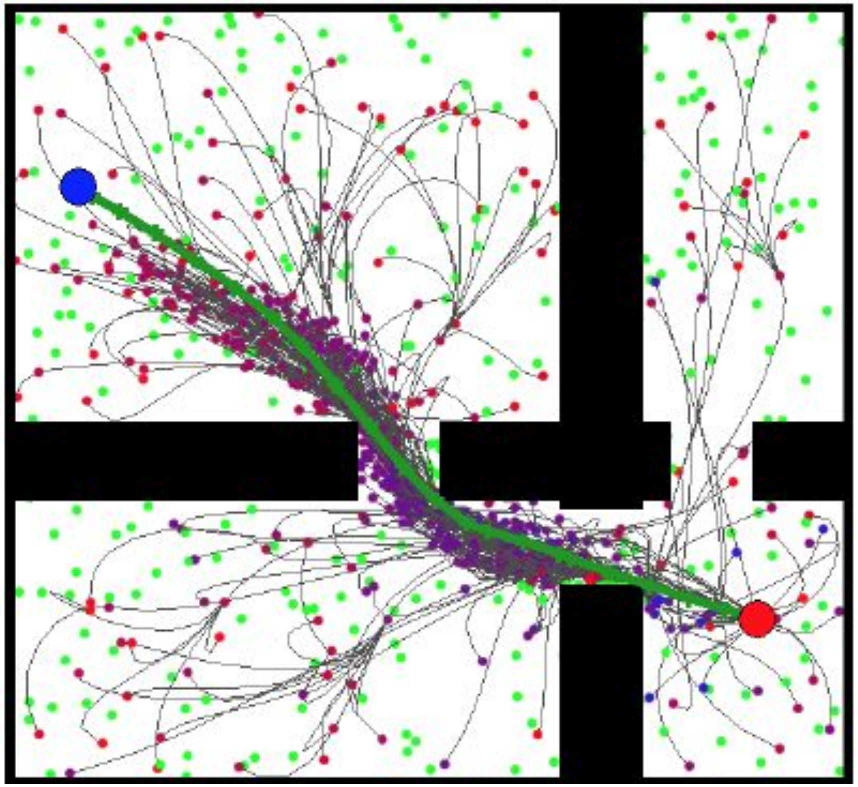
Related work:

Learning heuristics



2017 Bhardwaj, Mohak, Sanjiban Choudhury, and Sebastian Scherer use a clairvoyant oracle to **learn heuristics** for search-based planners

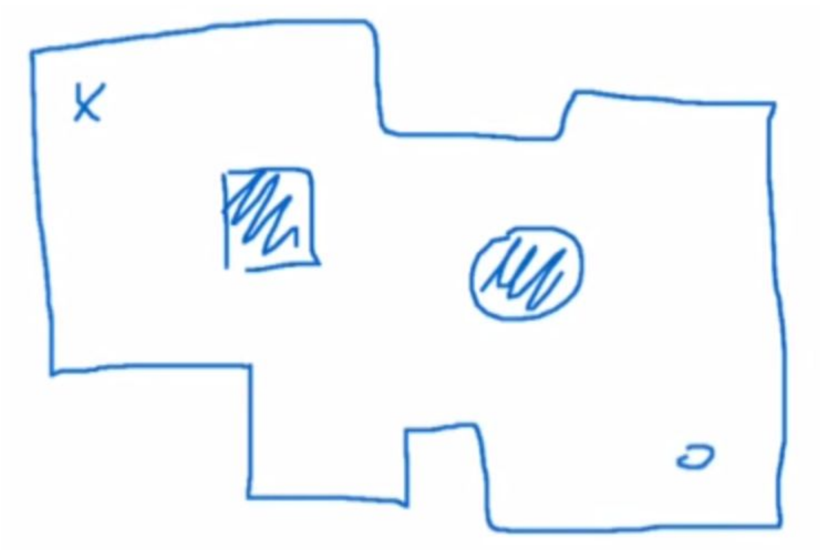
Learning distribution bias



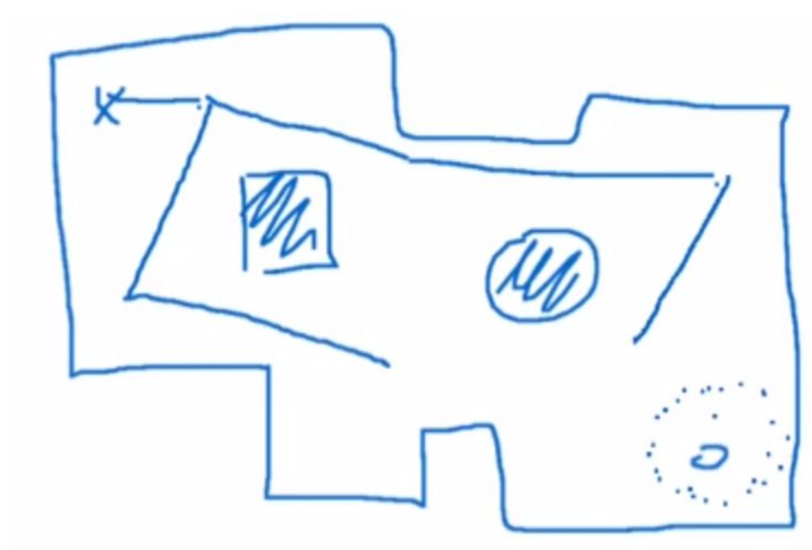
Brian Ichter, James Harrison, and Marco Pavone, "Learning Sampling Distributions for Robot Motion Planning." *arXiv preprint* 2017

Contributions

Where should I sample here?



What about now?



- Learn a **sampling distribution** using a representation **jointly** over the
  - fully-observable **environment** and
  - the **current state of the search tree**

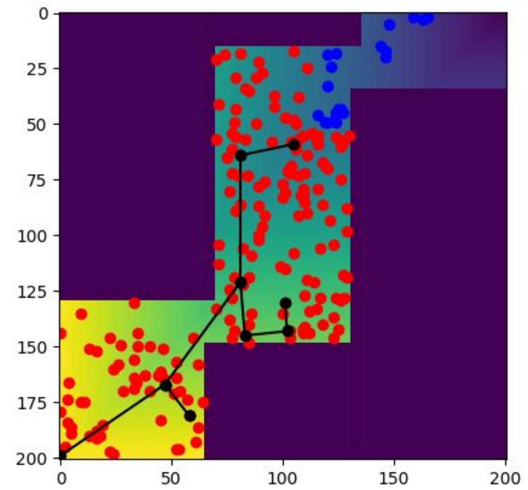
What is a good sample?

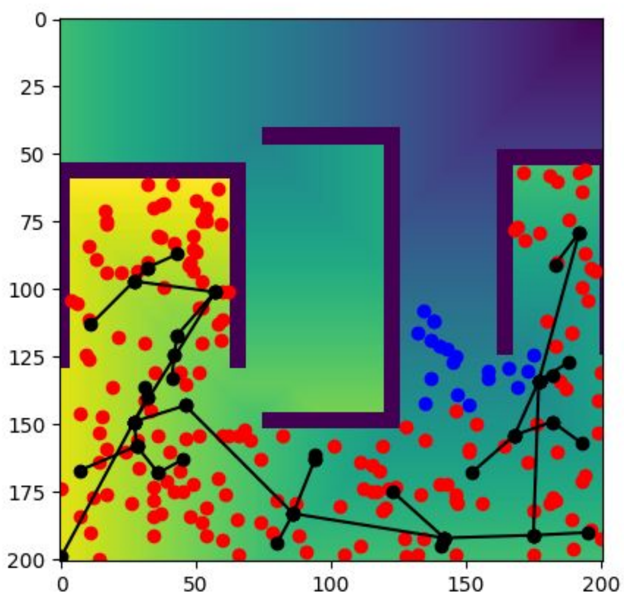
Supervised sampling

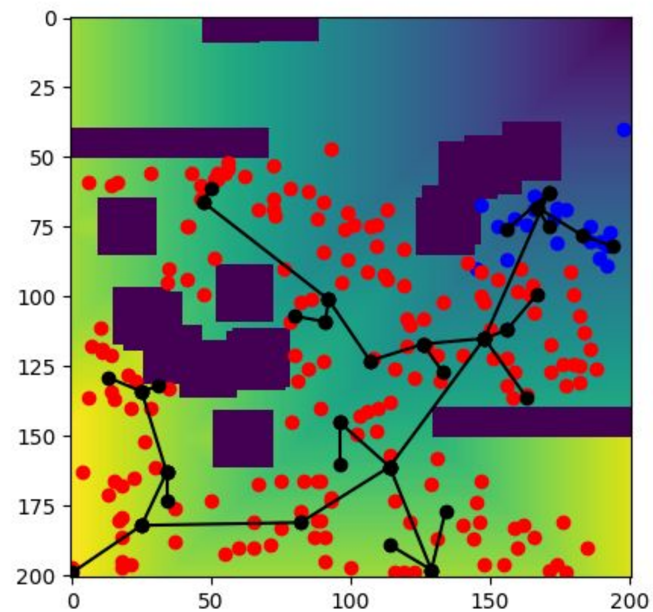
Sample N valid states

Rank by fitness

Pick best K samples

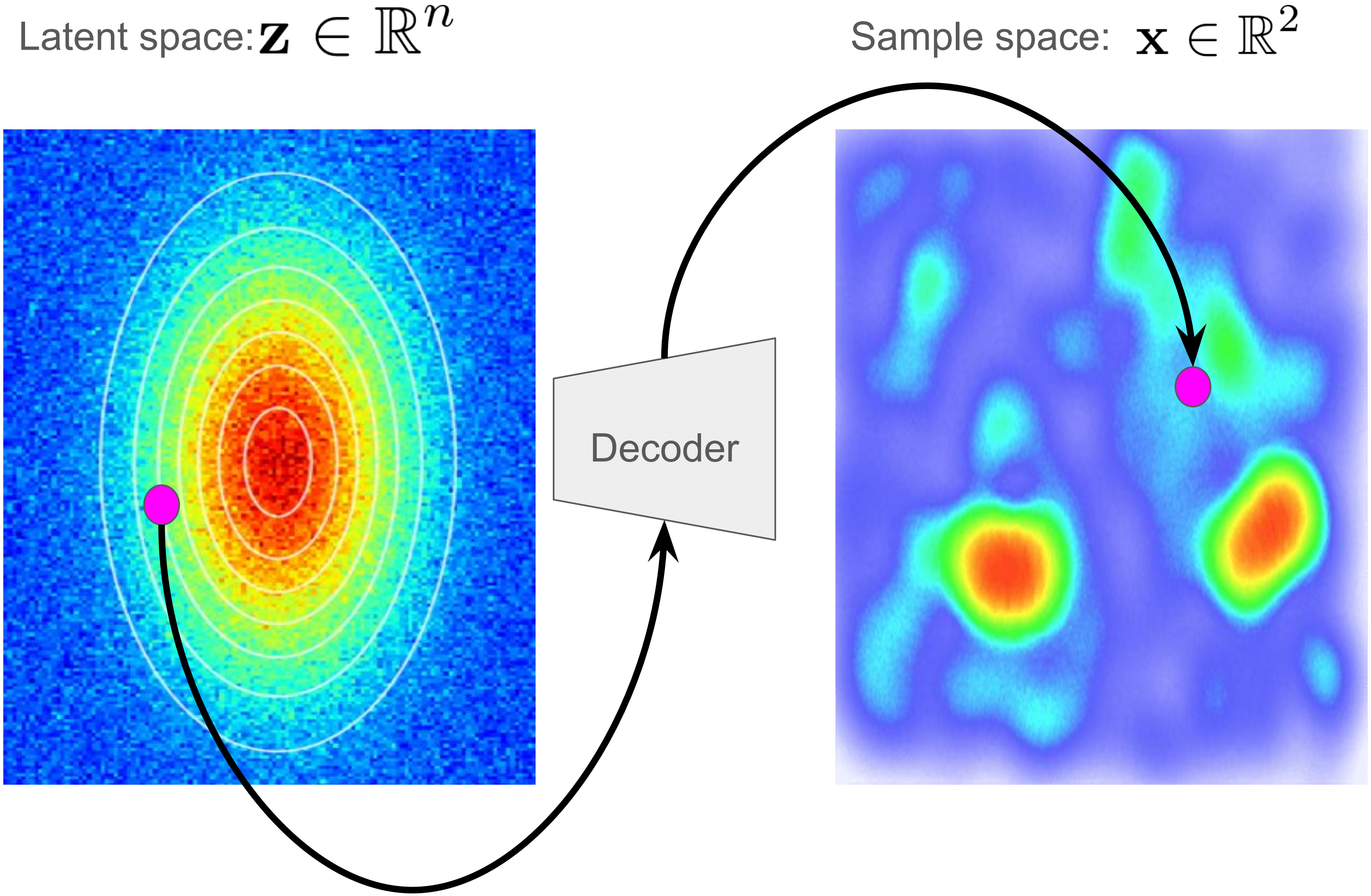




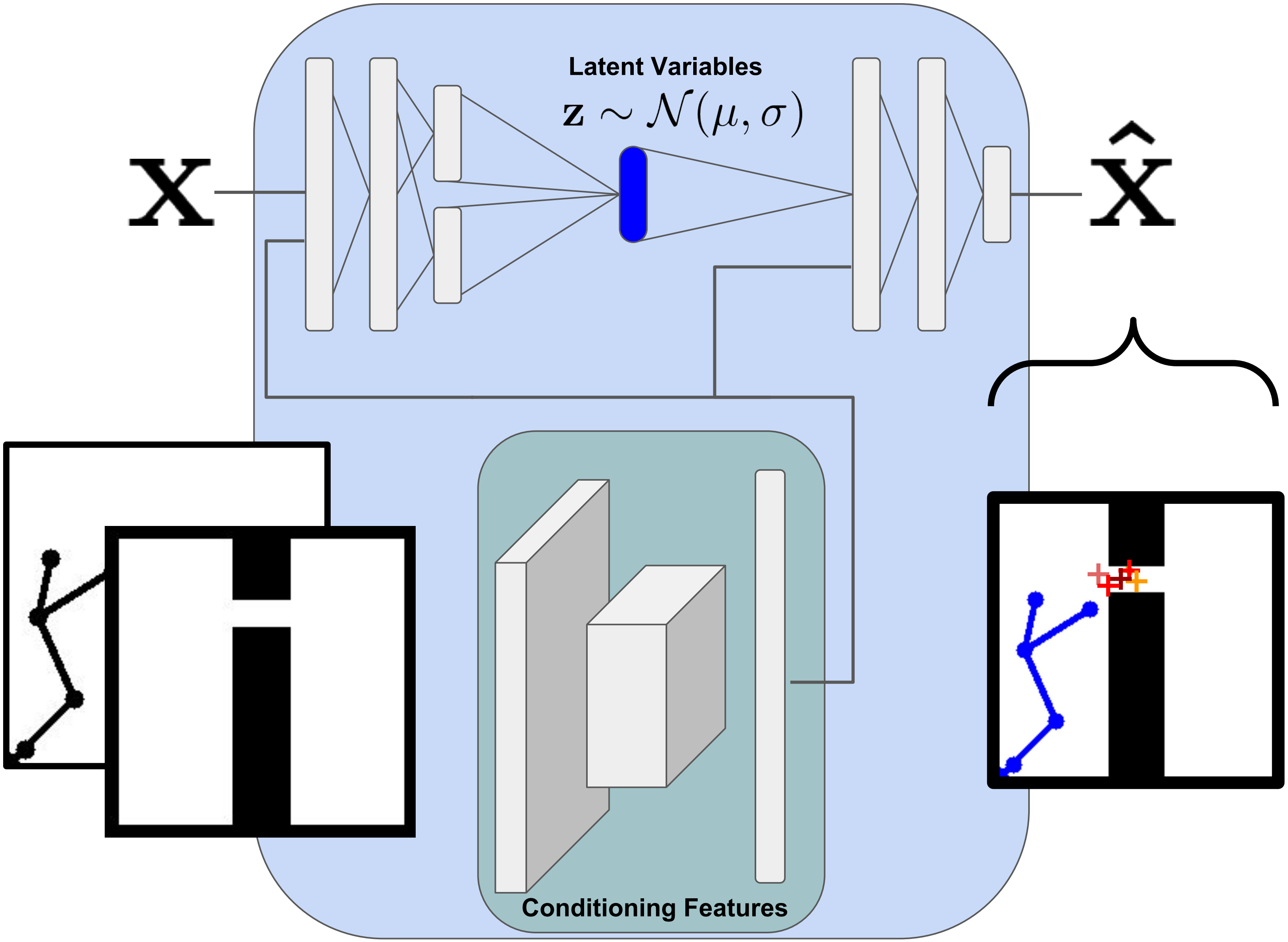


How do we mimic this distribution?

Variational auto-encoder

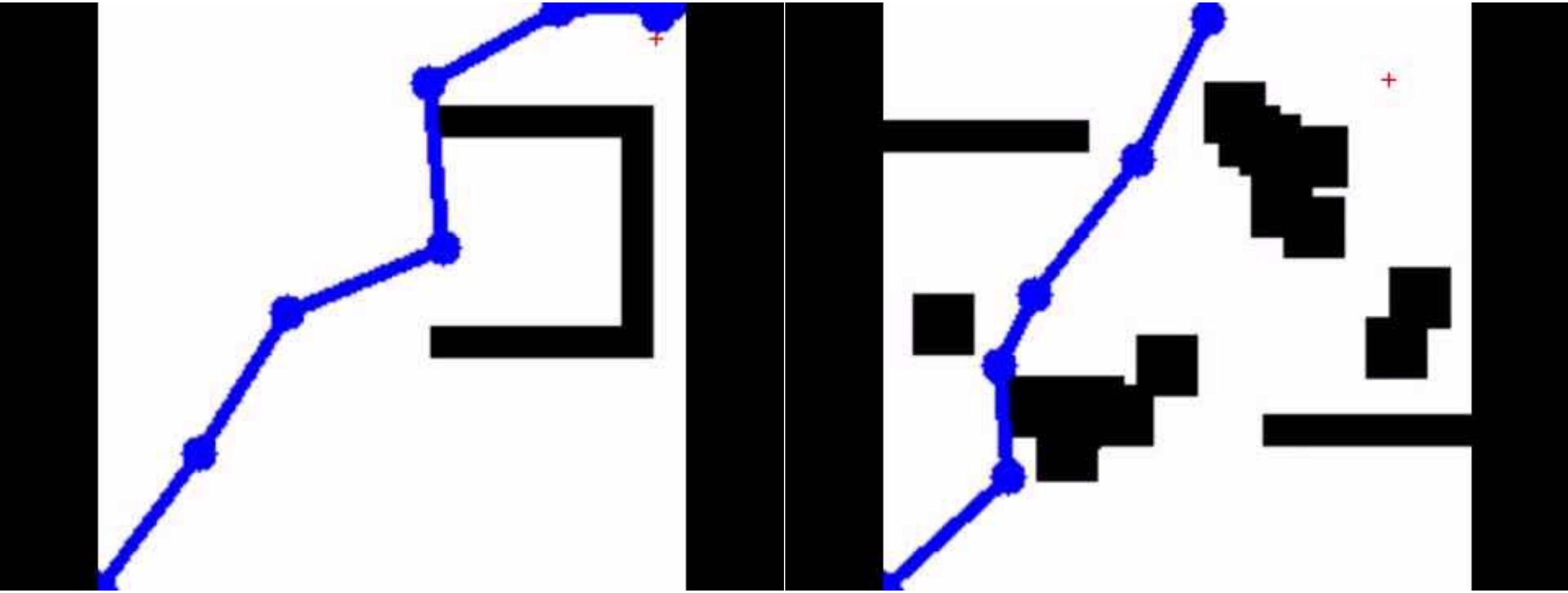


RRT Sampling CVAE Network

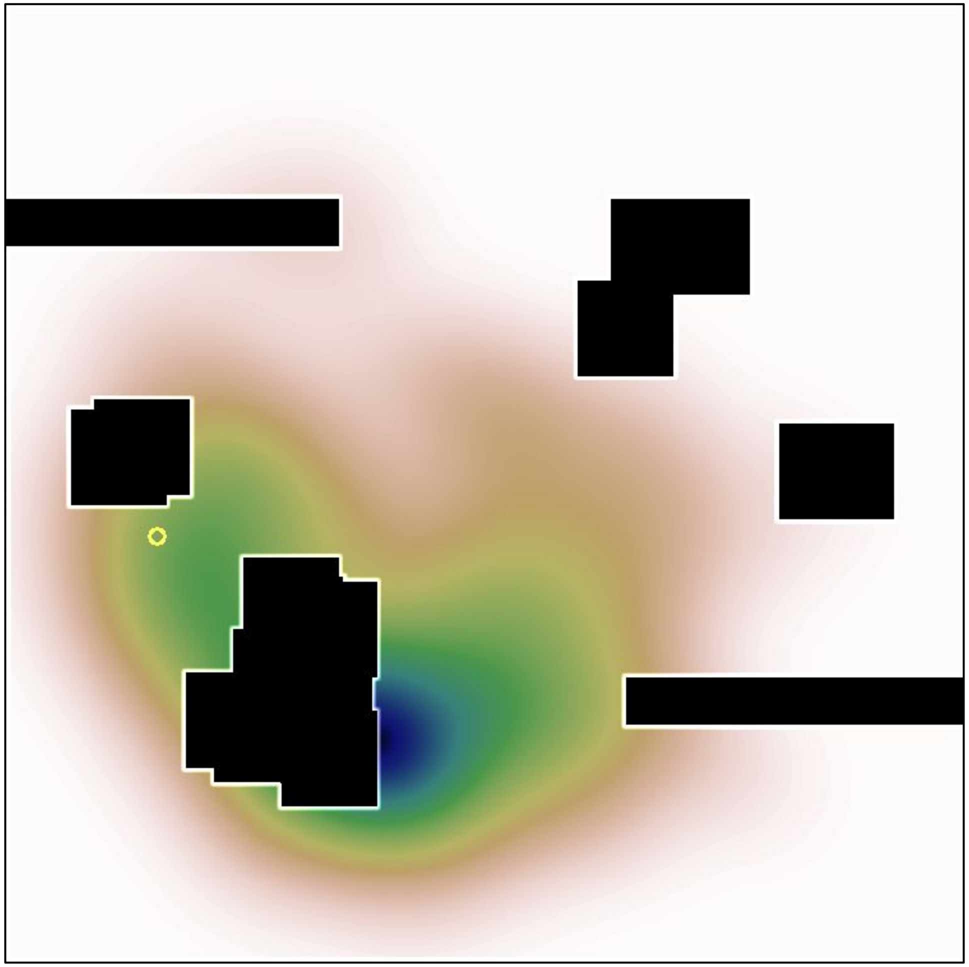


Results: supervised training

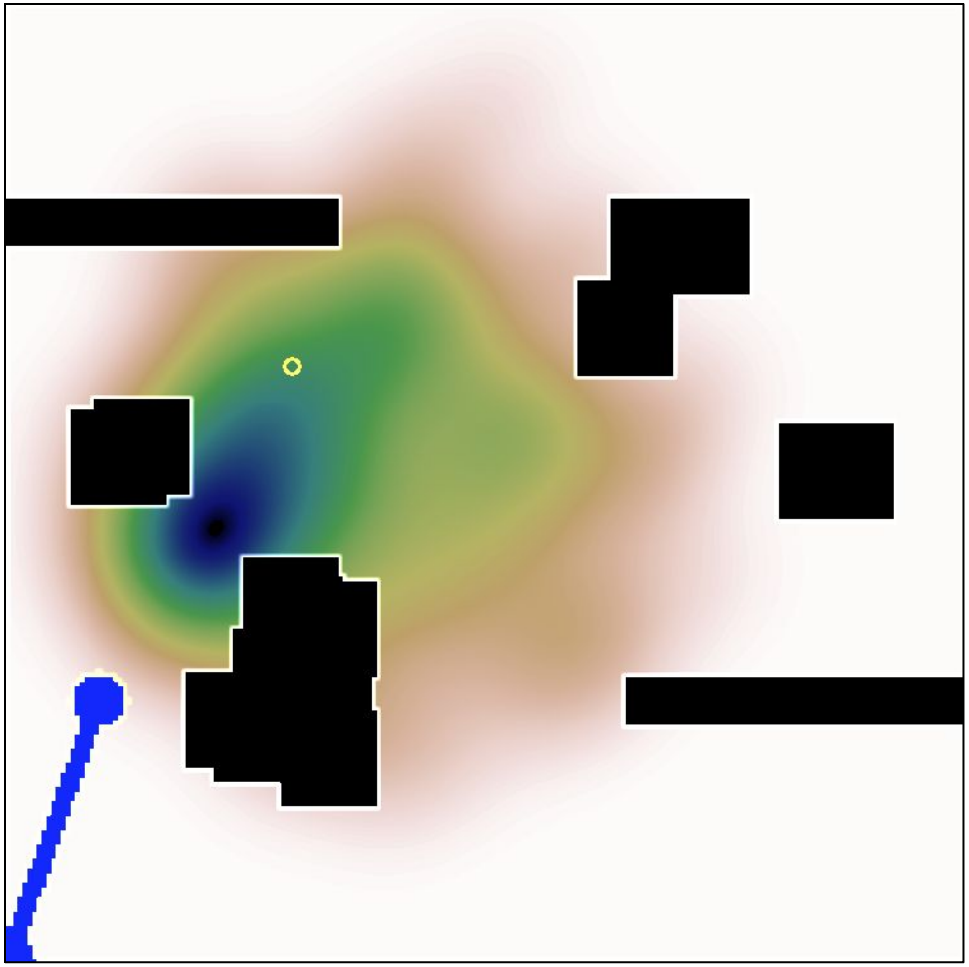
Sequence of test time



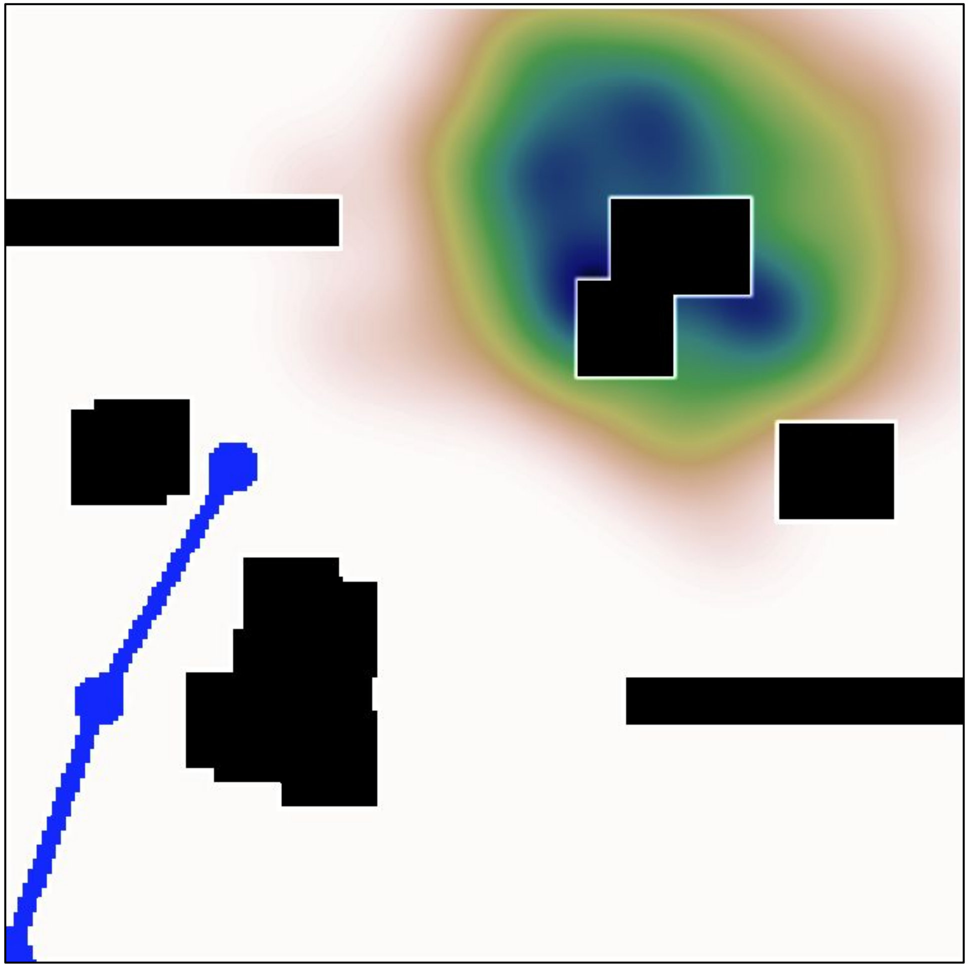
Solving the planning problem



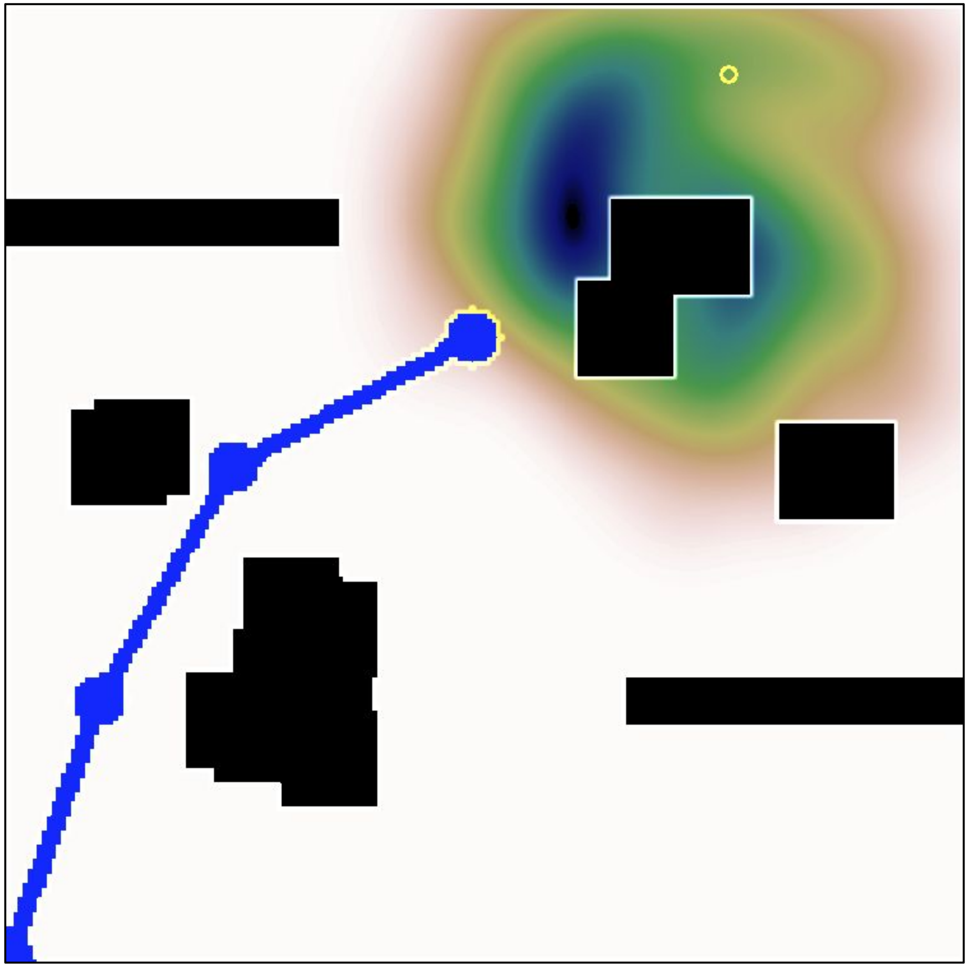
Solving the planning problem



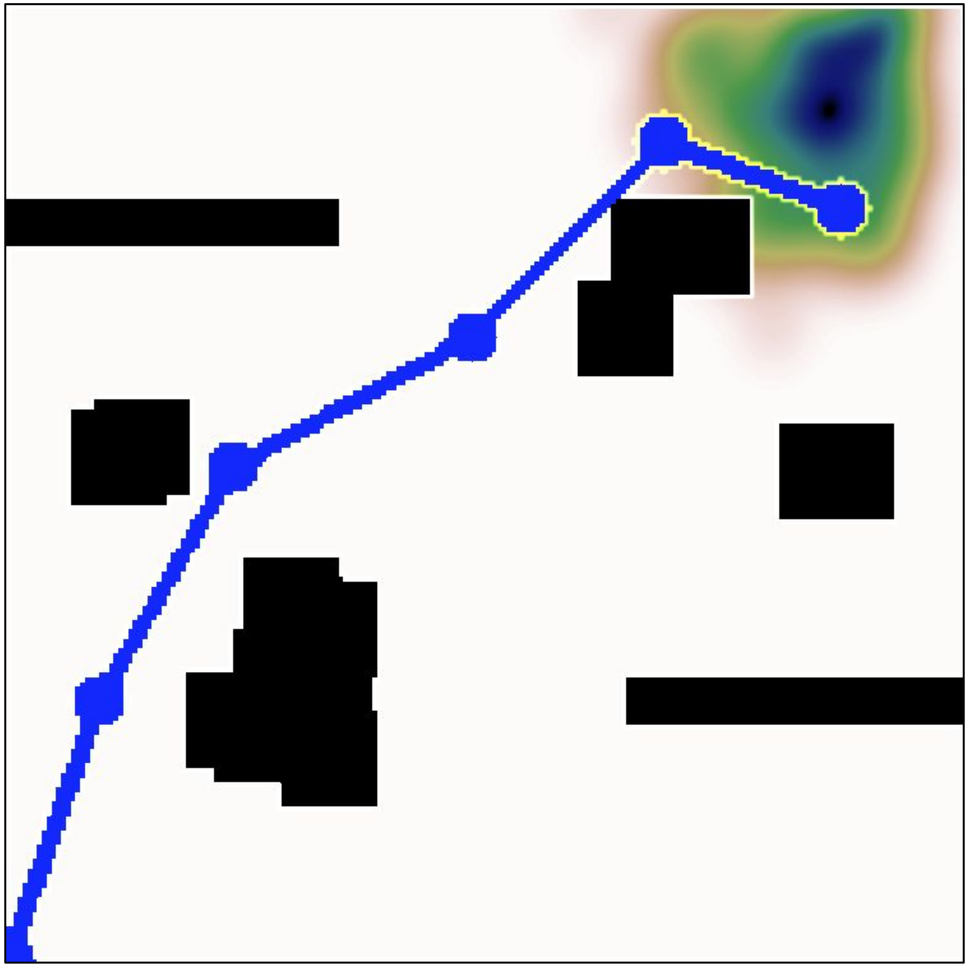
Solving the planning problem



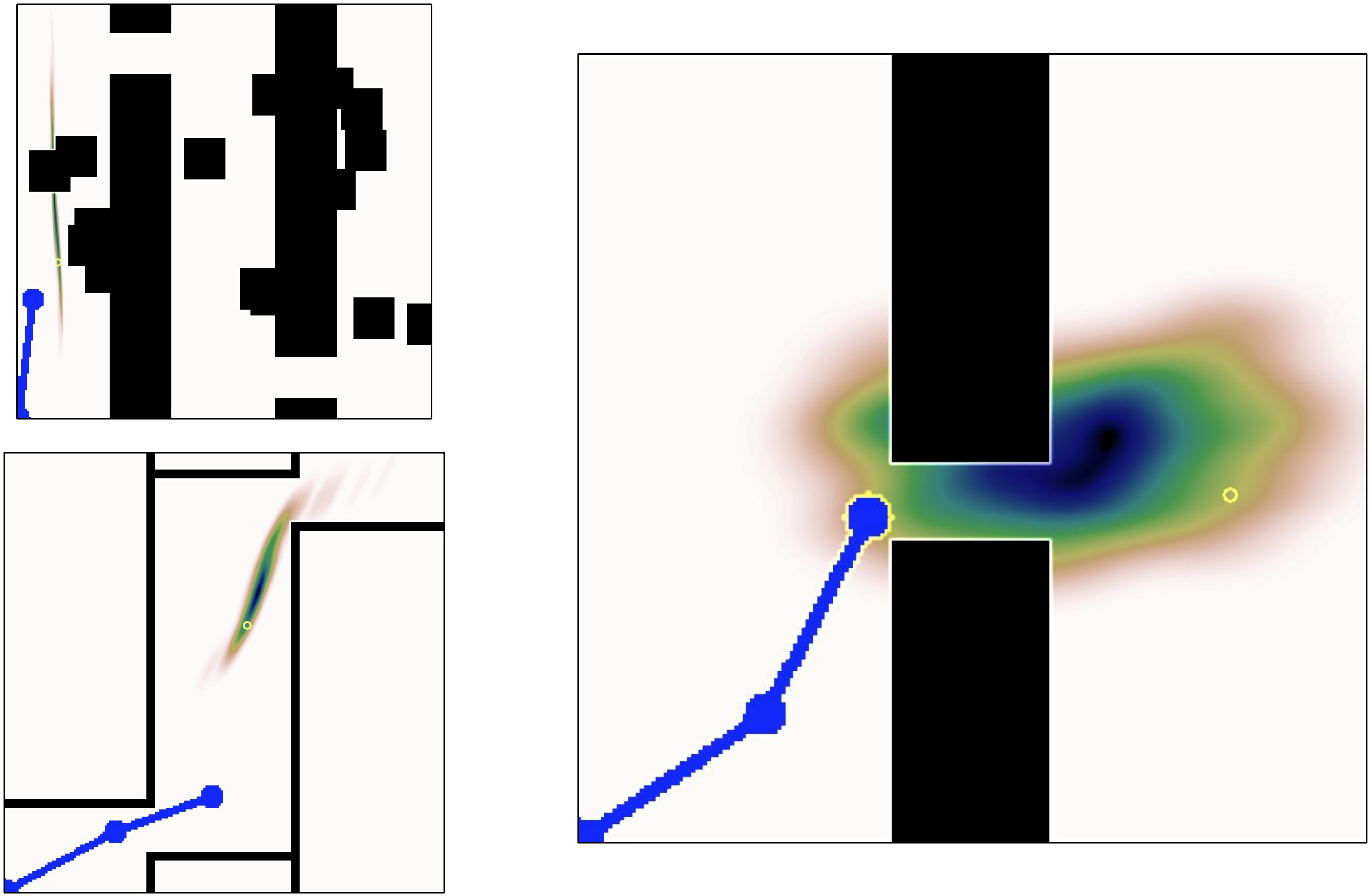
Solving the planning problem



Solving the planning problem

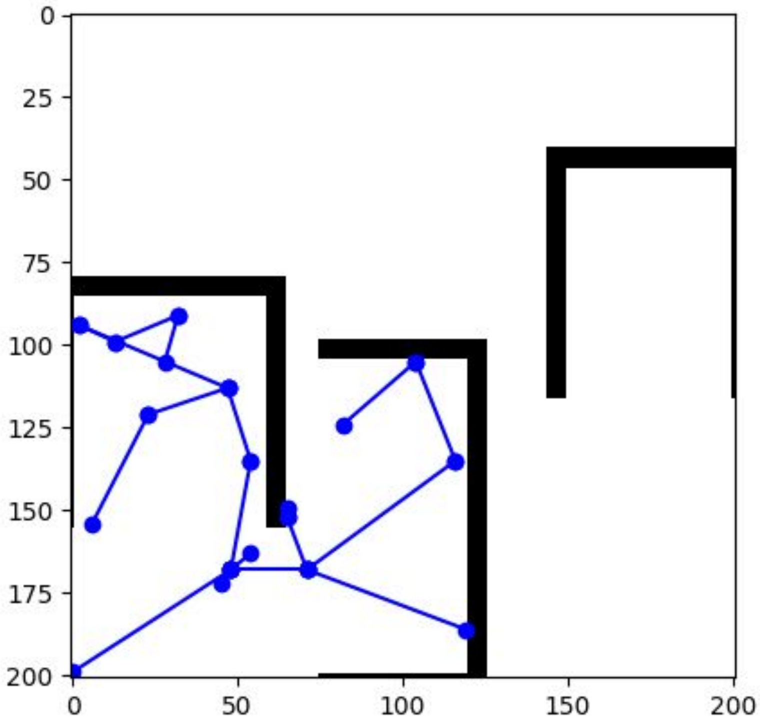


Example of distributions in other environments

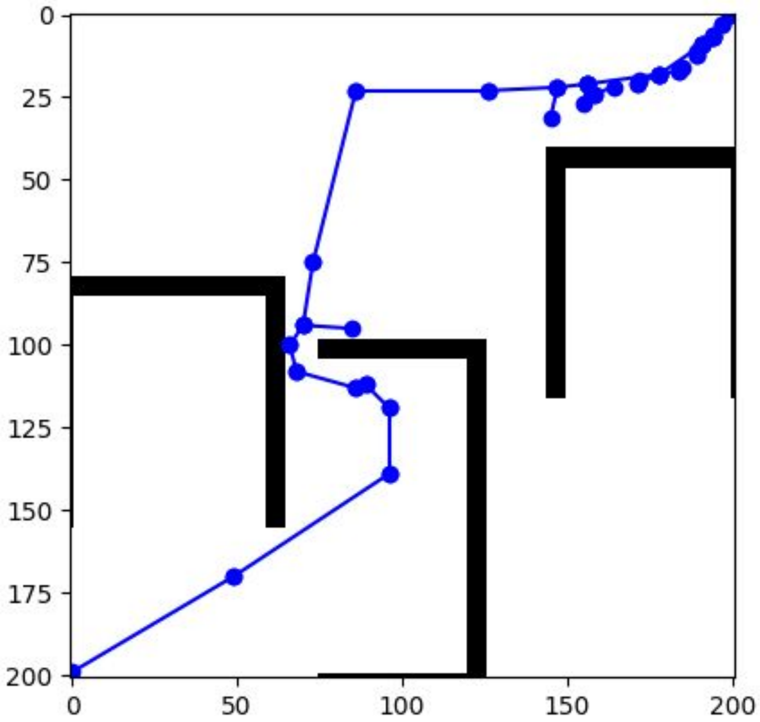


DAgger Training

RRT tree

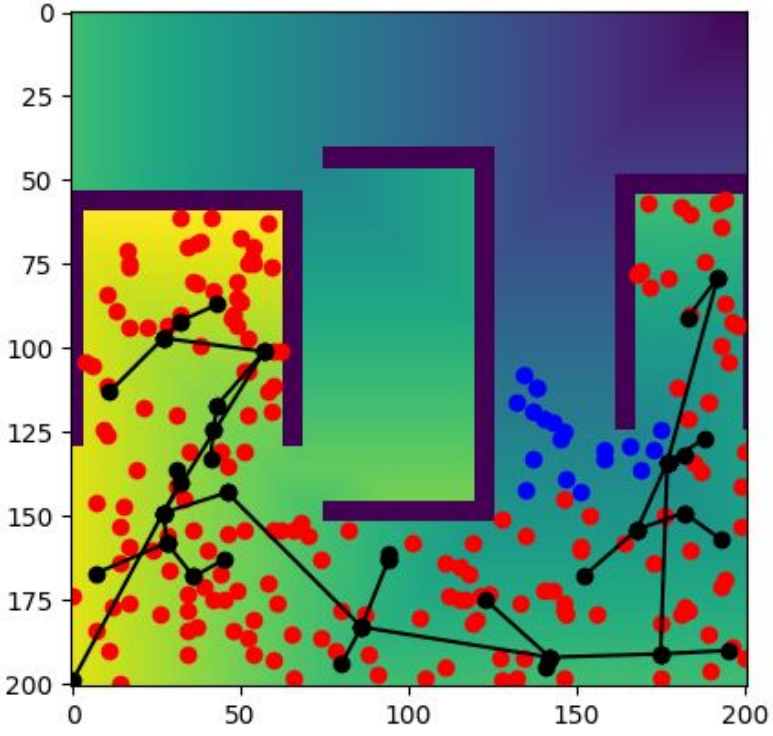


Our trees

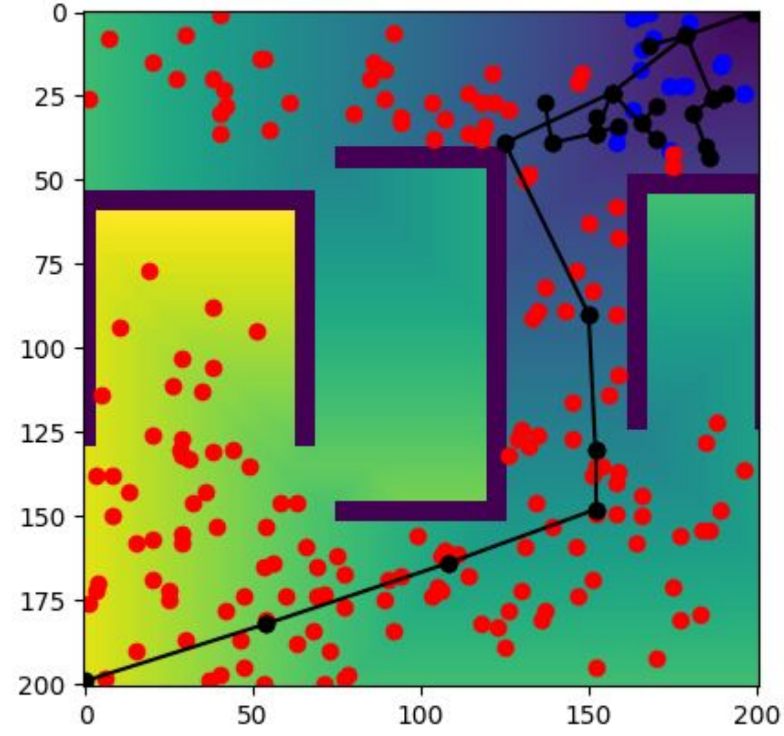


DAgger results

RRT tree



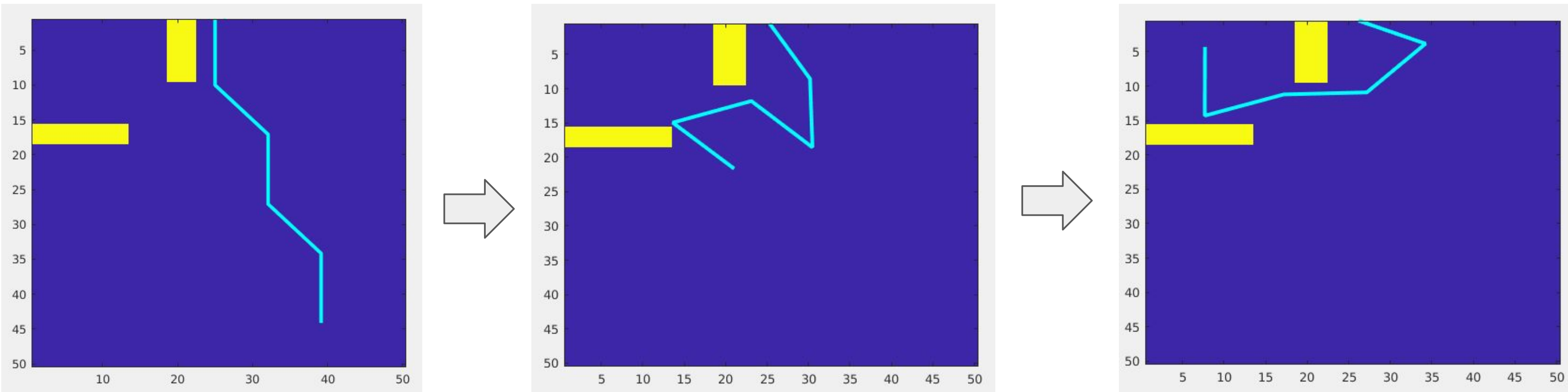
DAgger tree (7 iterations)



What about harder problems?

Extension to higher state-spaces: 5DOF manipulator

Start Samples Goal

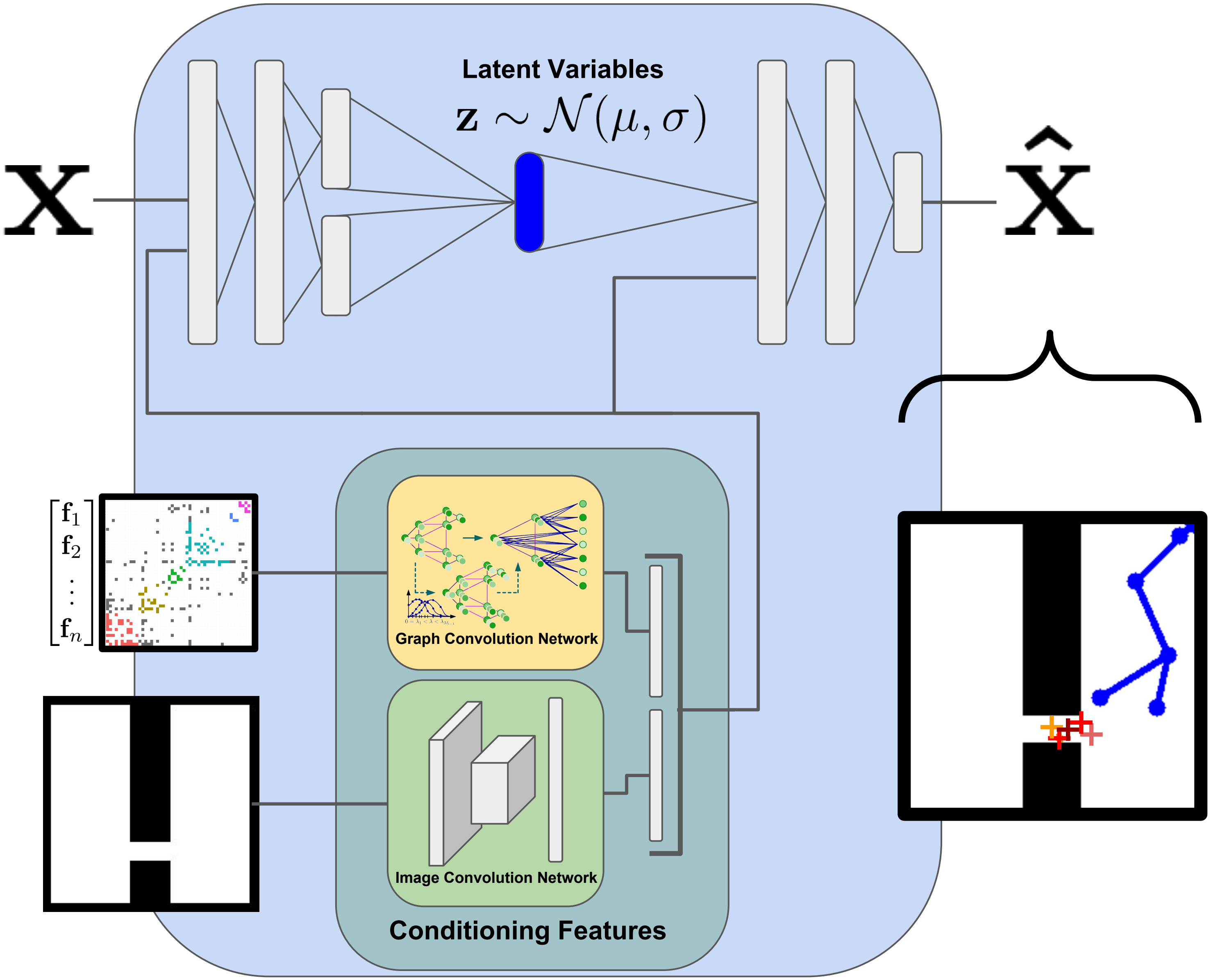


Differences with respect to current approach (2D point robot):

- Current search tree is passed to CVAE as graph instead of an image
  - Adjacency map
  - Featurization (values of each state)
- Cost-to-go function is obtained by solving an optimal planner from sample to goal

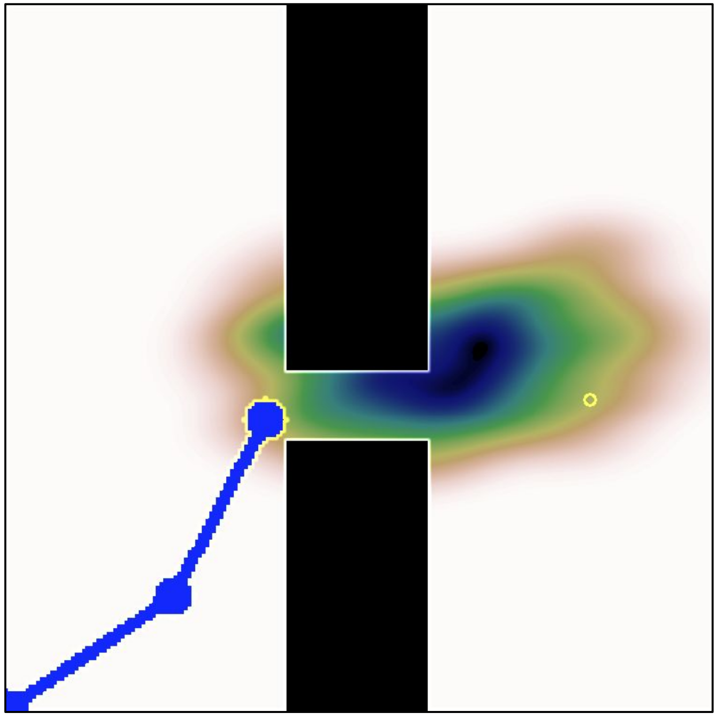
Currently being implemented in OMPL for final report

RRT Sampling With Graph CVAE Network

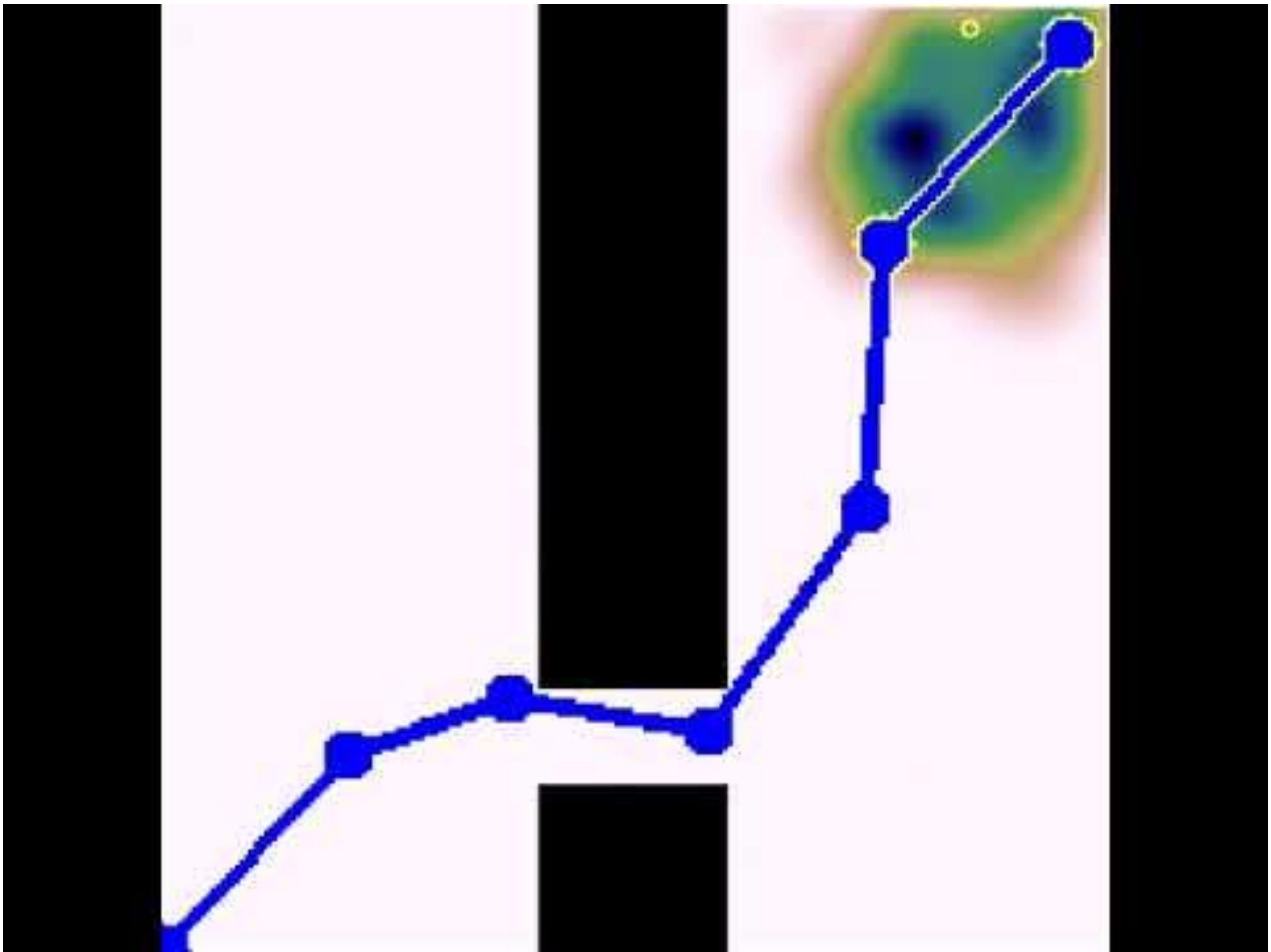


Questions

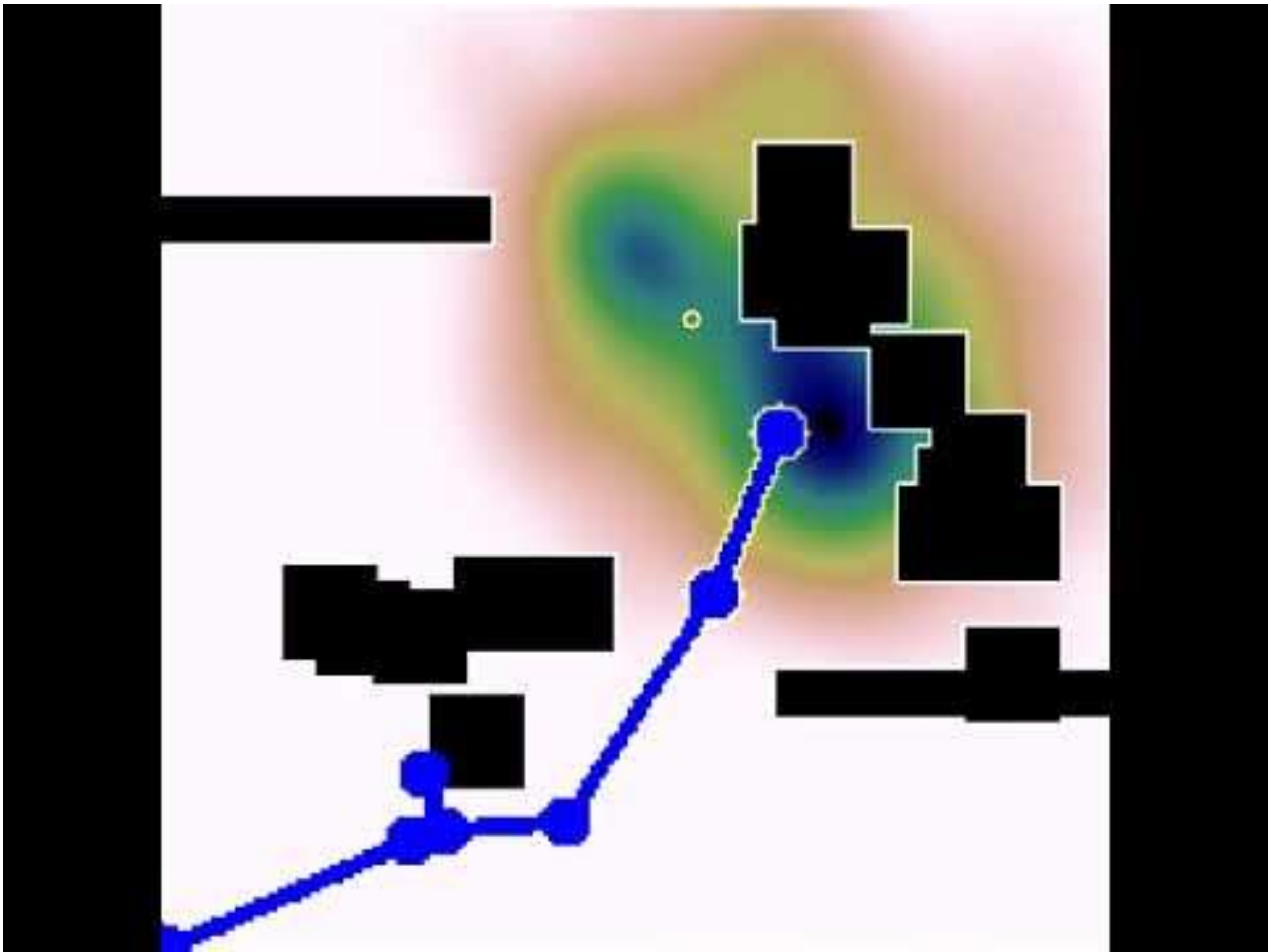
- RRT Planning
- CVAE Sampling
- Convolutional Sampler
- DAgger Training
- Higher Dimensions
- Graph Sampler



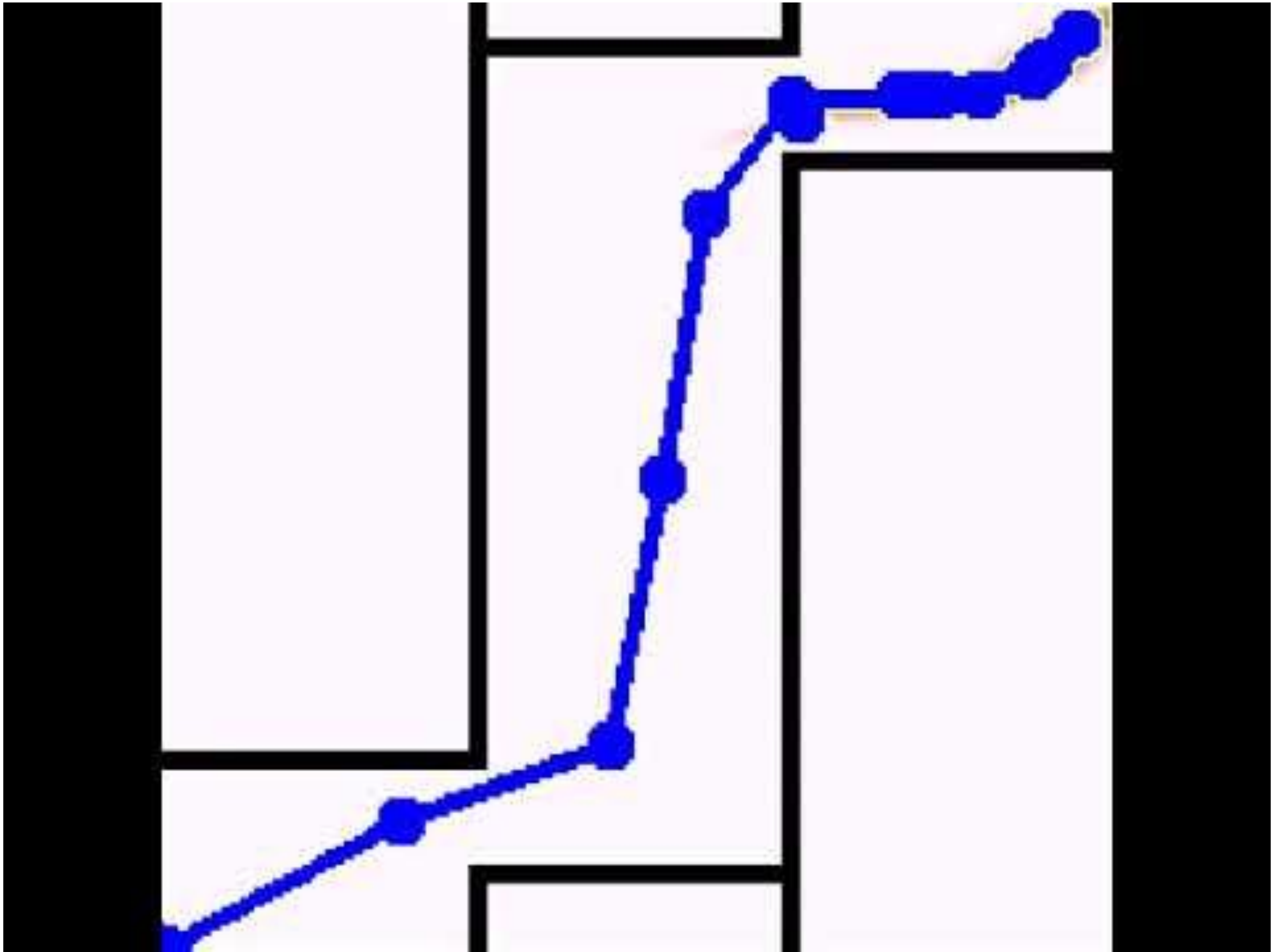
Shifting Gaps



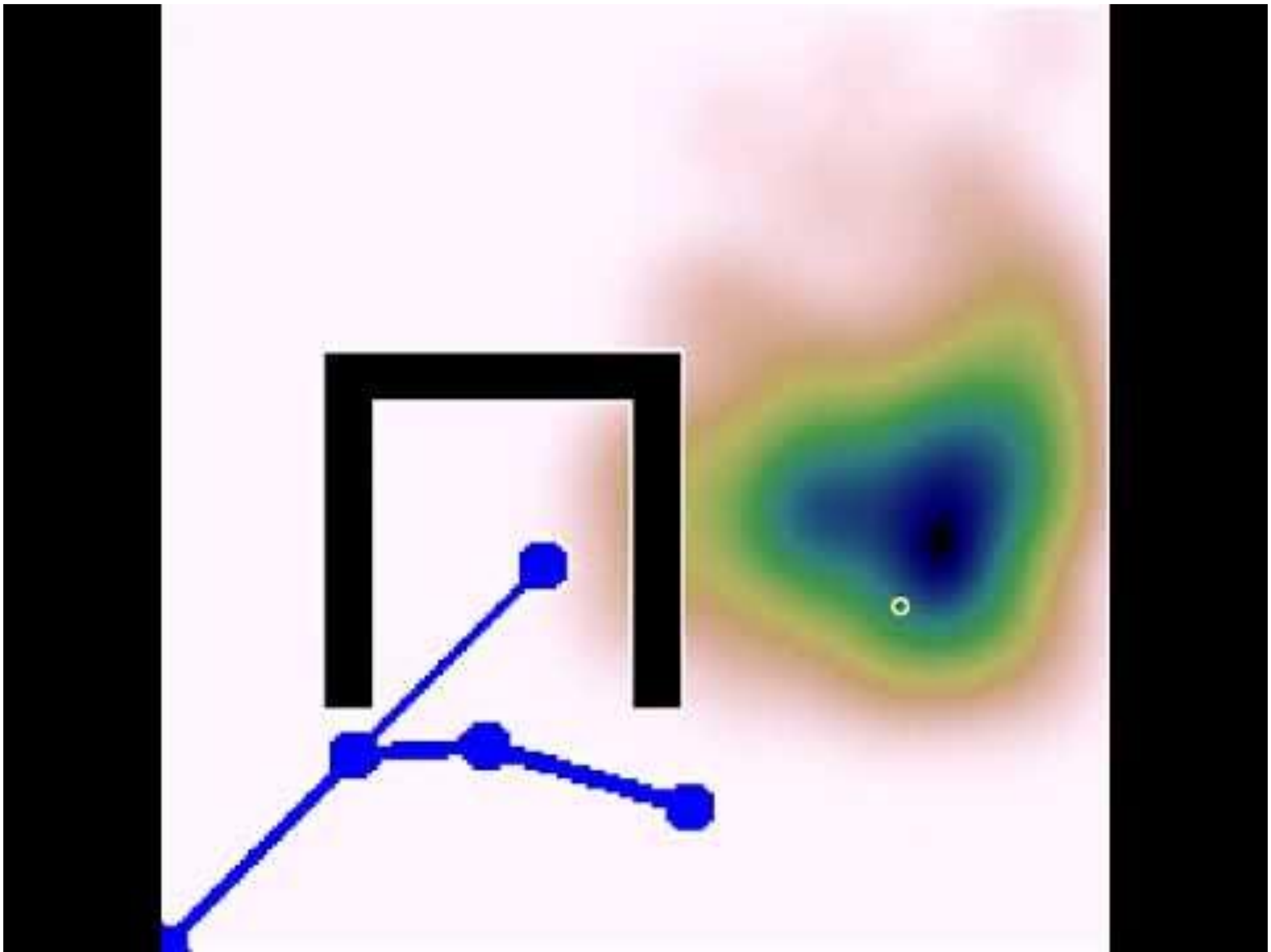
Forest



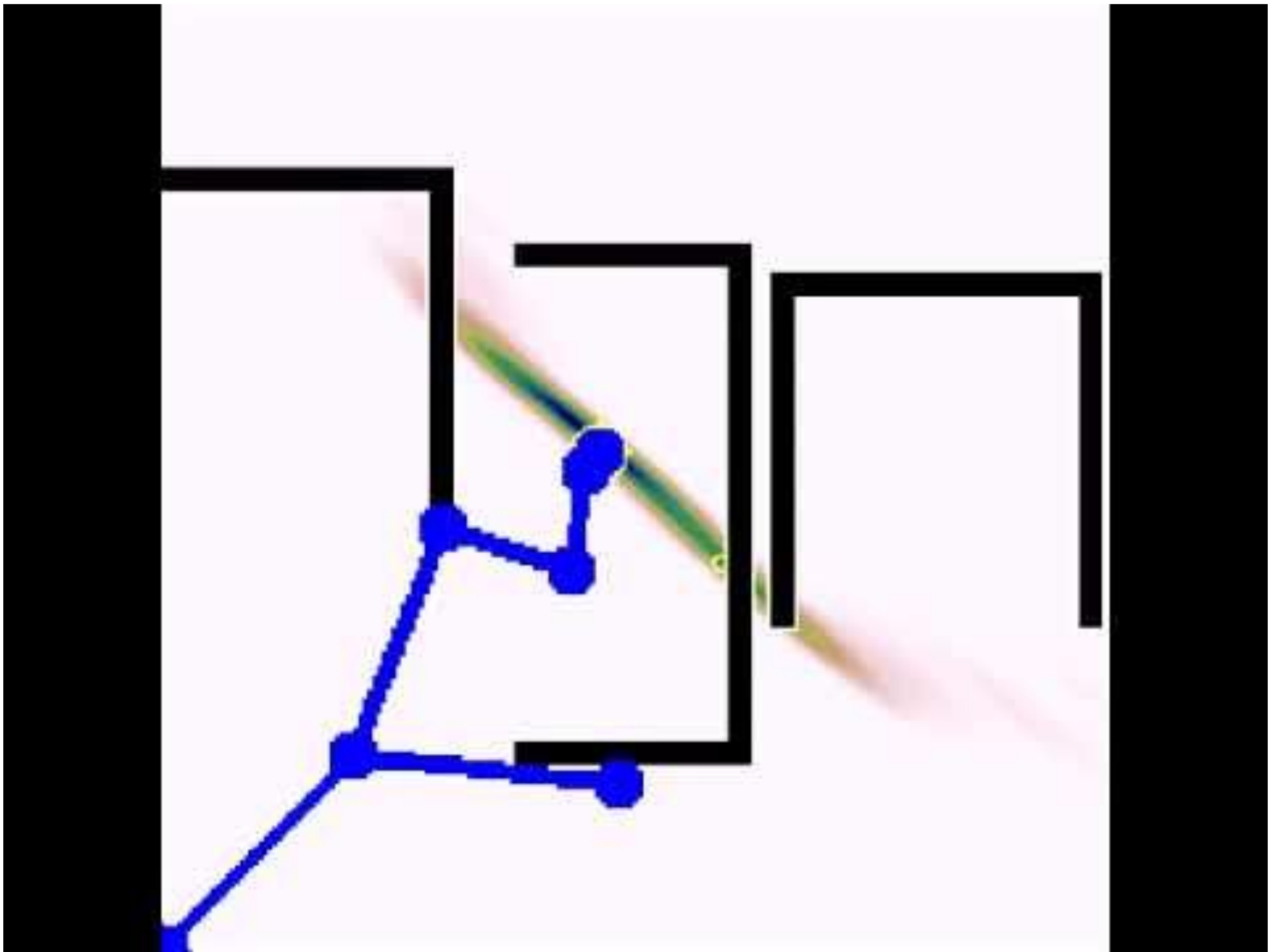
Mazes



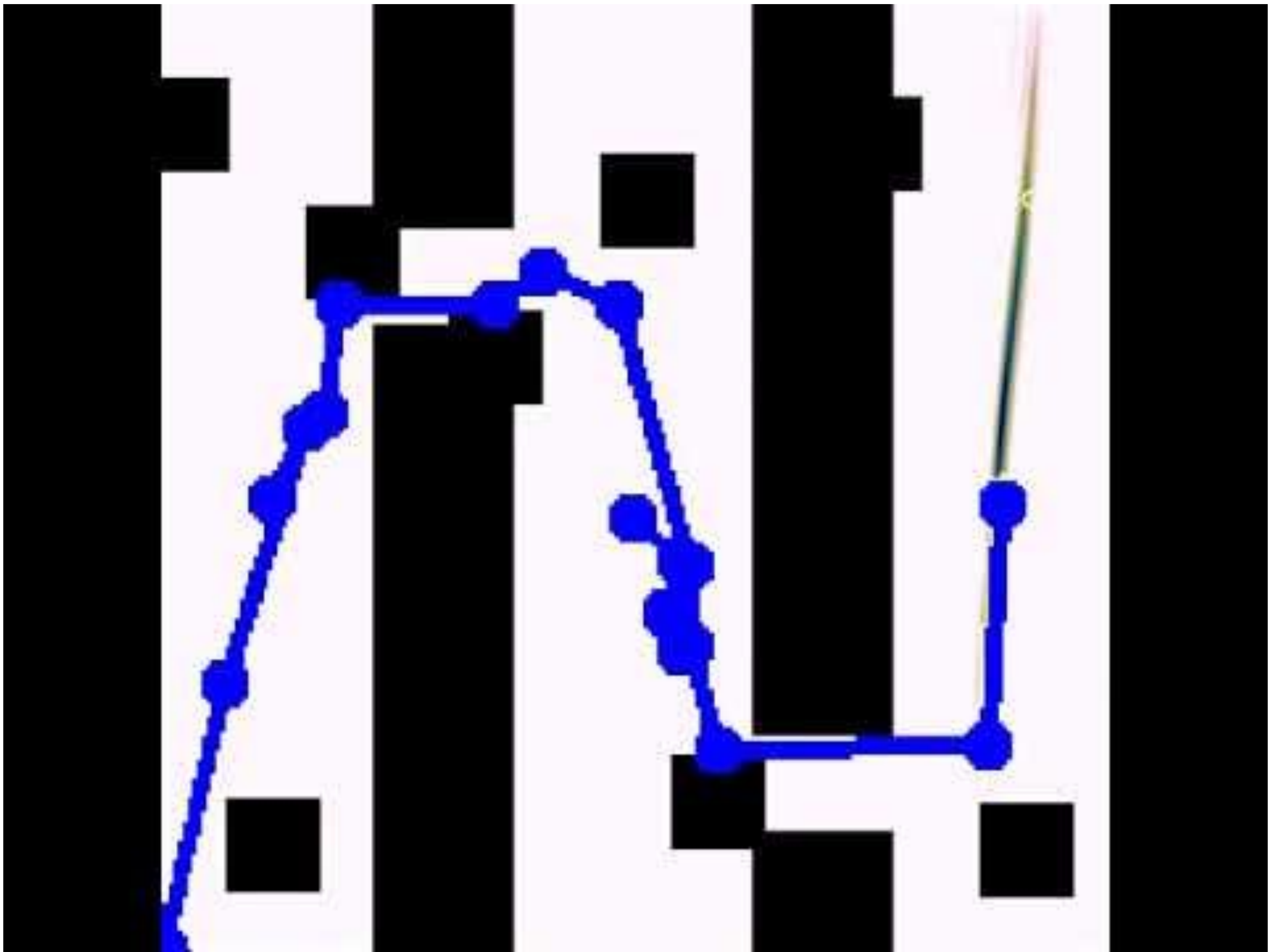
Single Bugtrap



Multiple Bugtraps



Gaps and Forest



BACKUP

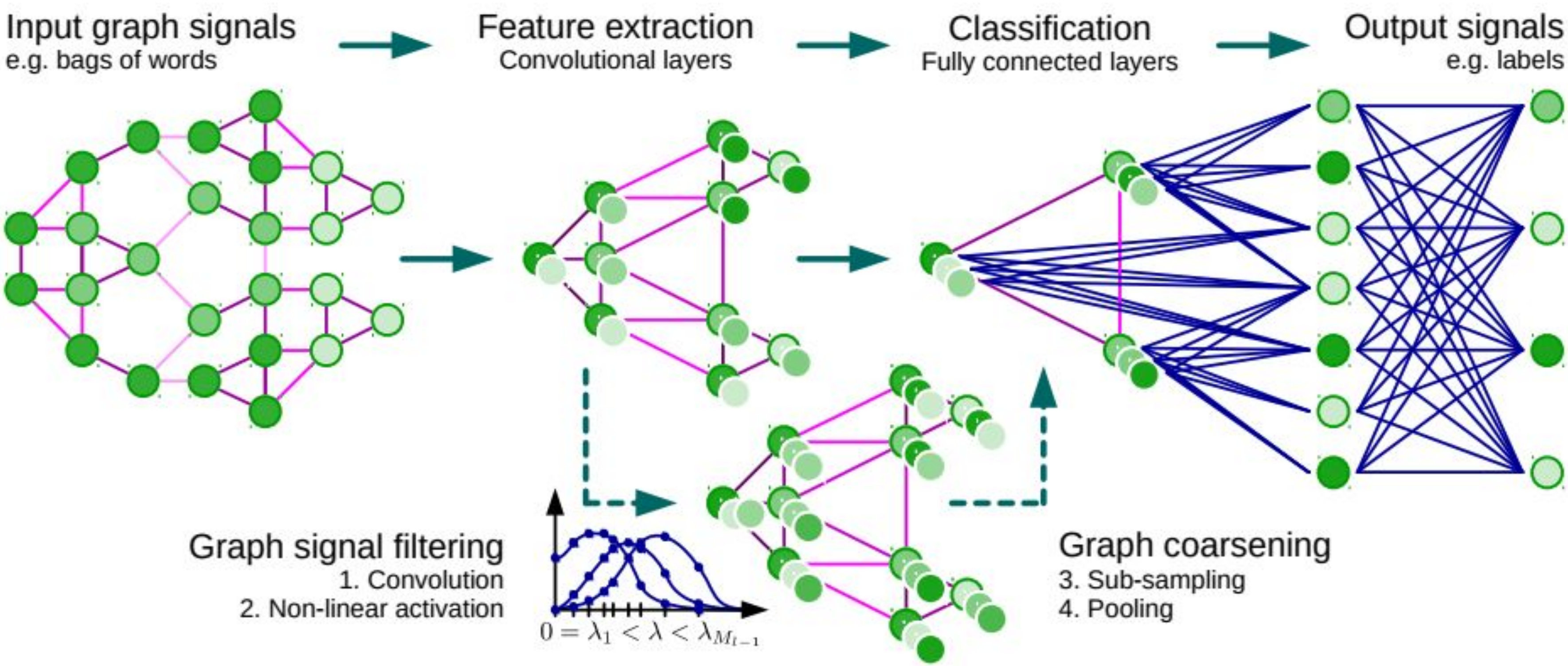
Some samples are more optimistic-in-the-face-of-partial-observability than others

- Ideally, we would want the learnt policy to take risks aka be “greedy” sometimes, depending on the planning progress, (number of (failed)) collision checks, etc.

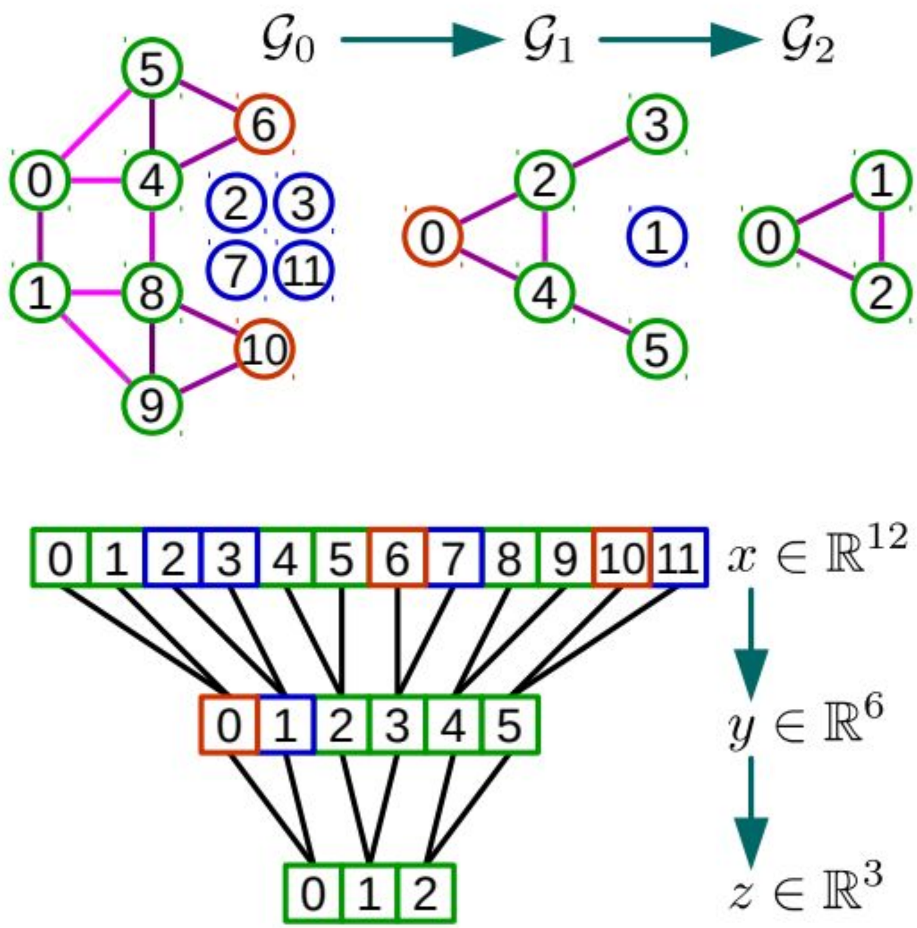
- For instance, a simple conditioning variable could be the number of collision checks failed till now. If this is high, ideally the distribution should have more density near the graph.
  On the contrary, if a lot of collisions were happening, the distribution can take risks

- A simple #gupta way of encoding this is add another channel with circles on the states that failed collision checks
  (it’s kinda weird - one channel has black for obstacles - a bad region indicator, one channel has black for the graph - sample close to this indicator, one channel has black for the failed nodes - sample away from this indicator.
  I guess we should have some color for good things, and bad things?

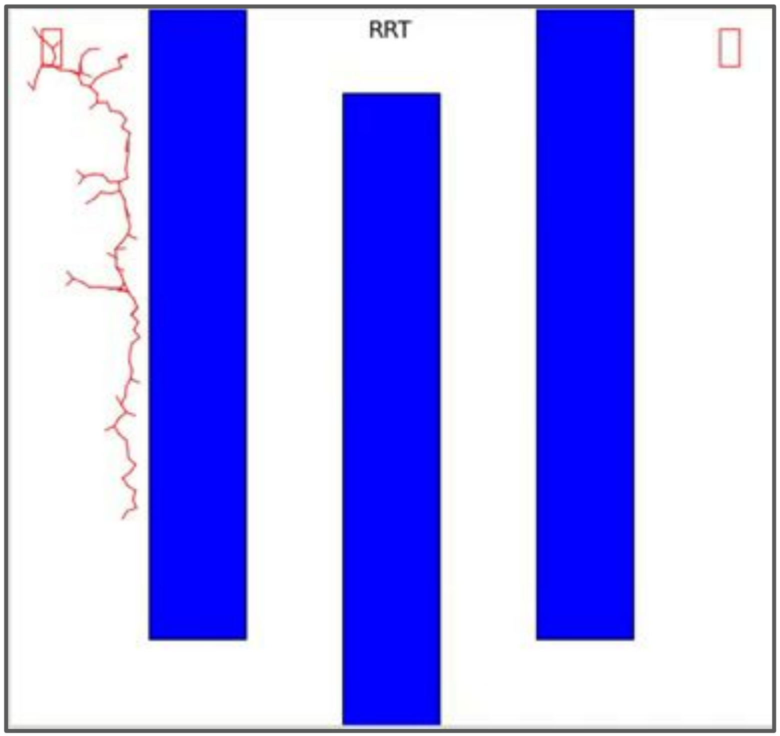
Graph-based Features: Localized Spectral Filtering



Graph Coarsening using Heavy Edge Matching



What are Rapidly-Exploring Random Trees?



RRT: The Piano-Movers Problem
https://www.youtube.com/watch?v=rPgZyq15Z-Q&