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)

- Motion Planner
- Behaviour Planner
- Local Planner

high-level decision

reference trajectories

No learning component ...

(abstracted)

measurements, perceptions, etc.
Deep models are trained by deep reinforcement learning.
**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’,

\[ G((\text{has\_stopped\_in\_stop\_region} \quad \text{and in\_stop\_region}) \cup \text{highest\_priority}) \]

**Road Scenario**

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

---

**Deep Model for Decision Making**

**Deep Model for Trajectory Generation**

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

Motion Planner (w/o MCTS)

Runtime Verifier

• Checks LTL-like strings until violated.

✓ preconditions, e.g., in an option ‘Wait’,

\[
G((\text{has\_stopped\_in\_stop\_region} \\
\quad \text{and in\_stop\_region}) \cup \text{highest\_priority})
\]

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

\[
G(\text{in\_stop\_region} \Rightarrow \\
\quad (\text{in\_stop\_region} \cup \text{has\_stopped\_in\_stop\_region}))
\]

An episode ends when:

✓ Ego reaches the right end on the road,
✓ a traffic rule is violated, or
✓ a collision happens.
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
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.

Motion Planner (w/o MCTS)

Deep Model for Decision Making

Option (high-level decision)

Next Option?
WISEMOVE Architecture

Motion Planner (w/o MCTS)

Deep Model for Decision Making

Deep Model for Trajectory Generation

reference trajectory “——”

To the road scenario
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.

Diagram:
```
Option "○"
├── Deep Model for Trajectory Generation
│   └── reference trajectory “—”
```
After 100,000 steps training ...

KeepLane

Ego Attributes:
v: 3.27

Stop

Ego Attributes:
in_stop_region: 0
hasStopped_in_stop_region: 0
v: 3.67

Follow

Ego Attributes:
veh_ahead: 1
in_stop_region: 1
v: 1.21

Wait

Ego Attributes:
highest_priority: 0
intersection_is_clear: 1
v: 0.0
After 100,000 steps training …

Ego Attributes:
- $v$: 3.27

KeepLane

Ego Attributes:
- in_stop_region: 0
- has_stopped_in_stop_region: 0

Follow

Ego Attributes:
- highest_priority: 0
- intersection_is_clear: 1

v: 0.0

Wait

mean (std) % success after 100,000 training

<table>
<thead>
<tr>
<th></th>
<th>KeepLane</th>
<th>Stop</th>
<th>Wait</th>
<th>Follow</th>
<th>ChangeLane</th>
</tr>
</thead>
<tbody>
<tr>
<td>% success</td>
<td>78.1 (29.4)</td>
<td>87.6 (20.4)</td>
<td>78.3 (28.8)</td>
<td>81.0 (15.4)</td>
<td>92.8 (14.3)</td>
</tr>
</tbody>
</table>

(averaged over 100 trials of 100 episodes)
After 1,000,000 steps training ...

**KeepLane**

Ego Attributes:
- v: 11.27

**Stop**

Ego Attributes:
- in_stop_region: 1
- has_stopped_in_stop_region: 0
- v: 0.23

**Follow**

Ego Attributes:
- vehAhead: 1
- in_stop_region: 0
- v: 3.26

**Wait**

Ego Attributes:
- highest_priority: 0
- intersection_is_clear: 1
- v: 0.0
Each low-level deep model is trained \textit{a priori} for 1,000,000 steps.

One deep model, trained by reinforcement learning (DQN), outputs an \textit{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)

<table>
<thead>
<tr>
<th>Success</th>
<th>LTL Violation</th>
<th>Collision</th>
</tr>
</thead>
<tbody>
<tr>
<td>92.0 (2.0)</td>
<td>5.40 (1.9)</td>
<td>2.60 (1.6)</td>
</tr>
</tbody>
</table>

(averaged over 1000 episodes)

Ego Attributes:
in_stop_region: 0
has_stopped_in_stop_region: 0
close_to_stop_region: 1
intersection_is_clear: 1
in_intersection: 0
highest_priority: 0
intersection_is_clear: 1
veh_ahead: 0
lane: 0

v: 8.84
With MCTS over Options ...

Traverse until the leaf node, with exploration & exploitation

Overall performance

<table>
<thead>
<tr>
<th>Without MCTS</th>
<th>With MCTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>success</td>
<td>success</td>
</tr>
<tr>
<td>LTL violation</td>
<td>LTL violation</td>
</tr>
<tr>
<td>collision</td>
<td>collision</td>
</tr>
<tr>
<td>92.0 (2.0)</td>
<td>98.5 (1.5)</td>
</tr>
<tr>
<td>5.40 (1.9)</td>
<td>0.9 (0.9)</td>
</tr>
<tr>
<td>2.60 (1.6)</td>
<td>0.6 (0.8)</td>
</tr>
</tbody>
</table>

(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.