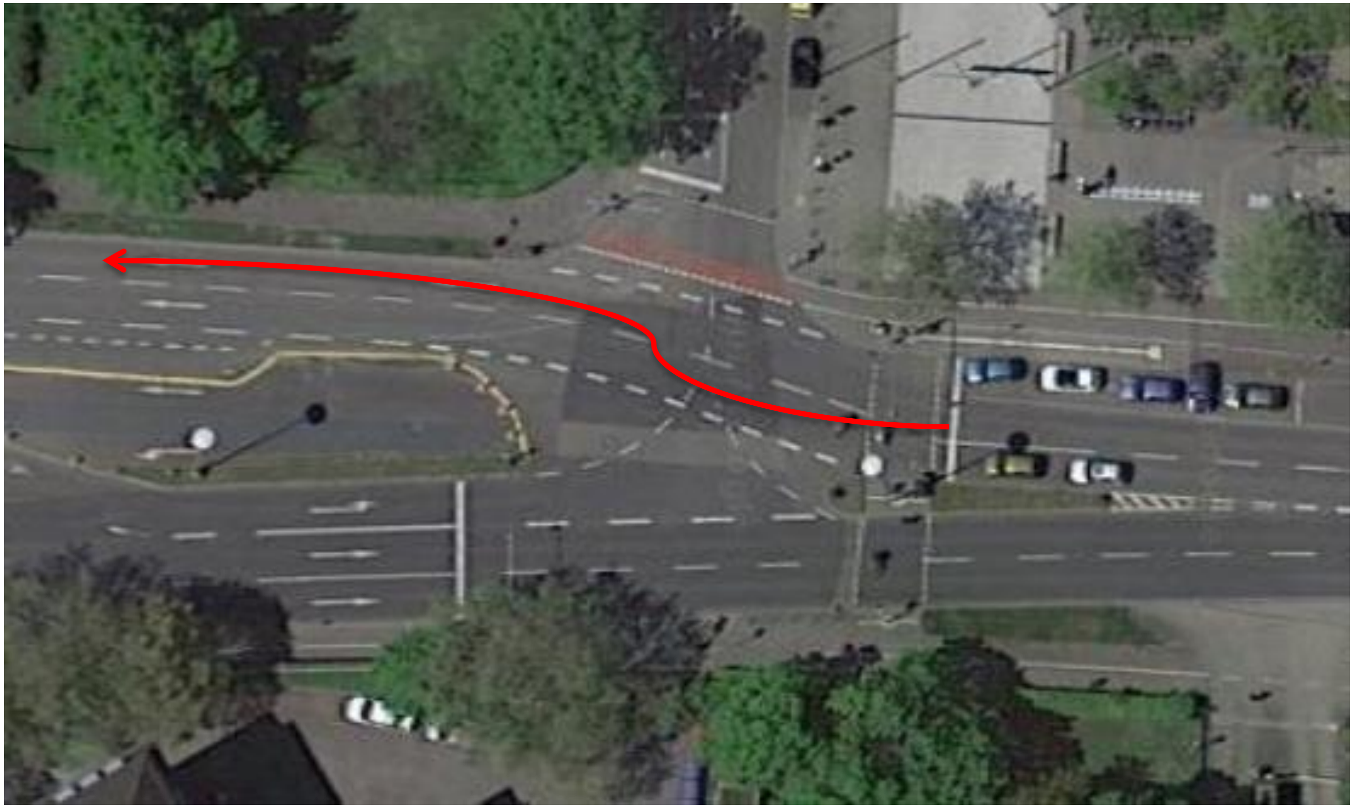
Driving Like a Human: Imitation Learning for Path Planning using CNNs

Eike Rehder, Jannik Quehl and Christoph Stiller

eike.rehder@daimler.com

IROS 2017 | 24.09.2017

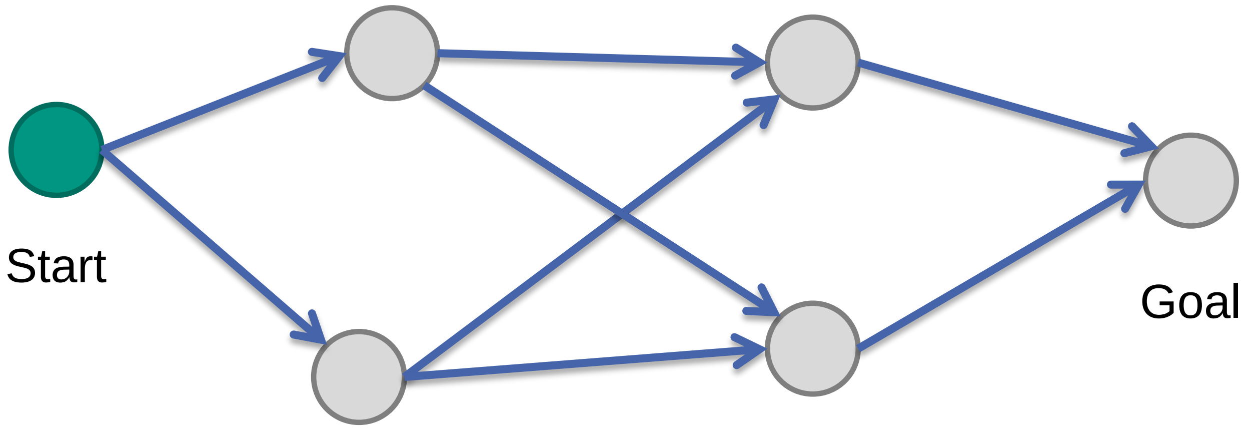
Introduction: Path Planning



Short Review:
Dijkstra‘s Algorithm

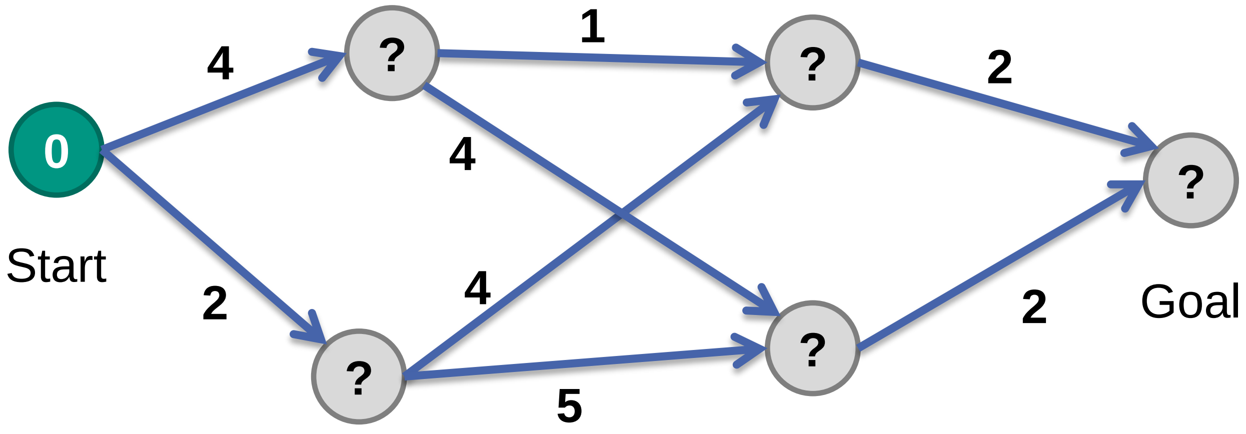
Short Review: Dijkstra‘s Algorithm

Find shortest path from start to goal:



Short Review: Dijkstra‘s Algorithm

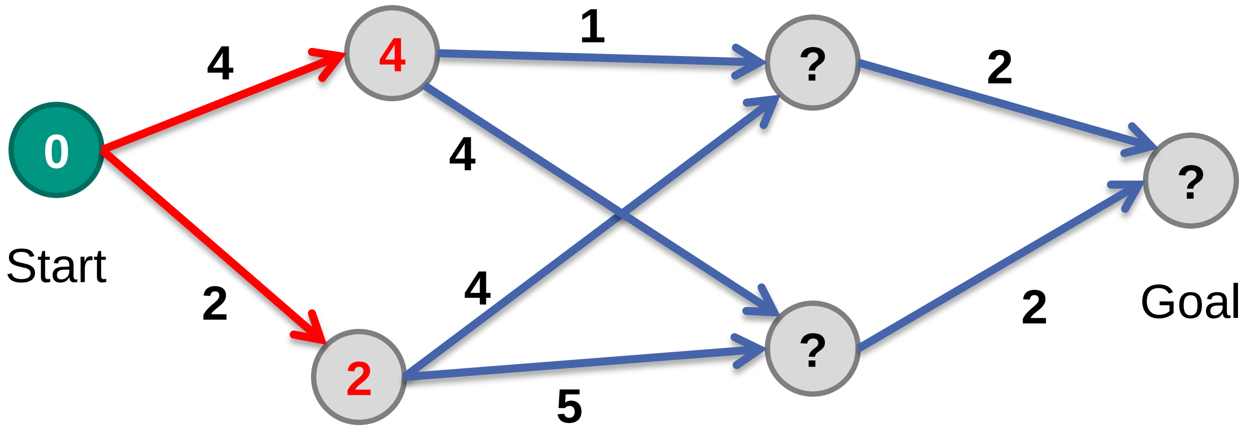
Find shortest path from start to goal:



- Assign edge costs, node costs, Start = 0

Short Review: Dijkstra‘s Algorithm

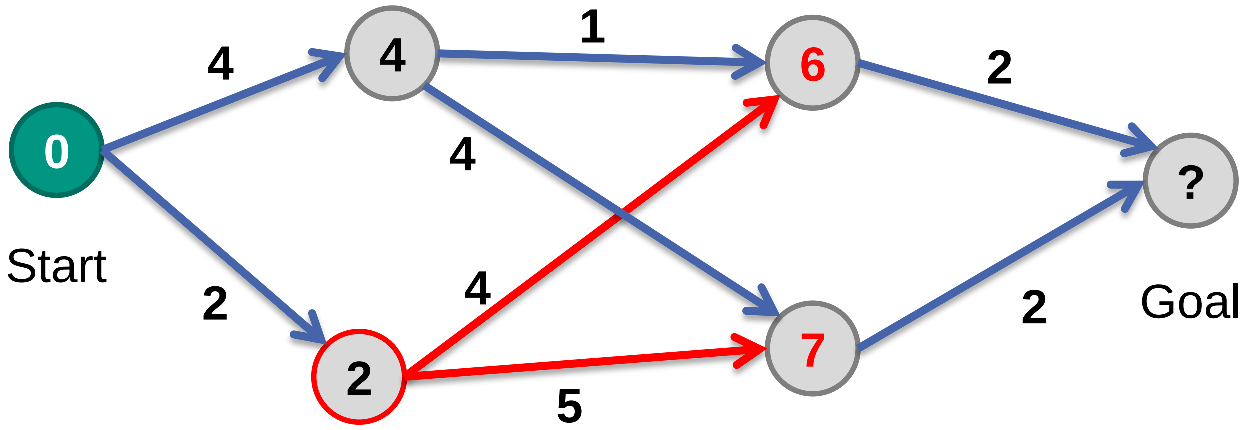
Find shortest path from start to goal:



- Assign edge costs, node costs, Start = 0
- Propagate and sum costs

Short Review: Dijkstra‘s Algorithm

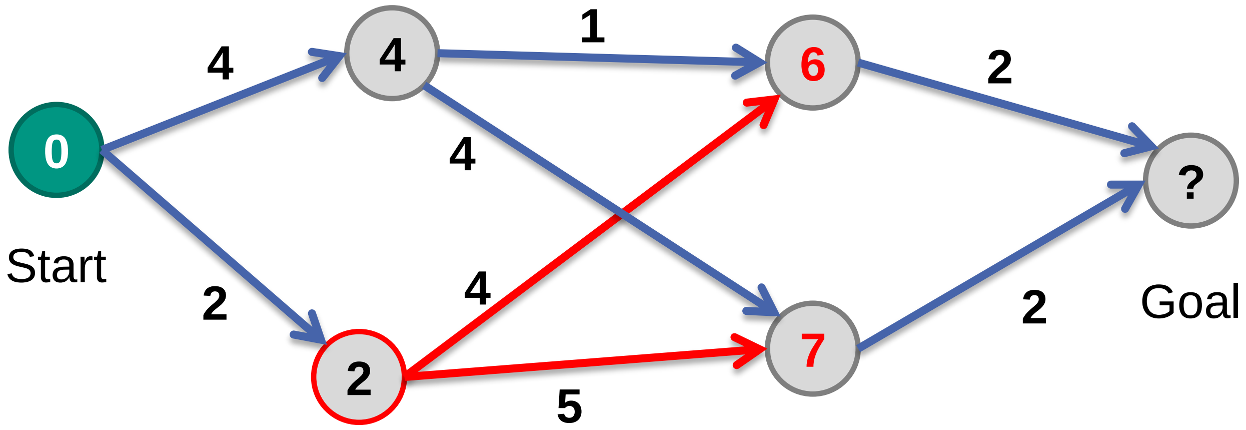
Find shortest path from start to goal:



- Assign edge costs, node costs, Start = 0
- Propagate and sum costs
- Expand cheapest node

Short Review: Dijkstra‘s Algorithm

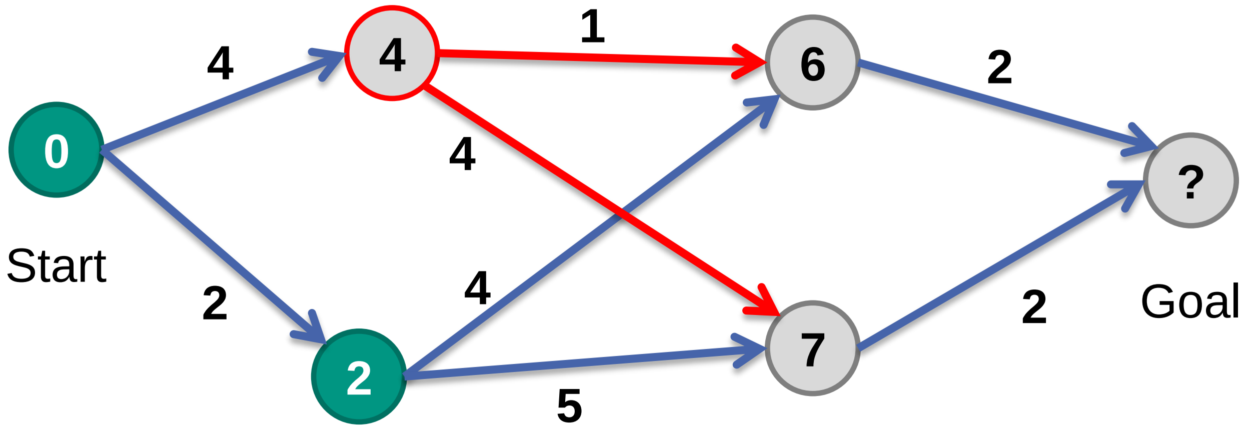
Find shortest path from start to goal:



- Assign edge costs, node costs, Start = 0
- Propagate and sum costs
- Expand cheapest node

Short Review: Dijkstra‘s Algorithm

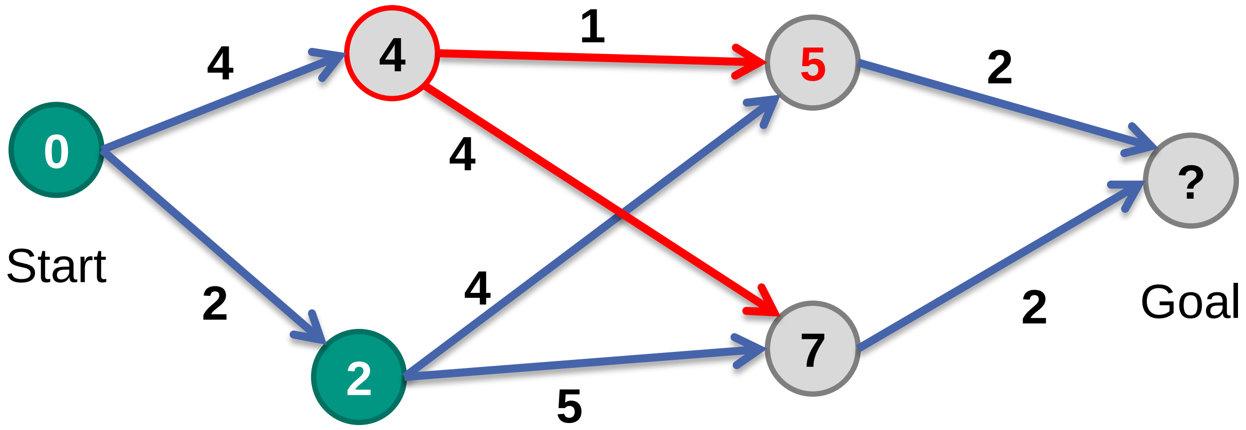
Find shortest path from start to goal:



- Assign edge costs, node costs, Start = 0
- Propagate and sum costs
- Expand cheapest node

Short Review: Dijkstra‘s Algorithm

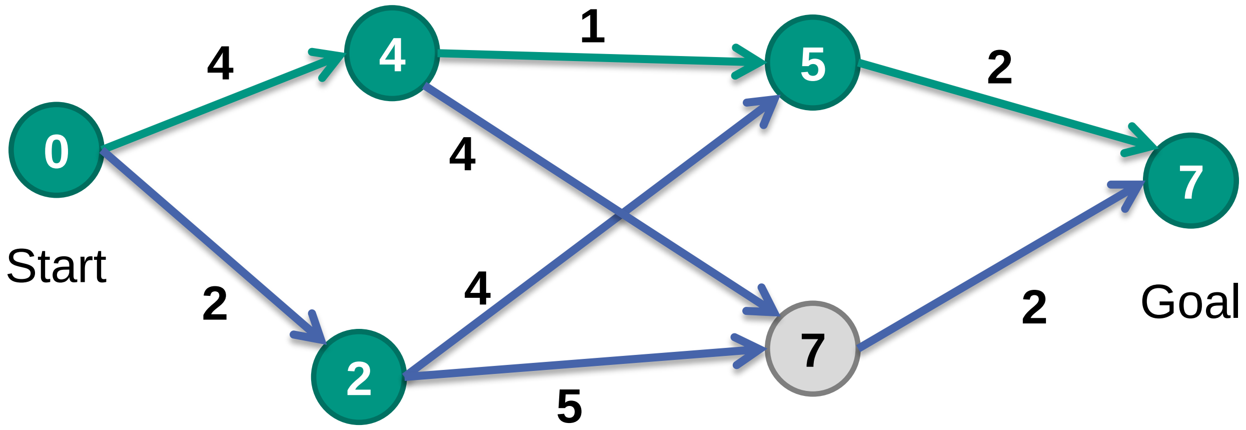
Find shortest path from start to goal:



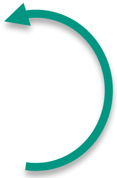
- Assign edge costs, node costs, Start = 0
- Propagate and sum costs
- Expand cheapest node
- Re-assign minimum cost

Short Review: Dijkstra's Algorithm

Find shortest path from start to goal:

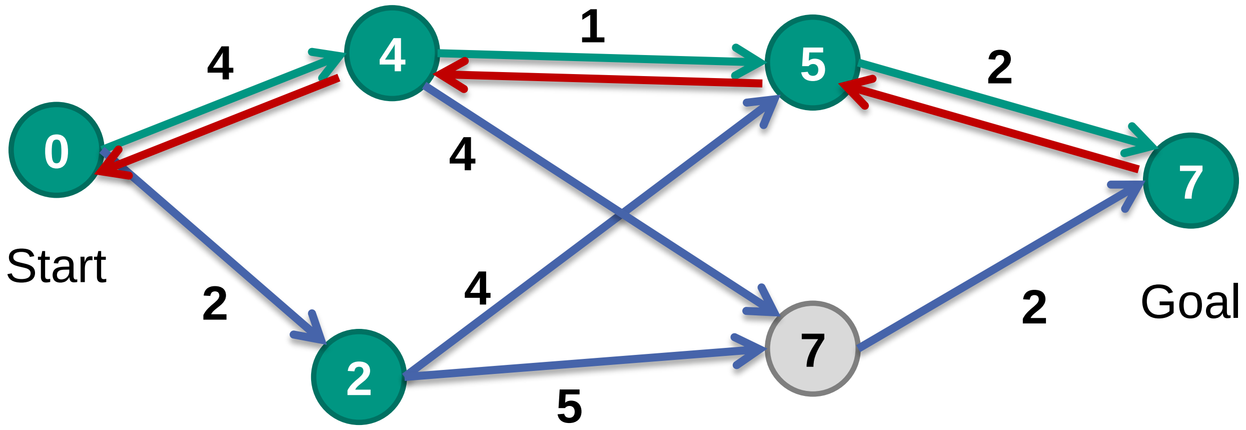


- Assign edge costs, node costs, Start = 0
- Propagate and sum costs
- Expand cheapest node
- Re-assign minimum cost



Short Review: Dijkstra‘s Algorithm

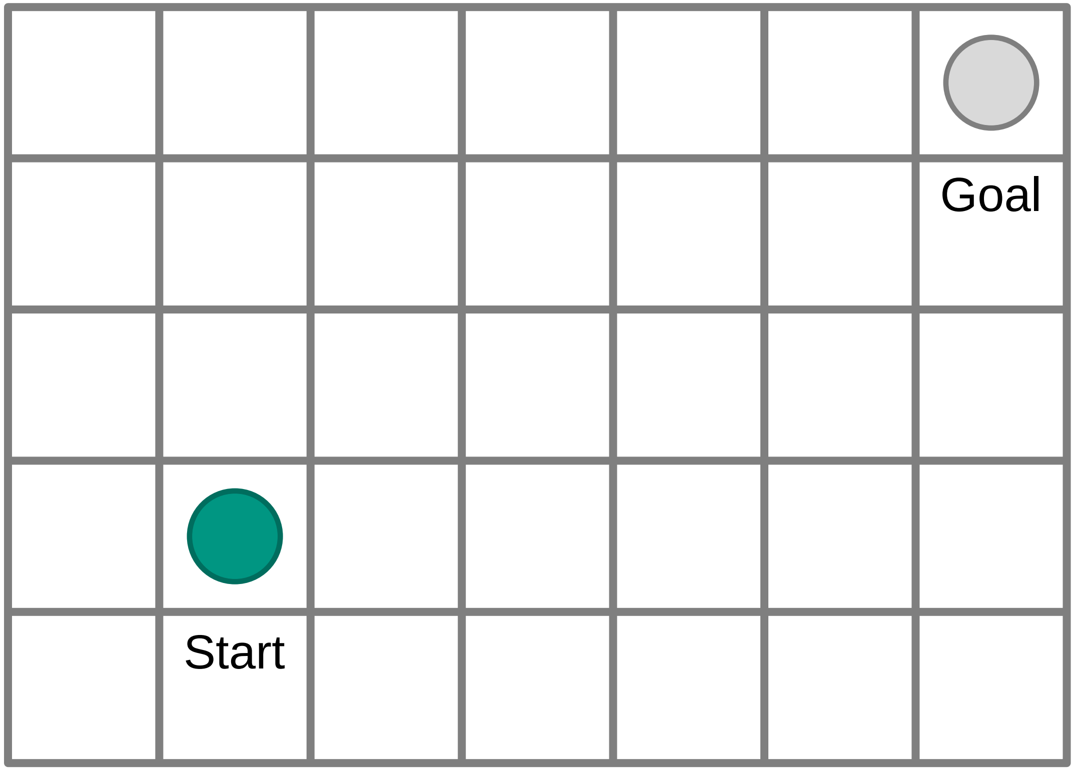
Find shortest path from start to goal:



- Assign edge costs, node costs, Start = 0
- Propagate and sum costs
- Expand cheapest node
- Re-assign minimum cost
- Trace back shortest path

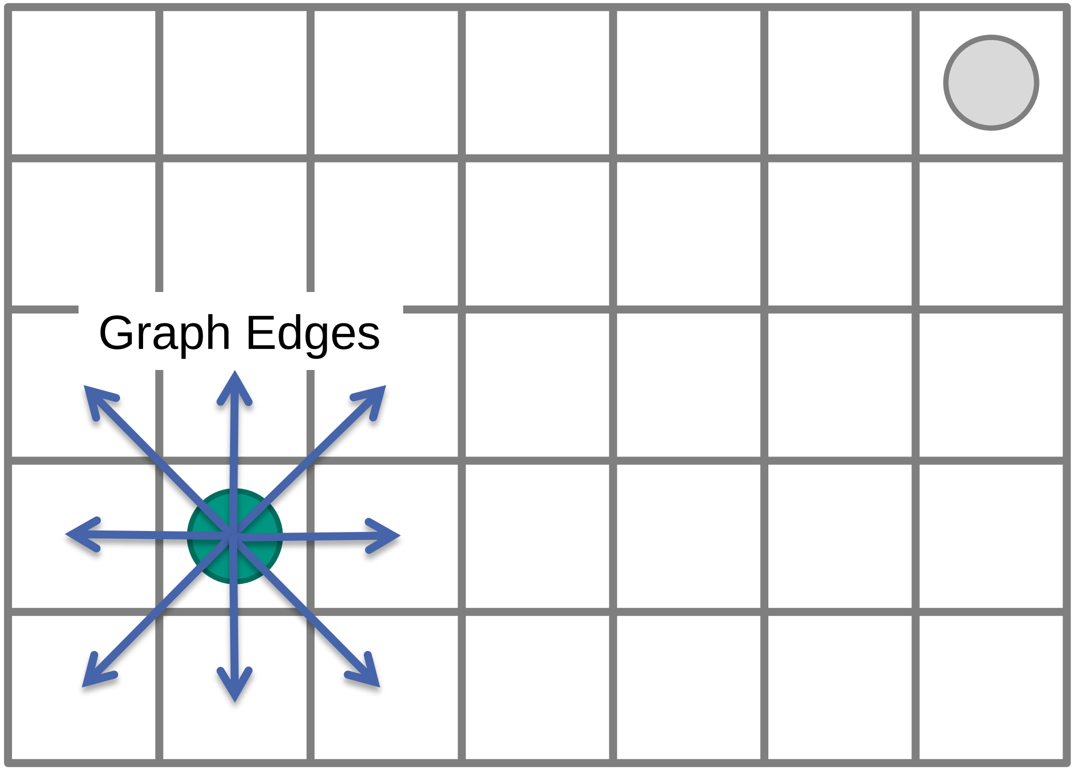
Short Review: Dijkstra‘s Algorithm

Find shortest path from start to goal:



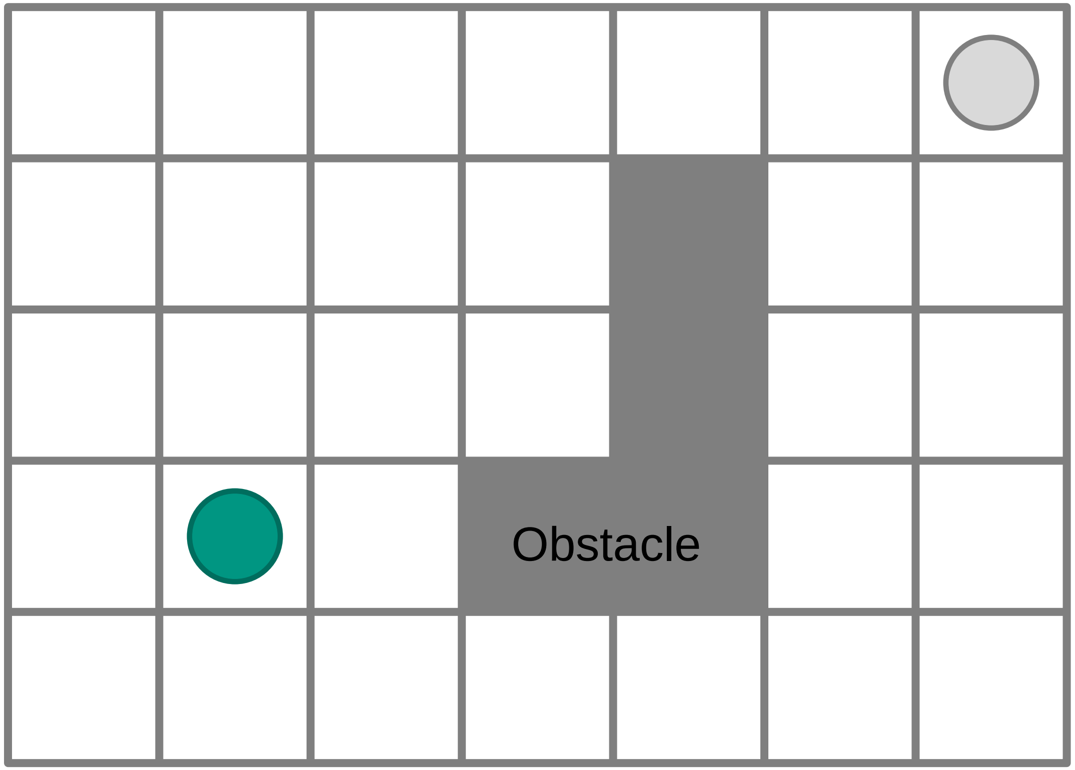
Short Review: Dijkstra‘s Algorithm

Find shortest path from start to goal:



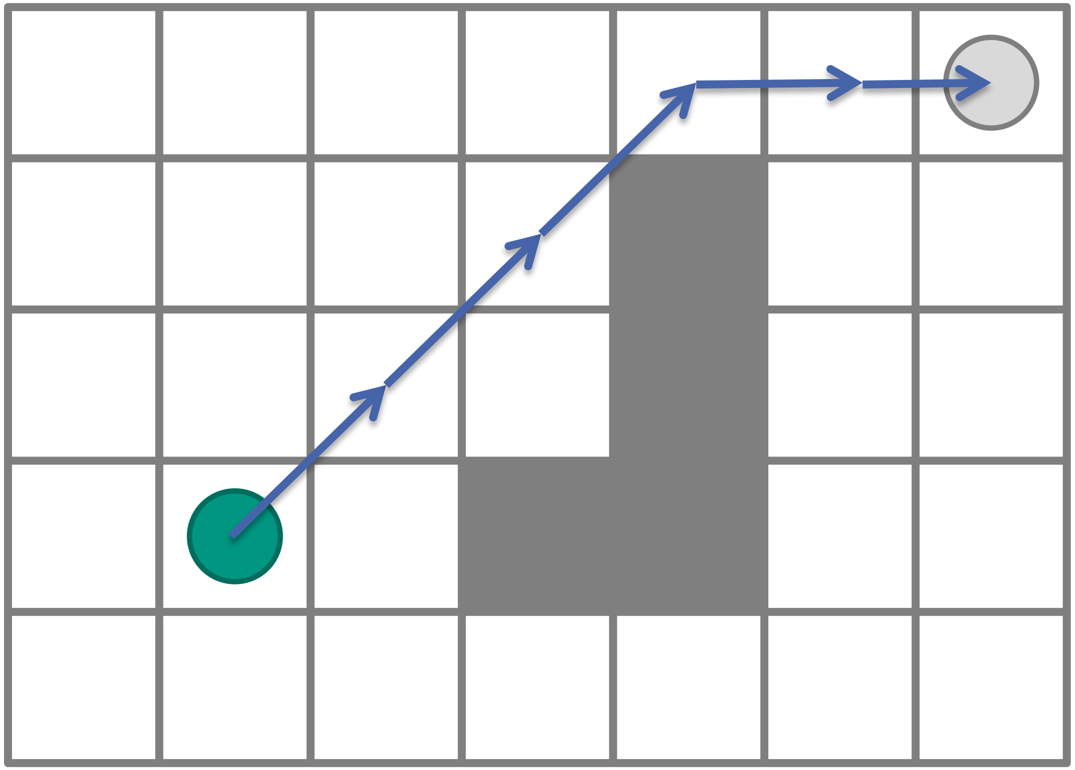
Short Review: Dijkstra‘s Algorithm

Find shortest path from start to goal:



Short Review: Dijkstra‘s Algorithm

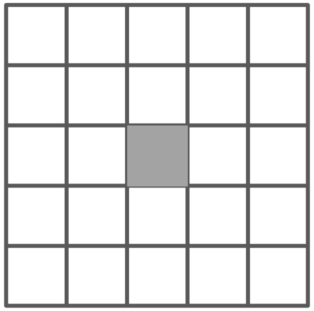
Find shortest path from start to goal:



Shortest Path with a CNN

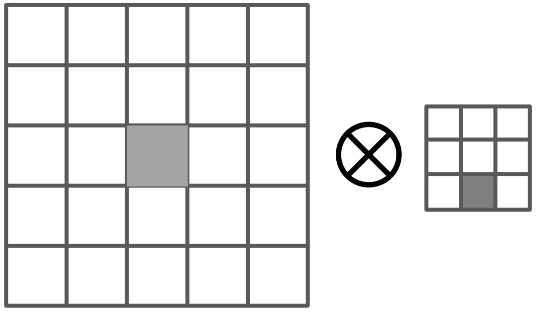
Finding the Shortest Path with a CNN

- Assign edge costs, node costs, Start = 0



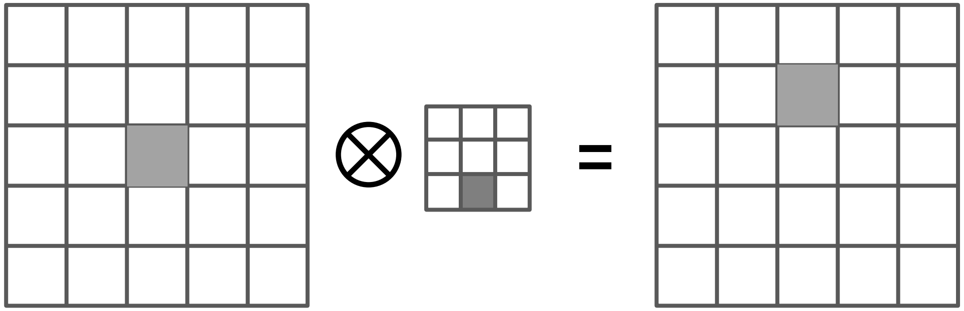
Finding the Shortest Path with a CNN

- Assign edge costs, node costs, Start = 0
- Propagate



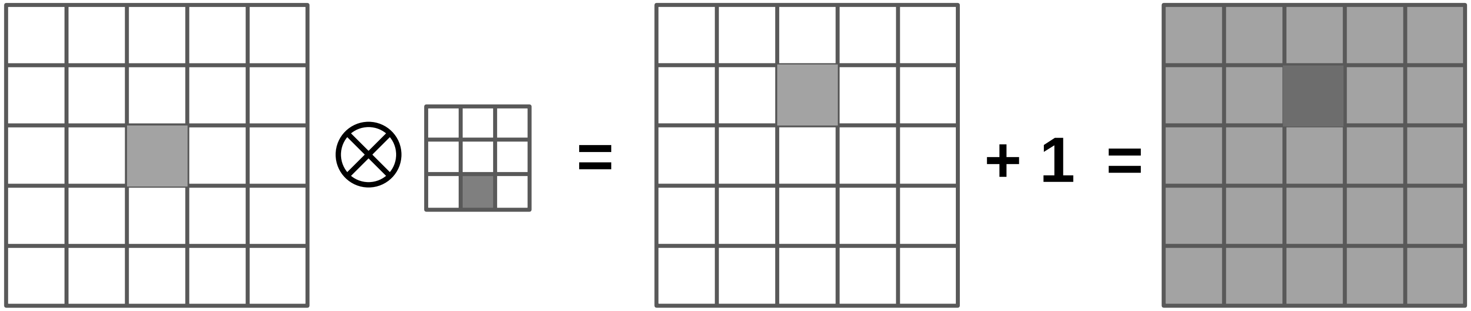
Finding the Shortest Path with a CNN

- Assign edge costs, node costs, Start = 0
- Propagate



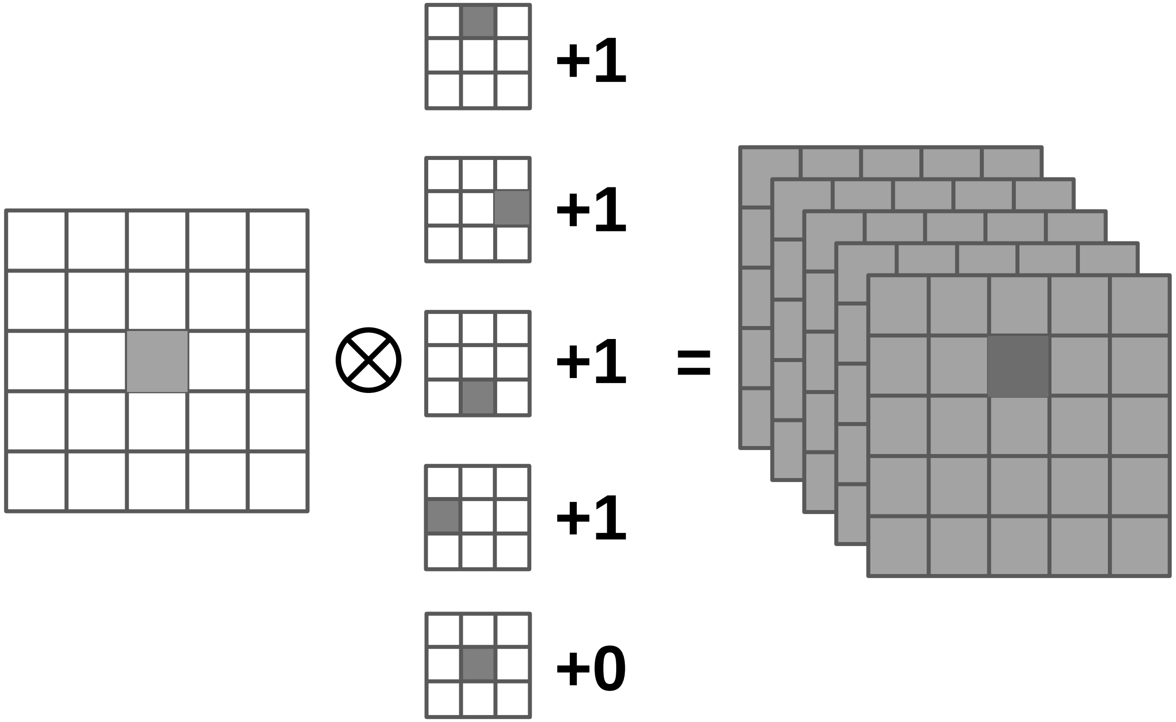
Finding the Shortest Path with a CNN

- Assign edge costs, node costs, Start = 0
- Propagate and sum costs



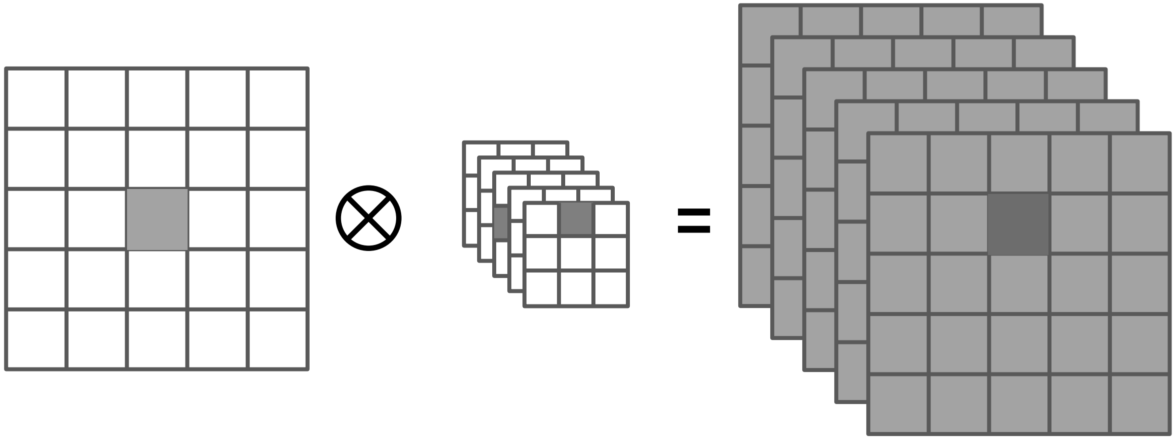
Finding the Shortest Path with a CNN

- Assign edge costs, node costs, Start = 0
- Propagate and sum costs



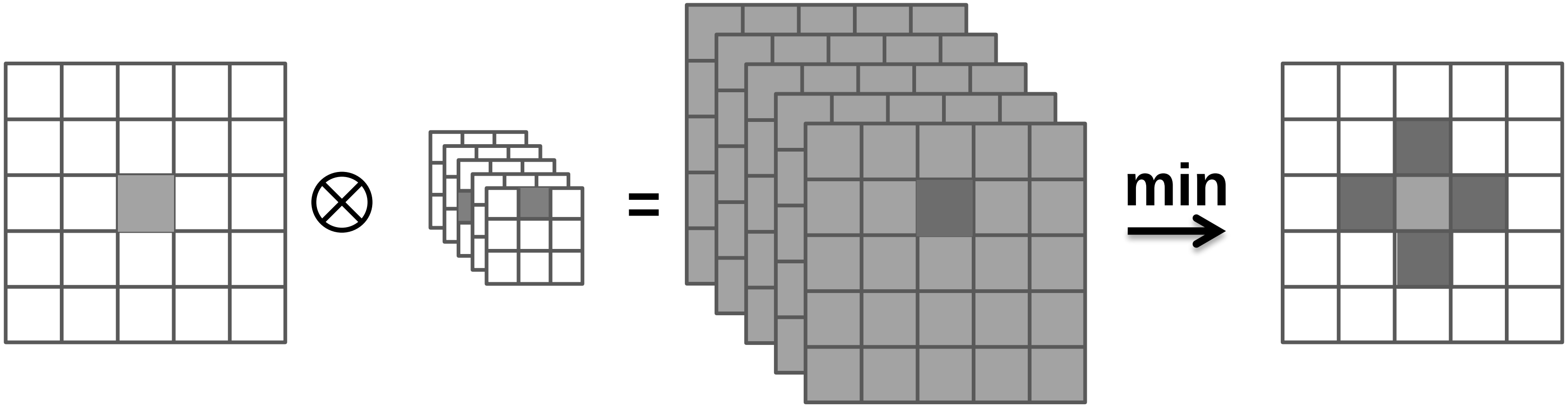
Finding the Shortest Path with a CNN

- Assign edge costs, node costs, Start = 0
- Propagate and sum costs



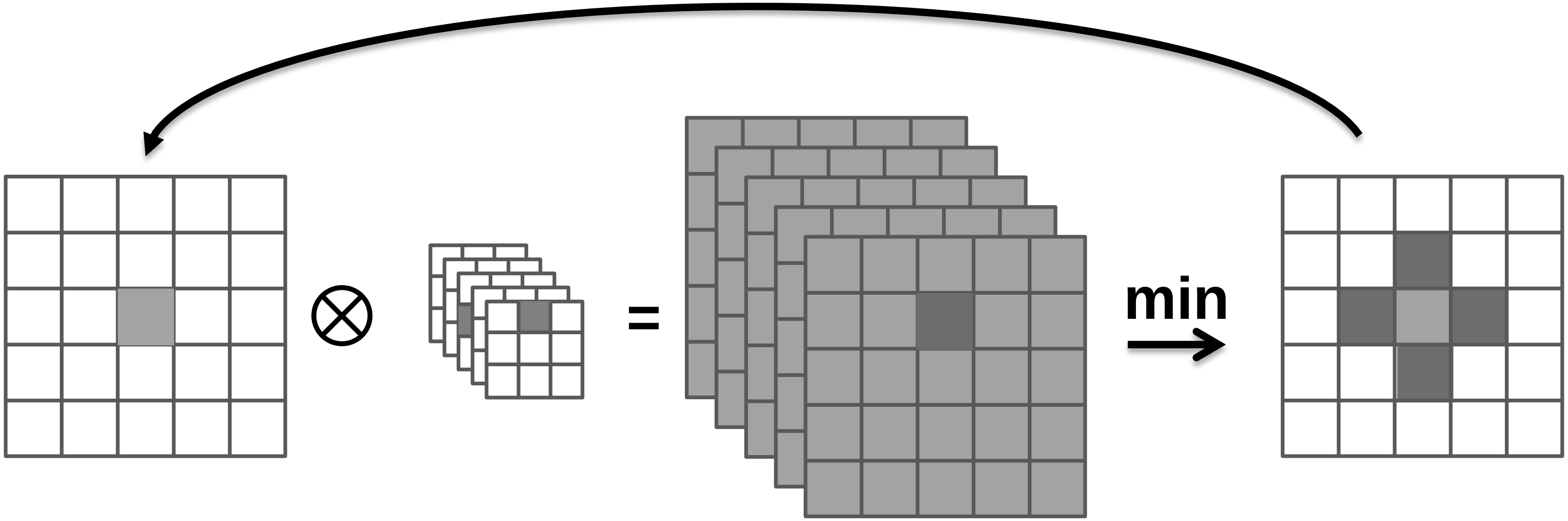
Finding the Shortest Path with a CNN

- Assign edge costs, node costs, Start = 0
- Propagate and sum costs
- Re-assign minimum cost

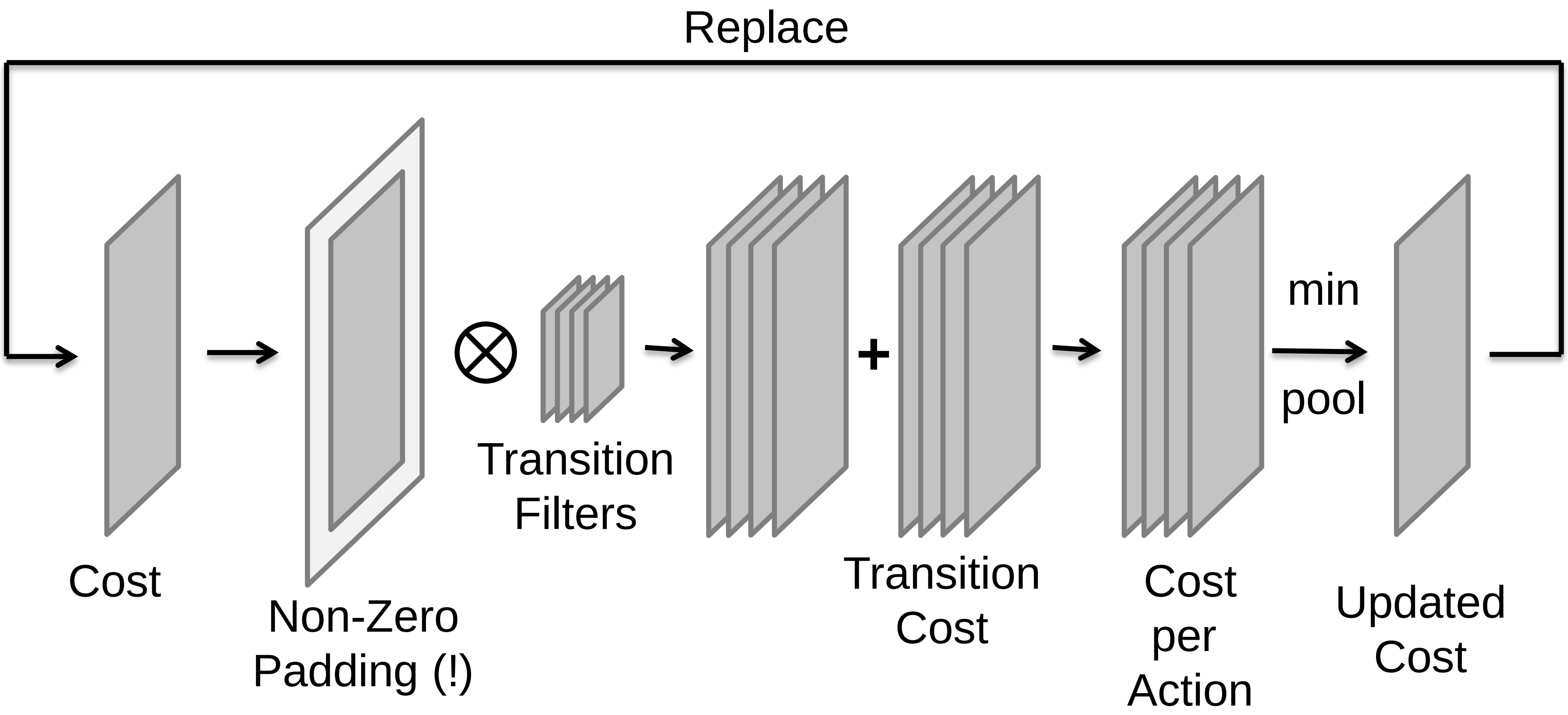


Finding the Shortest Path with a CNN

- Assign edge costs, node costs, Start = 0
- Propagate and sum costs
- Re-assign minimum cost

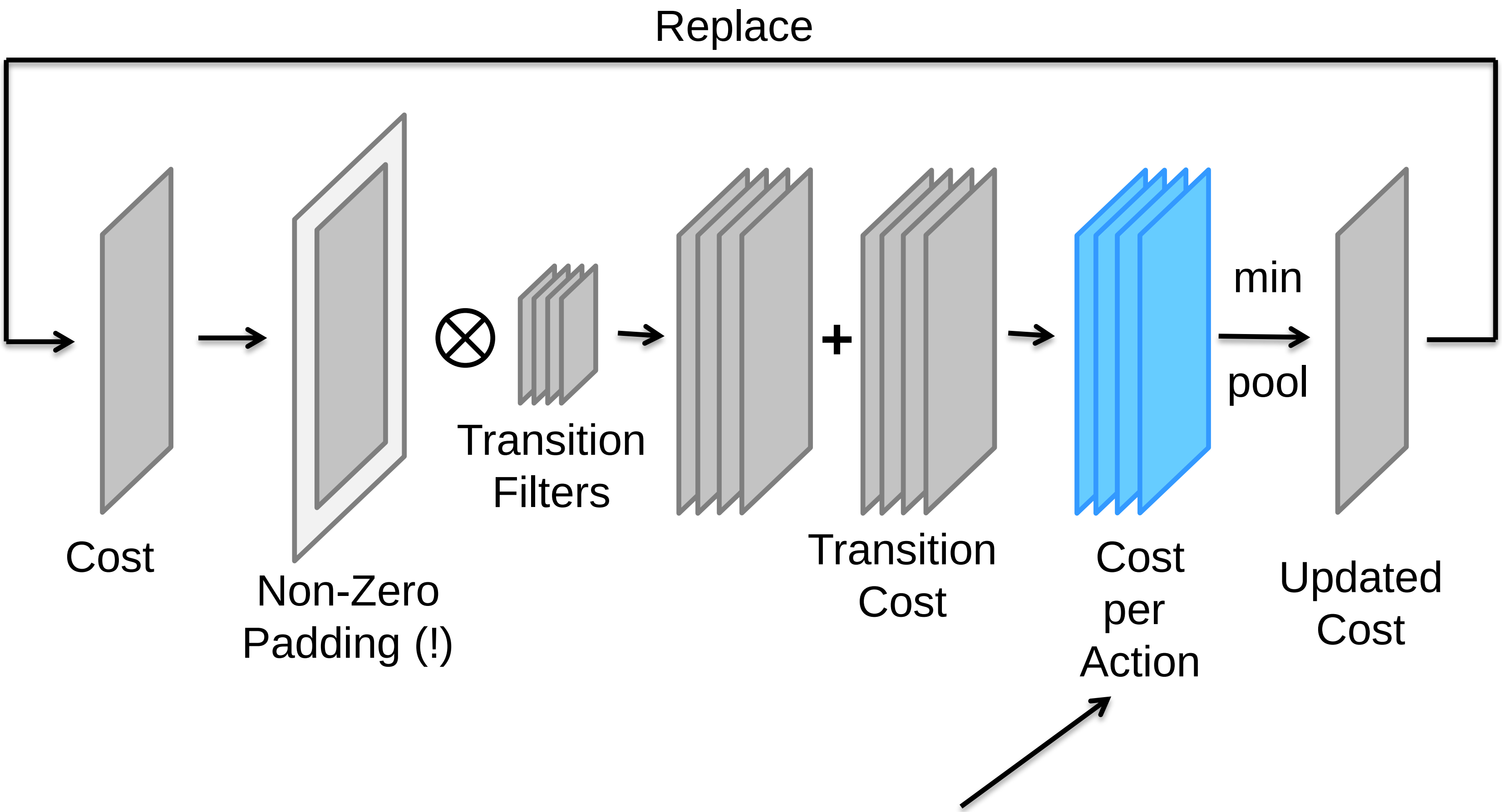


Finding the Shortest Path with a CNN



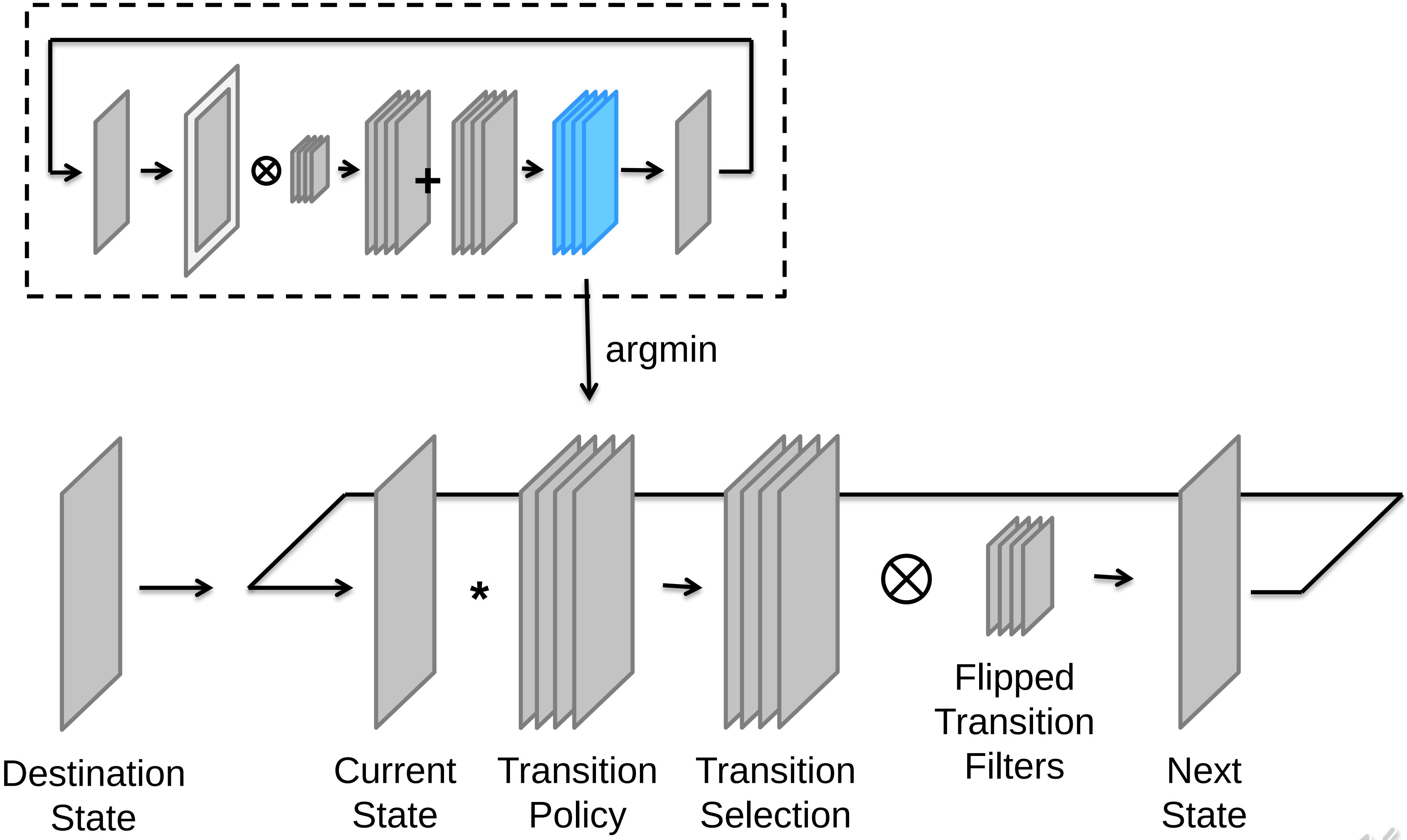
"Reinforcement learning via recurrent convolutional neural networks", arXiv:1701.02392

Finding the Shortest Path with a CNN



Argmin of this layer is transition policy

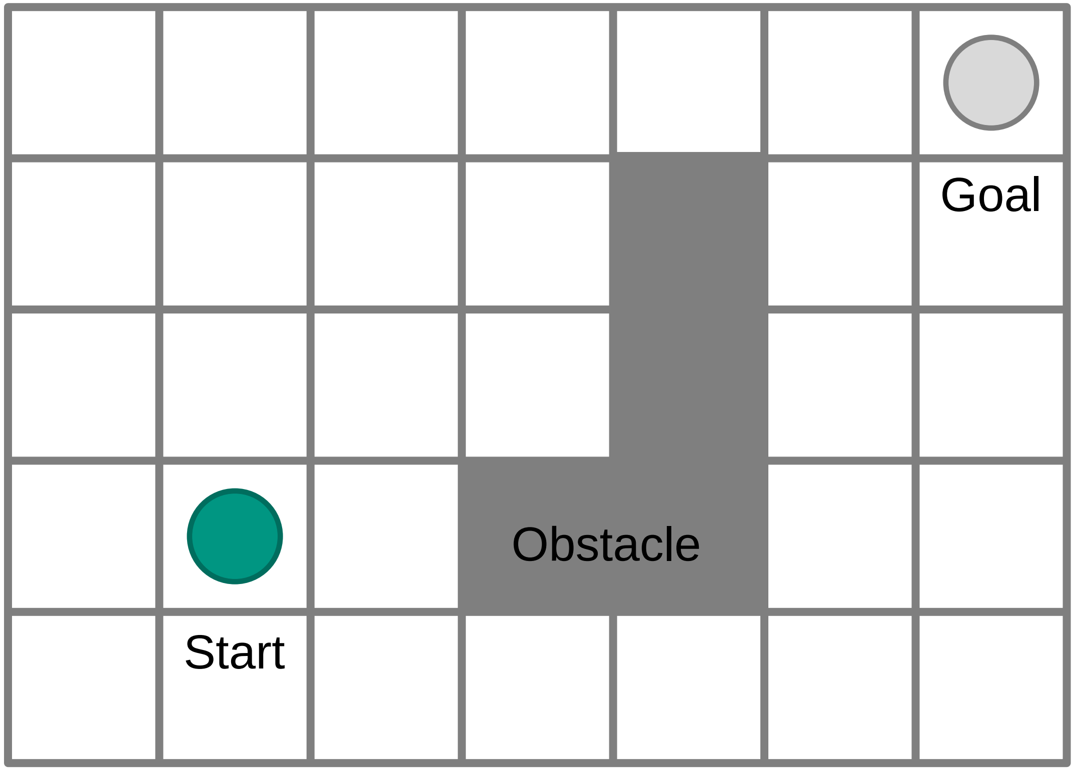
Evaluating the Shortest Path with a CNN



Example:
Simple Path Planning

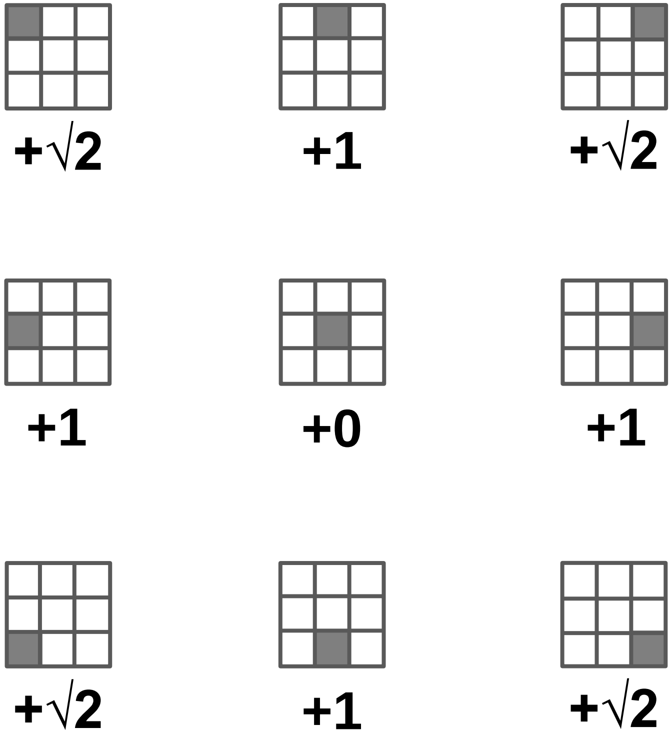
Example: Path Planning

Find shortest path from start to goal:



Example: Path Planning

- Nine possible transition filters
- Cost is the traversed distance



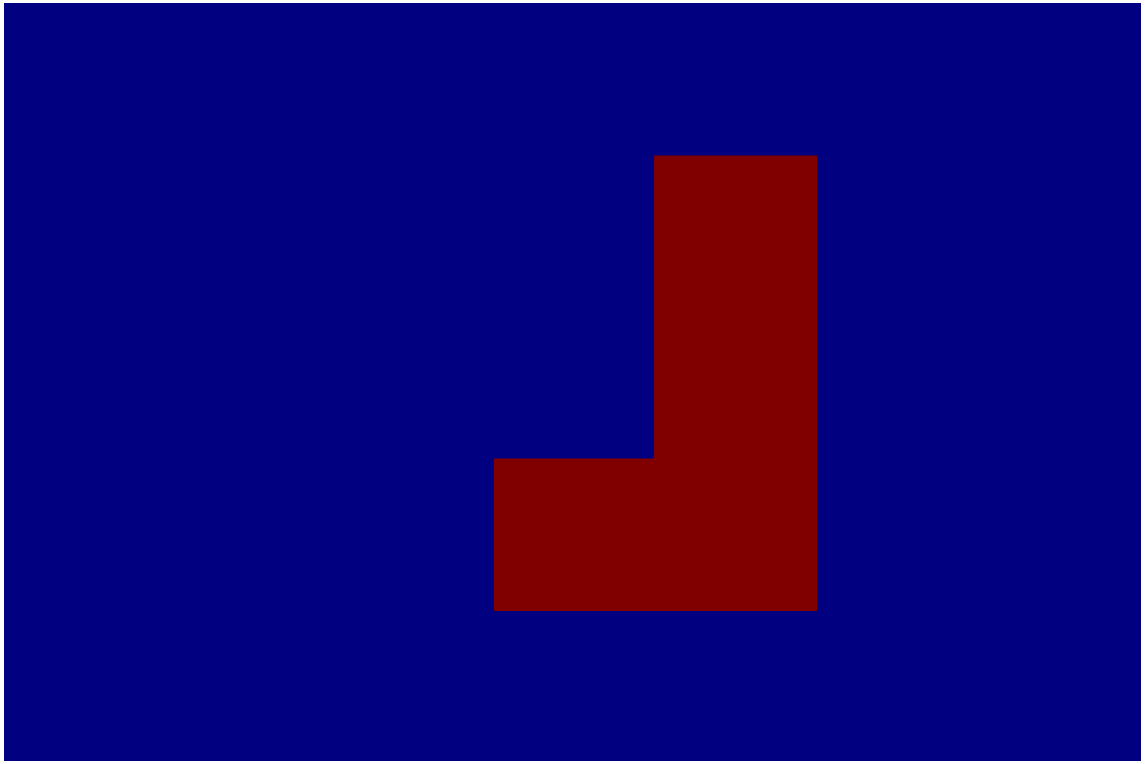
+√2 +1 +√2

+1 +0 +1

+√2 +1 +√2

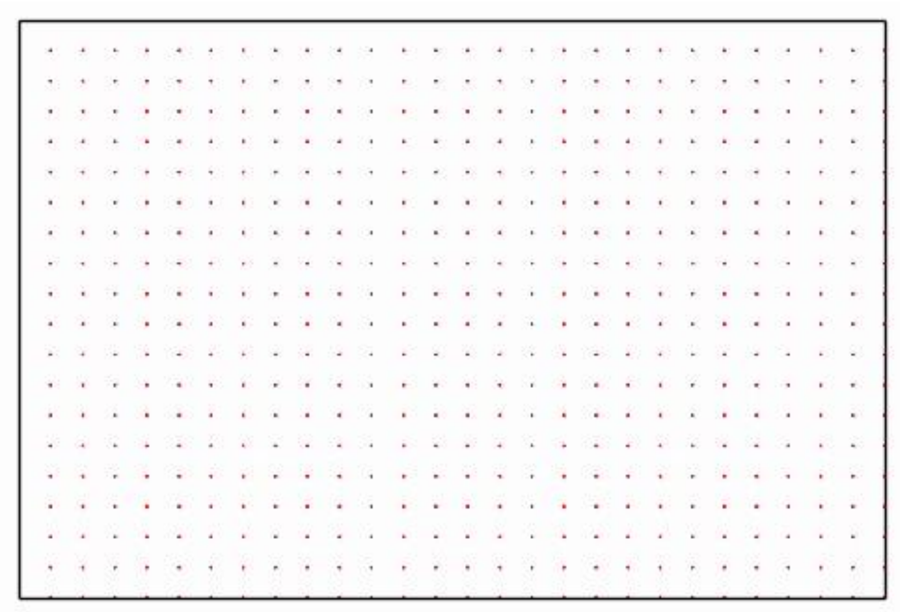
Example: Path Planning

- Cost Model
  - Additive layer
  - High cost where obstacle is located



Example: Path Planning

Cost Map



State Transition Map

Finding the Shortest Path with a CNN

- If you use Dijkstra:
  - Graph traversal with known transitions is faster
  - States can be updated selectively
  - Visited nodes will never be touched again

- **Why would you do it then?**

Driving Like a Human: Imitation Learning

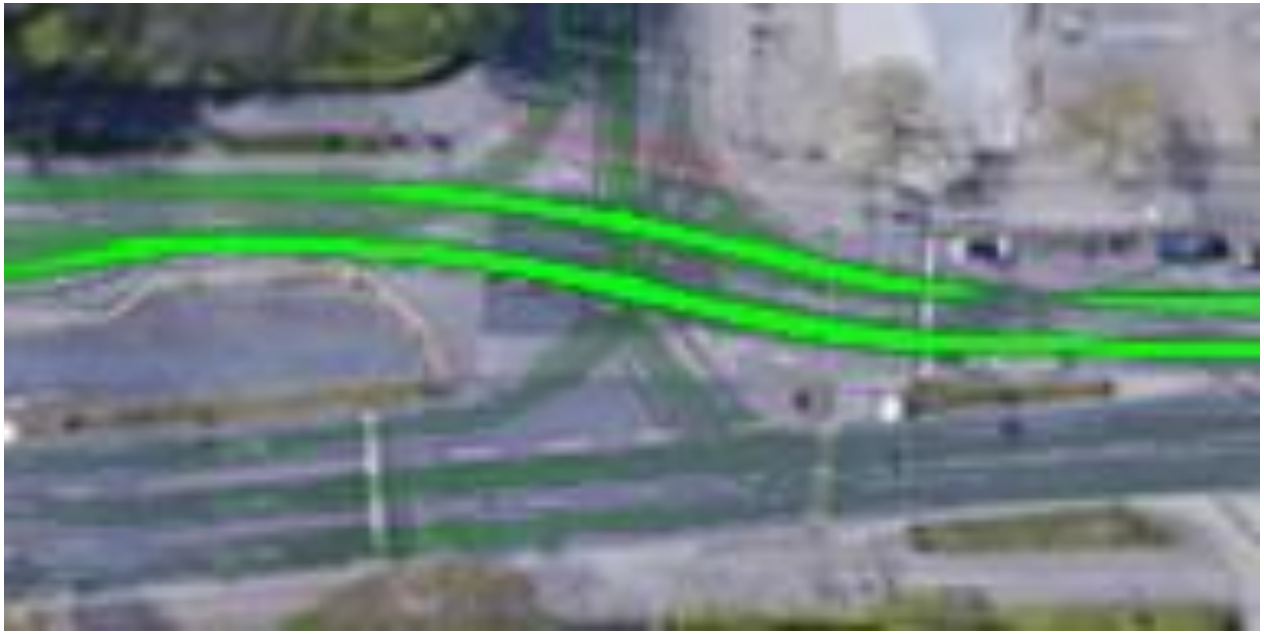
Imitation Learning

Intersection in Karlsruhe

Arial view: Google Maps

Imitation Learning

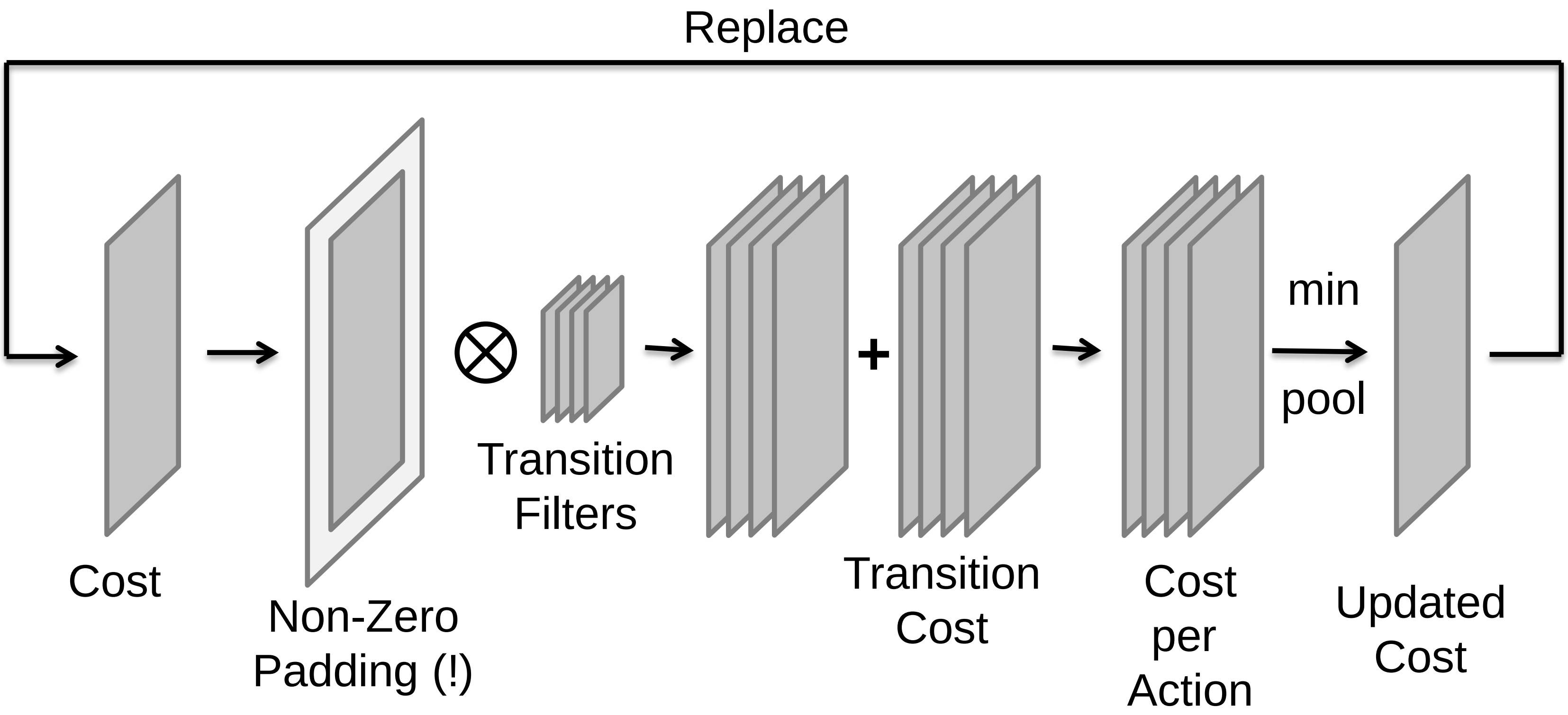
- Recorded trajectories
- **Teach a network to imitate human behavior**



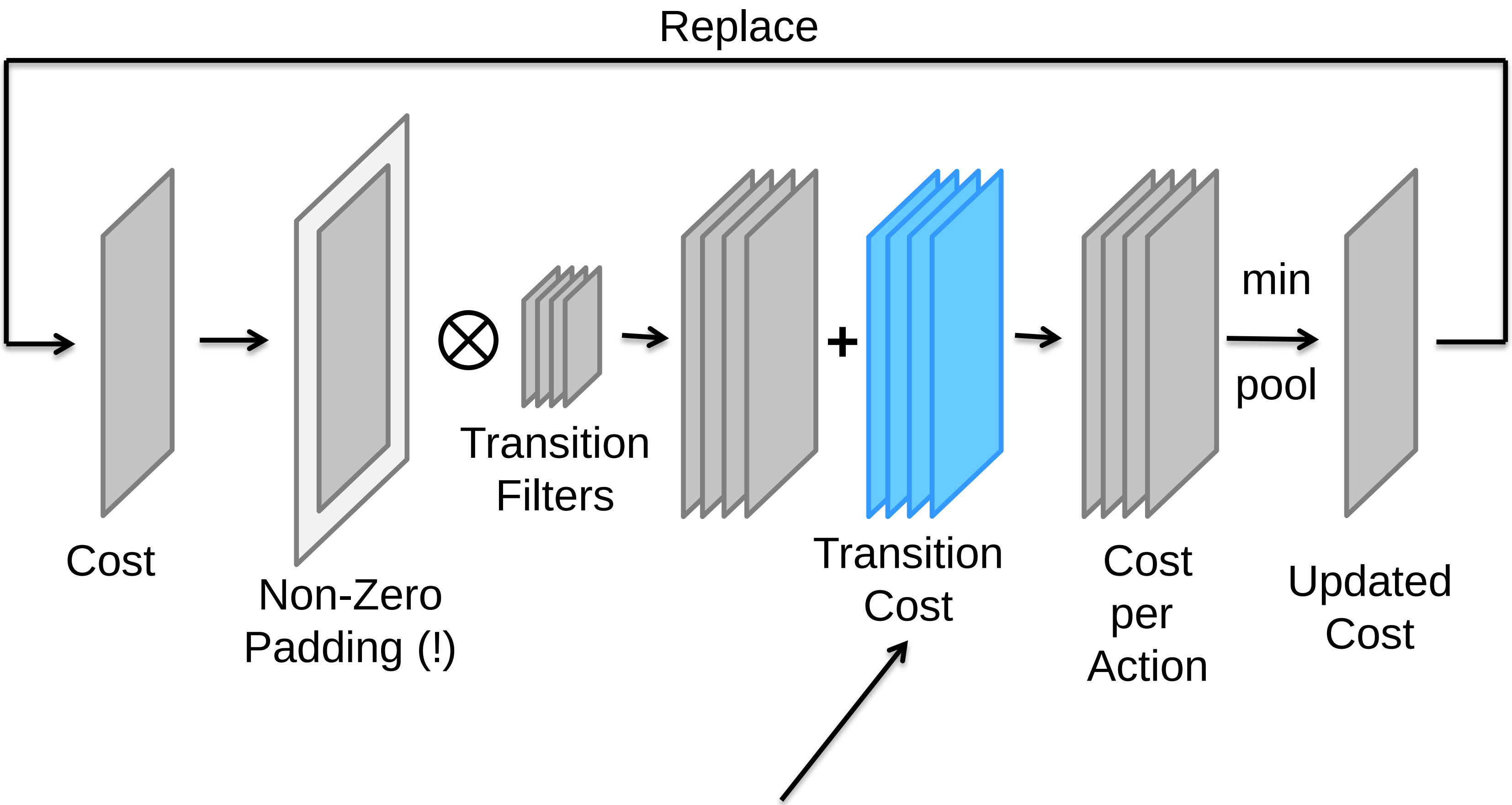
Intersection in Karlsruhe

Arial view: Google Maps

Imitation Learning

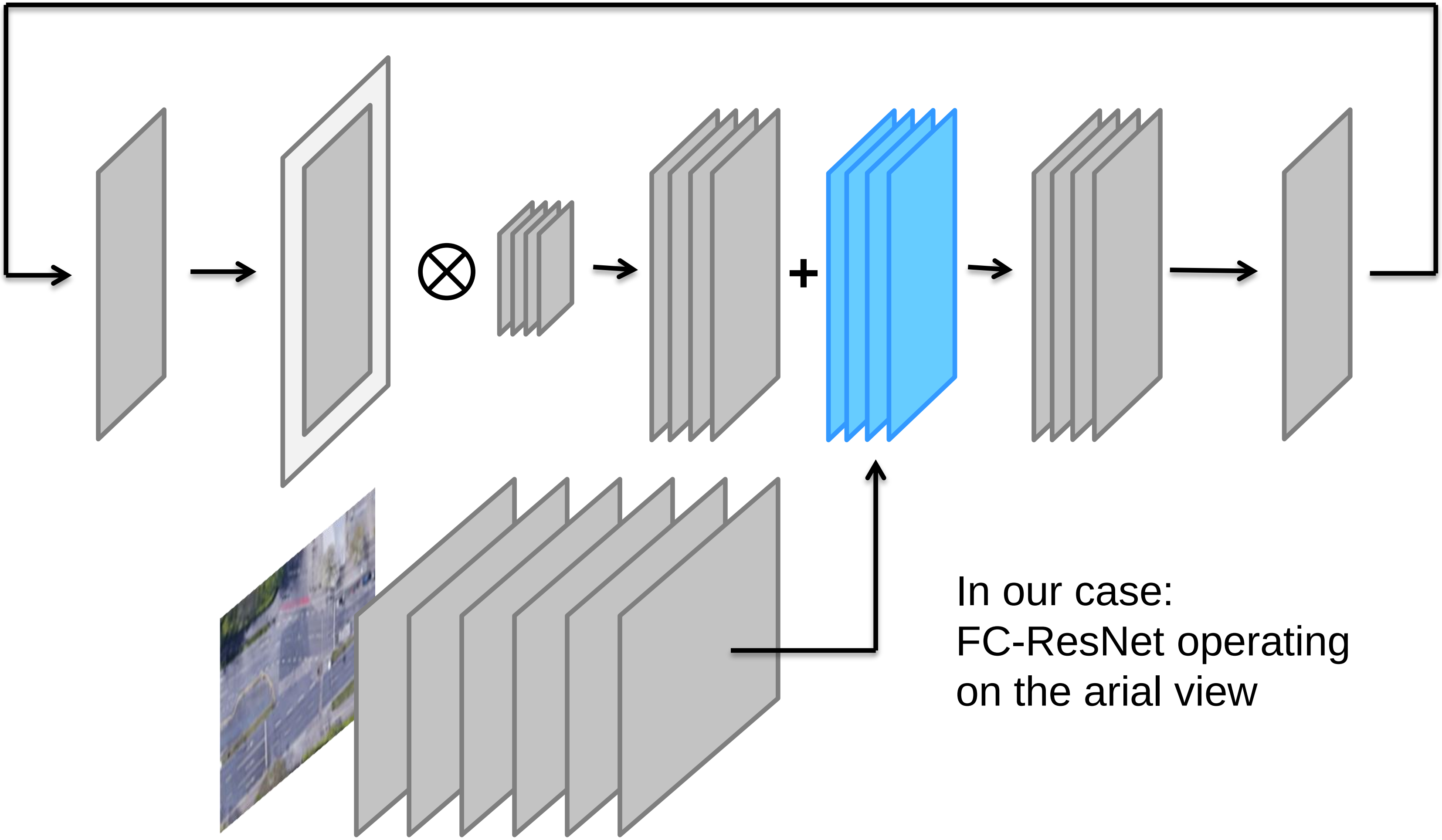


Imitation Learning



Fill in the whole bunch of CNN techniques

Example II: Imitation Learning

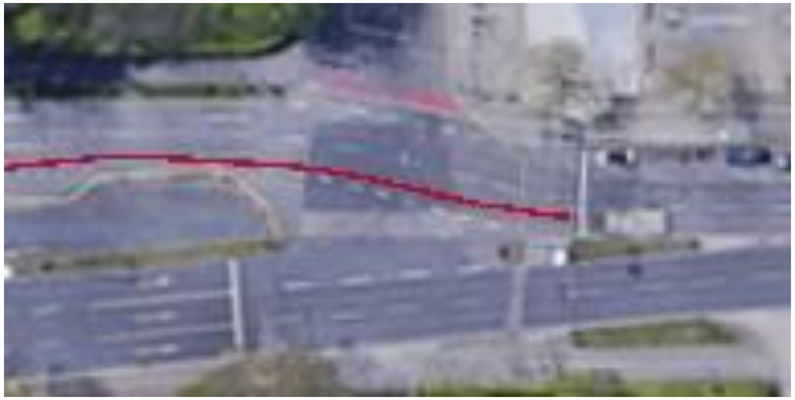


In our case:
FC-ResNet operating
on the arial view

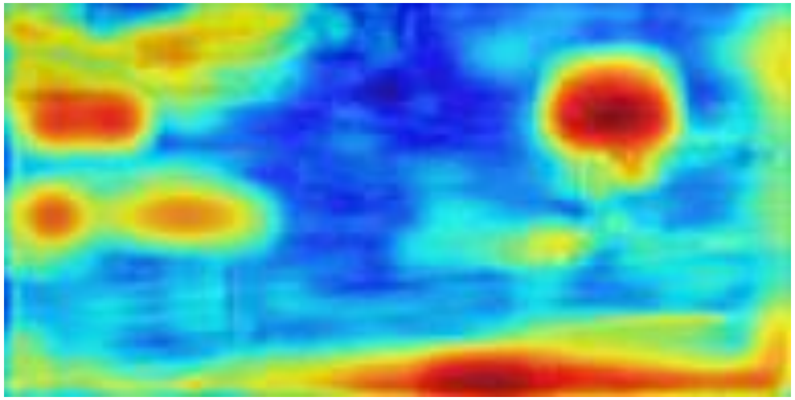
Imitation Learning

Path driven by human

Imitation Learning

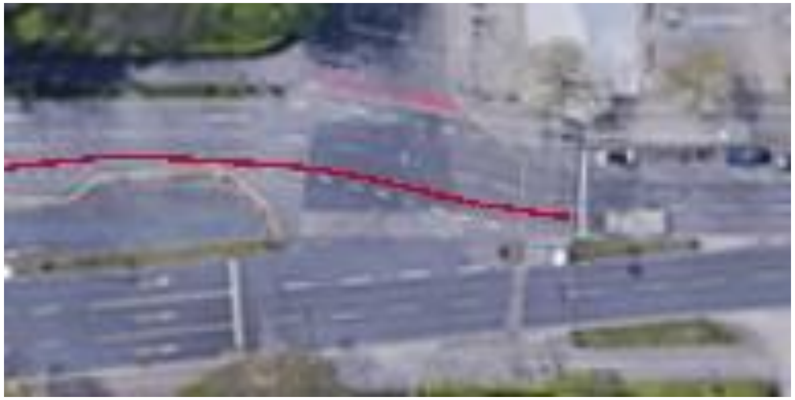


Path driven by human

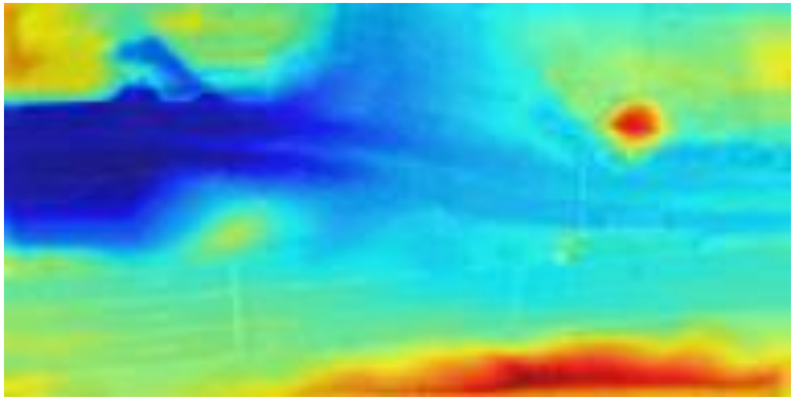


Cost map from arial image

Imitation Learning

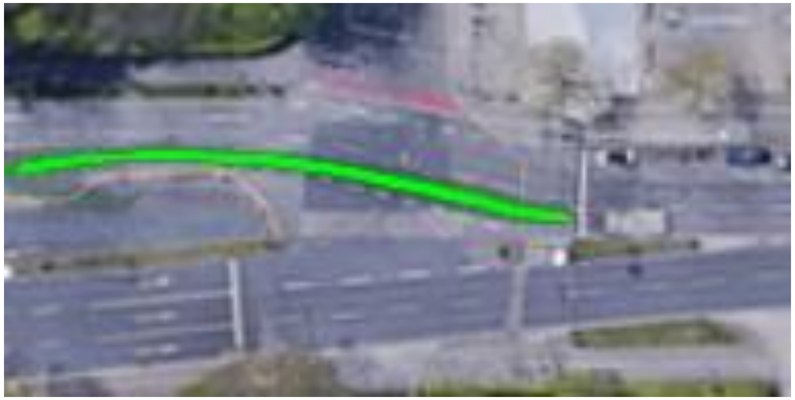


Path driven by human

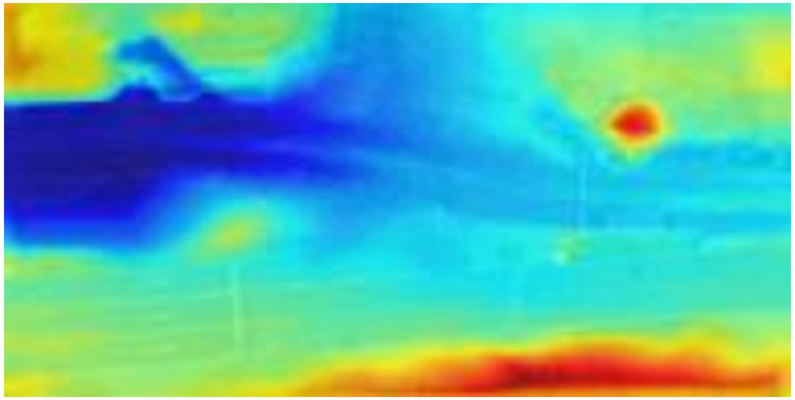


Cost map after planning

Imitation Learning

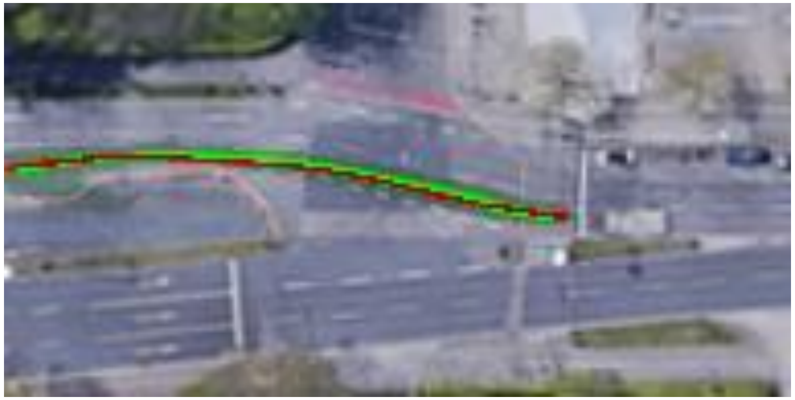


Path planned by network



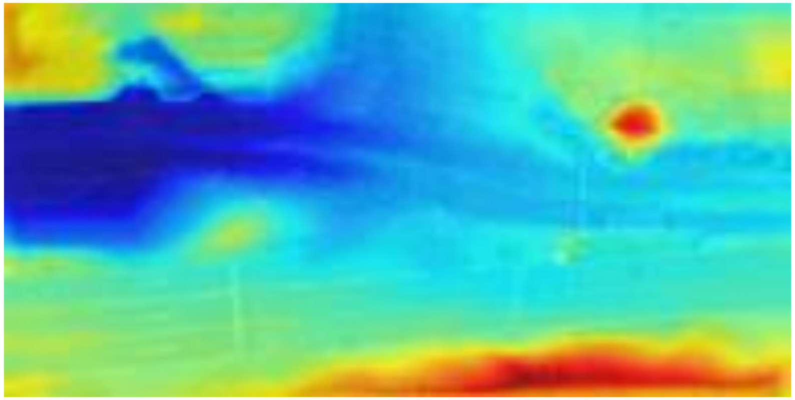
Cost map after planning

Imitation Learning



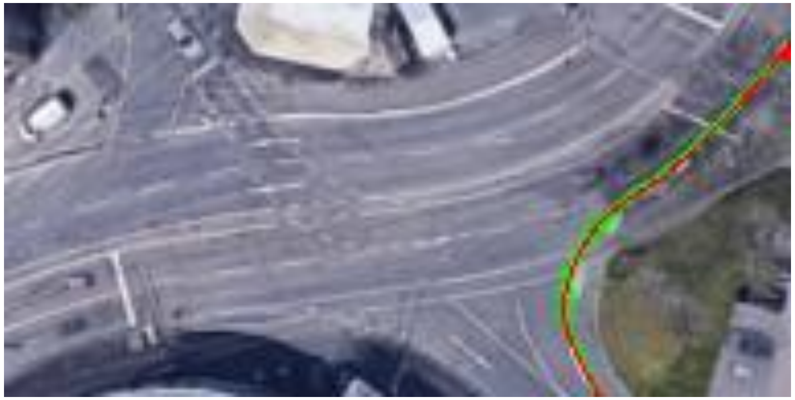
Path planned by network

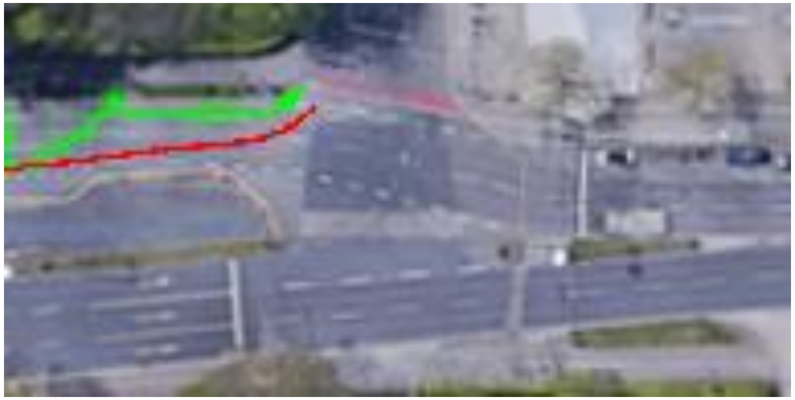
Path driven by human



Cost map after planning

Imitation Learning

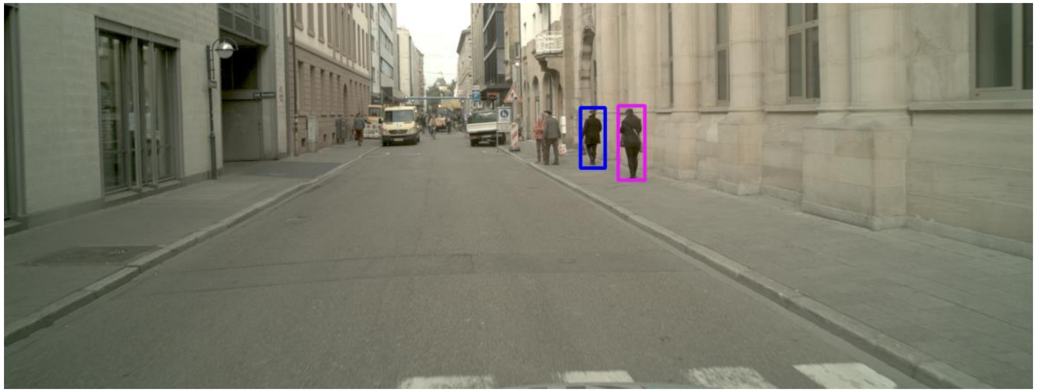




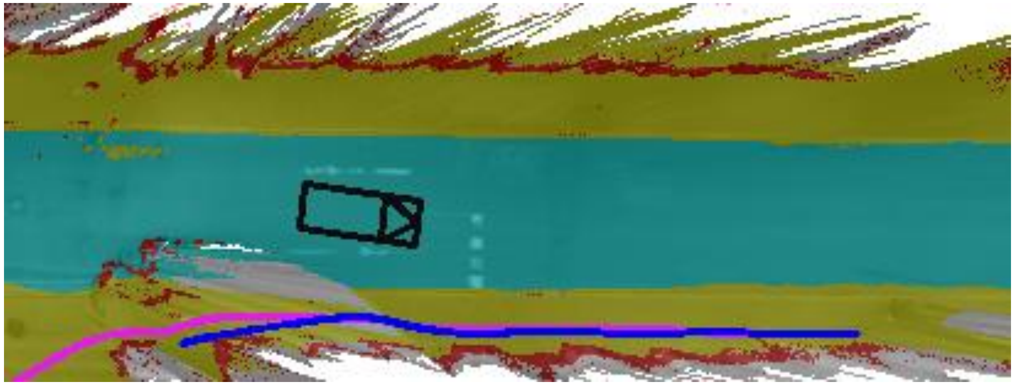
Path planned by network
Path driven by human

Outlook: Prediction and Cooperation

Outlook: Pedestrian Prediction



Camera image



Road

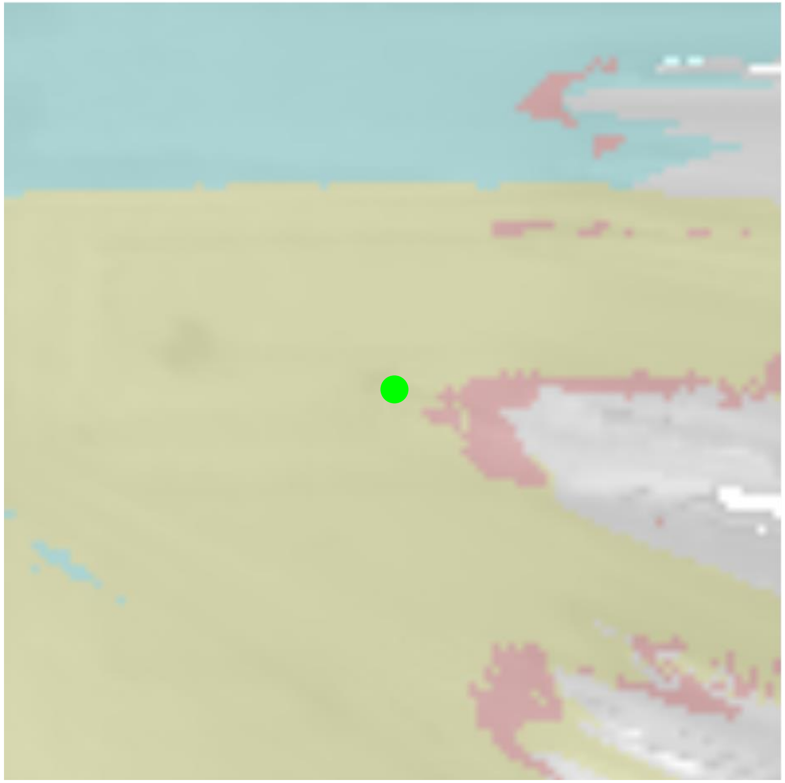
Sidewalk

Obstacles

Semantic map and top view

Teach a network to predict human motion by planning

Outlook: Pedestrian Prediction



Crop of map centered around pedestrian

🟦 Road

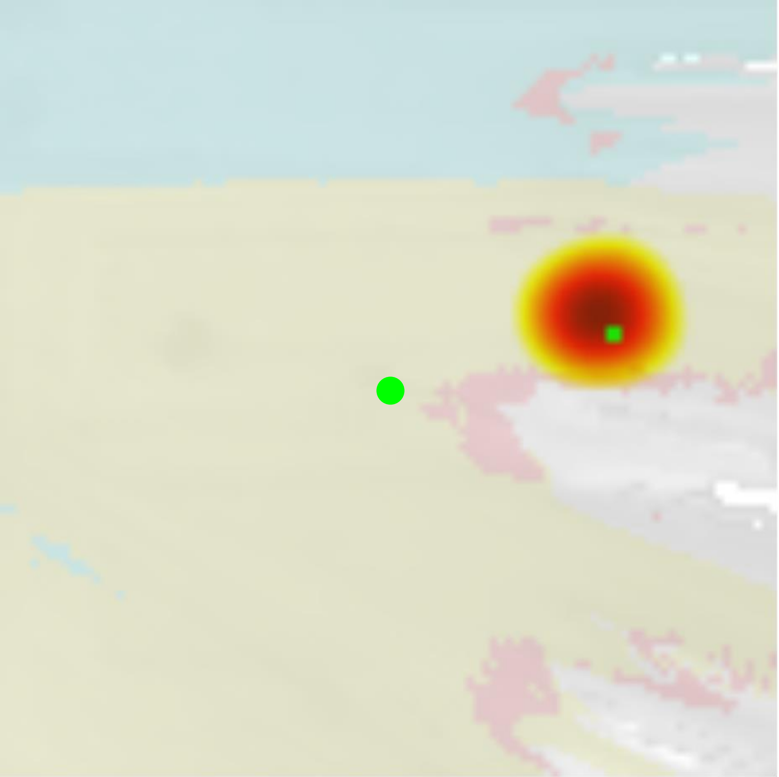
🟧 Sidewalk

🟥 Obstacles

🟩 Pedestrian

“Pedestrian Prediction by Planning using Deep Neural Networks”, arXiv:1706.05904

Outlook: Pedestrian Prediction

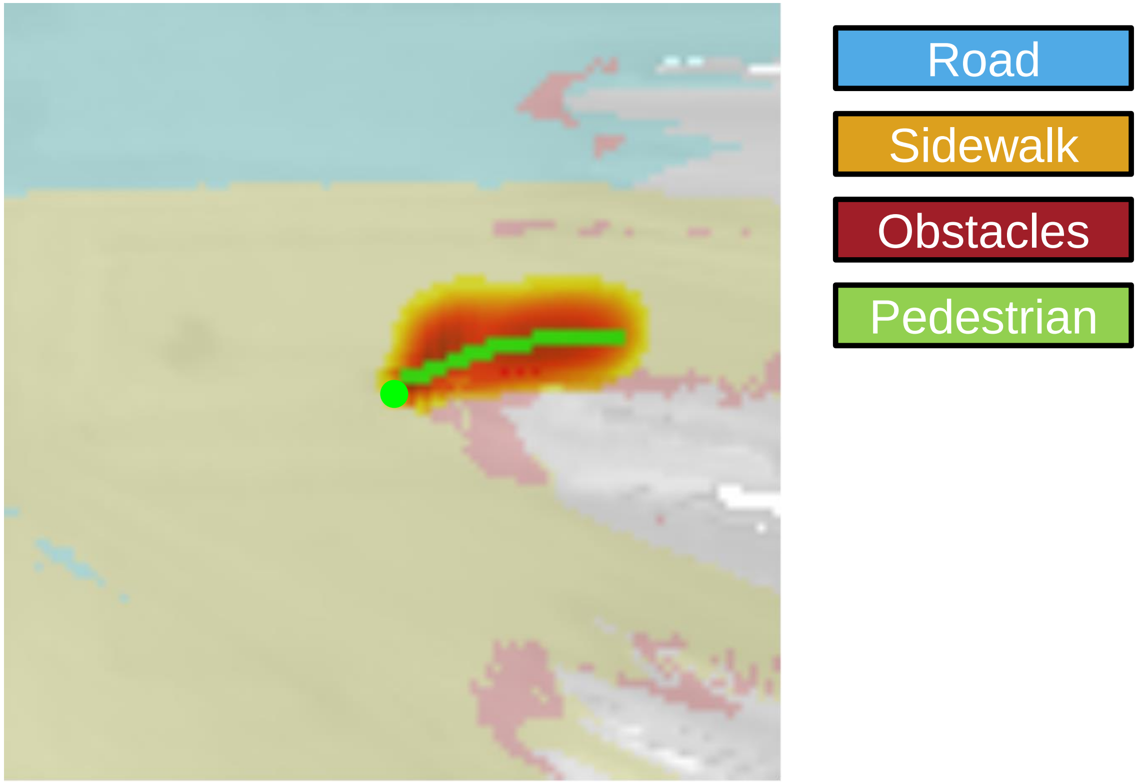


- 🟦 Road
- 🟧 Sidewalk
- 🟥 Obstacles
- 🟩 Pedestrian

Predict destination for planning

“Pedestrian Prediction by Planning using Deep Neural Networks”, arXiv:1706.05904

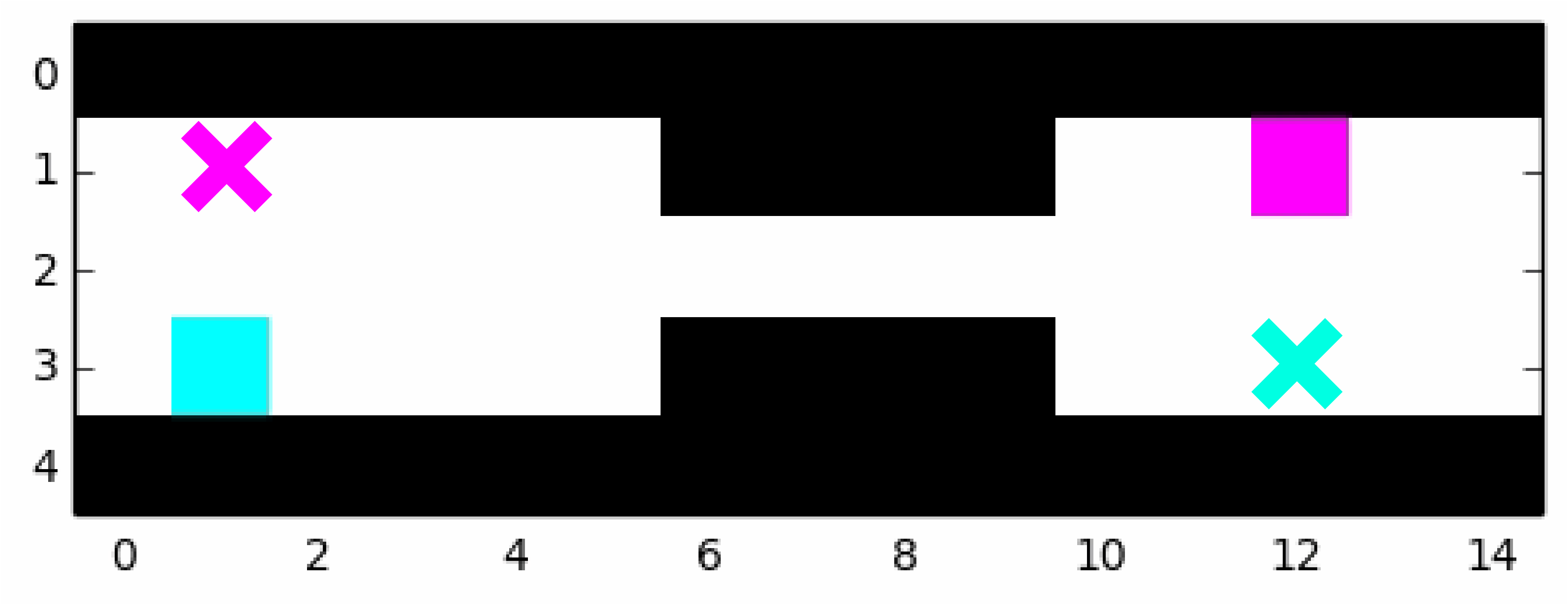
Outlook: Pedestrian Prediction



Predicted with Net

“Pedestrian Prediction by Planning using Deep Neural Networks”, arXiv:1706.05904

Outlook: Cooperative Planning



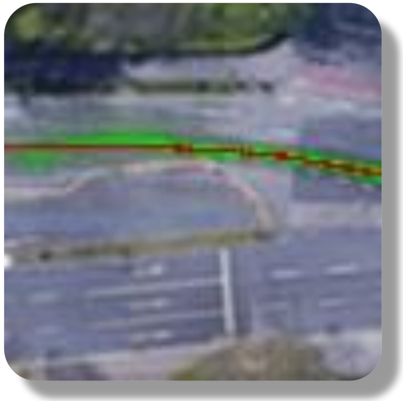
Teach a network resolve conflicts

"Cooperative Motion Planning for Non-Holonomic Agents with Value Iteration Networks", arXiv:1706.05904

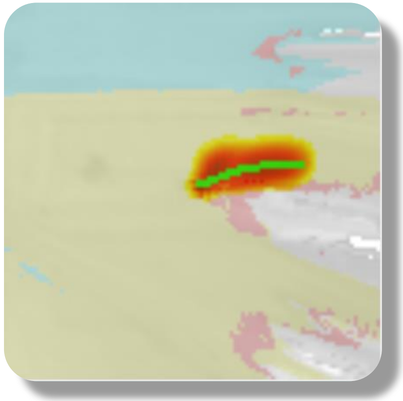
Summary

Summary

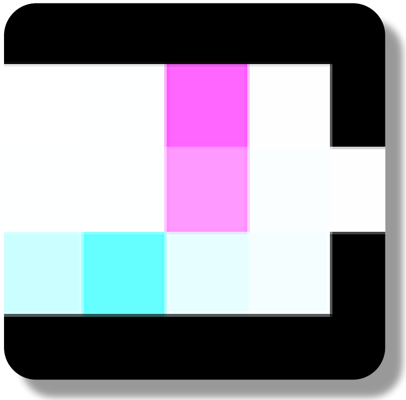
Planning Net…



… for imitation



… for prediction



… for cooperation

The People

Jannik Quehl
Trajectory Data

Maximilian Naumann
Cooperative Planning

Florian Wirth
Destination Prediction