Traffic & Transportation

Optimisation of Decision Efficiency for Autonomous Driving at Unsignalised Intersections Based on DRL and GPT

Bojun LIU[1]

Original Scientific Paper
Submitted: 27 Apr 2025
Accepted: 29 Aug 2025
Published: 28 Apr 2026

[1] LbaijJun@163.com, Faculty of Engineering, University of Sydney, Sydney, Australia

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT
The rapid increase in urban vehicle numbers has intensified traffic congestion and safety challenges. Unsignalised intersections pose significant difficulties for autonomous vehicle decision-making. To enhance decision efficiency and safety in such scenarios, this study proposes a decision optimisation method for autonomous driving at unsignalized intersections. The approach first employs a generative pre-trained transformer (GPT) to learn complex interactive behaviour patterns from driving data and acquire prior knowledge. This prior knowledge is then used to initialise the policy network of a deep reinforcement learning (DRL) agent, specifically deep q-network (DQN), which is further optimised through interaction within a simulated environment. This framework aims to combine the powerful sequence modelling capability of GPT with the goal-directed optimisation strength of DRL. Experimental results demonstrate that the proposed method achieves superior performance: The median safe distance reaches 19.58 m (maximum 32.50 m, minimum 8.46 m), the collision rate is as low as 1.07%, and the success rate exceeds 98%. Compared to baseline methods, the proposed approach significantly improves decision-making efficiency and safety for autonomous vehicles at unsignalised intersections, validating its effectiveness.

KEYWORDS
deep reinforcement learning; driving decisions; intelligent driving; generate pre-trained transformation models.

1. INTRODUCTION

The advent of autonomous driving (AD) promises to revolutionise transportation, offering the potential to dramatically enhance road safety and traffic efficiency. However, realizing this potential requires vehicles to navigate a wide array of complex and unpredictable real-world scenarios. With the swift advancement of the social economy, the growing prevalence of vehicles has intensified problems of traffic jams and safety concerns in cities [1]. The progress of AI technology is driving the automotive industry towards a highly intelligent and automated direction. In this context, enhancing the security and productivity of autonomous driving decision-making (ADDM), especially in complex scenarios such as unsignalised intersections, has become a key research hotspot.

Unsignalised intersections represent a critical bottleneck for the widespread adoption of AD. These environments lack explicit traffic signals, forcing human drivers to engage in a complex ballet of implicit negotiations, relying on subtle cues to infer intent and safely cross. For an AD system, this unstructured environment poses an immense challenge. This not only requires efficient path planning, but also reliable interactive DM strategies. From a policy and societal perspective, the inability to master these scenarios has profound implications. Accidents at intersections are a major source of traffic fatalities and injuries, and inefficient navigation contributes to congestion and emissions. Therefore, developing robust ADDM systems

for unsignalised intersections is not merely a technical challenge; it is a critical step toward achieving the societal benefits of autonomous mobility and shaping future traffic policy.

Deep reinforcement learning (DRL) has received widespread attention in the area of AD because of its potential in end-to-end control tasks [2]. However, applying DRL directly to complex real-world traffic scenarios, especially at unsignalised intersections, faces several fundamental challenges: (1) Low sample efficiency: DRL typically necessitates extensive engagement with the environment to acquire effective tactics; (2) High exploration risk: Random exploration in complex traffic environments may lead to unsafe behaviour; and (3) Difficulty of modelling complex interactions: Traditional DRL methods may struggle to capture fine interaction patterns with other traffic participants (vehicles, pedestrians, etc.) [3]. Existing DRL methods often simplify the environmental model or fail to accurately simulate complex human driving behaviour when dealing with unsignalised intersections, which limits their performance in practical applications [4].

Concurrently, the generative pre-trained transformer (GPT), as a representative of large language models, has demonstrated remarkable abilities in tasks related to natural language processing and sequence modelling [5]. Its transformer-based architecture is adept at capturing long-range dependencies in sequential data, and this ability is also applicable to modelling complex driving behaviour sequences.

Drawing from the aforementioned analysis, this study proposes a novel hybrid framework that integrates the advantages of GPT and DRL to tackle the difficulties associated with unsignalised intersections. Our approach first leverages GPT's powerful pattern recognition and complex sequence data modelling to learn a behavioural prior from a substantial quantity of driving data. Then, this pre-trained GPT model is used as the initialisation for the policy network for the DRL agent (using the deep Q network (DQN) algorithm), which is further optimised through interaction with a simulation environment to maximise cumulative rewards (balancing safety and efficiency) while retaining human-like driving characteristics.

The main contributions of this article are threefold. First, we propose a novel hybrid DRL-GPT architecture that synergises the sequence modelling power of transformers with the goal-oriented optimisation of reinforcement learning, addressing the limitations of using either approach in isolation. Second, we demonstrate that pre-training a GPT model on expert driving data to initialise the DRL policy network significantly improves sample efficiency and reduces unsafe exploration during training. Third, through extensive simulation and comparative analysis, we validate that our method yields superior performance in safety, efficiency and decision stability, providing a robust blueprint for developing next-generation decision-making systems for autonomous vehicles.

2. RELATED WORKS

2.1 Automatic driving methods at unsignalised intersections

The decision control of unsignalised intersections has always been a research hotspot. Shi et al. [6] proposed a coordinated control method based on proximal policy optimisation (PPO), which utilises RL to adapt to dynamic traffic flow. Liu et al. [7] designed a rule-based high-precision intelligent transportation system that detects lanes and generates intersection maps through sliding windows. This method relied on precise maps and rule definitions, which can limit its adaptability to unforeseen situations. Maadi et al. [8] used RL for adaptive traffic signal control. These studies each have their own focus, but there is still room for improvement in modelling complex interactions between vehicles and dealing with high levels of uncertainty. More recently, to better understand the underlying challenges, Qu et al. [9] proposed a generalised linear mixed effects model to understand the behaviour patterns of human drivers at signalised and unsignalised urban intersections. The results showed that the speed of the drivers at the intersection followed a "deceleration acceleration" pattern, confirming that driver attributes and traffic conditions have a major impact on driving behaviour and highlighting the need for adaptive, learning-based models.

2.2 Application of DRL in AD

DRL plays an important role in ADDM. Huang et al. [10] derived a strategy based on DRL and tested it in simulated urban scenarios, demonstrating its potential to improve safety but also highlighting challenges with sample efficiency. Chen et al. [11] proposed an interpretable end-to-end DRL method that introduced latent environment models to generate semantic bird's-eye views, which performed well in crowded city scenes but at the cost of high model complexity. Fuchs et al. [12] used DRL to plan minimum-time trajectories, demonstrating the feasibility of controlling vehicles in dynamic environments. Although DRL shows great

potential, a persistent challenge remains in efficiently and safely learning optimal strategies, especially for complex scenarios like unsignalised intersections. To address the low learning efficiency of traditional DRL methods, an innovative trend involves integrating expert guidance. For instance, Pang et al. [13] recently proposed an LLM-guided deep reinforcement learning (LGDRL) framework where a large language model acts as a driving expert to provide intelligent guidance to the DRL agent. Their results showed a significant improvement in both task success rate and learning efficiency, underscoring the value of combining expert priors with DRL.

2.3 Application of GPT in behaviour prediction

GPT and its transformer architecture have shown outstanding performance in sequence modelling. While its direct application to ADDM is an emerging area, its potential is significant. A recent 2024 survey by Al-Sharman et al. confirms that developing robust, non-overcautious decision-making schemes for unsignalised intersections remains an open challenge where advanced models are needed [14]. This has led to a trend towards using transformer-based models to directly address complex driving decisions. A prime example is the MTD-GPT model by Liu et al., which frames multi-task decision-making at unsignalised intersections (e.g., turning left, going straight) as a sequence modelling problem [15]. By training a GPT model on expert driving data, they demonstrated strong generalisation performance, validating the approach of using transformer architectures to learn complex behavioural patterns in driving. Other applications have also shown the utility of GPT-like models in related domains, such as predicting behaviours from emotional cues or analysing genomic data [16, 17].

2.4 Research gaps

The literature reveals a clear trajectory towards more intelligent, data-driven decision-making models. However, a critical research gap remains. Traditional approaches struggle to fully capture complex dynamic interactions. Pure DRL methods encounter difficulties including low sample efficiency and unsafe exploration. Although GPT/transformer has great potential in sequence modelling, prior works have focused on using them for direct behaviour cloning (e.g., MTD-GPT) or for high-level guidance from a generalist LLM (e.g., LGDRL). There is currently insufficient research on how to effectively integrate a pre-trained, task-specific transformer into a DRL framework – specifically, by using its learned behavioural priors to initialise and guide the policy network of the DRL agent. The objective of this research is to bridge this void by proposing and validating a new paradigm for combining imitation learning and reinforcement learning through a hybrid DRL-GPT architecture.

3. AD DECISION OPTIMISATION BASED ON DRL AND GPT

3.1 Construction of markov decision process (MDP) model

To formally structure the decision-making problem at an unsignalised intersection, we model the interaction between the autonomous vehicle and its environment as a markov decision process (MDP). The MDP provides a mathematical framework for modelling decision-making in situations where outcomes are partly random and partly under the control of a decision-maker. It is defined by a set of states, actions, transition probabilities and rewards. In the DM process of autonomous vehicles, signalised intersections provide clear indications, allowing vehicles to make dynamic and efficient decisions based on real-time information such as traffic signals, traffic flow, vehicle speed and location. However, in actual traffic environments, there is still a large number of intersections without signal control, and the environment of these intersections is more complex and unpredictable. To effectively address this issue, the study adopts MDP to model AD decisions. The diagrammatic representation of MDP state transfer is presented in *Figure 1*.

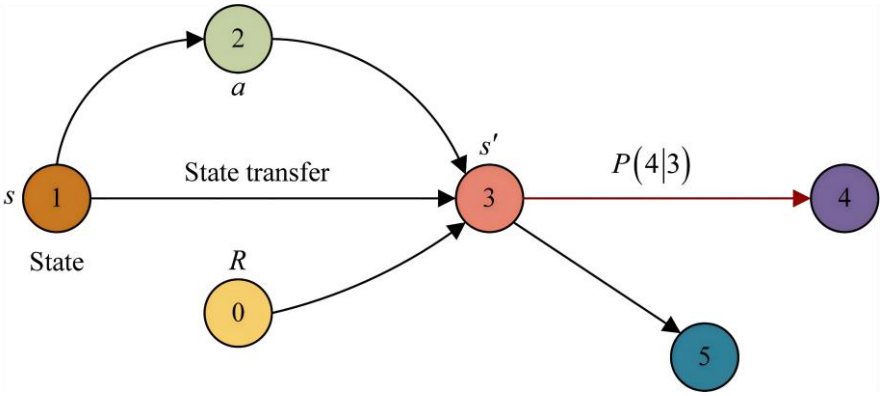


Figure 1 – Conceptual illustration of the Markov decision process (MDP) state transition model

Figure 1 provides a conceptual illustration of the state transition dynamics within the MDP. As shown in *Figure 1*, the MDP comprises a state space (S), an action space (A), a reward function (R) and state transitions (P). Each state $s \in S$ represents a snapshot of the environment (e.g., vehicle positions, velocities). From state s , the agent can take an action $a \in A$ (e.g., accelerate, turn), which leads to a new state s' with a certain probability. In each state transition process, the agent receives an immediate reward. The decision rule, or policy π , describes the probability of implementing action a in state s . The ultimate goal is to find an optimal policy, π^* , that maximises the expected cumulative reward. The probability of state to action is presented in *Equation 1*.

$$\pi(a|s) = P(A_t = a|S_t = s) \tag{1}$$

In *Equation 1*, $\pi(a|s)$ is the probability of implementing action a in state s , and $P(A_t = a|S_t = s)$ represents the probability of implementing action a in state s at time t . In each state, the goal of the agent is to select actions that maximise the expected cumulative reward, so research can use state value functions and action value functions to quantify the quality of each state and action.

To evaluate the quality of a policy, we define value functions. The state-value function (SVF), $V^{\pi}(s)$, represents the expected cumulative discounted reward starting from state s and following policy π . It provides a measure of how good it is to be in a particular state. The calculation of the SVF is presented in *Equation 2*.

$$V^{\pi}(s) = E^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_0 = s\right] \tag{2}$$

In *Equation 2*, E^{π} represents the expected value under strategy π , γ represents the discount factor, the value of this factor is $0 \le \gamma < 1$, this factor takes into account the discount of long-term rewards, ensuring that more recent rewards receive higher attention. R_{t+1} is the immediate reward obtained at time $t+1$.

Similarly, the action-value function (AVF), $Q^{\pi}(s,a)$, represents the expected return after taking action a in state s and thereafter following policy π . This function is crucial as it tells us how good it is to perform a specific action in a given state. The calculation of the AVF is presented in *Equation 3*.

$$Q^{\pi}(s,a) = E^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_0 = s, A_0 = a\right] \tag{3}$$

Our objective is to find the optimal policy, π^* , which has a corresponding optimal SVF (V^*) and optimal AVF (Q^*). These represent the maximum possible expected return achievable from any state or state-action pair, respectively, and are defined by the Bellman optimality equations (*Equations 4 and 5)*.

$$V^*(s) = \max_{\pi} V^{\pi}(s) \tag{4}$$

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{5}$$

By solving for the optimal AVF, the optimal strategy can be found. In our approach, we use DRL to learn an approximation of the optimal action-value function, Q^*, which in turn defines our optimal policy.

3.2 GPT behaviour modelling

In the scenario of unsignalised intersections, DRL often requires a vast amount of environmental interaction for training to converge to effective driving strategies. This process is not only computationally expensive but also poses significant safety risks during the initial exploration phase.

To mitigate these challenges, we employ a generative pre-trained transformer (GPT) model to learn a behavioural prior from expert driving data. We chose GPT over other temporal models like LSTMs for two primary reasons. First, the transformer architecture's self-attention mechanism is exceptionally effective at capturing long-range dependencies in sequential data, which is critical for understanding complex traffic interactions. Second, the pre-training and fine-tuning paradigm allows the model to learn a generalisable representation of driving behaviour that can be adapted to specific tasks. By pre-training on successful driving trajectories, the GPT model learns the patterns of human-like driving, creating an efficient and interpretable initial policy for the DRL agent. The GPT model structure is presented in *Figure 2*.

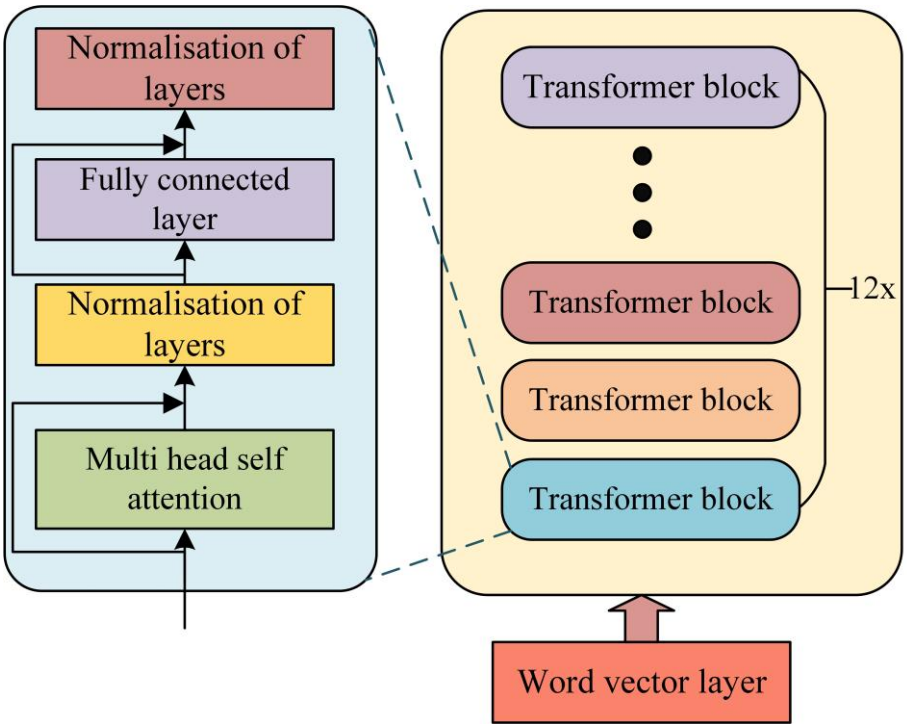


Figure 2 – The architecture of the generative pre-trained transformer (GPT) model used for behaviour modelling

Figure 2 presents the diagrammatic representation of the GPT model structure. As presented in *Figure 2*, GPT is based on the transformer architecture and is mainly composed of multiple decoder layers stacked together. Unlike traditional transformers, GPT only retains the masked multi head attention mechanism and removes additional multi head attention modules. This enables GPT to strongly model complex temporal dependencies and adapt to dynamic driving DM tasks [18]. In ADDM tasks, in addition to textual information, a large amount of data from the driving environment needs to be processed, such as vehicle status, traffic signals and the behaviour of surrounding traffic participants. The GPT model adopts an unsupervised learning pre training method, which first trains on a large-scale driving behaviour dataset to extract patterns from historical driving behaviours. Then, it fine-tunes the specific tasks to adapt to specific driving scenarios. During the training process, the GPT model predicts corresponding actions based on the input state sequence, thereby generating the optimal decision strategy.

The core of the GPT model is the self-attention mechanism (SAM), illustrated for GPT-2 in *Figure 3*.

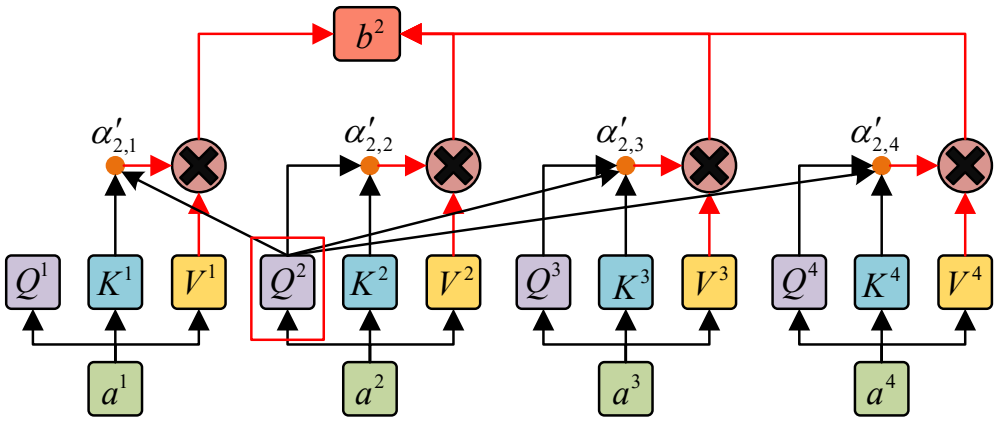


Figure 3 – The self-attention mechanism (SAM) of GPT-2

This mechanism allows the model to weigh the importance of different elements in the input sequence when making a prediction, enabling it to focus on the most relevant environmental cues. This not only improves performance but also offers a degree of interpretability, as attention weights can be analysed to understand the model's focus. The calculation of the SAM is presented in *Equation 6*.

$$Attention(Q,K,V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

In *Equation 6*, QK^T is the dot product of the query vector and the key vector, d_k is the dimension of the key vector, and T is transposition. In the SAM of GPT-2, the dot product of query vector Q and key vector K reflects the relationship between the current position and other positions, which determines what information the model needs to focus on in the current state. Before using GPT for AD behaviour modelling, it is necessary to first build a high-quality dataset to ensure that the model can learn effective driving decision patterns. The study collected driving data of autonomous vehicles at unsignalised intersections, including sensor data, vehicle status data and surrounding traffic environment data. Subsequently, the raw data are cleaned and organised, such as removing outliers, aligning time series, normalising etc., to ensure the stability and consistency of the input data. Next, these data are organised into sequences of states, actions and rewards for training the GPT-2 model. This enables the model to understand different driving scenarios and how to choose appropriate actions based on the current state and adjust strategies according to the reward mechanism. The research selects successful decision samples and converts them into trajectory sequences to describe the agent's path from one state to another in the environment. In the training process of GPT, the diversity and quality of data directly affect the model's generalisation ability and decision-making effectiveness. While our approach leverages large public datasets (NGSIM, INTERACTION) to build a robust prior, the model's performance on completely novel, out-of-distribution scenarios remains a limitation that requires ongoing research and data collection. The calculation of the trajectory sequence is presented in *Equation 7*.

$$\zeta = (s_1, a_1, r_1, s_2, a_2, r_2, ..., s_T, a_T, r_T) \tag{7}$$

In *Equation 7*, ζ represents the driving trajectory and T represents the time step. During the training process, the model will continuously optimise its strategy based on the states, actions and reward sequences in the driving trajectory, thereby improving the accuracy and safety of AD behaviour. The strategy function generated by the GPT model over time steps is presented in *Equation 8*.

$$\mu_{gpt} = (a_T \,|\, s_{n,T}, R_{n,T}) \tag{8}$$

In *Equation 8*, μ_{gpt} represents the policy function of the GPT model, a_T represents the actions at time steps, $S_{n,T}$ is the state information at time step n, and $R_{n,T}$ represents the reward information at time step n. Through this strategy function, the model can make optimal decisions at each time step based on the current state and reward, thereby achieving optimal AD behaviour in a dynamic environment. The study uses the cross entropy loss function (LF) for model evaluation, as presented in *Equation 9*.

$$L = \frac{1}{N}\sum_{t=1}^{n} P(a_T)\log\left(\mu_{gpt}\left(s_{n,T}, R_{n,T}\right)\right) \tag{9}$$

In *Equation 9*, L represents the cross entropy LF, N is the length of the time step sequence, and $P(a_T)$ is the probability of the target action a_T. The cross entropy loss function is used to evaluate the difference between the probability of actions generated by the model and the target action. By minimising this loss function, the model can gradually optimise its decision strategy. In the DRL environment, the GPT model optimises its strategy by minimising cross entropy loss, enabling it to generate action sequences that are suitable for the current environment. When integrating GPT into DRL environment, the objective of the model is to achieve the highest possible total rewards through learning how to make optimal decisions in a specific environment.

3.3 Decision optimisation based on DRL

To further optimise DM efficiency, DRL is used for reinforcement learning in the DM process of AD. DRL combined with MDP framework optimises the DM strategy of intelligent agents through interaction with the environment, thereby making optimal choices in complex traffic environments. In the DRL environment, the agent not only relies on historical state and reward information to generate actions but also needs to further optimise its strategy through interaction with the environment [19]. DRL, as a machine learning approach, can learn optimal decision strategies through the interaction between intelligent agents and the environment. In DRL, decision problems are typically modelled as MDPs, which guide policy optimisation through explicit state space, action space and reward functions. Therefore, the research is based on the DRL framework for decision optimisation in AD. The DRL framework is presented in *Figure 4*.

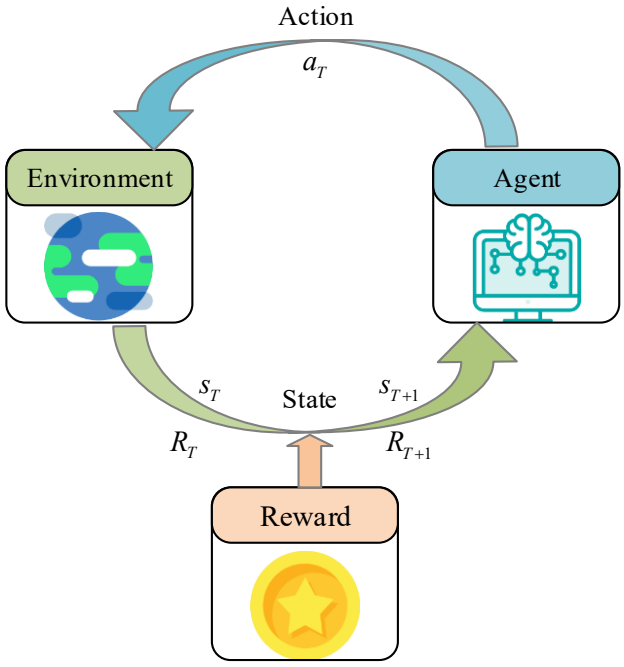


Figure 4 – The deep reinforcement learning (DRL) framework

Figure 4 shows a schematic diagram of the DRL framework. As presented in *Figure 4*, in the DRL framework, the agent continuously interacts with the environment and optimises its behaviour strategy based on feedback from actions and states to maximise overall benefits. In ADDM, the state space represents the current state of the vehicle in the environment, the action space defines the actions that the vehicle can take at each time step, and the reward function aims to encourage safe, efficient and energy-saving driving behaviour. Through this MDP definition, the DRL algorithm can learn the optimal driving strategy in a constantly changing traffic environment. With the rapid development of DRL, combining MDP with deep learning has emerged as a widely-studied area within the domain of reinforcement learning. Common DRL algorithms include DQN, PPO and asynchronous advantage actor critic (A3C) [20]. Among them, DQN has significant advantages in dealing with high-dimensional state space and action space problems by introducing neural networks [21].

Therefore, the study is based on DQN structure for decision optimisation of AD. The DQN structure diagram is presented in *Figure 5*.

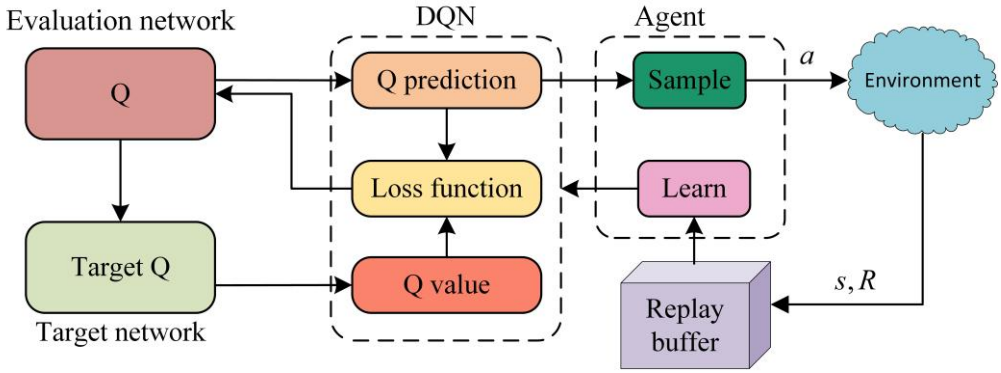


Figure 5 – The deep q-network (DQN) architecture

Figure 5 shows a schematic diagram of the DQN structure. As shown in *Figure 5*, the DQN structure mainly includes an evaluation network (EN) and a target network (TN), where the EN is mainly employed to estimate the Q-values of all possible actions in the current state and select the optimal action. The TN is employed to generate target Q-values, which are employed to calculate the LF of the EN, avoiding overfitting of the EN and increasing training stability. The experience replay unit generates a large amount of experience data during its interaction with the environment. Throughout the training phase, the agent engages with the environment, producing experiential data that gets saved in the experience replay buffer. By randomly extracting experience samples from the buffer, the correlation between data can be reduced, thereby improving the efficiency and stability of training. The mathematical formula for the target Q-value is presented in *Equation 10*.

$$y_T = R_T + \gamma \max_a \left(s_{T+1}, a'; \theta\right) \tag{10}$$

In *Equation 10*, y_T represents the target Q-value of the time step, a' represents all possible actions calculated by the TN, and θ is the parameter of the TN. These target Q-values are generated by the TN and serve as update targets for the EN. The Q-value function is presented in *Equation 11*.

$$Q^{\pi}\left(s_T, a_T\right) = E_{sT+1}\left[R_{T+1} + \gamma Q^{\pi}\left(s_{T+1}, \pi\left(s_{T+1}\right)\right)\right] \tag{11}$$

In *Equation 11*, $Q^{\pi}(S_T, a_T)$ represents the Q-value function under strategy π, and E_{sT+1} represents the expected value of the next time state s_{T+1}. The Q-value function represents the expected total reward for an agent to perform a certain action in a given state under a given strategy. The formula for updating the Q-value function is presented in *Equation 12*.

$$Q^{\pi}\left(s_T, a_T\right) \leftarrow Q^{\pi}\left(s_T, a_T\right) + \alpha\delta \tag{12}$$

In *Equation 12*, α and δ both represent update terms, which are learning rate and temporal difference error, respectively. The value of the parameter learning rate is $0 \le \alpha \le 1$. By minimising the temporal differential error, the model continuously optimises its Q-value function, gradually improving the accuracy of decision-making. The calculation of the LF is presented in *Equation 13*.

$$L\left(\theta\right) = E_{\left(s_T, a_T, R_{T+1}, S_{T+1}\right) \square D}\left[\left(y_T - Q\left(s_T, a_T; \theta\right)\right)^2\right] \tag{13}$$

In *Equation 13*, $L(\theta)$ represents the LF of the EN with parameter θ, $(s_T, a_T, R_{T+1}, S_{T+1}) \sim D$ represents a quadruple sampled from the experience replay buffer D, and $Q(s_T, a_T; \theta)$ is the Q-value for evaluating the network's choice of action a_T in state s_T. The loss function is used to measure the difference between the Q-value output by the EN and the target Q-value. By reducing the loss, the accuracy of the EN can be gradually improved. The reward function is presented in *Equation 14*.

$$R_{total} = \omega_1 R_1 + \omega_2 R_2 + \omega_3 R_3 \tag{14}$$

In *Equation 14*, R_{total} represents cumulative reward, R_2 and R_3 represent collision penalty, efficiency reward and distance reward, respectively, ω_1, ω_2, and ω_3 represent the weights of collision penalty, efficiency reward and distance reward, respectively [22]. The design of the reward function can affect the learning objectives of the agent, and a reasonable reward design can guide the agent to choose the optimal behaviour. The study adopts DQN as the implementation method of DRL. DQN is capable of efficiently tackling intricate issues within high-dimensional state and action spaces by utilising neural networks to approximate Q-values and subsequently refine strategies. A depiction of DQN's training procedure can be found in *Figure 6*.

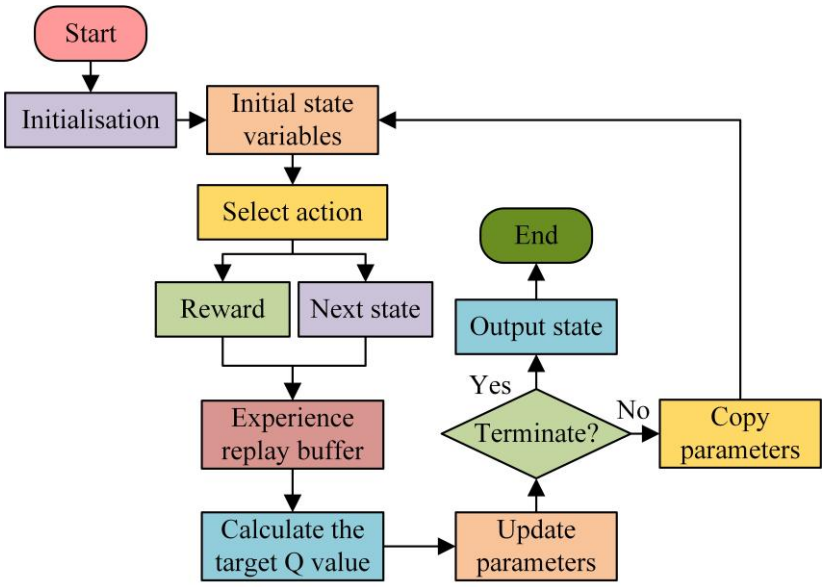


Figure 6 – The pseudo-code for the DQN training algorithm

Figure 6 shows the training process of DQN. As illustrated in *Figure 6*, during the training process of DQN, the Q network and TN are initialised first, and an experience replay buffer is set. Then, an initial state is randomly selected in the environment, and the agent selects actions based on the current strategy and executes them. Next, the Q network is updated based on the rewards from environmental feedback and the next state. Then the stability of training is improved through experience replay and continuously optimised DM strategies. After training, the agent is able to select the optimal action based on the current state in a signal free intersection environment. To assess the efficacy of the suggested approach, the experimental process can be broken down into the subsequent procedures: Firstly, the process collects actual driving data of autonomous vehicles at unsignalised intersections, including sensor data, vehicle status, behaviour data of surrounding traffic participants etc., and preprocesses the collected raw data, including removing outliers, aligning data, normalising etc., to ensure the stability and consistency of the data. Then, based on successful decision samples, the process generates a trajectory sequence. The trajectory sequence describes the path of an intelligent agent from one state to another. Continuing with model training, the pre-processed data is organised into state, action and reward sequences in the MDP model for training the GPT-2 model. Driving behaviour is modelled through unsupervised learning and fine-tuned to adapt to specific tasks. Additionally, DQN is utilised for reinforcement learning of decisions to optimise the DM strategies of the agent. Finally, the trained model is validated in a simulated signal-free intersection environment.

4. VALIDATION OF AD DECISION OPTIMISATION BASED ON DRL AND GPT

4.1 Experimental environment configuration

To confirm the validity of the AD decision optimisation method based on DRL and GPT, experiments were conducted on the Ubuntu 16.04.7 LTS operating system using Intel Core i7-9700 processor and NVIDIA GeForce GTX 1080Ti GPU, equipped with 64GB of memory to ensure sufficient computing resources. The simulation of urban mobility (SUMO) software was used to simulate the environment for simulating urban traffic scenarios. A typical unsignalised intersection scene was constructed in a simulation environment to test the performance of decision optimisation methods in complex road conditions. In addition, the study selected

publicly available NGSIM and INTERACTION datasets for experimentation. After mixing the dataset, it was divided into a test set (TES) and a training set (TRS) in a 3:7 ratio to ensure fairness in model training and validation. *Table 1* shows the detailed setup of the experimental environment.

Table 1 – Specific experimental environment configuration

Environment	Configuration
Operating system	Ubuntu 16.04.7 LTS
Processor	Intel Core i7-9700
Memory	64GB
GPU	NVIDIA GeForce GTX 1080Ti
Simulation environment	SUMO
Deep learning framework	PyTorch
Simulation software	SUMO
Scientific computing library	NumPy

4.2 Analysis of the effectiveness of ADDM

To verify the effectiveness of ADDM based on DRL and GPT, a comparative analysis was conducted with other advanced DM methods. Other methods included rule-based system (RBS) methods, fuzzy logic system (FLS) methods and Dijkstra algorithm-based DM methods. *Figure 7* illustrates a comparison of safety distances achieved by various methods. From *Figure 7a*, under the TRS, the maximum, median and minimum safe distances of the DRL and GPT-based DM methods were 32.50 m, 19.58 m and 8.46 m, respectively. However, the maximum values of the other three methods did not exceed 26.00 m and there were outliers. In the TES of *Figure 7b*, the maximum, median and minimum safe distances of the decision methods based on DRL and GPT were 31.98 m, 19.37 m and 8.97 m, respectively. Similarly, other methods had outliers, and the maximum value did not exceed 26.50 m. Overall, the safety distance of the research method was significantly higher than other methods, indicating that it could more effectively cope with various driving scenarios in dynamic environments, maintain a larger safe distance, and reduce the risk of vehicle collisions.

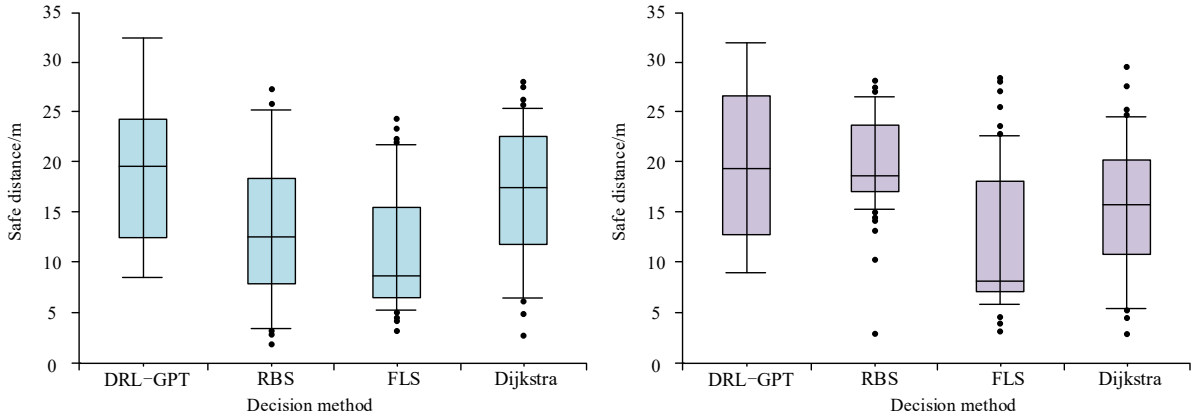


Figure 7 – Comparison of safety distances for different methods: a) training set; b) test set

To conduct additional verification of the effectiveness of the ADDM methods based on DRL and GPT, a comparative analysis of the driving performance of different DM methods was conducted, as shown in *Table 2*. From *Table 2*, the methods based on DRL and GPT performed well in all indicators. In terms of collision rate, this method had a collision rate of only 1.07% in the TES, which was significantly lower than other methods. In terms of success rate, the research method had a success rate of over 98% in both the training and testing sets, far higher than other methods. In regard to the number of sudden braking attempts, there were only 2 in the TES, while all other methods exceeded 5, indicating that this method could handle driving situations more smoothly. The average travel time was around 180 seconds, which exhibited greater efficiency in comparison

to the others. In regard to path accuracy, both the average displacement error and the final displacement error were controlled within 0.5 m. In summary, the methods based on DRL and GPT not only improved safety and efficacy but also achieved high-precision path planning.

Table 2 – Comparison of driving performance using different DM methods

Data set	Method	Success rate (%)	Collisions (%)	Average displacement error (m)	Final displacement error (m)	Average travel time (s)	Number of sudden braking cycles
TRS	DRL-GPT	98.92±0.61	1.18±0.30	0.35±0.23	0.45±0.19	180.06±2.15	3
	RBS	85.16±3.67	8.56±1.46	0.75±0.51	1.12±0.48	210.64±6.69	12
	FLS	88.23±3.34	6.83±1.64	0.65±0.25	0.95±0.33	200.06±5.91	9
	Dijkstra	90.23±1.64	5.91±0.89	0.52±0.24	0.73±0.26	190.97±7.93	6
TES	DRL-GPT	98.56±0.80	1.07±0.53	0.32±0.11	0.41±0.23	177.69±3.67	2
	RBS	85.34±4.42	8.50±1.89	0.77±0.68	1.18±0.62	235.40±7.96	12
	FLS	84.61±2.85	6.55±1.73	0.58±0.30	0.87±0.26	205.41±6.25	9
	Dijkstra	92.59±1.85	5.62±1.25	0.61±0.34	0.79±0.33	185.91±7.65	6

The cumulative reward scores for different DM methods are shown in *Figure 8*. From *Figure 8a*, in the TRS, the AD decision method based on DRL and GPT exhibited relatively stable scores, fluctuating around 40 points with little volatility. The scores of the RBS, FLS and Dijkstra methods had significant fluctuations, fluctuating around 35 points, 28 points and 23 points, respectively. From *Figure 8b*, the trend in the TES was similar to that in the TRS. The cumulative reward score of the research method in the TES remained stable at around 40 points, while the scores of the other three methods did not exceed 38 points. In summary, the ADDM methods based on DRL and GPT could maintain efficient and stable performance in different environments.

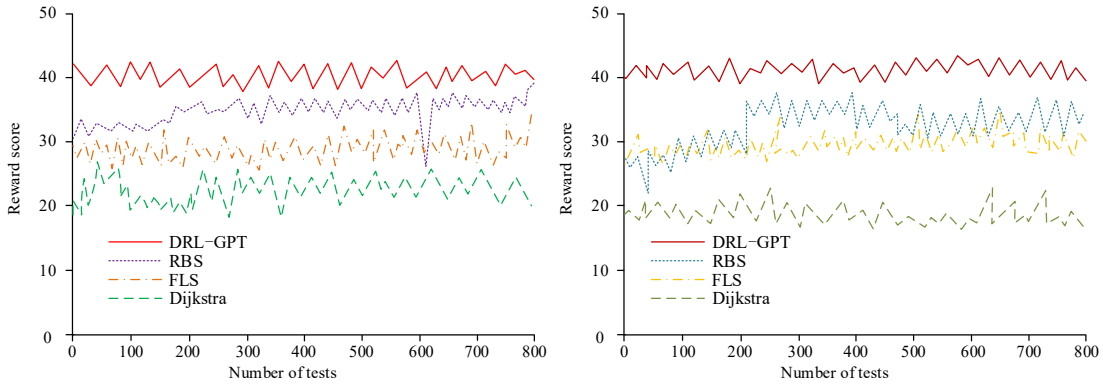
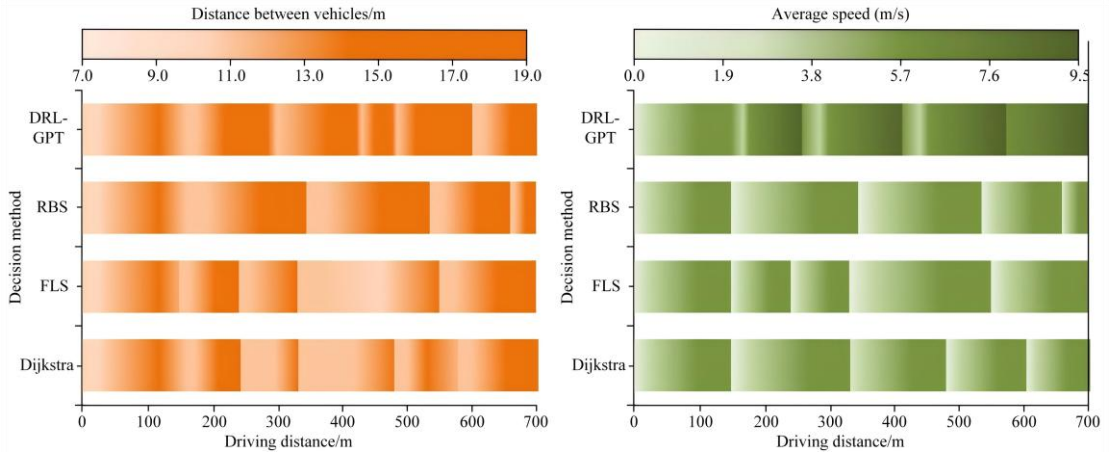


Figure 8 – Cumulative reward scores for different DM methods: a) comparison of scores for different methods in training set; b) comparison of scores for different methods in the test set

4.3 Practical application verification

To confirm the practical effectiveness of the ADDM method based on DRL and GPT, an unsignalised intersection was selected for traffic validation. The test took place on a secluded stretch of road to ensure data fairness. The comparison of the safety and efficiency of various DM methods is presented in *Figure 9*. From *Figure 9a*, in the upstream section, the heatmap colours of all methods were similar, indicating that the average distance between vehicles was similar. However, as the driving distance increased, the colour of the heatmap based on DRL and GPT methods was significantly darker, indicating higher operational safety. In *Figure 9b*, the heatmap based on DRL and GPT methods showed that the vehicle speed changed less, and the traffic was smooth with higher efficiency. However, the heat maps of other methods showed a decrease in speed. Overall, the methods based on DRL and GPT demonstrated higher safety and efficiency when passing through intersections.



(a) *(b)*

Figure 9 – Comparison of safety and efficiency of different decision making methods: a) thermal maps of vehicle spacing using different decision-making methods; b) average velocity heatmap of different decision-making methods

To further assess the effectiveness of AD based on DRL and GPT, ablation experiments were conducted, as presented in *Table 3*. From *Table 3*, the success rate of model construction using MDP was only 65.34%. On this basis, the success rate of behaviour modelling using GPT-2 was further improved to 88.43%. After further introducing the self-attention mechanism, the success rate increased to 92.64%. After combining DQN for decision optimisation, the success rate was increased to 95.06%. The success rate of the final complete DRL-GPT method was as high as 98.97%. In summary, the addition of each module optimised the DM process for AD.

Table 3 – Ablation study results

MDP	GPT-2	SAM	DQN	DRL-GPT	Success rate (%)
√	/	/	/	/	65.34
√	√	/	/	/	88.43
√	√	√	/	/	92.64
√	√	√	√	/	95.06
√	√	√	√	√	98.97

4.4 Discussion

Our research proposed and validated a hybrid DRL-GPT model for autonomous decision-making at unsignalised intersections. The results clearly show that our method maintains larger safety distances, minimises collisions and achieves a significantly higher success rate than traditional approaches.

A key finding is the powerful synergy between imitation learning (via GPT) and reinforcement learning (via DQN). By initialising the DRL agent's policy with a pre-trained GPT model, we provide it with a strong behavioural prior learned from expert data. This two-stage process explains the superior performance of our hybrid model, as it directly addresses the critical DRL challenges of poor sample efficiency and unsafe exploration that were identified as primary barriers to developing robust AD systems for unsignalised intersections. The DRL phase then fine-tunes this policy, optimising it for specific objectives like maximising safety margins and traffic throughput, which may not be perfectly captured in the imitation data alone.

This robustness stands in contrast to pure DRL or rule-based methods, which often exhibit more erratic performance when faced with the stochastic nature of unsignalised intersections. The stability of the cumulative reward scores for our method further underscores its reliability. In summary, the DRL-GPT method demonstrated excellent performance across multiple dimensions, verifying its effectiveness in the complex and challenging environment of unsignalised intersections.

5. CONCLUSION

To improve the performance of AD on unsignalised road sections, this study combined DRL and GPT to construct an AD decision model. The results showed that the proposed method was significantly better than other comparative methods across all key metrics. The collision rate was only 1.07%, the success rate exceeded 98%, and the cumulative reward scores were consistently higher and more stable. The ablation experiment showed that each module effectively optimised the ADDM, with the success rate of the final complete DRL-GPT method reaching 98.97%. In summary, the AD unsignalised intersection DM method based on DRL and GPT significantly improved the success rate of DM and vehicle safety.

However, the training and optimisation process of the research model required a substantial quantity of computing resources and time, presenting a potential barrier to widespread implementation. Future research will proceed in several key directions. First, a model compression and quantisation techniques will be investigated to create a more lightweight version suitable for deployment on automotive-grade hardware. Second, a critical next step is to validate the model's performance in real-world driving scenarios, moving beyond simulation to assess its robustness to the full complexity of physical environments. Third, a thorough analysis of the model's failure cases plans to be conducted to identify specific edge scenarios that require targeted data augmentation or architectural refinement. Finally, exploring the integration of more advanced transformer architectures and DRL algorithms could further enhance the model's adaptability and generalisation capabilities.

REFERENCES

[1] Kiran BR, et al. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(6):4909-4926. DOI: 10.1109/TITS.2021.3054625.

[2] Grigorescu S, et al. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*. 2020;37(3):362-386. DOI: 10.1002/rob.21918.

[3] Zhu Z, Zhao H. A survey of deep RL and IL for autonomous driving policy learning. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(9):14043-14065. DOI: 10.1109/TITS.2021.3134702.

[4] Liu P, et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Acm Computing Surveys*. 2023;55(9):1-35. DOI: 10.1145/3560815.

[5] Chavez MR, et al. Chat generative pre-trained transformer: Why we should embrace this technology. *American Journal of Obstetrics and Gynecology*. 2023;228(6):706-711. DOI: 10.1016/j.ajog.2023.03.010.

[6] Shi Y, et al. A control method with reinforcement learning for urban un-signalized intersection in hybrid traffic environment. *Sensors (Basel)*. 2022;22(3):779-782. DOI: 10.3390/s22030779.

[7] Liu H, et al. Automatic lane-level intersection map generation using low-channel roadside LiDAR. *IEEE-CAA Journal of Automatica Sinica*. 2023;10(5):1209-1222. DOI: 10.1109/JAS.2023.123183.

[8] Maadi S, et al. Real-time adaptive traffic signal control in a connected and automated vehicle environment: optimisation of signal planning with reinforcement learning under vehicle speed guidance. *Sensors*. 2022;22(19):7501-7509. DOI: 10.3390/s22197501.

[9] Qu S, et al. Behavioral patterns of drivers under signalized and unsignalised urban intersections. *Applied Sciences*, 2024, 14(5): 1802-1807. DOI: 10.3390/App14051802.

[10] Huang Z, Wu J, Lv C. Efficient deep reinforcement learning with imitative expert priors for autonomous driving. *IEEE Trans Neural Netw Learn Syst*. 2023;34(10):7391-7403. DOI: 10.1109/tnnls.2022.3142822.

[11] Chen J, Li SE, Tomizuka M. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(6):5068-5078. DOI: 10.1109/TITS.2020.3046646.

[12] Fuchs F, et al. Super-human performance in gran turismo sport using deep reinforcement learning. *IEEE Robotics and Automation Letters*. 2021;6(3):4257-4264. DOI: 10.1109/LRA.2021.3064284.

[13] Pang H, Wang Z, Li G. Large language model guided deep reinforcement learning for decision making in autonomous driving. *Arxiv Preprint Arxiv*. 2024. DOI: 10.48550/arXiv.2412.18511.

[14] Al-Sharman M, et al. Autonomous driving at unsignalised intersections: A review of decision-making challenges and reinforcement learning-based solutions. *Arxiv Preprint Arxiv*. 2024. DOI: 10.48550/arXiv.2409.13144.

[15] Liu J, et al. MTD-GPT: A multi-task decision-making GPT model for autonomous driving at unsignalised intersections. In: Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC); 2023 Sep 24-28; Bilbao, Spain. p. 5154-5161. DOI: 10.1109/ITSC57777.2023.10421993.

[16] Kodati D, Tene R. Identifying suicidal emotions on social media through transformer-based deep learning. *Applied Intelligence*. 2023;53(10):11885-11917. DOI: 10.1007/s10489-022-04060-8.

[17] Ji Y, et al. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*. 2021;37(15):2112-2120. DOI: 10.1093/bioinformatics/btab083.

[18] Smarandache F. Plithogeny, plithogenic set, logic, probability and statistics: a short review. *Journal of Computational and Cognitive Engineering*. 2022;1(2):47-50. DOI: 10.47852/bonviewJCCE2202191.

[19] Guo Y, Mustafaoglu Z, Koundal D. Spam detection using bidirectional transformers and machine learning classifier algorithms. *Journal of Computational and Cognitive Engineering*. 2023;2(1):5-9. DOI: 10.47852/bonviewJCCE2202192.

[20] Lei L, et al. Deep reinforcement learning for autonomous internet of things: Model, applications and challenges. *IEEE Communications Surveys and Tutorials*. 2020;22(3):1722-1760. DOI: 10.1109/COMST.2020.2988367.

[21] Kuutti S, et al. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*. 2021;22(2):712-733. DOI: 10.1109/TITS.2019.2962338.

[22] Aradi S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(2):740-759. DOI: 10.1109/TITS.2020.3024655.