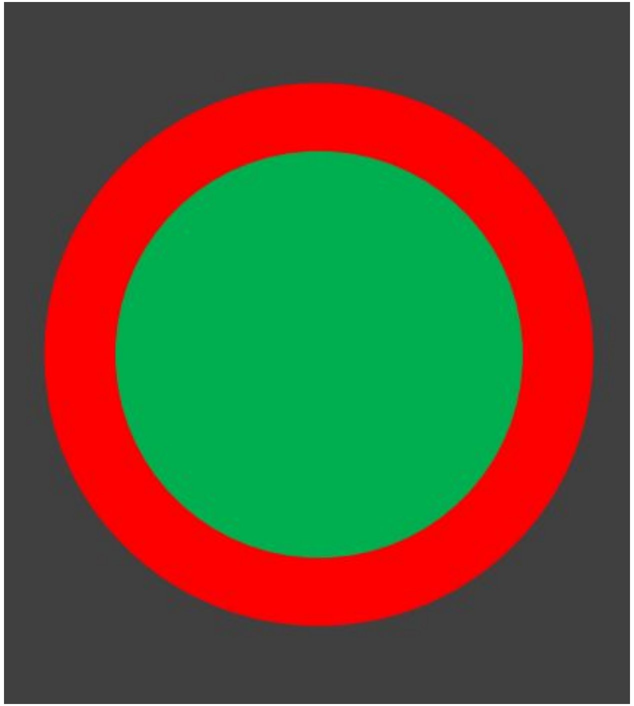
Generalizing Assured AI for Traffic Light Control

Daniel Stambler and Evan Leung

Assuring AI Systems

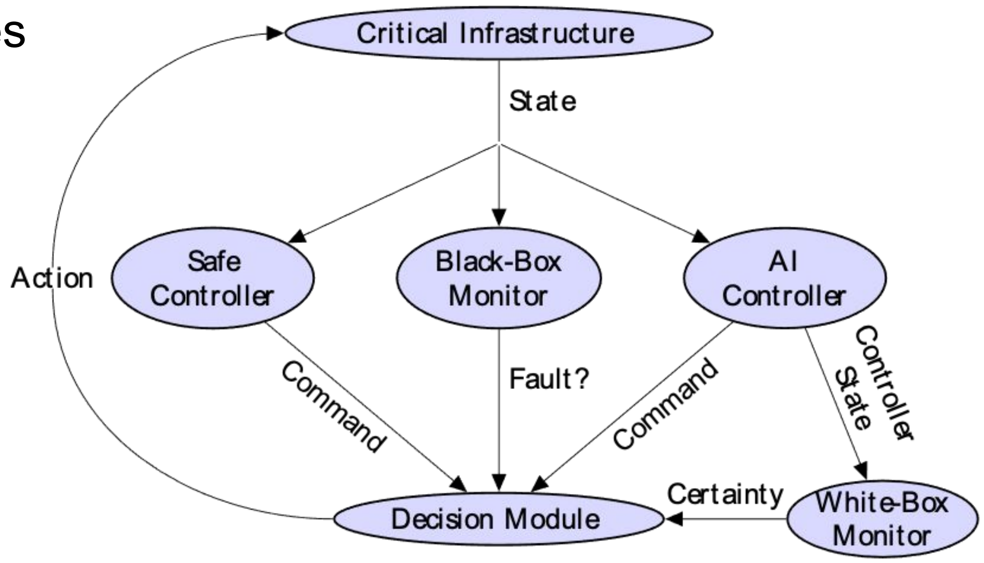
- AI systems becoming ubiquitous
- Cannot be used in critical systems
  - Need to guarantee the worst case
  - AI fails on edge cases
- Assuring AI
  - Switch to safe algorithm to handle situations when AI fails



AI Traffic Light Controllers

- Potential for more efficient travel through intersections
- Non-fatal errors still considered unacceptable
    - Unreasonably long wait times

- Blackbox Monitor
- Whitebox Monitor



Outline

- Motivation
- **Background**
- Previous Work
- Problem Definition
- Approaches and Results
- Future Work

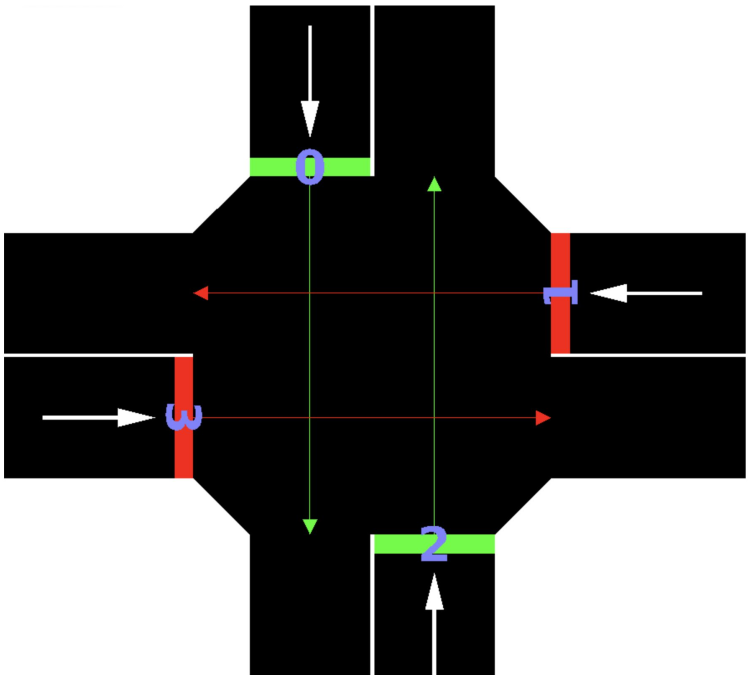
Definitions

- **Model:**
  - Differentiable mathematical formula for fitting input data
- **Training:**
  - Process of optimizing the weights of a given model
- **Evaluation:**
  - Process of using a trained model. Pass in inputs, get results
- **Monolithic Model:**
  - Inflexible model. To evaluate an m x n grid, the model needs to be trained on an m x n grid
- **General Model:**
  - Flexible model that can be evaluated on any m x n grid

Definitions: Defining our Environment

- Four way intersection, bidirectional roads
- Four incoming edges
- Four outgoing edges

- Straight + right turn on green
- Separate green light for left turns
- All lights (straight + left) must switch to yellow lights

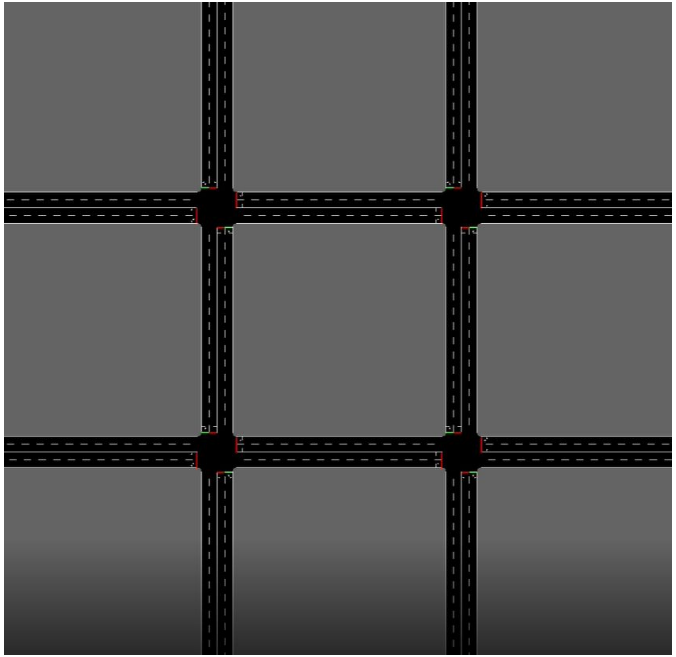


Outline

- Motivation
- Background
- **Previous Work**
- Problem Definition
- Approaches and Results
- Future Work

Previous Work

- Monolithic 2x2 model using SUMO/FLOW
  - Can’t scale up
- Generalized model in Gym CityFlow
  - Solves scaling problem
  - Couldn’t replicate results



Early Challenges

- Onboarded to Gym CityFlow
  - Didn’t see any learning

- Switched back to SUMO/FLOW
  - Replicated previous monolithic model success
  - Spent a while learning very large codebase
  - Understood Jerry’s approach and its possible flaws, brainstormed new approaches

Outline

- Motivation
- Background
- Previous Work
- **Problem Definition**
- Approaches and Results
- Future Work

Problem Definition

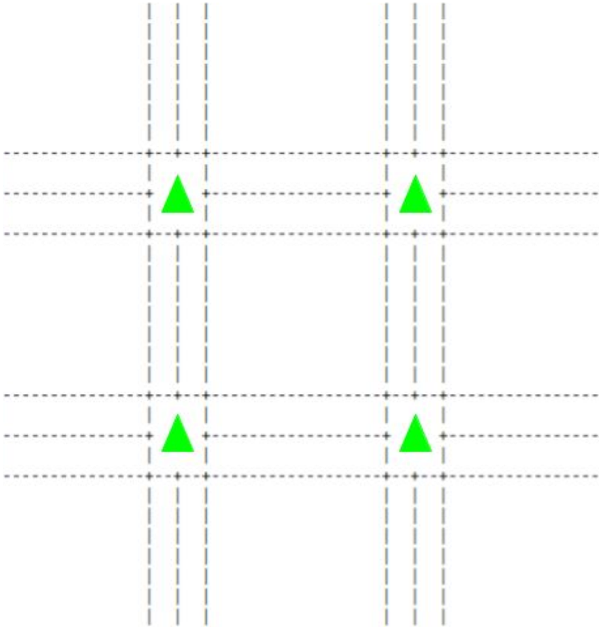
- Monolithic model takes too long to train for any topology larger than 2x2

- Goal - generalized model that:
  - outperforms Safe Controller
  - can be applied to any *n* x *m* topology
  - comparable performance to 2x2 monolithic

- Measuring performance in terms of average speed of all cars in intersection

Previous Attempts at Generalized Model

- 2x2 grid where each intersection employs the same model
  - AI learns at each intersection
  - Keeps feature vector small



- Training was unsuccessful
  - Every intersection is a corner case

Outline

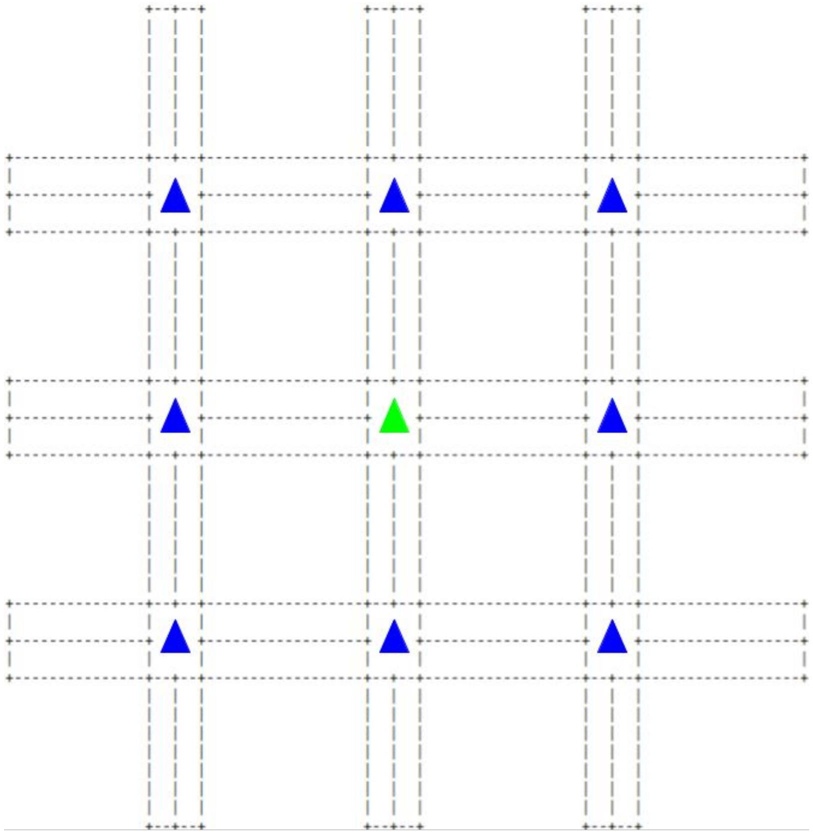
- Motivation
- Background
- Previous Work
- Problem Definition
- **Approaches and Results**
- Future Work

Key Terms

- ***NxM* training environment:** AI controller placed in the center of *NxM* grid of intersections, all other intersections safe or random
- ***NxM* evaluation environment:** trained AI controller placed at every intersection of *NxM* grid
- **Pertinent avg speed:** average speed over cars that enter edges connected to AI-controlled intersections
- **Grid padding:** add an extra layer of safe-controlled intersections around an *NxM* grid

Approach 1

- 3x3 training environment
- Fixed controllers on outer 8 intersections
- AI controller in the center



Inputs to the Model

- Modifying feature vector

- Changes to the system
  - Change number of traffic lights for AI to update
  - Update RL actions function to manually update traffic lights that aren’t at center node

Variable/Size	3 x 3 Monolithic Vals	Our Implementation Vals
Speeds	216	24
Distance to Intersection	216	24
Edge Number	216	24
Density	24	8
Velocity Average	24	8
Last Change	9	1
Direction	9	1
Currently Yellow	9	1
Total	**723**	**91**

Experiments

- Three runs with 3 x 3 grid all safe controllers to establish a baseline
- Three runs with 3 x 3 grid **all safe controllers except for center node**
- Three runs with 3 x 3 grid **all random controllers except for center node**

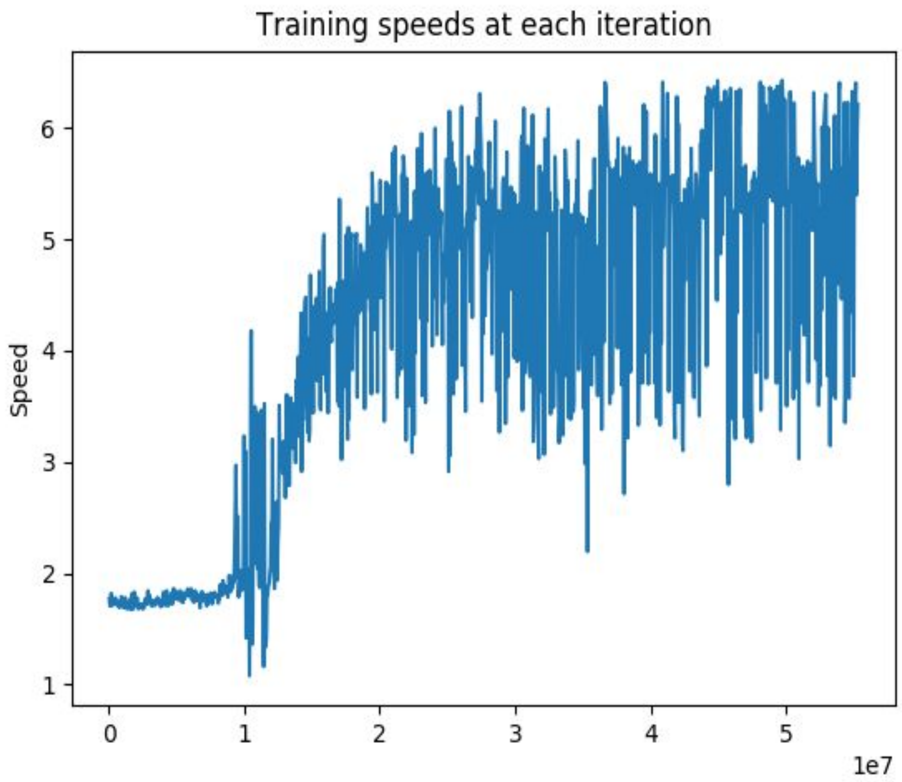
Safe Controller Baselines

- Need to compare our 3x3 results with 3x3 safe controller
- Three Safe Controller runs under different seeds
  - Result 1: Approx **5.59** m/s
  - Result 2: Approx **5.57** m/s
  - Result 3: Approx **5.55** m/s
- Average across runs: **5.57 m/s**

Monolithic 2 x 2 Baseline Results

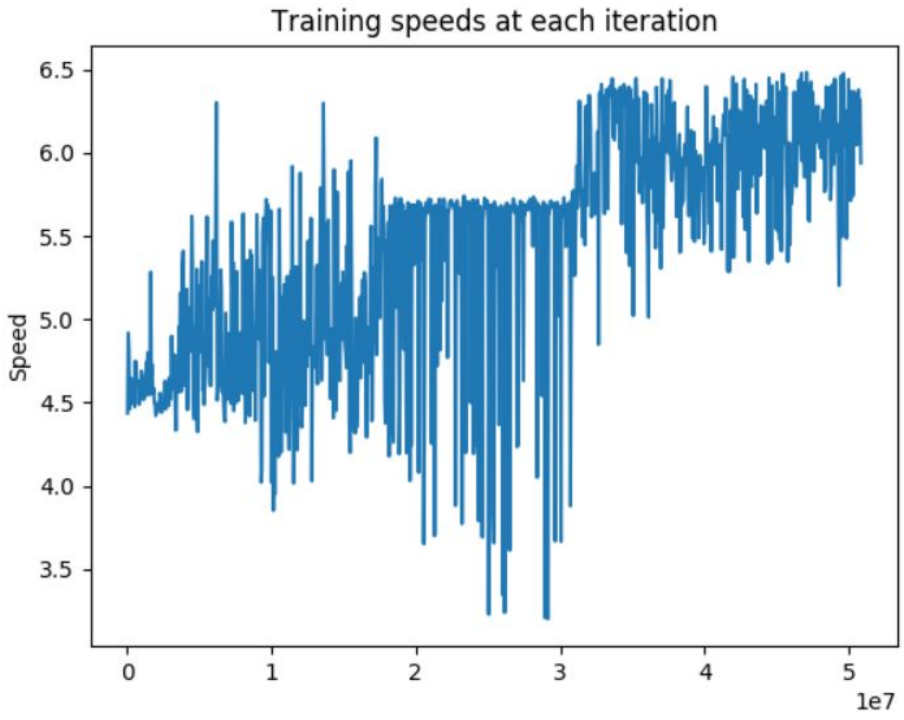
Best Average Speed:

6.429 m/s (49,700,000 steps)

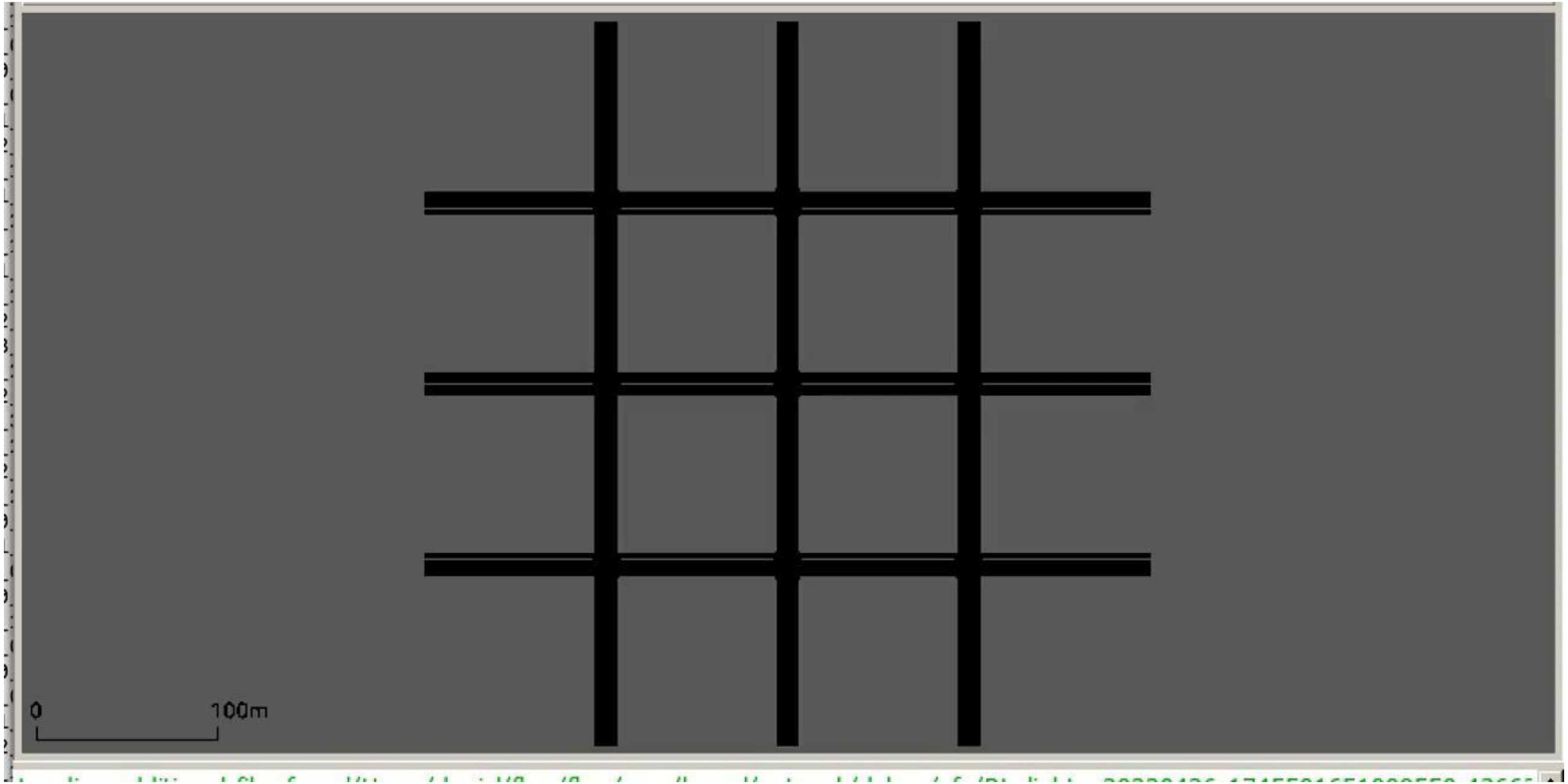


Safe Controller 3x3 Results

Best Pertinent Average Speed: 6.484 m/s (47,050,000 steps) from seed B

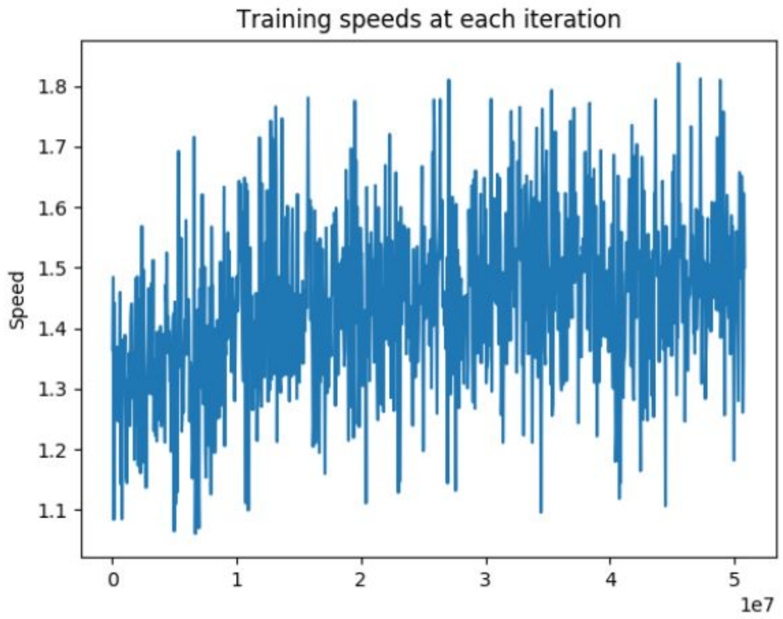


Safe Controller 3x3 Results



Random Controller 3 x 3 Results

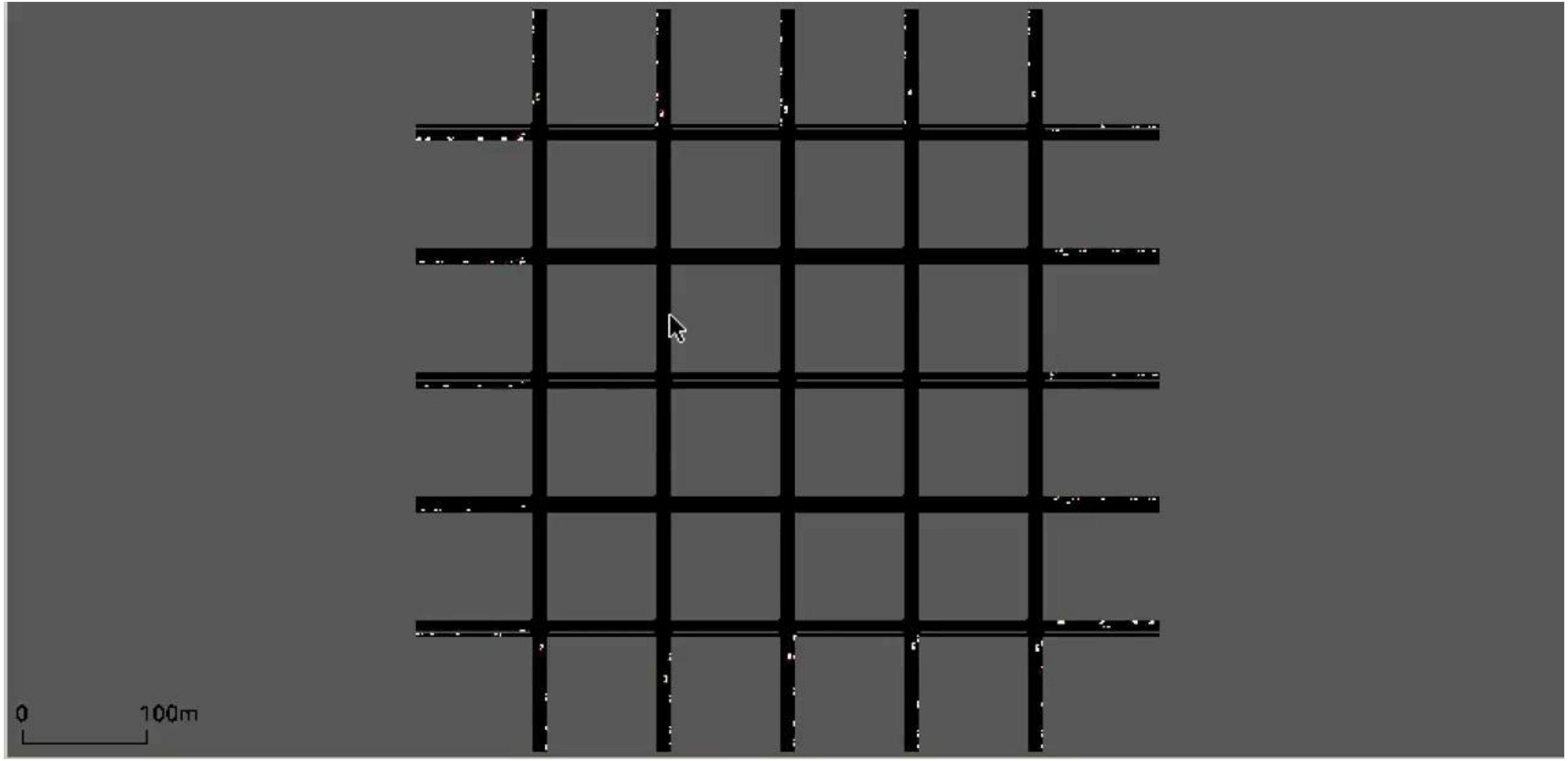
Best Pertinent Average Speed: 1.837 m/s (45,500,000 steps) from seed C



3x3 Evaluation Results

- Applied models to 3x3 evaluation environment
- Discovered models did not learn to generalize
- AIs trained on different environments and with different reported training speeds all yield ~4.3 m/s

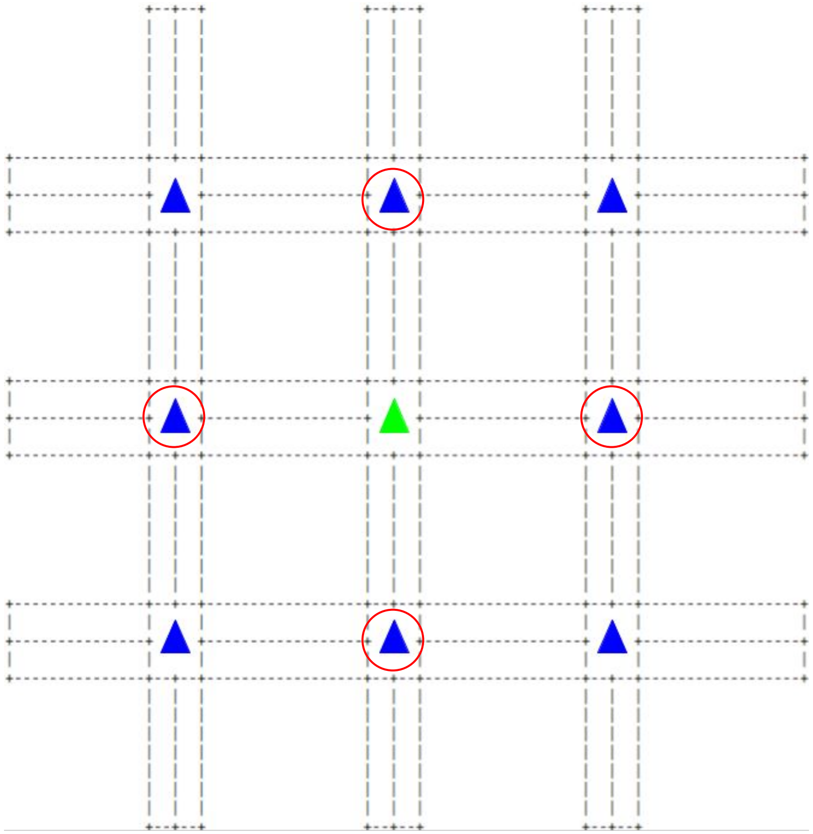
Evaluation Results



Approach 2

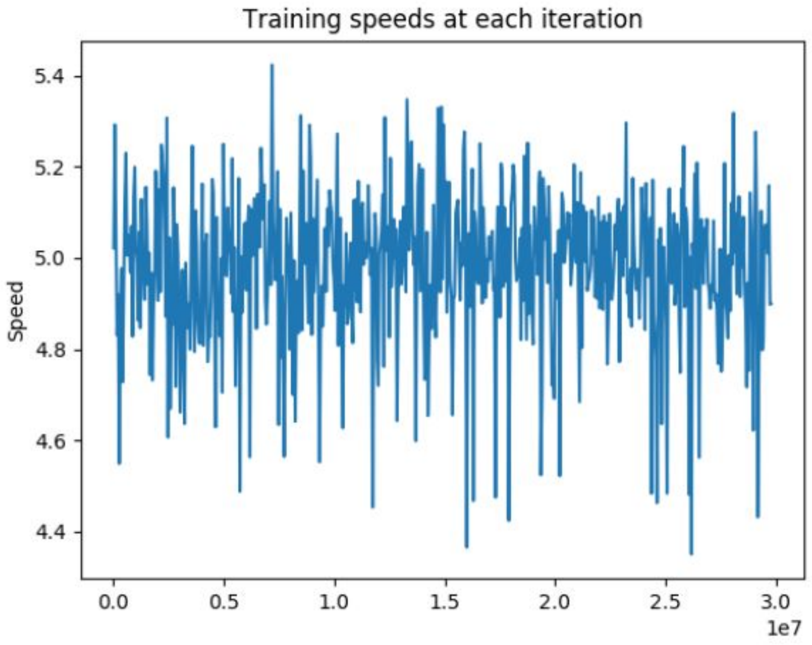
- Increase AI's observation space to "look ahead" 1 intersection
  - Work in tandem with other AIs on evaluation environment

- 5x5 training environment to avoid edge cases



Look Ahead Controller Results

Best Pertinent Average Speed:
5.423 m/s (7,200,000 steps)
from seed B



Look Ahead Evaluation Results

- Same results, failing to generalize/work with other AI controllers
- Average speed on 3x3 eval environment is 4.3 m/s

Outline

- Motivation
- Background
- Previous Work
- Problem Definition
- Approaches and Results
- **Future Work**

Future Work

- Add trained AI neighbors to training environment
- Training on multiple environments in parallel
- Replace Sumo with more efficient environment (possibly reconcile our implementations with gym-cityflow)

Questions

Thank You