A Deep Reinforcement Learning Approach to Roundabout Traffic Signal Control

Cheng-Yu Chen[1], Daniil Buryakov[1], Valentinus Roby Hananto[1], Victor Kryssanov[1]
[1]Graduate School of Information Science and Engineering, Ritsumeikan University, Japan

***Abstract*—Urban traffic congestion is a significant challenge for modern cities, leading to increased travel times, energy consumption, and environmental pollution. While conventional traffic signal control methods, such as fixed-time or rule-based strategies, are still widely used, they often fail to adapt to complex traffic dynamics, especially in urban roundabouts. This study proposes a deep reinforcement learning (DRL) framework for intelligent traffic signal control, specifically trained and tested in the Gongguan Roundabout in Taipei, Taiwan—an intersection with high traffic volumes and asymmetric geometry. A simulation setup is created using the SUMO software, and a Recurrent Proximal Policy Optimization (R-PPO) algorithm with LSTM memory is used to address partial observability and temporal dependencies in the traffic flow. Experiments conducted under different traffic conditions, including peak traffic hours, demand fluctuations, and incident disruptions, demonstrated that the proposed R-PPO framework significantly outperforms baseline strategies. These findings highlight the effectiveness of memory-enhanced DRL in achieving adaptive, stable, and fair traffic signal control in real-world roundabout scenarios.**

I. Introduction

Traffic congestion in urban areas has become a serious issue worldwide, caused by population growth and the rising number of private vehicles. This situation results in longer travel time, increased operating costs, and deteriorating air quality, which negatively impacts human health [1]. Numerous studies have demonstrated that effective traffic signal control is a key solution to alleviating congestion and improving urban mobility [2], [3]. In recent years, the advent of new communication technologies, such as the Internet of Things (IoT), has enabled vehicles and roadside units to exchange information rapidly [4]. These technologies provide a foundation for intelligent traffic control, and many adaptive traffic control systems (ATCS) have been developed, for example, InSync [5], RHODES [6], and systems based on decentralized cycle-free methods [7]. However, as ATCS rely on rule-based models designed by domain experts, even slight changes in traffic patterns can lead to significant drops in their performance. To overcome this limitation, reinforcement learning (RL) [8], [9] has emerged as a powerful alternative, where an agent learns a control policy through trial-and-error interactions with the environment. Most existing RL studies, however, were conducted on simple intersections with synthetic traffic flows. In contrast, roundabouts are inherently more complex due to their circular layout, multiple entry lanes, and high merging pressure. Additionally, in many Asian cities, motorcycles constitute a large portion of traffic flow, further increasing the unpredictability and difficulty of effective control.

To address the issues outlined above, this study focuses on the Gongguan roundabout in Taipei, Taiwan. This roundabout features six entry and exit approaches and handles over 10,000 vehicles during peak hours. Notably, motorcycles account for approximately 76% of the total traffic flow. According to government records, Gongguan has been ranked as the most accident-prone intersection in Taipei for seven consecutive years [10], [11]. The complex configuration and high traffic volume make Gongguan an ideal, yet highly challenging case for advanced traffic signal control. Therefore, this study aims to develop a more adaptive and intelligent control framework. The authors aimed to develop a deep reinforcement learning (DRL) system capable of efficiently managing traffic signals in roundabout environments that would help reduce vehicle delay, improve speed, and control the traffic stability.

II. Related Work

The earliest AI applications in traffic signal control involved fuzzy logic, artificial neural networks (ANNs), and basic reinforcement learning, each offering distinct advantages. Fuzzy logic [12] allows for the representation of imprecise and qualitative knowledge, enabling systems to mimic human decision-making. However, fuzzy systems require rules designed by experts, and cannot adapt over time, limiting their effectiveness in highly dynamic environments. ANNs [13] provide function approximation capabilities, mapping traffic states to control actions, based on historical data. They can learn complex nonlinear relationships in traffic behavior, and were initially applied in supervised learning settings for signal timing prediction and classification. As for reinforcement learning, early work with tabular RL methods, such as Q-learning [14], [15], was aimed at determining action-value functions and then selecting actions by explicit maximization. The RL studies demonstrated that AI agents could achieve adaptive control policies without pre-defined rules or labels.

To handle high-dimensional state spaces in traffic control, deep Q-Network (DQN) [16]–[18] uses a neural network to replace the Q-table and applies target networks and experience replay to stabilize learning. Various actor-critic methods, including A3C [2], [19], DDPG [20], [21], and PPO [22], [23] were developed to allow the agent to learn policies and value functions separately, supporting continuous control and improving stability.

In these and many other DRL-based traffic signal control methods, agents operate under partial observability. For example, upstream congestion or residual queues may not be

979-8-3315-7206-8/25/$31.00

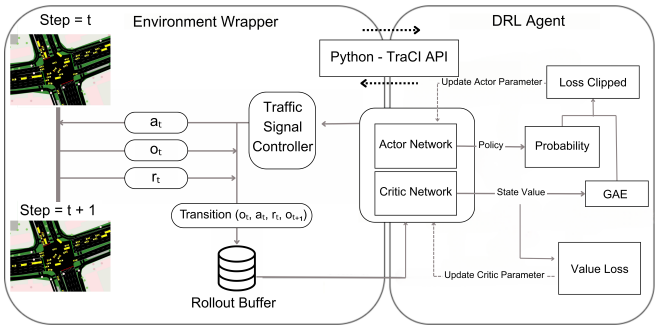


Fig. 2: Proposed system architecture.

TABLE I: Notation used in the DRL training framework

Symbol	Definition
o_t	Observation vector at timestep t (e.g., lane occupancy, queue length).
a_t	Action at timestep t (e.g., phase decision).
r_t	Reward signal based on traffic performance after executing a_t.
o_{t+1}	Observation after environment transition following a_t.
(o_t, a_t, r_t, o_{t+1})	Transition tuple stored in the rollout buffer.
$\{o_0, ..., o_t\}$	Historical observation sequence used by the policy.
$\pi_\theta(a_t \mid o_{\leq t})$	Stochastic policy with parameters θ generating actions.
$V_\phi(o_t)$	Value function estimating return from o_t.
A_t	Advantage estimate computed using GAE.

directly measurable, resulting in a partially observable Markov decision process (POMDP), where the agent must infer hidden dynamics from limited observations. Standard feedforward models, such as DQN and PPO, are not well-suited to these settings. To address the predicament, recurrent neural networks (RNNs) can be integrated into RL architectures, allowing agents to retain and reason over the past observations. Recurrent Proximal Policy Optimization (R-PPO) [24], [25], which incorporates LSTM layers into both the actor and critic networks, enables policy updates based on the current inputs and past states.

III. Proposed Method

Fig. 1 illustrates the closed-loop interaction of traffic signal control using deep reinforcement learning. The DRL agent continuously interacts with the environment in three sequential steps:

1) **Observation:** The agent receives the current traffic state.
2) **Action:** Based on the observation, the agent selects an action, such as switching the traffic light phase or extending the current (green) phase.
3) **Reward:** After the action is applied, the environment updates and returns a reward signal to evaluate the action. The agent then updates its policy, based on this feedback.

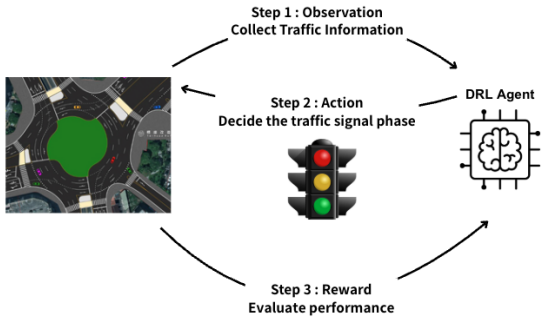


Fig. 1: Conceptual loop of DRL-based traffic signal control.

The proposed system integrates the traffic simulator SUMO [26] with a reinforcement learning agent, based on Recurrent Proximal Policy Optimization (R-PPO). SUMO provides for detailed urban traffic simulations, while the R-PPO agent learns to control traffic signals by interacting with the simulated environment. Fig. 2 illustrates the proposed system architecture, and Table I lists the notations used. The system operates in a closed loop where the SUMO simulator executes one step ($t \rightarrow t+1$) to update vehicle and signal states, followed by the environment module collecting traffic features to produce the observation vector o_t. The R-PPO agent utilizes the observation history $\{o_0, \ldots, o_t\}$ to generate an action a_t by applying a stochastic policy $\pi_\theta(a_t|o_{\leq t})$, which is then realized in SUMO via the TraCI API. After executing the action, the environment transitions to the next state o_{t+1} and provides the reward r_t, reflecting the traffic performance. The tuple (o_t, a_t, r_t, o_{t+1}) is stored in the rollout buffer, and after collecting a batch, the agent updates its policy π_θ and value function $V_\phi(o_t)$, using PPO with advantage estimates A_t computed through Generalized Advantage Estimation (GAE).

A. Reinforcement Learning Component Design

At each simulation timestep t, the environment provides an observation vector $o_t \in \mathbb{R}^{26}$ that captures traffic and signal-related information. This includes 18 approach-level features (lane occupancy, vehicle inflow, and speed for six approaches), 3 ring-level features (mean occupancy ring, and the 25th and 75th percentile velocities on the ring), and 5 phase-related features (the phase mask, timer ratio, and episode progress). These are arranged as follows:

$$o_t = \big[\underbrace{o_{\text{occ}}^{(1)},\ o_{\text{flow}}^{(1)},\ o_{\text{speed}}^{(1)},\ \ldots,\ o_{\text{occ}}^{(6)},\ o_{\text{flow}}^{(6)},\ o_{\text{speed}}^{(6)}}_{\text{Approach-level (18D)}},$$
$$\underbrace{o_{\text{ring\_occ}},\ o_{\text{speed\_p25}},\ o_{\text{speed\_p75}}}_{\text{Ring-level (3D)}},$$
$$\underbrace{o_{\text{mask}},\ o_{\text{timer}},\ o_{\text{progress}}}_{\text{Phase-related (5D)}}\big] \tag{1}$$

At each timestep, the agent selects a discrete action $a_t = \phi_t$, where $\phi_t \in \{0, 1, 2\}$ indicates the target green phase index. A

currently active phase holds for at least 15 seconds before switching and is forcibly switched after 30 seconds to prevent starvation. A 3-second yellow buffer is applied between green phases to ensure safety during transitions.

The reward at timestep t is defined as:

$$r_t = r_{\text{outflow}} + r_{\text{speed}} + r_{\text{slow}} + r_{\text{qmax}} + r_{\text{switch}},$$

where, r_{outflow} increases when more vehicles leave the roundabout, r_{speed} reflects the average speed of all vehicles, and r_{slow} is a penalty term for when many vehicles move under 2 m/s. r_{qmax} is also a penalty for when the queue is long, while r_{switch} when the agent changes the phase. The latter penalty becomes greater during training to make the agent learn to switch only when needed.

B. Recurrent Proximal Policy Optimization

Recurrent Proximal Policy Optimization (RPPO) is employed to handle sequential traffic signal control under partial observability, which is a common condition in urban environments with evolving queues and delayed vehicle responses. In terms of RPPO implementation, this study deployed Stable-Baselines3 [27] for its modular PPO framework with LSTM policy. The policy network (Fig. 3) receives the observation vector o_t, which is first processed by a multi-layer perceptron. The encoded features are passed to the LSTM module and then forwarded to two processing heads, the Actor Head and Critic Head. Actor Head produces a categorical distribution over three traffic signal phases, from which an action $\phi_t \in \{0, 1, 2\}$ is sampled at each decision step. Critic Head outputs a state-value estimate $V(o_t, h_t)$ to evaluate expected future returns, which helps assess the advantages during updates. The training procedure of RPPO is detailed in Algorithm 1. In each timestep, the agent interacts with the SUMO environment and collects the transition data. These data are stored in a recurrent rollout buffer for training. After collecting enough samples, the Generalized Advantage Estimation (GAE) method is used to calculate the advantages $\hat{A}_t$ and the target returns $\hat{R}_t$. The agent then updates the policy and value networks by using the clipped surrogate loss and the value loss functions. An entropy term is added during training to encourage the agent to explore different actions.

IV. Experiments

A. Network and Traffic Settings

The simulated Gongguan Roundabout has six arms with lane configurations extracted from OpenStreetMap and adjusted, using SUMO's `NETEDIT` tool, to accurately reflect the real-world geometry 4. Traffic demand is modeled based on morning peak-hour survey data provided by the Taipei City Traffic Engineering Office. The vehicle mix includes motorcycles, cars, trucks, and buses, each assigned specific size and speed parameters. Vehicles are generated using `.rou.xml` files with probabilistic arrival patterns. To introduce variability across episodes while maintaining consistent demand ratios, different random seeds are applied during traffic generation.

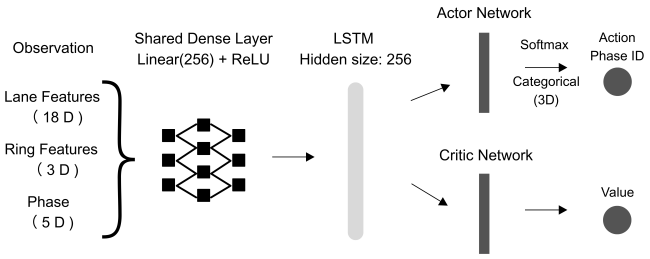


Fig. 3: RPPO policy network with observation encoder, LSTM memory, and actor-critic heads.

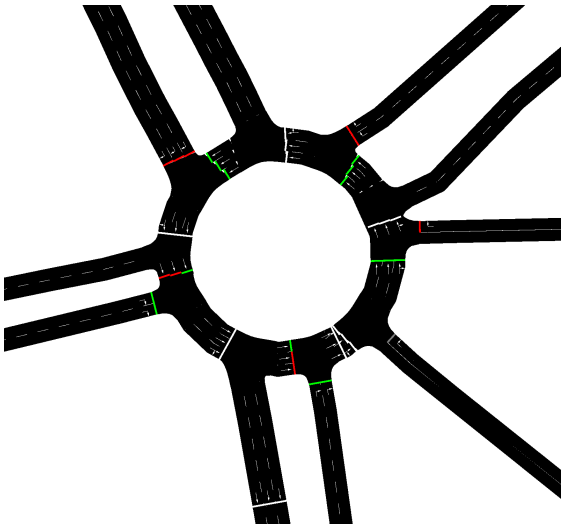


Fig. 4: Simulated Gongguan Roundabout.

B. Evaluation Scenarios

Two scenarios are designed to evaluate the proposed RPPO traffic signal control:

1) **High-Demand Scenario:** In this scenario, the overall traffic is increased to 130% of the baseline morning peak-hour traffic. Additionally, a noise of 10% is applied to the arrival rates for each route to introduce fluctuations in vehicle arrivals. This scenario evaluates the trained model's ability to handle high and varying traffic demand while maintaining a stable signal control performance.
2) **Accident Scenario:** This scenario introduces a realistic incident by randomly blocking one lane. The closure lasts for 15 minutes. This scenario tests the model's adaptability under partial road blockage and its ability to maintain traffic flow during unexpected disruptions.

C. Baselines

To evaluate the proposed RPPO method, three baseline methods are used for comparison. All reinforcement learning baselines use the same environment settings, observation space, reward structure, and action space to ensure fair

Algorithm 1: Recurrent PPO Training

Initialize: For each episode, generate a new dynamic `rou.xml`. Actor network $\mu_\theta(o_t, h_t^{\text{actor}})$ with parameters θ, and a recurrent critic network $V_\phi(o_t, h_t^{\text{critic}})$ with parameters ϕ. Recurrent rollout buffer $\mathcal{B} = \{(o_t, a_t, r_t, o_{t+1}, d_t, h_t^{\text{actor}}, h_t^{\text{critic}})\}_{t=0}^{T}$

for *episode* $e = 1$ *to* E **do**

  for *rollout step* $t = 1$ *to* T **do**

    Observe normalized state o_t and hidden state h_t.

    Select action $a_t \sim \pi_\theta(a_t \mid o_t, h_t)$.

    Execute a_t in SUMO, receive reward r_t, next state o_{t+1}.

    Store $(o_t, a_t, r_t, h_t, V_\phi(o_t))$ in the rollout buffer.

  Compute GAE advantages $\hat{A}_t$ and returns $\hat{R}_t$.

  for *each PPO update epoch* **do**

    Update policy θ by minimizing:

$$L^{\text{clip}} = \mathbb{E}_t\big[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\big],$$

    where $r_t(\theta) = \frac{\pi_\theta(a_t|o_t,h_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t,h_t)}$.

    Update value function ϕ by minimizing:

$$L^{\text{vf}} = \mathbb{E}_t\big[(V_\phi(o_t) - \hat{R}_t)^2\big].$$

    Add entropy bonus:

$$L^{\text{ent}} = \mathbb{E}_t\big[\mathcal{H}(\pi_\theta(\cdot \mid o_t, h_t))\big].$$

    Perform gradient updates:

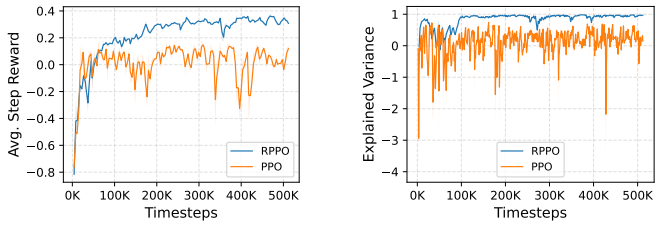
$$\theta \leftarrow \theta - \eta\nabla_\theta(L^{\text{clip}} - \beta L^{\text{ent}})$$

$$\phi \leftarrow \phi - \eta_v \nabla_\phi L^{\text{vf}},$$

    where η and η_v are learning rates, and β is the entropy coefficient.

Output: Trained Recurrent PPO policy π_θ, value function V_ϕ.

comparison. The fixed-time control applies a 30-second green time for each phase without real-time adaptation, serving as a lower-bound reference. The Double Deep Q Network (DDQN) baseline is a value-based reinforcement learning method where the agent selects phases by using double Q-learning to learn Q-values and reduce overestimation during updates. The Proximal Policy Optimization (PPO) adopts a policy gradient-based structure. The agent learns a stochastic policy, using a clipped surrogate objective, allowing for stable updates. This comparison isolates the effect of the recurrent structure in the proposed RPPO. All the above methods are trained under the same simulation settings, using consistent random seeds across runs, with the same traffic variations. The same reward structure is used to ensure that differences in results come from algorithm



(a) Training curves of reward.

(b) Explained variance of training.

Fig. 5: Comparison of RPPO and PPO performance.

differences, but not from reward shaping. As for the evaluation metrics, each method is characterized in terms of average speed and travel time to measure the traffic efficiency. The average queue length is used to assess the congestion levels, and the average step reward to monitor the learning progress during training. The average phase duration is taken as an indicator for the phase-switching behavior of the signal controller. The source code and experimental scripts could be obtained from https://github.com/chenyuuuuuu/DRL_Gongguan

V. Results

Fig. 5a and Fig. 5b present training results of RPPO and PPO under the morning peak-hour traffic scenario. RPPO gradually increases the average step reward to approximately 0.3 after 200K timesteps and maintains it between 0.2 and 0.4, whereas PPO fluctuates near zero with higher variance. This suggests that RPPO achieves more stable policy updates during training. It also maintains a higher explained variance, stabilizing around 0.8-1.0 after 250K timesteps, while PPO remains lower at approximately 0.3-0.7. A higher explained variance means that the value function predictions are closer to the target returns, implying that RPPO provides for more accurate advantage estimates and more reliable gradient updates.

Tables II and III show results obtained for the accident and high-demand scenarios, respectively. In both scenarios, RPPO demonstrated the best performance. It yielded the highest average step reward, the lowest average queue lengths, the shortest travel times, and the highest average speeds. In comparison, PPO and DDQN showed moderate improvements but failed to reach optimal performances. Fixed-Time Control performed the worst, especially under the high demand conditions. When the overall traffic demand increased to 130%, the network became nearly deadlocked, with new vehicles attempting to enter the roundabout while the vehicles already there could not leave. These results showed that RPPO effectively reduces congestion and improves traffic flow in both the accident and high-demand scenarios.

A. Discussion

Results obtained in the experiments confirmed that traffic exhibits temporal patterns, such as gradual increases in flow

TABLE II: Performance under High-Demand Scenario

Method	Avg. Step Reward	Queue (veh)	Speed (m/s)	Travel Time (s)	Phase Duration (s)
Fixed-Time	-1.08 ± 0.09	70.48 ± 13.81	2.43 ± 0.27	119.18 ± 6.34	30.00 ± 0.00
DDQN	-0.84 ± 0.04	26.70 ± 1.35	3.71 ± 0.10	75.27 ± 2.70	27.59 ± 0.82
PPO	-0.63 ± 0.04	23.42 ± 1.05	4.18 ± 0.08	66.57 ± 2.10	21.90 ± 0.27
RPPO	$\mathbf{-0.13 \pm 0.02}$	$\mathbf{12.84 \pm 0.48}$	$\mathbf{5.18 \pm 0.06}$	$\mathbf{49.76 \pm 0.90}$	21.25 ± 0.24

TABLE III: Performance under Accident Scenario

Method	Avg. Step Reward	Queue (veh)	Speed (m/s)	Travel Time (s)	Phase Duration (s)
Fixed-Time	-1.37 ± 0.14	122.70 ± 33.00	1.61 ± 0.34	94.80 ± 39.90	30.00 ± 0.00
DDQN	-1.25 ± 0.09	42.60 ± 8.50	2.87 ± 0.28	108.70 ± 20.40	26.02 ± 1.18
PPO	-0.37 ± 0.08	17.10 ± 2.10	4.65 ± 0.18	65.00 ± 4.90	21.23 ± 0.19
RPPO	$\mathbf{-0.17 \pm 0.11}$	$\mathbf{13.40 \pm 2.70}$	$\mathbf{5.17 \pm 0.25}$	$\mathbf{55.90 \pm 5.30}$	20.60 ± 0.13

and queue buildups. Recurrent models are capable of capturing these trends, enabling the agent to proactively adjust signal phases before congestion worsens. However, under a persistently heavy congestion, queues remain long and vehicle speeds drop to near zero. In such conditions, reward signals exhibit minimal variation, making it difficult for recurrent models to learn effective control strategies from the temporal dynamics. In contrast, DDQN performs better in these scenarios by relying on single-step Temporal Difference (TD) errors to select efficient greedy actions without depending on the historical patterns. This suggests that the effectiveness of a control method depends on characteristics of the traffic environment. Recurrent models are better suited to intersections with variable demand and observable traffic fluctuations, whereas value-based methods like DDQN can be more effective in consistently congested settings. In addition to model selection, the design of the reward structure also influences learning and control behavior. The phase-switch penalty in the reward function introduces a trade-off between immediate congestion relief and long-term signal stability. Frequent switching may temporarily reduce queue lengths but can lead to stop-and-go driving conditions and reduced intersection efficiency due to the increased number of yellow signal phases. By gradually increasing the switch penalty during training, the agent learns to assess whether a phase change is necessary, based on the current traffic conditions.

VI. Conclusions

This study applied deep reinforcement learning to improve traffic signal control at the Gongguan Roundabout in Taipei, Taiwan, under varying demand conditions. To address the unique characteristics of the roundabout traffic, the observation space and reward structure were designed to enable the agent to effectively learn and adapt to dynamic traffic patterns. The experimental results demonstrated that the proposed approach outperformed the popular traffic control strategies tested. Future work will extend this method to urban traffic networks involving multiple intersections and roundabouts to study coordinated control in more complex traffic situations. In these networks, mixed traffic modes, such as bicycles and pedestrians, will also be included to make the method more practical for real-world deployment.

References

[1] M. Tincani, "London congestion charge: The impact on air pollution," *Population and Environment*, vol. 44, no. 2, pp. 238–267, 2022. DOI: 10.1007/s11111-022-00401-4. [Online]. Available: https://doi.org/10.1007/s11111-022-00401-4.

[2] A. Haydari and Y. Yılmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 11–32, 2022. DOI: 10.1109/TITS.2020.3008612.

[3] W. Tong, A. Hussain, W. X. Bo, and S. Maharjan, "Artificial intelligence for vehicle-to-everything: A survey," *IEEE Access*, vol. 7, pp. 10 823–10 843, 2019. DOI: 10.1109/ACCESS.2019.2891073.

[4] Z. Xia, J. Wu, L. Wu, Y. Chen, J. Yang, and P. S. Yu, "A comprehensive survey of the key technologies and challenges surrounding vehicular ad hoc networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, no. 4, pp. 1–30, 2021. DOI: 10.1145/3451984.

[5] R. Engineering, "Insync adaptive traffic control shows initial safety benefits," in *ITE Western District Annual Meeting*, 2014. [Online]. Available: https://www.westernite.org/annualmeetings/14_Rapid_City/Presentations/4C-Gannaway.pdf.

[6] P. Mirchandani and L. Head, "A real-time traffic signal control system: Architecture, algorithms, and analysis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001, ISSN: 0968-090X. DOI: https://doi.org/10.1016/S0968-090X(00)00047-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X00000474.

[7] H. M. Abdelghaffar and H. A. Rakha, "A novel decentralized game-theoretic adaptive traffic signal controller: Large-scale testing," *Sensors*, vol. 19, no. 10, p. 2282, 2019. DOI: 10.3390/s19102282. [Online]. Available: https://www.mdpi.com/1424-8220/19/10/2282.

[8] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network," *arXiv preprint arXiv:1705.02755*, 2017. [Online]. Available: https://arxiv.org/abs/1705.02755.

[9] M. Muresan, L. Fu, and G. Pan, "Adaptive traffic signal control with deep reinforcement learning: An exploratory investigation," *arXiv preprint arXiv:1901.00960*, 2019. [Online]. Available: https://arxiv.org/abs/1901.00960.

[10] T.-E. Chou, *Reviewing taipei roundabout accident data and causes*, 2025. [Online]. Available: https://www.thenewslens.com/article/251978.

[11] W. E. Team, *How to build a safe and smoothly flowing gongguan roundabout?* 2024. [Online]. Available: https://www.wisdomfun.com.tw/insight1050.html.

[12] Z. Liu, "A survey of intelligence methods in urban traffic signal control," *International Journal of Computer Science and Network Security*, vol. 7, no. 7, pp. 105–112, 2007.

[13] R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, "Applications of artificial intelligence in transport: An overview," *Sustainability*, vol. 11, no. 1, p. 189, Jan. 2019. DOI: 10.3390/su11010189.

[14] X. Liang, X. Du, G. Wang, and H. Zhu, "Deep reinforcement learning for traffic light control in intelligent transportation systems," *arXiv preprint arXiv:2302.03669*, 2023. [Online]. Available: https://arxiv.org/abs/2302.03669.

[15] "Adaptive traffic signal control based on dueling deep q-learning," in *Proceedings of ACM Conference on Intelligent Transportation*, 2024. DOI: 10.1145/3584376.3584431.

[16] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, ACM, 2018, pp. 2496–2505. DOI: 10.1145/3219819.3220096.

[17] W. Genders and S. Razavi, *Using a deep reinforcement learning agent for traffic signal control*, arXiv preprint, 2016. arXiv: 1611.01142 [cs.LG].

[18] D. Ma, J. Ou, and Y. Ju, "Intelligent traffic signal control strategy based on rainbow algorithm," in *2023 IEEE 8th International Conference on Intelligent Transportation Engineering (ICITE)*, 2023. DOI: 10.1109/ICITE59717.2023.10733910.

[19] W. Genders and S. Razavi, "Evaluating reinforcement learning state representations for adaptive traffic signal control," in *Proceedings of the International Conference on Computational Science (ICCS)*, ser. Procedia Computer Science, vol. 130, Elsevier, Jan. 2018, pp. 26–33.

[20] N. Casas, *Deep deterministic policy gradient for urban traffic light control*, 2017. arXiv: 1703.09035 [cs.LG].

[21] T. Wu, P. Zhou, K. Liu, *et al.*, "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8243–8256, 2020. DOI: 10.1109/TVT.2020.2997896.

[22] L. Huang and X. Qu, "Improving traffic signal control operations using proximal policy optimization," *IET Intelligent Transport Systems*, vol. 17, pp. 592–605, 3 2022. DOI: 10.1049/itr2.12286.

[23] Y. Zhu, H. Liu, and J. Wang, "Intelligent traffic light via policy-based deep reinforcement learning," *arXiv preprint arXiv:2101.12345*, 2021. [Online]. Available: https://arxiv.org/abs/2101.12345.

[24] D. Kakitahi, R. Xu, and M. Chen, "Adaptive traffic signal control based on multi-agent reinforcement learning," *arXiv preprint arXiv:2504.09876*, 2025. [Online]. Available: https://arxiv.org/abs/2504.09876.

[25] A. e. a. She, "Safe, efficient, comfort, and energy-saving automated driving," *arXiv preprint arXiv:2306.11465*, 2023. [Online]. Available: https://arxiv.org/abs/2306.11465.

[26] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/.

[27] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html.