WiseMove: A Framework to Investigate Safe Deep Reinforcement Learning for Autonomous Driving

Jaeyoung Lee, Aravind Balakrishnan, Ashish Gaurav,
Krzysztof Czarnecki, Sean Sedwards

University of Waterloo

September, 14th, 2019

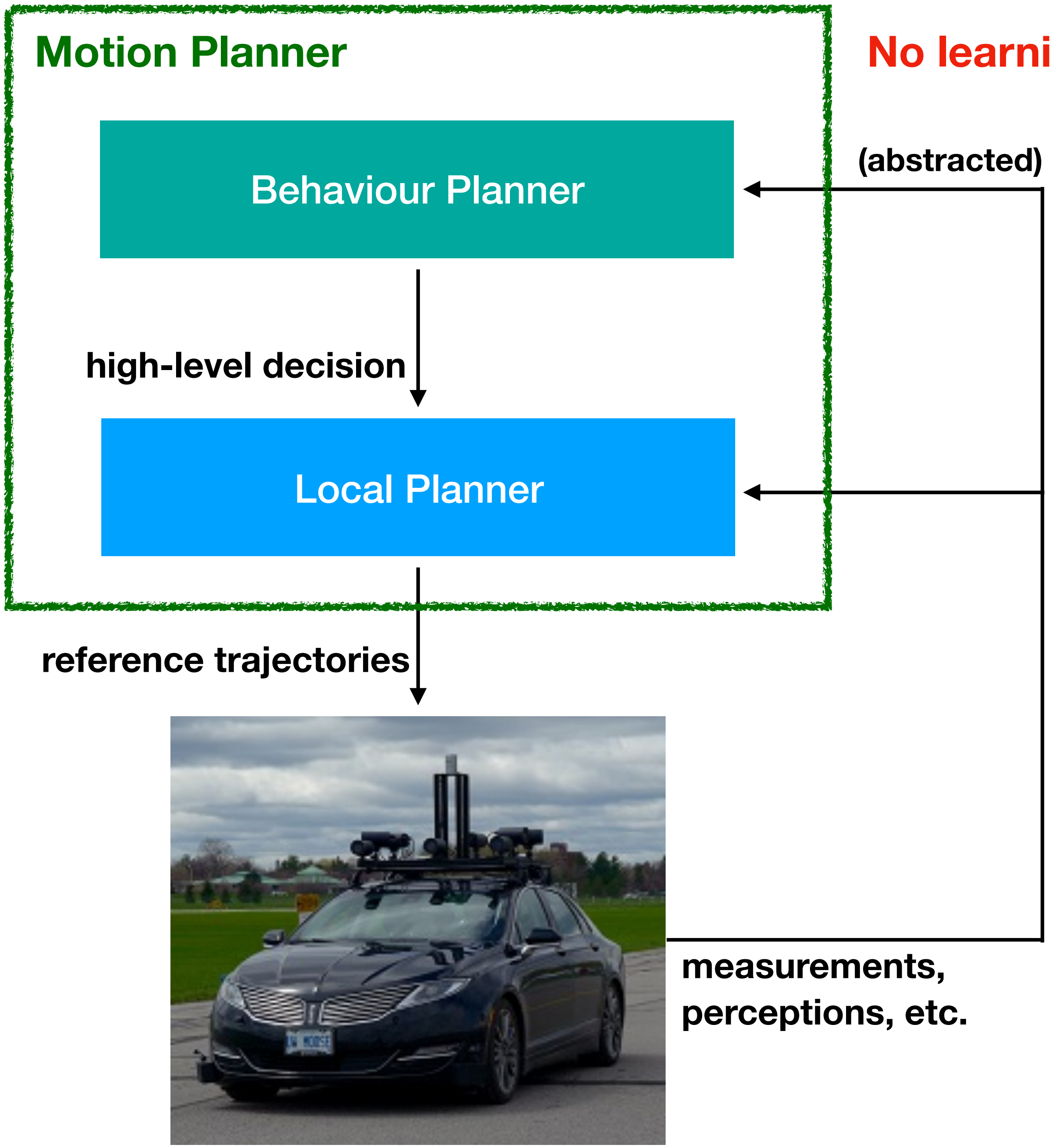
WiseMove?

- A research platform that mimics our autonomous driving stack.

- Objective: investigate the safety and performance of motion planners trained using deep reinforcement learning

- Features:
  - ✓ Hierarchical Decision Making
  - ✓ Runtime Verification
  - ✓ Reinforcement Learning / Monte Carlo Tree Search (MCTS)

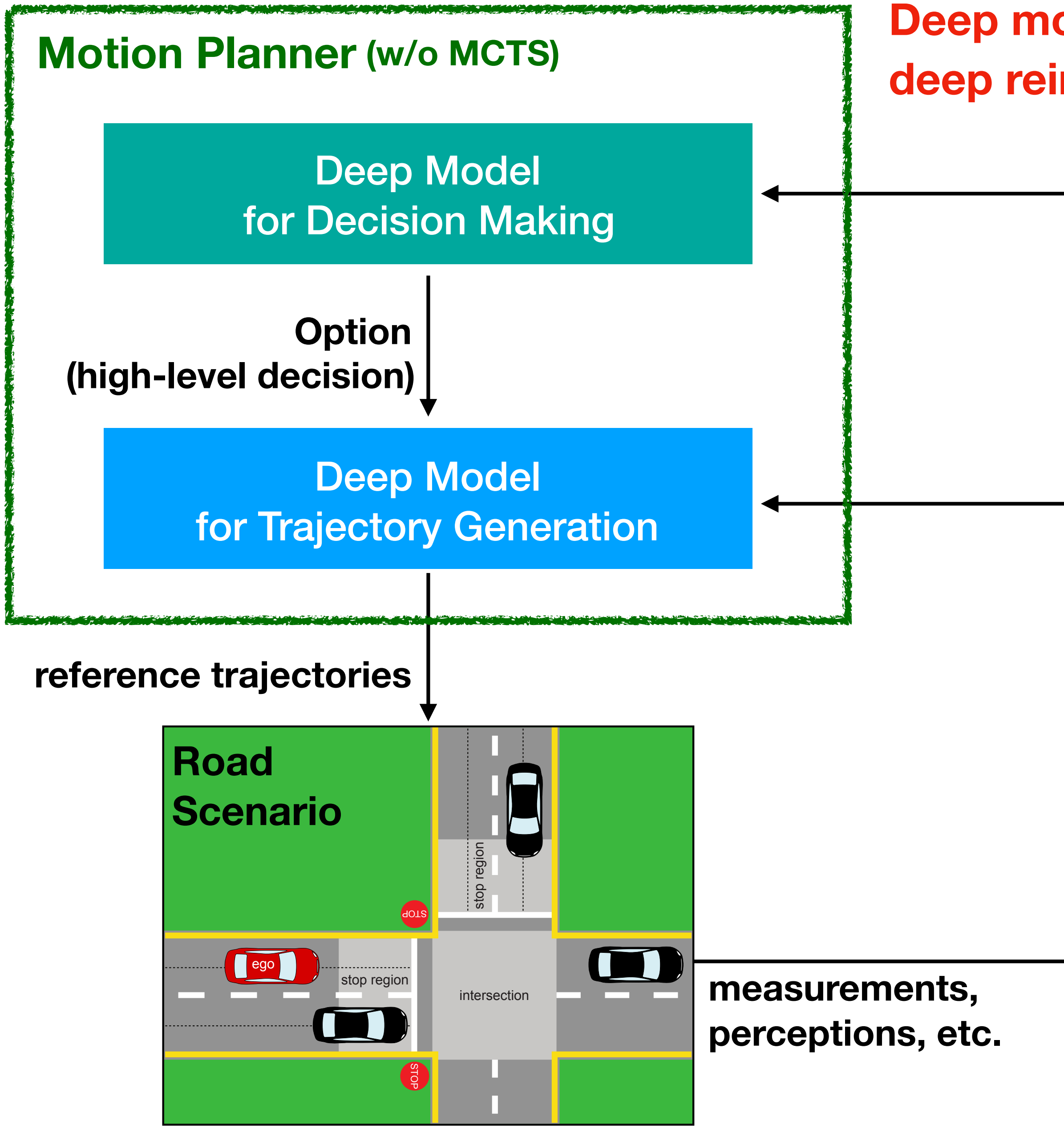
Motion Planning Architecture in 100 km Public Drive (2018)



No learning component ...

WISEMOVE Architecture

Deep models are trained by deep reinforcement learning.



WiseMove Architecture

Motion Planner (w/o MCTS)

Deep Model
for Decision Making

Option

- **Five Options:**

KeepLane, Stop, Wait, Follow, ChangeLane

- **Components**

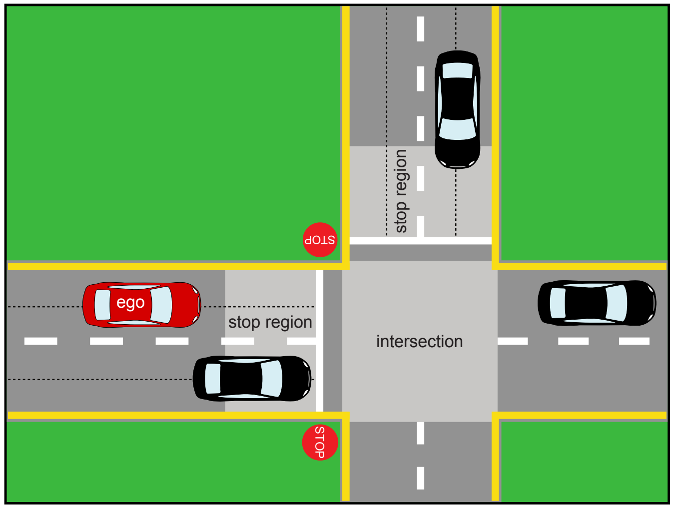
✓ speed limit, target lane

✓ time-out (e.g., 1 sec.)

✓ preconditions, e.g., in an option 'Wait',

```
G((has_stopped_in_stop_region
            and in_stop_region) U highest_priority)
```

Road Scenario



- Two "two-lane and one-way" roads
- All-ways stop implemented by the stop region
- 0~5 other vehicles

Deep Model
for Trajectory Generation

WISEMOVE Architecture

Motion Planner (w/o MCTS)

Deep Model
for Decision Making

Runtime Verifier

- **<u>Checks LTL-like strings until violated.</u>**

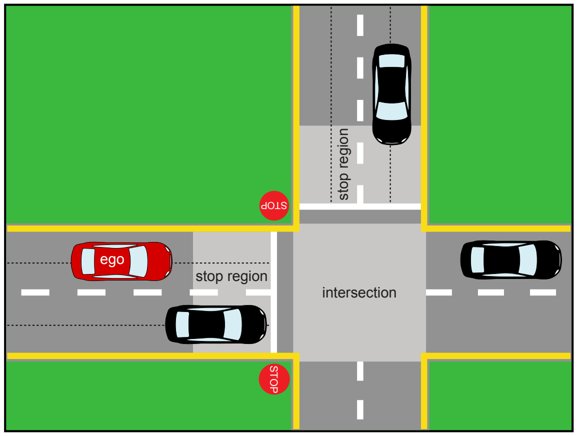
✓preconditions, e.g., in an option 'Wait',

```
G((has_stopped_in_stop_region
          and in_stop_region) U highest_priority)
```

✓traffic-rules, e.g., in a stop region,

```
G(in_stop_region =>
    (in_stop_region U has_stopped_in_stop_region))
```

Road Scenario



- **An episode ends when:**
  - **✓ Ego reaches the right end on the road,**
  - **✓ a traffic rule is violated, or**
  - **✓ a collision happens.**

Deep Model
for Trajectory Generation

WiseMove Architecture

Motion Planner **(w/o MCTS)**

Deep Model
for Decision Making

- **Choose the 'best' Option.**
  Input: a state representation
  Output: the learnt 'best' Option
- **Act upon the termination of the current Option.**

Option
(high-level decision)

Next Option?

Deep Model
for Trajectory Generation

WISEMOVE Architecture

Motion Planner (w/o MCTS)

Deep Model
for Decision Making

Option
(high-level decision)

Next Option?

Deep Model
for Trajectory Generation

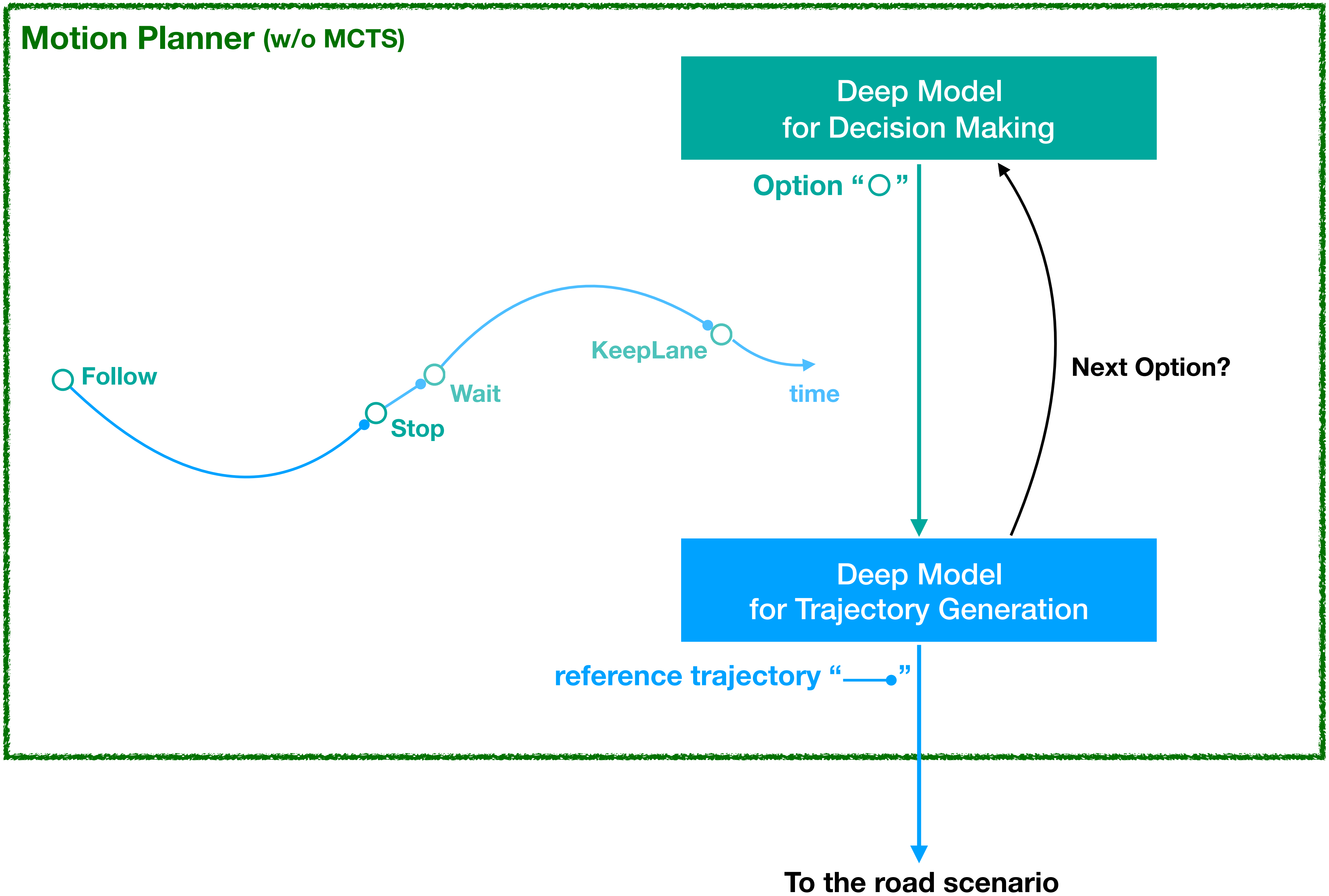
- **A deep model is stored for each Option.**

  Input: a state representation (simplified)

  Output: reference trajectories, *given an Option*

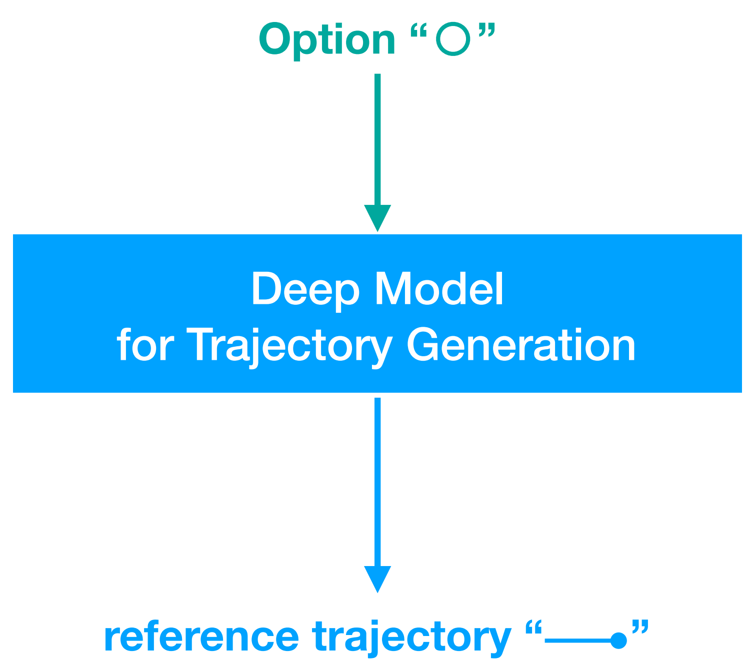
- **Trajectories generated with simplified vehicle model.**

WiseMove Architecture



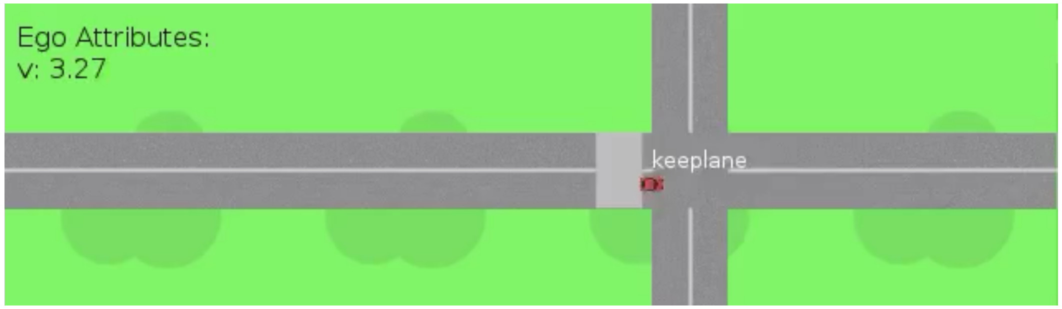
Training & Testing Low-level Deep Models

- **Five Deep Models —one for each Option.**
- **Each model**
  - ✓ **outputs continuous control commands generating the trajectories**
  - ✓ **was trained by reinforcement learning (DDPG) with**
    - ✓ **20 sec. timeout**
    - ✓ **(additional) preconditions and, if necessary, traffic rules.**

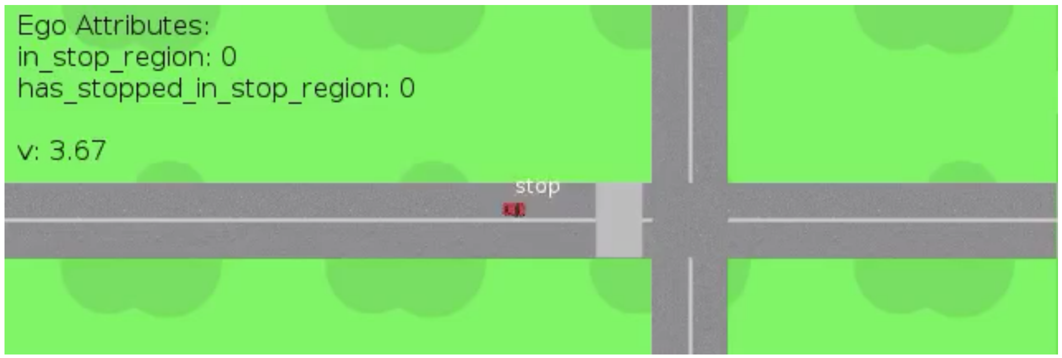


After 100,000 steps training …

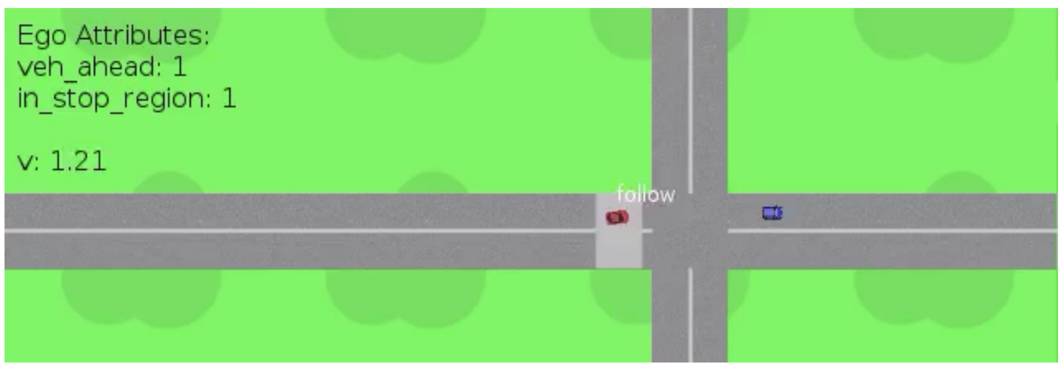
KeepLane



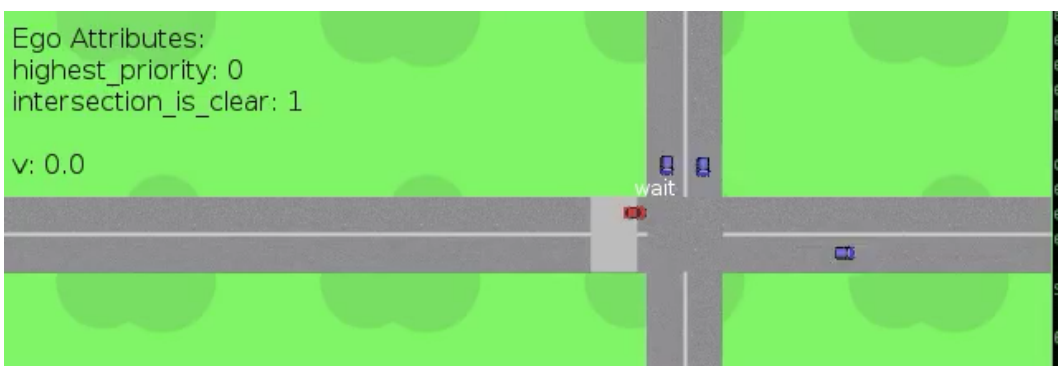
Stop



Follow

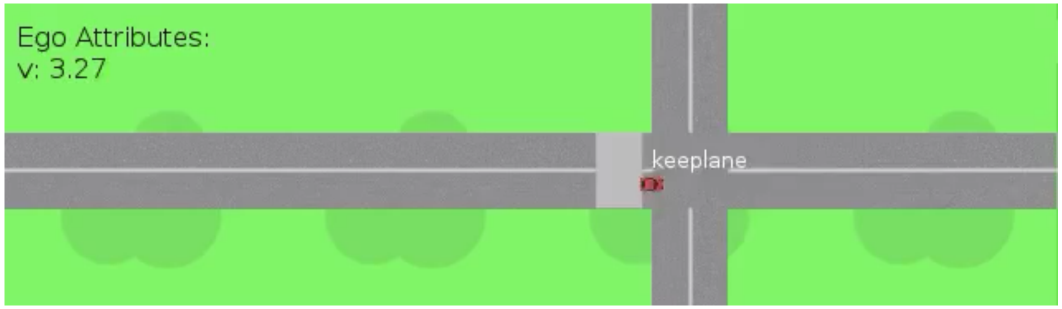


Wait



After 100,000 steps training …

KeepLane



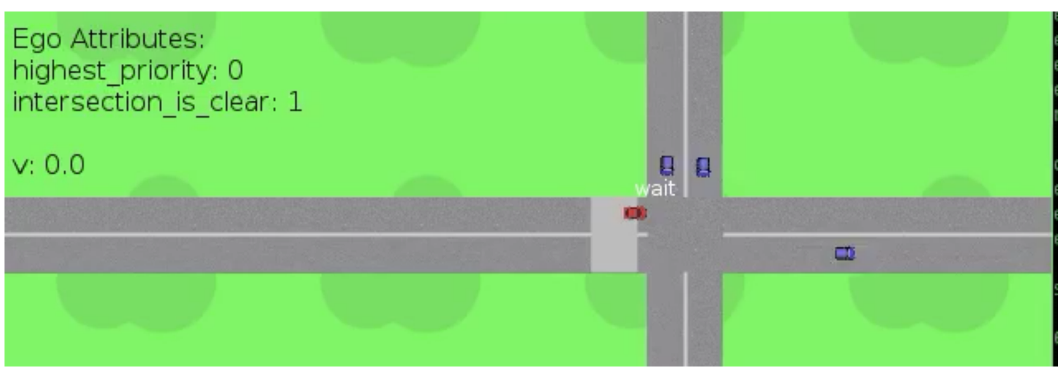
mean (std) % success after 100,000 training

KeepLane	Stop	Wait	Follow	ChangeLane
78.1 (29.4)	87.6 (20.4)	78.3 (28.8)	81.0 (15.4)	92.8 (14.3)

(averaged over 100 trials of 100 episodes)

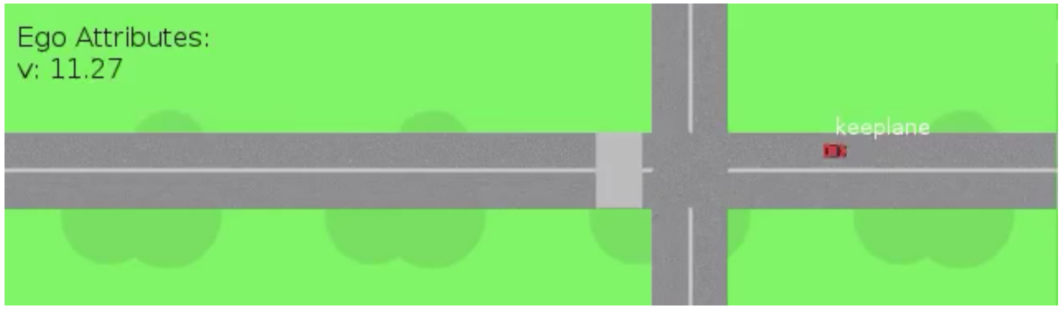
Follow

Wait



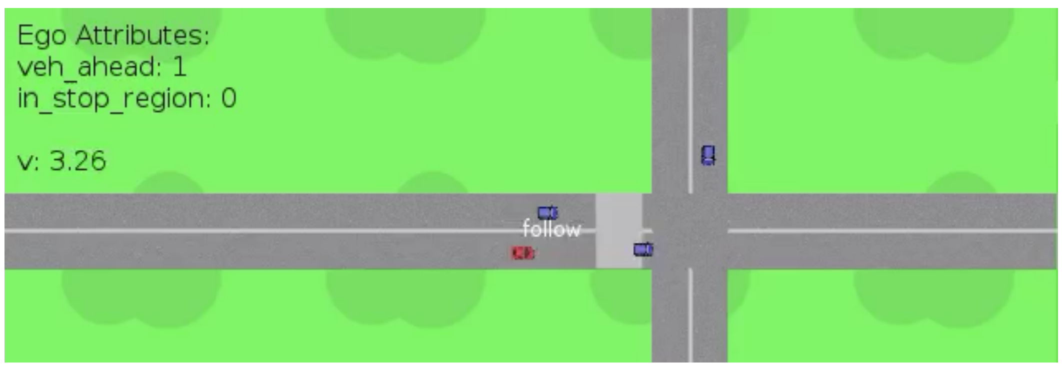
After 1,000,000 steps training …

KeepLane

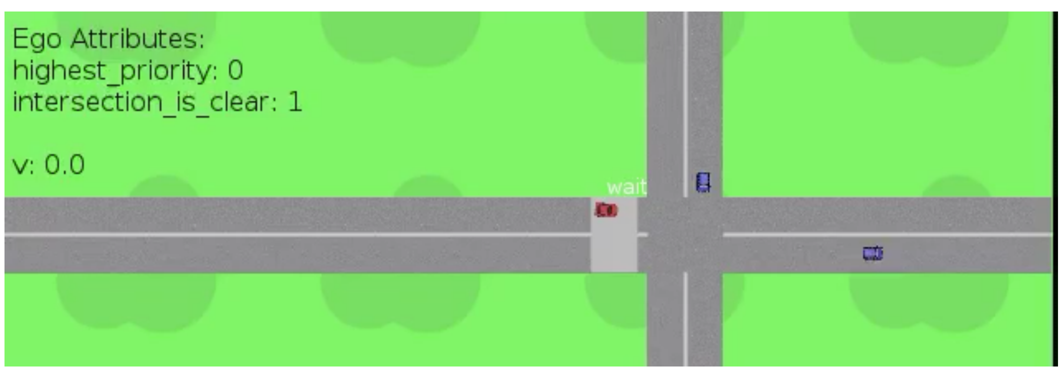


Stop

Follow

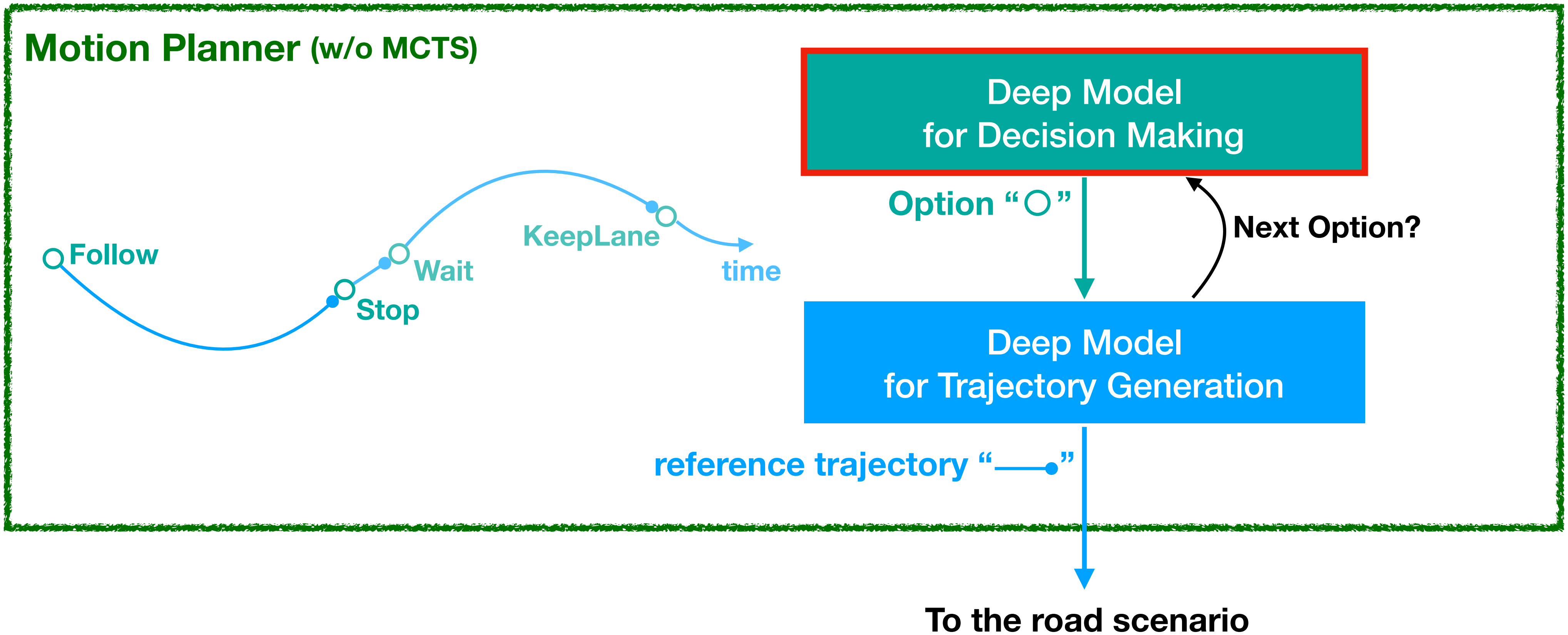


Wait



Training & Testing High-level Deep Model

- Each low-level deep model is trained *a priori* for 1,000,000 steps.
- One deep model, trained by reinforcement learning (DQN), outputs an Option.
- 1 sec. time-out for each option; 20 sec. time-out for an entire episode.

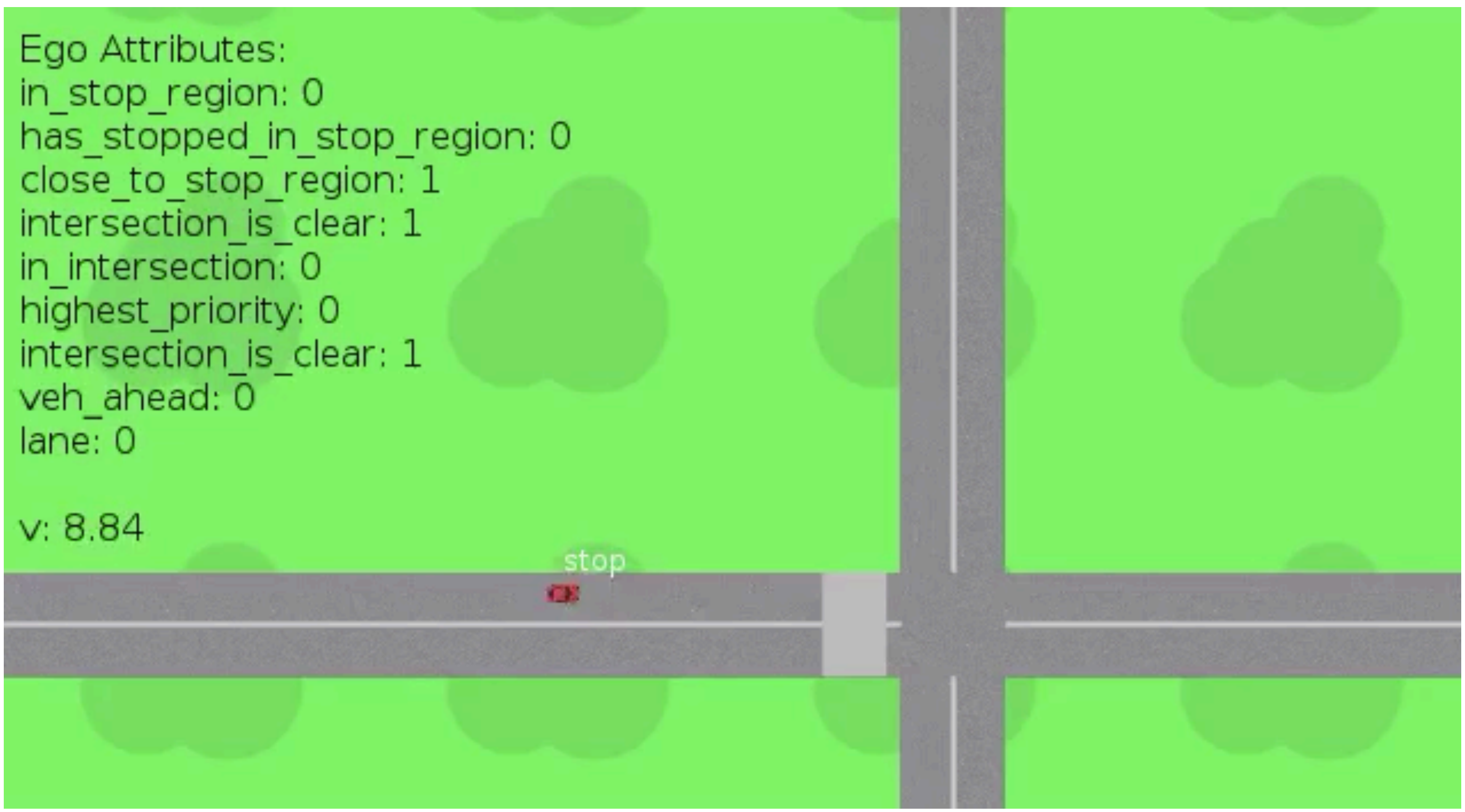


Training & Testing High-level Deep Model

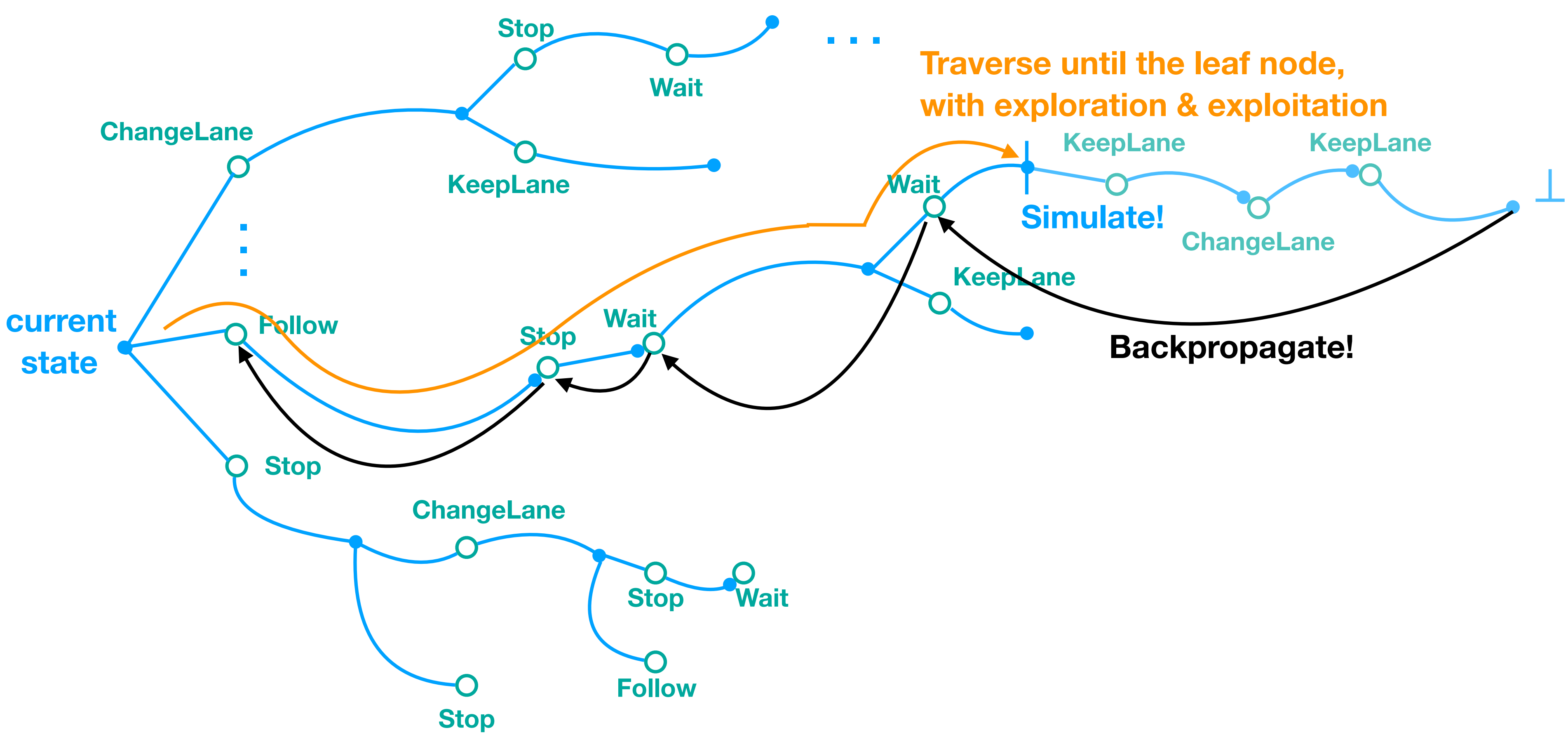
Overall performance (after 200,000 steps training)

success	LTL violation	collision
92.0 (2.0)	5.40 (1.9)	2.60 (1.6)

(averaged over 1000 episodes)



With MCTS over Options ...



Overall performance

Without MCTS			With MCTS		
success	LTL violation	collision	success	LTL violation	collision
92.0 (2.0)	5.40 (1.9)	2.60 (1.6)	98.5 (1.5)	0.9 (0.9)	0.6 (0.8)

(averaged over 1000 episodes)

Concluding Remarks

- **Features:**

Options / Reinforcement Learning / Runtime Verification / Monte Carlo Tree Search (MCTS)

- **The results are reproducible using the publicly available code at**

git.uwaterloo.ca/wise-lab/wise-move/

- **Future works**
  - ✓ **Comparisons of RL and hand-coded motion planners.**
  - ✓ **Different scenarios, realistic vehicle dynamics, etc.**
  - ✓ **Simulation-to-Real**

Thank you for attention!

Q & A

Acknowledgment

This work is supported by the Japanese Science and Technology agency (JST) ERATO project JPMJER1603: HASUO Metamathematics for Systems Design, and by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant: Model-Based Synthesis and Safety Assurance of Intelligent Controllers for Autonomous Vehicles.