Reinforcement Learning Based Autonomous Intersection Management: A Survey

Muntasir Hossain (210041265)
Ibrahima (210041259)
Kazi Shakkhar Rahman (210041240)

Islamic University of Technology

October 13, 2025

Outline

Time to Conflict (TTC) Heuristic

- **Problem:** When should ego vehicle cross intersection?
- **Method:** Calculate time for cross-traffic to reach ego vehicle
- **Decision:** Proceed if TTC ¿ threshold for consecutive checks

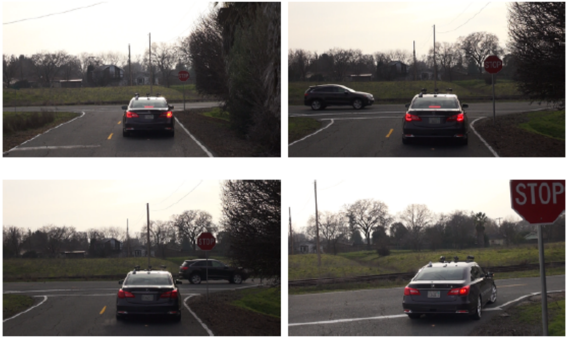


Fig. 6. Steps for handling a right turn with cross-traffic present. a) Stop sign is taken into account and the AD vehicle gets into **STOP** state before the intersection. b) Cross-traffic is detected and the AD vehicle waits until TTC algorithm deems the turn safe. c) Cross-traffic vehicle has passed the intersection, and the AD vehicle resumes to the **GO** state. d) vehicle makes the right turn.

TTC Decision Process

Parameters

- Safety threshold: 3.0 seconds
- Required: 5 consecutive safe checks
- Update rate: 10 Hz

Time (s)	**TTC Value**	**Safe?**	**Count**
0.0	2.5 s	No	0
0.1	3.2 s	Yes	1
0.2	3.1 s	Yes	2
0.3	2.8 s	No	0
0.8	4.0 s	Yes	5 $\rightarrow$ **GO**

Main Contributions

- Deep RL outperforms heuristic and rule-based approaches
- RL agents discover extrapolative measures for unforeseen circumstances

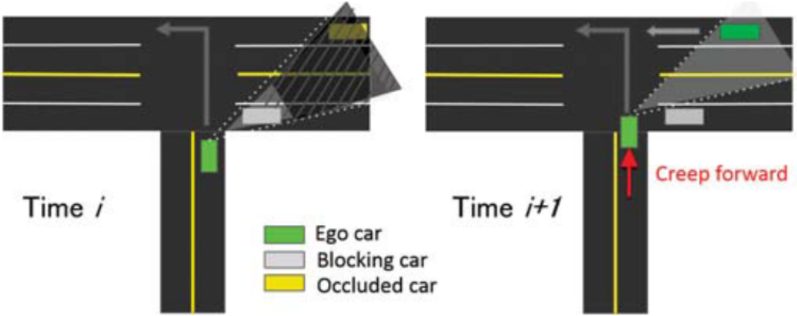


Fig. 1: Using creeping behavior to actively sense occluded obstacles. The objective is to determine the acceleration profile along the path while safely avoiding collisions.

Experiment Scenario (2018)

(a) *Right* (b) *Left* (c) *Left2* (d) *Forward* (e) *Challenge*

Fig. 2: Visualizations of intersection tasks used for our experiments.

T-Intersection with 5 scenarios:

- Right turn, Left turn (single/double lane), Straight (single/multiple lanes)

State Space (2018)

Representation

- **Perspective:** Bird's eye view
- **Space:** Discretized grid (Cartesian coordinates)
- **Vehicle Encoding:** Heading angle, velocity, occupancy indicator

DQN Time to Go: 18×26 grid

DQN Sequential: 5×11 grid

Action Space and Rewards (2018)

DQN Time to Go

Actions: {Wait, Go}

DQN Sequential

Actions: {Accelerate, Decelerate, Constant Velocity}

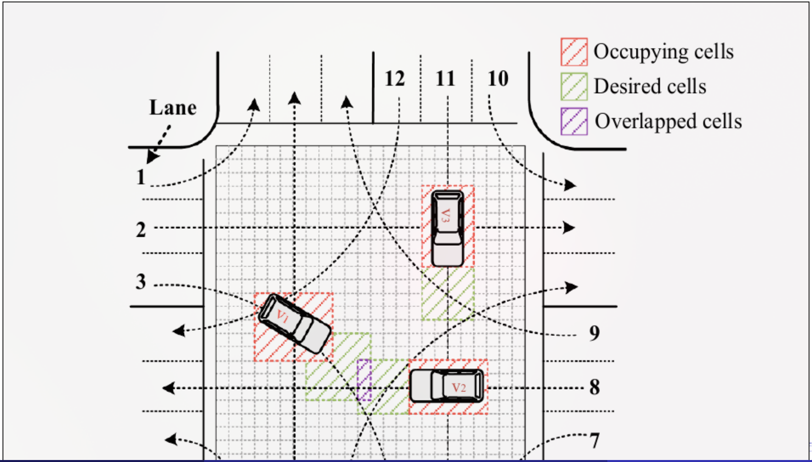
Reward Function

$$R(t) = \begin{cases} +1 & \text{success} \\ -10 & \text{collision} \\ -0.01 & \text{step cost} \end{cases}$$

Algorithm: Standard DQN

Main Contribution

Multi-agent reinforcement learning for AIM decision-making



State Representation (DCL-AIM)

Individual State Components

- Current position (occupied cells)
- Speed
- Moving intention (reserved cells ahead)
- Queue length of current lane

$$DCs = \begin{cases} max(\lceil V^2/2a \rceil, \lceil V\Delta t + \frac{1}{2} a \Delta t^2 \rceil), & V < V_m \\ max(\lceil V^2/2a \rceil, \lceil V\Delta t \rceil), & V = V_m \end{cases}$$

Action Space and Reward (DCL-AIM)

Current speed	V_0	V_m
Actions	$\{+a, 0\}$	$\{-a, 0\}$

Reward Function

Minimize intersection delay $\rightarrow$ use negative of delay as reward

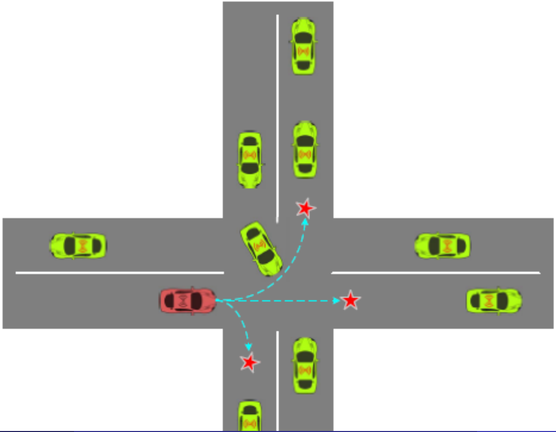
$$r(S, A) = -\sum_{v_i \in C} \left(\Delta t - \frac{L_i}{V_m} \right)$$

Algorithm: Q-learning with dual Q-table (independent + joint)

2020 - Multi-Task RL for Unsignalized Intersections

Main Contribution

Single unified learning framework for all navigation tasks



Task Representation (2020)

Component	Meaning	Values
g_l	Turn Left	0 or 1
g_r	Turn Right	0 or 1
g_s	Go Straight	0 or 1
g_c	Minimize Delay	Always 1

Table: Task Vector $G = [g_l, g_r, g_s, g_c]$

State and Action Space (2020)

State

$S = [S_e, S_1, \ldots, S_5]$

Ego vehicle: $S_e = [V_e]$

Each social vehicle: $S_i = [X_i, Y_i, V_i, \cos\theta_i, \sin\theta_i]$

Action Space

$A = [0, 3, 6, 9]$ m/s

Reward

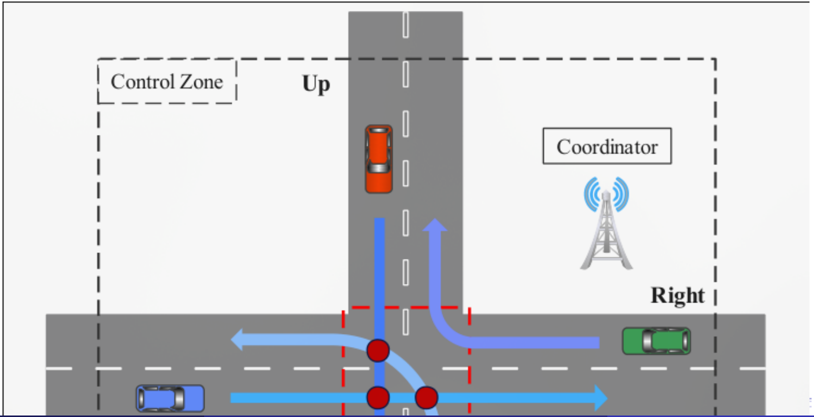
$$R(t) = \begin{cases} +50 & \text{success} \\ -500 & \text{collision} \\ -0.15 & \text{step cost} \end{cases}$$

Algorithm: Multi-task DQN

2022 - Deep RL for CAV Collaboration

Main Contributions

- Modeled as partially observable stochastic game
- Cooperative multi-agent PPO algorithm



State and Action Space (2022 - CAV)

Observation Space

$O_i = \{S_{own}, S_i, S_{i+1}, \ldots, S_m\}$

$S_{own/i} = \{v, x, y, \sin\theta, \cos\theta\}$

Action Space

Acceleration: $a_i \in \{-a_{max}, +a_{max}\}$

Position and heading control calculated automatically

Reward Function (2022 - CAV)

Centralized Reward

$r_k = w_1 R_v + w_2 R_c + w_3 R_t$

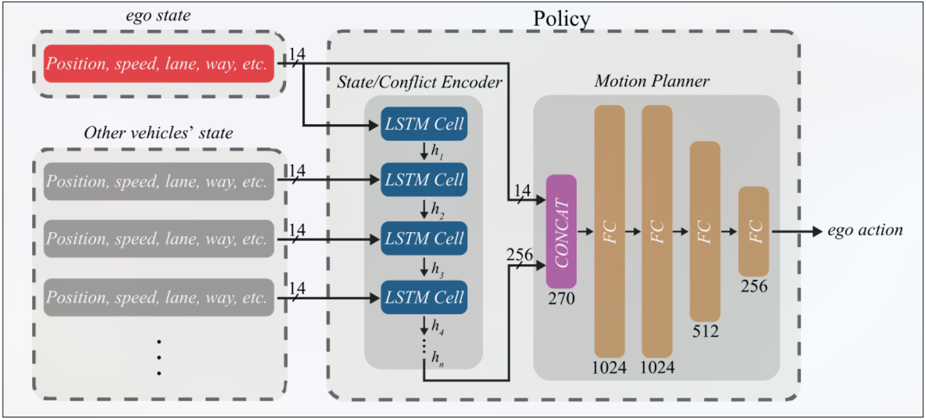
- **Efficiency:** $R_v = \sum_i^N -(\Delta t - \frac{v^i \Delta t}{V_m})$
- **Comfort:** $R_c = \sum_{i \in N} \frac{\|a_i^k\|^2}{\|a_{max}\|}$
- **Terminal:** $R_t = \begin{cases} +2 & \text{all success} \\ -5 & \text{collision} \\ 0 & \text{otherwise} \end{cases}$

Algorithm: Cooperative Multi-Agent PPO

2022 - Multi-Agent DRL for CAV Management

Main Contribution

Advanced Reinforced AIM (adv.RAIM) system



Two modules:

Action and Reward (2022 - adv.RAIM)

Action Space

Normalized speed: $a \in [0, 1]$

Denormalized to max road speed of 13.9 m/s (50 km/h)

Reward (Individual)

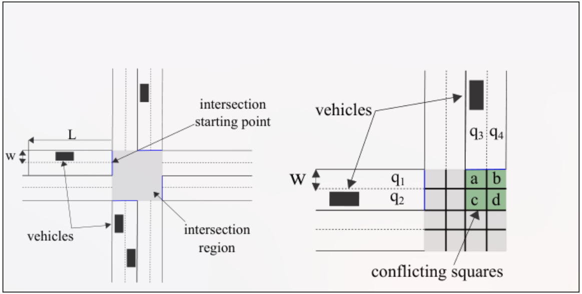
$$R = \begin{cases} -100 & \text{collision} \\ +100 & \text{crossed intersection} \\ -\text{timestep} & \text{otherwise} \end{cases}$$

Algorithm: Twin Delayed DDPG (TD3)

2022 - Real-Time Intelligent AIM

Main Contribution

Polling-based controller + RL agents for scheduled arrivals



Two tasks:

1. Reach intersection at scheduled time
2. Maintain safe distance from front vehicle

State, Action, and Reward (2022 - Real-Time)

State Space

- Current speed, distance to intersection, remaining time
- Front vehicle: speed, distance, acceleration

Action Space

$a = [-1, 0, 1]$

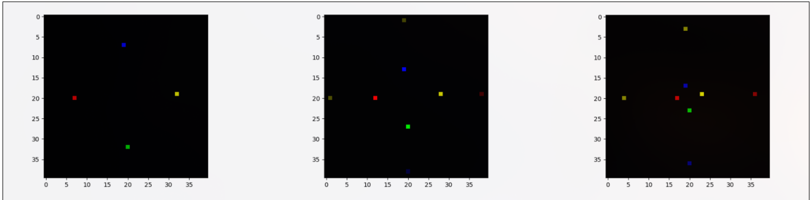
Multi-Objective Reward

$\mathbf{r} = \{r_1, r_2\}$ where r_1 for timing, r_2 for safety gap

Algorithm: Multi-Discount DQN

Main Contribution

Centralized solution using CNNs



(a) S_t (b) S_{t+1} (c) S_{t+2}

State representation: Picture-like 2D grid

- Color $\rightarrow$ vehicle route
- Luminosity $\rightarrow$ vehicle speed

Design Rationale (2023)

Why Picture Grid?

- Fixed state size regardless of vehicle count
- Conveys road geometry information

Action Space

For each lane: Issue ROW for 2 closest vehicles

Reward

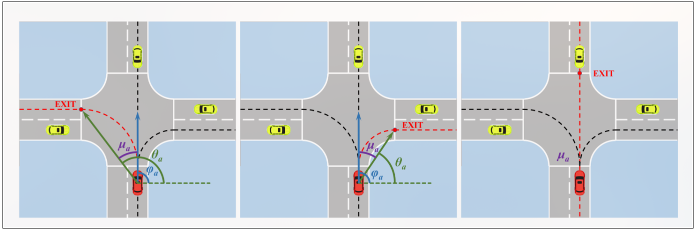
$r(s, a) = 100 \times T_{\text{avg wait}} + 10 \times n_{\text{out}}$

Algorithm: DQN with CNN

2024 - Decision-Making with Attention Mechanism

Main Contributions

- Mix-Attention neural network filters relevant information
- New state input for driving task differentiation



Improved State Space (2024)

Standard State

$$S_{ego} = \{p_{ego}, x_{ego}, y_{ego}, v_{x,ego}, v_{y,ego}, \phi_{ego}\}$$

Improved State (with driving intention)

$$S_{ego} = \{p_{ego}, x_{ego}, y_{ego}, v_{x,ego}, v_{y,ego}, \phi_{ego}, \mu_{ego}\}$$

$$\mu_{ego} = \begin{cases} |\phi_{ego} - \theta_{ego}| & \text{turn} \\ \arctan(\frac{d_{ego}}{d_{total}}) & \text{straight} \end{cases}$$

Detection: 9 closest vehicles within 48m radius

Action and Reward (2024)

Action Space

{accelerate, idle, decelerate}

$v_{\text{target}} = v_{\text{dis}} + \{1, 0, -1\}$

Reward

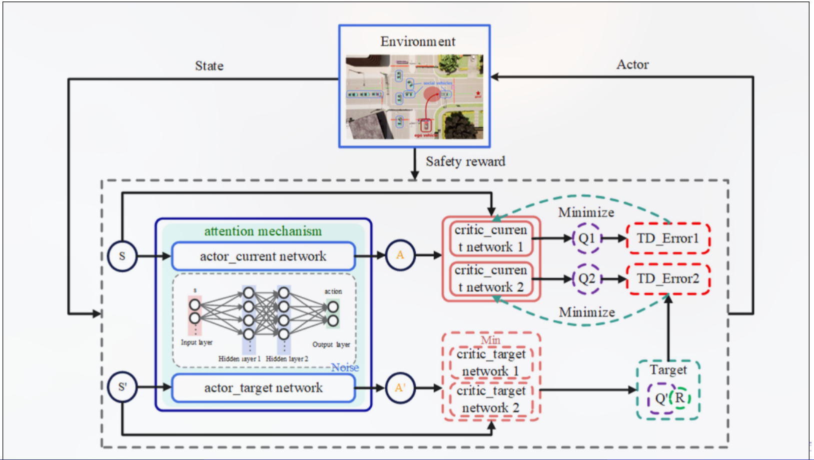
$$R = \begin{cases} 30 & \text{reached destination} \\ -120 & \text{collision} \\ 0.1 & \text{reached target speed} \end{cases}$$

Algorithm: Soft Actor-Critic (SAC)

2024 - Local Attention Safety RL (LA-SRL)

Main Contribution

Ego-attention model in actor network captures interdependencies



State, Action, and Reward (LA-SRL)

State

$S = [s_e, s_1, \ldots, s_5]$
$S_{e/i} = [v_x, v_y, x, y, \cos\theta, \sin\theta, d]$ where d = distance from risk area

Action (Continuous)

2D vector: (a_{accel}, a_{brake}) normalized to $[0, 1]$

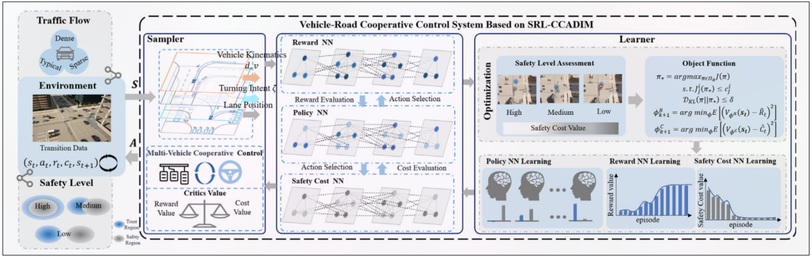
Reward

$R = \mu_r R_{risk} + \mu_a R_{avail} + R_e$ (safety + availability + events)

Algorithm: TD3

Main Contribution

Constrained Policy Optimization (CPO) ensures safety while optimizing performance



Three networks: Policy, Reward, Safety

State and Action (2025)

State

$S_i = \{d_i, v_i, \delta_i, l_i, k_i\}$

- d_i: distance to exit
- δ_i: current lane
- l_i: driving direction
- k_i: communication delay

$S = \prod_i^n S_i$

Action

$A_i = [v_i, \Delta\omega_i]$ (velocity and heading change)

Algorithm: MAPCPO (Multi-Agent Proximal CPO)

2022 - Driving Tasks Transfer using Deep RL

Main Contribution

Decision-making framework based on transfer learning + Dueling DQN

State, Action, and Reward (Transfer Learning)

State

$s_i = \{x, y, v_x, v_y\}$
State: $[s_{ego}, s_1, \ldots, s_n]$

Action

$a_t \in [-5, 0, 5]$ m/s²

Reward

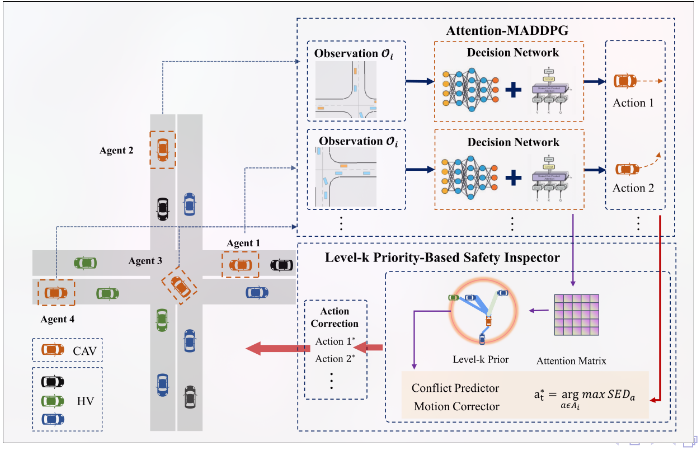
$$r = \begin{cases} 1 \cdot \text{highest-speed} - 5 \cdot \text{collision} & \text{not reached} \\ 1 & \text{reached endpoint} \end{cases}$$

Algorithm: Dueling DQN

2024 - Cooperative Decision-Making with Game Priors

Main Contribution

Multi-Agent Game Prior Attention DDPG (MAGPA-DDPG)



Summary of Approaches

Year	Method	Algorithm	Key Feature
2018	Single Agent	DQN	Occluded intersections
2019	Multi-Agent	Q-learning	Decentralized coordination
2020	Single Agent	Multi-task DQN	Unified framework
2022	Multi-Agent	MA-PPO	CAV collaboration
2022	Multi-Agent	TD3	Conflict encoding
2023	Centralized	CNN+DQN	V2X communication
2024	Single Agent	SAC	Attention mechanism
2024	Single Agent	TD3	Ego-attention safety
2025	Centralized	CPO	Safety constraints

Key Trends and Insights

Evolution of Approaches

- From single-agent $\rightarrow$ multi-agent coordination
- From discrete $\rightarrow$ continuous action spaces
- From simple grids $\rightarrow$ attention mechanisms
- From unconstrained $\rightarrow$ safety-constrained optimization

Common Elements

- State: Position, velocity, heading of ego + surrounding vehicles
- Rewards: Balance efficiency, safety, and comfort
- Increasing use of neural attention for relevant information filtering

Future Directions

- **Robustness:** Handling sensor noise and communication delays
- **Scalability:** Managing high-density traffic scenarios
- **Generalization:** Transfer learning across different intersection types
- **Safety:** Formal verification and guaranteed safety bounds
- **Human-AV Interaction:** Mixed traffic scenarios
- **Real-world Deployment:** Bridging sim-to-real gap

Thank You!
Questions?