Self-play makes LLM a generalizable safe and strategic driver

Xiangyu Li
Dept. Electrical and Computer Engineering
Northwestern University
Evanston, USA
xiangyuli2027@u.northwestern.edu

Ruochen Jiao
Dept. Electrical and Computer Engineering
Northwestern University
Evanston, USA
ruochen.jiao@u.northwestern.edu

Xiangyu Shi
Dept. Electrical and Computer Engineering
Northwestern University
Evanston, USA
xiangyushi2029@u.northwestern.edu

Qi Zhu
Dept. Electrical and Computer Engineering
Northwestern University
Evanston, USA
qzhu@northwestern.edu

***Abstract*—This project addresses the challenge of optimizing decision-making strategies for autonomous vehicles (AVs) in complex multi-agent traffic scenarios, particularly where interactions involve cooperation or competition. Traditional supervised learning methods often require large-scale labeled datasets, which are costly and insufficient for capturing dynamic, real-world traffic interactions. Furthermore, existing models lack adaptability and reasoning capabilities when facing novel or adversarial behaviors from other agents. To address these limitations, we propose a novel framework that integrates reinforcement learning (RL) from self-play with Large Language Models (LLMs) as decision-makers. Inspired by advancements in self-playing adversarial language games, this approach enables LLMs to dynamically simulate interactions between AVs, generating diverse and realistic training data.**

In this framework, vehicles engage in iterative self-play to predict and adapt to the strategies of other agents. For example, one vehicle simulates merging strategies while another evaluates cooperative or competitive responses, fostering mutual adaptation and robust decision-making. This process not only advances autonomous driving strategies through scalable and efficient self-play but also enhances the reasoning capabilities of LLMs, improving their problem-solving skills and transferability across domains. Moreover, by generating training data through simulated interactions, the framework reduces reliance on large-scale supervised datasets, significantly lowering computational and data acquisition costs. This method represents a novel and efficient approach to autonomous driving and offers broader implications for multi-agent systems and general AI reasoning. The code is available at https://github.com/Xiangyu-Li-nu.

***Index Terms*—self-play, large-language model, reinforcement learning, autonomous driving**

I. Introduction

The rapid advancements in artificial intelligence have greatly accelerated progress in autonomous driving. However, optimizing decision-making strategies in dynamic and complex traffic environments remains a significant challenge. Traditional methods, including rule-based systems, machine learning models, and reinforcement learning, exhibit notable limitations in handling complex scenarios and adapting to novel situations.

Rule-based systems rely on predefined logic rules, which, while suitable for simple traffic scenarios, lack flexibility and are unable to address rare or complex interactions. Moreover, as scenarios grow in complexity, the scalability and maintainability of rule-based systems become increasingly difficult. Supervised learning-based machine learning models depend heavily on large-scale annotated datasets,

978-1-7281-4387-3/19/$31.00 © 2019 IEEE

leading to high data acquisition costs and susceptibility to data bias and incompleteness. These models also struggle with edge cases and exhibit limited capacity for exploration in unknown situations.

Reinforcement learning (RL), as a promising alternative, focuses on learning strategies through interaction with environnment to optimize decision-making. However, traditional RL approaches demand a vast amount of high-fidelity simulation data, posing significant computational and temporal challenges. Additionally, RL methods may exhibit unsafe exploratory behavior during training, limiting their applicability to real-world autonomous driving scenarios. Furthermore, RL models often face generalization issues, making it difficult to adapt to dynamic and unpredictable traffic environments.

The emergence of large language models (LLMs) offers a new paradigm for addressing these challenges. LLMs, trained on extensive text datasets, have demonstrated exceptional capabilities in language understanding and reasoning. Recent studies highlight their applications in transportation and robotics, where they can provide interpretable decisions and enable strategic reasoning in multi-agent systems. These unique characteristics make LLMs a promising tool for tackling the complex interactions inherent in autonomous driving.

This project introduces an innovative framework that integrates reinforcement learning with large language models to optimize decision-making in multi-agent traffic scenarios. Specifically, it employs a self-play mechanism, where autonomous agents simulate cooperative or competitive interactions to dynamically generate diverse and realistic training data. This approach reduces reliance on large-scale labeled datasets and significantly lowers the costs of data acquisition and model training. Additionally, the self-play process enhances the reasoning capabilities of LLMs, enabling them to predict the strategies of other agents and adapt effectively.

In the proposed scenario, two autonomous vehicles (AVs) interact: a ramp autonomous vehicle (RAV) attempting to merge onto the main road and a main road autonomous vehicle (MAV) deciding whether to cooperate (e.g., slowing down or changing lanes) or compete (e.g., maintaining speed or accelerating). Each vehicle uses its LLM to analyze the other's historical behaviors and optimize its strategy in real-time. The RAV adjusts its merging strategy based on its prediction of the MAV's actions, while the MAV decides its actions to optimize its objectives, such as efficiency or minimizing delays. This dynamic interaction fosters mutual strategy adaptation, ultimately improving safety, efficiency, and merging success.

By combining reinforcement learning, self-play, and the reasoning abilities of large language models, this study aims to develop a scalable and efficient decision-making framework for autonomous vehicles. The expected outcomes include safer and more efficient driving strategies and enhanced reasoning capabilities of LLMs, opening new avenues for applications in logistics, robotics, and other domains. This framework aspires to advance autonomous driving and provide novel solutions for complex decision-making in multi-agent systems.

II. Literature Review

A. Traditional Methods in AVs Decision Making

In the development of decision-making systems for autonomous vehicles (AVs), traditional approaches such as rule-based systems, machine learning models, and reinforcement learning (RL) have been extensively studied. Each method provides a distinct framework for AV decision-making, but they also exhibit specific limitations that impede their performance in complex real-world scenarios.

Rule-based systems rely on explicitly defined sets of instructions to determine vehicle behavior. These systems are advantageous in their interpretability and ease of implementation, as described in [1], but they struggle with adaptability. Specifically, the inability of rule-based systems to handle unforeseen scenarios, such as edge cases or highly dynamic environments, makes them inflexible in real-world applications [2]. Moreover, as the complexity of the driving environment grows, the number of rules increases exponentially, leading to a combinatorial explosion of possibilities that complicates system updates and maintenance [3]. The lack of a learning

mechanism further limits these systems, as they cannot adapt or improve based on new data, requiring manual adjustments to address emerging scenarios [4].

Machine learning models, particularly those based on supervised learning, have shown promise in addressing some of the limitations of rule-based systems. By leveraging large-scale annotated datasets, these models can learn complex decision-making patterns. However, as noted in [5], the high dependency on annotated datasets introduces challenges in scalability due to the cost and time associated with data collection and labeling. Furthermore, the performance of these models is often constrained by the quality and diversity of the training data. Biases inherent in the dataset can result in poor generalization to underrepresented scenarios, such as rare traffic conditions or adverse weather. Additionally, machine learning models typically exhibit limited robustness in unfamiliar environments, as they are optimized for the distribution of their training data and fail to extrapolate effectively to novel situations.

Reinforcement learning (RL) presents a more dynamic approach to decision-making by allowing agents to learn optimal policies through interactions with their environment. This method has been extensively explored in AV decision-making tasks, as discussed in [6]. RL offers the advantage of enabling agents to discover strategies that maximize long-term rewards, even in the absence of explicit supervision. However, the practical implementation of RL faces significant hurdles. First, RL algorithms are highly data-intensive, requiring large amounts of interaction data to converge on optimal policies. This issue is compounded by the need for high-fidelity simulation environments to generate realistic training scenarios, which are computationally expensive to develop and maintain. Additionally, RL agents often exhibit unsafe behaviors during exploration phases, rendering them unsuitable for direct deployment in real-world settings without extensive safety precautions [7]. The problem of poor generalization also persists in RL, as policies trained in simulated environments frequently fail to transfer effectively to real-world conditions due to the ”reality gap”.

In summary, while traditional methods such as rule-based systems, machine learning models, and RL have contributed significantly to the evolution of AV decision-making, their limitations highlight the need for more robust, scalable, and adaptable solutions. Future research must address these challenges to enable AVs to operate reliably in diverse and complex environments.

B. Development of LLM in Relevant Applications

Large Language Models (LLMs) have become integral to various domains due to their ability to understand and generate human-like text. In autonomous vehicle (AV) applications, LLMs are employed for interpretable decision-making, enhancing interaction between AVs and users. For instance, LLMs improve communication by explaining decisions and predicting human intent [8]. They also enable strategic reasoning in multi-agent systems, such as fleet management and resource allocation [9]. Additionally, LLMs assist in end-to-end planning by interpreting semantic-level cues like traffic rules or human instructions [10].

However, these applications face challenges, including reliance on costly annotated datasets and difficulty generalizing to unseen scenarios. These limitations highlight the need for innovative methods to enhance adaptability and reasoning in LLM-based AV systems.

C. Self-Play and Its Impact on LLM Reasoning

Recently, the work ”Self-play” [11] has gained much research interests, which introduces a mechanism where models engage in adversarial or cooperative interactions with themselves to refine reasoning and decision-making. By simulating diverse scenarios, self-play improves generalization and reduces the need for extensive external data. This method has proven effective in enhancing model robustness, particularly in multi-agent and dynamic environments.

Inspired by self-play, we propose integrating this mechanism into LLM-based AV agents. By simulating adversarial or cooperative driving scenarios, LLM agents can autonomously explore edge

cases and refine their strategies. This approach reduces dependence on annotated datasets while enhancing adaptability and decision-making. Furthermore, self-play improves multi-agent interactions, enabling AVs to coordinate effectively in complex, real-world environments. Our design leverages self-play to bridge traditional AV methods and the dynamic requirements of autonomous driving.

III. Problem Definition

A. Scenario Description

In this scenario, an autonomous vehicle on a ramp named ramp autonomous vehicle (RAV) aims to merge into the main road, while an autonomous vehicle named main road autonomous vehicle (MAV) on the main road faces a decision:

- **Cooperation:** The MAV can choose to yield by changing lanes or reducing its speed, facilitating the RAV's merge. The RAV may decelerate and wait for the MAV to pass before merging.
- **Competition:** The MAV can maintain its current lane and speed, or accelerate, focusing more on its own efficiency without providing convenience to the RAV. The RAV may accelerate to merge.

The RAV determines its strategy based on a prediction of the MAV's behavior:

- If the RAV predicts that the MAV will **cooperate**, it may choose to merge aggressively.
- If the RAV predicts that the MAV will **compete**, it may choose to delay its merge or adjust its acceleration.

Both vehicles use their respective large language models (LLMs) to analyze the situation and make decisions:

- The MAV uses its LLM to analyze the historical behavior of the RAV, assess whether cooperation or competition would optimize its objectives, such as maintaining efficiency or avoiding delays.
- The RAV leverages its LLM to analyze the historical behavior of the MAV, predict its current strategy, and decide its own merging approach.

This interaction leads to a dynamic decision-making process where both AVs adapt their strategies in real-time to achieve their respective goals of safety, efficiency, and merging success.

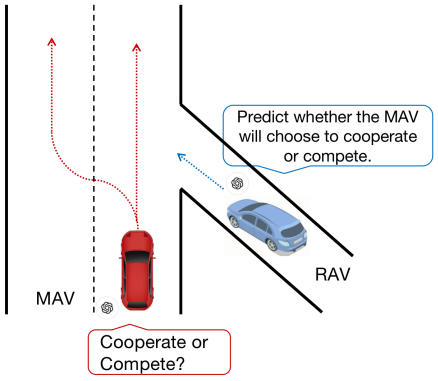

Fig. 1. Transportation scenario setup

IV. Game Setup

Players:

- **RAV:** Chooses acceleration/merging strategy based on MAV's predicted behavior.
- **MAV:** Decides whether to cooperate or compete based on historical observation of RAV.

A. State

In this simulation environment, we define the states for the two vehicles as follows:

- **RAV State:** The state of RAV, denoted as $s_{r,t}$, includes:

$$s_{r,t} = [x_{r,t}, lane_{r,t}, v_{r,t}, a_{r,t}, P_{cooperate,r}],$$

where:

- $x_{r,t}$: position of the RAV in $lane_r$ at timestep t.
- $lane_{r,t}$: The lane number where the RAV is located at timestep t.
- $v_{r,t}$: The speed of the RAV at timestep t.
- $a_{r,t}$: The accelerations of the RAV at timestep t.
- $P_{cooperate,r}$: Probability of cooperation as predicted by the LLM for the MAV at timestep t.

- **MAV State:** The state of MAV, denoted as $s_{m,t}$, includes:

$$s_{m,t} = [x_{m,t}, lane_{m,t}, v_{m,t}, a_{m,t}, P_{cooperate,m}],$$

where:

- $x_{m,t}$: position of the MAV in $lane_r$ at timestep t.
- $lane_{m,t}$: The lane number where the MAV is located at timestep t.
- $v_{m,t}$: The speed of the MAV at timestep t.
- $a_{m,t}$: The accelerations of the MAV at timestep t.
- $P_{cooperate,m}$: Probability of cooperation as predicted by the LLM for the RAV at timestep t.

B. Dynamic Updates

The states for both vehicles evolve over time based on their respective dynamics:

$$s_{t+1} = f(s_t, u_t)$$

$$s_t = [s_{r,t}, s_{m,t}]$$

where:

- s_t: Current state of the vehicle.
- $u_t = [a_{m,t}, a_{r,t}, lane_{m,t}, lane_{r,t}]$: Control inputs, representing the accelerations and lane-changing behavior of MAV and RAV.
- f: Vehicle dynamics model.

C. Lane-Specific Behavior

For lane information, we define:

$$lane_{m,t}, lane_{r,t}, = \begin{cases} 0 & \text{if the vehicle is in the left lane of the main road.} \\ 1 & \text{if the vehicle is in the right lane of the main road.} \\ 2 & \text{if the vehicle is located at the ramp.} \end{cases}$$

LLM Integration for Predictions

The LLM predicts the probability of cooperation based on historical observations:

$$P_{cooperate,r} = \text{LLM}\left(\text{history of MAV's actions}\right),$$
$$P_{cooperate,m} = \text{LLM}\left(\text{history of RAV's actions}\right).$$

D. Prompt Framework for Action Generation by LLMs

The decision is based on analyzing lane positions, velocities, accelerations, and the predicted cooperation probabilities.

MAV Action Generation Prompt

The MAV's decision-making process involves the following steps:

$$\text{Action}_{\text{MAV}} = \arg \max_{a_{\text{MAV}} \in \{\text{Cooperate, Compete}\}} R_{\text{MAV}}(s_t, a_{\text{MAV}}, a_{\text{RAV}}),$$

where:

- $R_{\text{MAV}}(s, a_{\text{MAV}}, a_{\text{RAV}})$: The reward function for the MAV, dependent on the state s, its own action a_{MAV}, and the RAV's predicted action a_{RAV}.

a) LLM Prompt for MAV Action::

- Input: $(s_{m,input} = [x_{m,t}, lane_{m,t}, v_{m,t}, a_{m,t}, P_{\text{cooperate},m}, x_{r,t}, lane_{r,t}, v_{r,t}, a_{r,t},])$
- Query:

Based on the current state:
Lane: $lane_{m,t}, lane_{r,t}$ Position: $x_{m,t}, x_{r,t}$
Velocity: $v_{m,t}, v_{r,t}$
Acceleration: $a_{m,t}, a_{r,t}$
Predicted cooperation probability from RAV: $P_{\text{cooperate},m}$,
Should MAV cooperate or compete?

- Output: Action recommendation: *Cooperate* or *Compete*? *Accelerate*, *Decelerate*, or *Lane-changing*.

Query

In this scenario, you are driving a **Main Autonomous Vehicle (MAV)** on the main road while a **Remote Autonomous Vehicle (RAV)** is attempting to merge into your lane. Your decision will impact both safety and efficiency. Should you cooperate (e.g., decelerate or change lanes) or compete (e.g., maintain speed or accelerate)?
Input State Information:
Historical MAV State: - Position: 116, Lane: 3, Speed: 24.7767, Acceleration: 0
- Position: 103, Lane: 2, Speed: 22.0, Acceleration: 0

Current MAV State: - Position: 103, Lane: 2, Speed: 22.0, Acceleration: 0

Historical RAV State: - Position: 116, Lane: 3, Speed: 24.7767, Acceleration: 0
- Position: 103, Lane: 2, Speed: 22.0, Acceleration: 0

Current RAV State: - Position: 116, Lane: 3, Speed: 24.7767, Acceleration: 0

RAV's Behavior Prediction:
Predict RAV's cooperation probability $P_{cooperate,r}$ based on its historical states:

$$P_{cooperate,r} = \begin{cases} 0, & \text{if RAV is likely to compete.} \\ 1, & \text{if RAV is likely to cooperate.} \end{cases}$$

Decision Explanation:

Output

1. **RAV's Decision Prediction:** Based on the historical behavior of the RAV, I predict that the RAV will cooperate. The probability of cooperation is high, $P_{cooperate} = 1$, since the RAV has a consistent pattern of merging into the lane in a controlled manner.
2. **Decision and Explanation:** Based on the prediction that the RAV will cooperate, my decision is to also cooperate. This means I will decelerate to allow the RAV to merge smoothly into my lane.

- **Action:** Decelerate (This action is selected from the cooperation options: decelerate, maintain speed). Decelerating ensures a safe and efficient merging process for both vehicles, minimizing the risk of collision and improving the overall traffic flow. Maintaining speed or accelerating could lead to a conflict, as it would block the RAV's attempt to merge, which could reduce safety and efficiency.

E. RAV Action Generation Prompt

The RAV's decision-making process involves:

$$\text{Action}_{\text{RAV}} = \arg \max_{a_{\text{RAV}} \in \{\text{Accelerate, Decelerate, Merge}\}} R_{\text{RAV}}(s, a_{\text{RAV}}, a_{\text{MAV}}),$$

where:

- $R_{\text{RAV}}(s, a_{\text{RAV}}, a_{\text{MAV}})$: The reward function for the RAV, dependent on the state s, its own action a_{RAV}, and the MAV's predicted action a_{MAV}.

a) LLM Prompt for RAV Action::

- Input:$(s_{m,input} = [x_{m,t}, lane_{m,t}, v_{m,t}, a_{m,t}, P_{\text{cooperate},r}, x_{r,t}, lane_{r,t}, v_{r,t}, a_{r,t},])$
- Query:

  Based on the current state:
  Lane: $lane_{m,t}, lane_{r,t}$ Position: $x_{m,t}, x_{r,t}$
  Velocity: $v_{m,t}, v_{r,t}$ Acceleration: $a_{m,t}, a_{r,t}$
  Predicted cooperation probability from MAV: $P_{\text{cooperate},r}$,
  Should RAV cooperate, compete?

- Output: Action recommendation: *Cooperate* or *Compete*? *Accelerate*, *Decelerate*, or *Merge*.

Query

In this scenario, you are driving a **Remote Autonomous Vehicle (RAV)** on the main road while a **Main Autonomous Vehicle (MAV)** is in the same lane. Your decision will impact both safety and efficiency. Should you cooperate (e.g., decelerate, change lanes) or compete (e.g., maintain speed or accelerate)?
Input State Information:
Historical RAV State: - Position: 116, Lane: 3, Speed: 24.7767, Acceleration: 0
- Position: 103, Lane: 2, Speed: 22.0, Acceleration: 0

Current RAV State: - Position: 116, Lane: 3, Speed: 24.7767, Acceleration: 0

Historical MAV State: - Position: 103, Lane: 2, Speed: 22.0, Acceleration: 0
- Position: 116, Lane: 3, Speed: 24.7767, Acceleration: 0

Current MAV State: - Position: 103, Lane: 2, Speed: 22.0, Acceleration: 0

RAV's Behavior Prediction:
Predict RAV's cooperation probability $P_{cooperate,r}$ based on its historical states:

$$P_{cooperate,r} = \begin{cases} 0, & \text{if RAV is likely to compete.} \\ 1, & \text{if RAV is likely to compete.} \end{cases}$$

Decision Explanation:

Output

1. **RAV's Decision Prediction:** Based on the historical behavior of the RAV, I predict that the RAV will cooperate. The probability of cooperation is high, $P_{cooperate} = 1$, since the RAV has a consistent pattern of attempting to merge into the lane in a controlled manner.
2. **Decision and Explanation:** Based on the prediction that the MAV will cooperate, my decision is also to cooperate. This means I will either decelerate or change lanes to allow the MAV to pass or merge smoothly.

- **Action:** Change lane (This action is selected from the cooperation options: decelerate, change lane). Changing lanes ensures a safe and efficient merging process for both vehicles, minimizing the risk of collision and improving the overall traffic flow. Decelerating could also be an option, but changing lanes allows both vehicles to continue without affecting the flow of traffic too much. Maintaining speed or accelerating could lead to a conflict, as it could block the MAV's ability to merge, which would reduce safety and efficiency.

F. Reward Functions

G. Safety Component Using Time-to-Collision (TTC)

The Safety Component is represented using Time-to-Collision (TTC), defined as:

$$\text{Risk of Collision}(x_{m,t}, x_{r,t}, v_{m,t}, v_{r,t}) = \begin{cases} \frac{|x_{r,t} - x_{m,t}|}{|v_{m,t} - v_{r,t}|}, & \text{if } v_{m,t} \neq v_{r,t} \text{ and } |v_{m,t} - v_{r,t}| > \epsilon, \\ & \text{if } lanem, t = 1, lane_{r,t} = 1 \\ & \text{and } x_{r,t} >= 0, \text{else if } lanem, t = 1, \\ & lane_{r,t} = 2 \text{ and } x_{m,t} >= 0, x_{r,t} < 0 \\ \infty, & \text{otherwise,} \end{cases}$$

where:

- $x_{m,t}, x_{r,t}$: Positions of two vehicles (e.g., MAV and RAV).
- $v_{m,t}, v_{r,t}$: Velocities of two vehicles.
- ϵ: A small threshold to avoid division by zero, here we use 10^{-6}.

When TTC falls below a threshold τ_{TTC}, a penalty is applied:

$$\text{R}_{\text{safety}} = \begin{cases} \frac{\tau_{\text{TTC}} - \text{TTC}}{\tau_{\text{TTC}}}, & \text{if TTC} < \tau_{\text{TTC}=2s}, \\ 0, & \text{otherwise.} \end{cases}$$

H. Game Outcome Component

The game outcome rewards the vehicle that achieves its objective (e.g., leading through the ramp or merging successfully):

$$R_{\text{outcome}} = \begin{cases} +\gamma_{\text{win}}, & \text{if the vehicle achieves its goal (e.g., merge first),} \\ -\gamma_{\text{lose}}, & \text{if the vehicle fails to achieve its goal.} \end{cases}$$

where:

- $\gamma_{\text{win}} > 0$: Positive reward for winning (e.g., successful merging or maintaining lead).
- $\gamma_{\text{lose}} > 0$: Negative reward for losing (e.g., failing to merge or being overtaken).

I. Additional Factors

a) Time Delay Penalty:: Encourages efficient behavior by penalizing time delays:

$$R_{\text{time}} = -\lambda_{\text{time}} \cdot \text{Time Delay},$$

where:

- $\lambda_{\text{time}} > 0$: Weight for time delay penalty.

b) Speed Control Penalty:: Encourages smooth traffic flow by penalizing extreme acceleration or deceleration:

$$R_{\text{speed}} = -\lambda_{\text{speed}} \cdot \left| a - a_{\text{desired}} \right|,$$

where:

- a: Current acceleration.
- a_{desired}: Desired acceleration for smooth traffic flow.
- $\lambda_{\text{speed}} > 0$: Weight for speed control penalty.

c) Global Traffic Efficiency Reward:: Promotes overall system efficiency:

$$R_{\text{system}} = +\lambda_{\text{system}} \cdot \text{Traffic Flow Efficiency},$$

where:

- $\lambda_{\text{system}} > 0$: Weight for system efficiency.
- Traffic Flow Efficiency: Measured by average vehicle speed or throughput.

J. Final Reward Functions for MAV and RAV

a) MAV Reward::

$$\begin{aligned} R_{\text{MAV}} = {} & R_{\text{safety}} + R_{\text{outcome}} \\ & - \lambda_{\text{time}} \cdot \text{Time Delay} \\ & - \lambda_{\text{speed}} \cdot |a_{\text{MAV}} - a_{\text{desired}}| + \lambda_{\text{system}} \cdot \text{Traffic Flow Efficiency}. \end{aligned}$$

b) RAV Reward::

$$\begin{aligned} R_{\text{RAV}} = {} & R_{\text{safety}} + R_{\text{outcome}} - \lambda_{\text{time}} \cdot \text{Time Delay} \\ & - \lambda_{\text{speed}} \cdot |a_{\text{RAV}} - a_{\text{desired}}| \\ & + \lambda_{\text{system}} \cdot \text{Traffic Flow Efficiency}. \end{aligned}$$

V. Explanation

- Safety Component: Ensures both vehicles prioritize avoiding collisions.
- Game Outcome: Incentivizes competition or cooperation based on their goals (e.g., merge success or lane holding).
- Time Delay Penalty: Encourages timely decision-making.
- Speed Control Penalty: Smoothens acceleration and deceleration to reduce disruptions.
- Global Efficiency: Aligns local decisions with overall traffic system performance.

A. Dynamic Updates for State Transitions

The state transitions are defined as:

$$\begin{aligned} x_{t+1} &= x_t + v_t \cdot \Delta t, \\ v_{t+1} &= v_t + a_t \cdot \Delta t, \\ l_{t+1} &= l_t + \Delta l, \end{aligned}$$

where:

- x_t, v_t, a_t: Current position, velocity, and acceleration.
- Δt: Time step.
- l_t: Current lane index.
- Δl: Lane change indicator.

VI. Imitation Learning

Due to the limited capabilities of current open-source LLMs, the generation policy $\pi_\theta(y|x)$ may not strictly follow the game rules specified in prompts for RAV and MAV decision-making. Therefore, before initiating self-play, we conduct **imitation learning** (behavior cloning) using a dataset of high-quality outputs generated by GPT-4. To ensure consistency with the game rules, we

separately train $\pi_\theta(u|\mathrm{f}_{\mathrm{MAV}}(s))$ and $\pi_\theta(v|\mathrm{f}_{\mathrm{RAV}}(s))$ on data selected based on reward values.

Specifically, we collect the data and divide it into two subsets:

- **RAV dataset:** A subset $\mathcal{T}_{\mathrm{im}}^{\mathrm{RAV}}$ where the RAV receives higher rewards, i.e., $R(\tau) > 0$.
- **MAV dataset:** A subset $\mathcal{T}_{\mathrm{im}}^{\mathrm{MAV}}$ where the MAV receives higher rewards, i.e., $R(\tau) > 0$.

The imitation learning objective for each vehicle is to maximize the log-likelihood of its actions based on the respective dataset:

$$\mathcal{L}_{\mathrm{im}}^{\mathrm{MAV}}(\pi_\theta) = -\mathbb{E}_{\tau \in \mathcal{T}_{\mathrm{im}}^{\mathrm{MAV}}} \left[\frac{1}{T} \sum_{t=1}^{T} \log \pi_\theta(u_t|\mathrm{f}_{\mathrm{MAV}}(s_t)) + \beta_1 \mathrm{KL}[\pi_\theta||\pi_{\mathrm{ref}}] \right],$$

$$\mathcal{L}_{\mathrm{im}}^{\mathrm{RAV}}(\pi_\theta) = -\mathbb{E}_{\tau \in \mathcal{T}_{\mathrm{im}}^{\mathrm{RAV}}} \left[\frac{1}{T} \sum_{t=1}^{T} \log \pi_\theta(v_t|\mathrm{f}_{\mathrm{RAV}}(s_t)) + \beta_1 \mathrm{KL}[\pi_\theta||\pi_{\mathrm{ref}}] \right].$$

Here:

- $\beta_1 > 0$ is a re-weighting parameter.
- The KL regularizer $\mathrm{KL}[\pi_\theta||\pi_{\mathrm{ref}}]$ prevents overfitting and maintains the general language modeling capabilities of the LLM.
- π_{ref} is the initial checkpoint of the LLM before training.

The overall imitation learning objective combines the two components:

$$\mathcal{L}_{\mathrm{im}}(\pi_\theta) = \frac{1}{2}\mathcal{L}_{\mathrm{im}}^{\mathrm{MAV}}(\pi_\theta) + \frac{1}{2}\mathcal{L}_{\mathrm{im}}^{\mathrm{RAV}}(\pi_\theta).$$

By training on datasets where the RAV and MAV achieve higher rewards, we ensure the model captures effective decision-making strategies for both vehicles. This imitation learning step establishes a solid foundation for subsequent self-play and further optimization.

VII. Reinforcement Learning from Self-Play

To ensure the LLM adheres to the rules of the MAV and RAV decision-making process, We generate self-play training data based on a reinforcement learning environment. Self-play allows the LLM policy $\pi_\theta(y|x)$ to alternate between the MAV and RAV roles, generating data for both vehicles. The high computational complexity of multi-turn text generation during self-play is mitigated using an offline learning scheme. This process consists of the following steps:

1) Copy the current LLM policy π_θ as $\pi_{\tilde{\theta}}$.
2) Conduct self-play episodes by setting $\mu_{\tilde{\theta}}(u|s_t) = \pi_{\tilde{\theta}}(u|\mathrm{f}_{\mathrm{MAV}}(s_t))$ and $\nu_{\tilde{\theta}}(v|s_t) = \pi_{\tilde{\theta}}(v|\mathrm{f}_{\mathrm{RAV}}(s_t))$.
3) Collect episodes $\mathcal{T}_{\tilde{\theta}} = \{\tau \sim \mu_{\tilde{\theta}} \times \nu_{\tilde{\theta}}\}$ and split them into:
   - $\mathcal{T}_{\tilde{\theta}}^{\mathrm{MAV}}$: MAV-winning episodes where $R(\tau) > 0$.
   - $\mathcal{T}_{\tilde{\theta}}^{\mathrm{RAV}}$: RAV-winning episodes where $R(\tau) < 0$.
4) Filter the collected data to identify high-quality episodes based on the reward function and use these for supervised fine-tuning.

The objective is to refine the LLM such that the latent semantic distribution in its vector space aligns more closely with good data and diverges from bad outcomes. For this, we define the supervised fine-tuning loss as follows:

$$\mathcal{L}_{\mathrm{sp}}^{\mathrm{MAV}}(\pi_\theta) = -\mathbb{E}_{\tau \in \mathcal{T}_{\tilde{\theta}}^{\mathrm{MAV}}} \left[\frac{1}{T} \sum_{t=1}^{T} \frac{\mu_{\tilde{\theta}}(u_t|s_t)}{\pi_\theta(u_t|s_t)} \hat{A}_t^{\mu_{\tilde{\theta}}} - \beta_2 \mathrm{KL}[\pi_\theta||\pi_{\tilde{\theta}}] \right],$$

$$\mathcal{L}_{\mathrm{sp}}^{\mathrm{RAV}}(\pi_\theta) = -\mathbb{E}_{\tau \in \mathcal{T}_{\tilde{\theta}}^{\mathrm{RAV}}} \left[\frac{1}{T} \sum_{t=1}^{T} \frac{\nu_{\tilde{\theta}}(v_t|s_t)}{\pi_\theta(v_t|s_t)} \hat{A}_t^{\nu_{\tilde{\theta}}} - \beta_2 \mathrm{KL}[\pi_\theta||\pi_{\tilde{\theta}}] \right],$$

where:

- $\hat{A}_t^{\mu_{\tilde{\theta}}}$ and $\hat{A}_t^{\nu_{\tilde{\theta}}}$ are the estimated advantages for MAV and RAV actions, respectively.
- $\beta_2 > 0$ is a re-weighting parameter to balance the KL regularizer.
- $\mathrm{KL}[\pi_\theta||\pi_{\tilde{\theta}}]$ ensures the updated policy remains close to the original to maintain stability.

The overall self-play optimization objective is:

$$\mathcal{L}_{\mathrm{SP}}(\pi_\theta) = \frac{1}{2}\mathcal{L}_{\mathrm{sp}}^{\mathrm{MAV}}(\pi_\theta) + \frac{1}{2}\mathcal{L}_{\mathrm{sp}}^{\mathrm{RAV}}(\pi_\theta).$$

Through this process, the LLM refines its ability to generate consistent and high-quality outputs for both MAV and RAV roles, ensuring alignment with the reward function while improving its decision-making capabilities.

VIII. Experiments

To evaluate the effectiveness of our MAV and RAV self-play framework, we designed experiments using open-source pretrained LLMs, specifically focusing on LLaMA-based models. The training process includes three main stages: in-context learning,

imitation learning using high-quality data generated by GPT-4, and reinforcement learning through self-play episodes. For baseline comparison, we considered both standard supervised fine-tuning (SFT) methods and imitation learning frameworks.

We chose two open-source LLaMA models (LLaMA 3.2-1b-Instruct, LLaMA 3.2-3b-Instruct, and LLaMA 3.1-8b-Instruct for training. These models were selected because they are smaller in size, making them easier to train, and their performance leaves much room for improvement. Testing showed that GPT-4 could generate answers with the expected format, which means a strong understanding of the problem. However, the open-source LLaMA models are hard to strictly follow prompts, which indicates gaps in comprehension. This highlights the importance of in-context learning and imitation learning during pretraining.

A. Experimental Setups

1) Training Data Preparation: The training data preparation process includes the following steps:

- **In-context Learning:** We used few-shot prompts to guide GPT-4 to generate structured question-answer datasets, enabling the smaller open-source models to learn the MAV and RAV decision-making process.

EXAMPLE: MAV DECISION

Question

In this scenario, you are driving a **Main Autonomous Vehicle (MAV)** on the main road while a **Remote Autonomous Vehicle (RAV)** is attempting to merge into your lane. Your decision will impact both safety and efficiency. Should you cooperate (e.g., decelerate or change lanes) or compete (e.g., maintain speed or accelerate)?
Input State Information: **Historical MAV State:** - Position: 105, Lane: 2, Speed: 25, Acceleration: -2 - Position: 80.0, Lane: 2, Speed: 25, Acceleration: 0.0 - Position: 55.0, Lane: 2, Speed: 25, Acceleration: 0.0
Current MAV State: - Position: 30.0, Lane: 2, Speed: 25, Acceleration: 0.0
Historical RAV State: - Position: 129, Lane: 2, Speed: 19, Acceleration: 1 - Position: 110.0, Lane: 2, Speed: 19, Acceleration: 1 - Position: 91.0, Lane: 2, Speed: 19, Acceleration: -1.0
Current RAV State: - Position: 72.0, Lane: 2, Speed: 19, Acceleration: -3.0

Consider the above information and justify your decision in terms of both safety and efficiency.

Answer

Decision: Cooperate
Detailed Reasoning: 1. **RAV's Predicted Behavior:** - The RAV's historical deceleration (-1 and -3) suggests a yielding tendency, indicating that the RAV is not aggressively competing for lane space. - By maintaining a cooperative stance, the MAV can ensure smooth integration of the RAV into traffic.
2. **Safety Consideration:** - Cooperating by maintaining speed avoids abrupt changes, reducing the risk of collision or sudden lane shifts. - With a safe buffer between the MAV and RAV, maintaining speed ensures continued momentum while aligning with the RAV's predicted cooperative behavior.
3. **Efficiency Consideration:** - Decelerating unnecessarily could disrupt the flow of traffic and reduce overall efficiency. - Maintaining speed optimizes both time and safety, avoiding unnecessary delays.

Recommended Action: Maintain Speed - **Why:** The MAV is currently ahead of the RAV (position 30.0 vs. 72.0) with no immediate risk of collision. The RAV's yielding behavior further supports the decision to maintain speed. - **Outcome:** This action supports both safety and system efficiency, minimizing unnecessary interventions while facilitating a smooth traffic flow.

IX. EXAMPLE: RAV DECISION

Question

As the driver of a **Remote Autonomous Vehicle (RAV)**, you are attempting to merge into the main road, where a **Main Autonomous Vehicle (MAV)** is already driving. Should you cooperate (e.g., decelerate or maintain speed) or compete (e.g., accelerate or block the MAV)?
Input State Information: **Historical RAV State:** - Position: 129, Lane: 2, Speed: 19, Acceleration: 1 - Position: 110.0, Lane: 2, Speed: 19, Acceleration: 1 - Position: 91.0, Lane: 2, Speed: 19, Acceleration: -1.0
Current RAV State: - Position: 72.0, Lane: 2, Speed: 19, Acceleration: -3.0
Historical MAV State: - Position: 105, Lane: 2, Speed: 25, Acceleration: -2 - Position: 80.0, Lane: 2, Speed: 25, Acceleration: 0.0 - Position: 55.0, Lane: 2, Speed: 25, Acceleration: 0.0
Current MAV State: - Position: 30.0, Lane: 2, Speed: 25, Acceleration: 0.0
Based on the above information, justify your decision and describe the reasoning behind your actions.

Answer

Decision: Cooperate
Detailed Reasoning: 1. **MAV's Predicted Behavior:** - The MAV has shown consistent deceleration in its historical behavior, indicating a tendency to yield when necessary. - Its current steady speed (25) and non-aggressive acceleration further suggest cooperative behavior.
2. **Safety Consideration:** - By cooperating, the RAV can create a safe distance from the MAV, reducing the risk of collision. - Avoiding aggressive acceleration ensures smoother integration into the lane without disrupting the MAV's flow.
3. **Efficiency Consideration:** - Accelerating aggressively could force the MAV to slow down, leading to inefficiency and potential conflicts. - Decelerating ensures smoother merging, aligning with the MAV's cooperative stance.
Recommended Action: Decelerate - **Why:** Decelerating allows the RAV to create sufficient space for the MAV to continue without conflict. This action prioritizes safety while maintaining smooth traffic flow. - **Outcome:** This minimizes the risk of collision and promotes a harmonious merging process, ensuring the RAV integrates into traffic efficiently and safely.

- **Filtering:** After generating 1,000 question-answer pairs using GPT-4's API, we manually filtered the data to ensure high quality and alignment with the task objectives.
- **Self-Play Data:** We used the reward function to filter good and bad data generated during self-play within the reinforcement learning environment, selecting only high-quality data for supervised fine-tuning.
- **Supervised Fine-Tuning (SFT):** The fine-tuning process aligns the latent semantic distribution in the model's vector space with good data and diverges from bad data, ensuring robust performance.

1) System Specifications: The experiments were conducted using the following hardware and software setup:

- **Operating System:** Ubuntu 22.04
- **Frameworks:** PyTorch 2.1.2, Python 3.10, CUDA 11.8
- **GPU:** NVIDIA A800-80GB GPUs $\times 2$
- **CPU:** 28 vCPU Intel(R) Xeon(R) Gold 6348 @ 2.60GHz
- **Memory:** 200GB RAM

A. *Evaluation*

To evaluate the performance of the Large Language Models (LLMs) used in our autonomous decision-making framework, we employed a variety of metrics that assess both their reasoning capabilities and effectiveness in dynamic, real-world scenarios. The following metrics were used:

- **Reasoning Benchmarks:** To assess the reasoning capabilities of the models, we utilized widely recognized benchmarks, including BIG-Bench Hard (BBH), ARC, MMLU, and PIQA. These benchmarks are specifically designed to evaluate the model's ability to comprehend complex tasks and generate accurate responses across a variety of reasoning domains. By testing the models on these well-established benchmarks, we gained insight into their ability to handle different levels of reasoning complexity, ranging from basic logical inferences to more advanced problem-solving tasks. The results provide a comprehensive measure of the model's cognitive abilities and generalization performance across a broad range of challenges.
- **Game Win Rates (Self-Play Performance):** To evaluate the effectiveness of the self-play framework, we measured the win rates of the MAV and RAV models across multiple testing episodes. Win rates were calculated based on the number of wins, losses, and ties, excluding

invalid episodes where the LLM failed to follow the game rules. This metric serves as an indicator of how well the models perform in competitive, real-world-like scenarios, such as decision-making in traffic, where agents must navigate complex interactions and make strategic choices. A high win rate suggests that the model has successfully learned to cooperate, compete, and adapt to changing circumstances within the environment.

- **Scenario Generalization:** In addition to reasoning and self-play, we evaluated the models' ability to generalize across different traffic and decision-making scenarios. This included a variety of dynamic environments, such as lane-changing, merging, and exit ramp scenarios. The models were trained to handle a range of unpredictable situations, and their performance was assessed in terms of how effectively they could adapt to new, unseen scenarios. A key focus was on evaluating the models' ability to balance cooperation and competition, and how well they maintained safety and efficiency under different traffic conditions.
- **Decision-Making Accuracy:** Another important aspect of evaluation was measuring the decision-making accuracy of the LLMs in various autonomous driving and task allocation scenarios. We tested the models in environments where quick, accurate decisions were required to ensure safety and efficiency. Decision-making accuracy was calculated based on the model's ability to choose the correct actions, such as lane merging, acceleration, deceleration, or collision avoidance, relative to the given traffic or environmental conditions. High accuracy in these tasks reflects the models' ability to understand context, make safe decisions, and predict optimal actions under uncertainty.
- **Adaptability to Real-World Variability:** To assess the robustness of the models, we also tested their adaptability to real-world variability. This involved training and testing the models in scenarios with varied traffic densities, environmental conditions (e.g., weather, lighting), and unpredictable behaviors from other agents (e.g., pedestrians, other vehicles). The models' ability to adapt to these changing conditions and maintain performance without significant degradation was a key metric for evaluating their practicality in real-world applications.

1) Training Hyperparameters: We followed these configurations for training:

- **Learning Rate:** 5×10^{-6} for imitation learning, 2×10^{-6} for SFT.
- **KL Penalty Coefficients:** $\beta_1 = 0.1$, $\beta_2 = 0.2$.
- **Batch Size:** 128.
- **Max Sequence Length:** 2048.
- **Epochs:** One epoch over the offline collected trajectories.
- **Decay Parameter:** $\gamma = 0.8$.

B. Expected Results

After applying imitation learning and self-play, we expect the MAV and RAV models to achieve the following outcomes:

- **Enhanced Decision-Making and Safety:** Post-training, both MAV and RAV are expected to demonstrate safe and efficient decision-making capabilities, avoiding collisions and ensuring smooth merging in dynamic traffic environments. The reinforcement learning approach fine-tunes their ability to balance cooperation and competition, aligning with predefined traffic safety metrics. This training enables the system to respond in real-time to complex and dynamic driving scenarios, ensuring both safety and smooth driving.
- **Improved Benchmark Performance:** The models are expected to outperform their pre-trained baselines on various reasoning benchmarks, such as BIG-Bench Hard (BBH) and ARC. Compared to traditional rule-based models and standard reinforcement learning models, those trained with imitation learning and self-play show enhanced reasoning abilities, reflected in higher win rates and improved generalization. Through comparison with multiple LLaMA models and other pre-trained models, we find that our method performs better in

handling complex traffic scenarios, achieving a better balance between inference and decision-making accuracy.

- **Superior Generalization and Cross-Scenario Adaptability:** Our models exhibit significant generalization capabilities across various scenarios, especially in handling complex traffic situations. Compared to rule-based systems, reinforcement learning models, and others, our approach demonstrates greater flexibility in adapting to different traffic scenarios, such as lane changes and exit ramps. In particular, during training, the models not only achieve excellent performance in known scenarios but also handle unseen situations with higher generalization ability. This generalization capability means that the system can adapt to more varied traffic environments without relying on extensive rule adjustments or additional scene-specific labeled data, reducing the need for manual tuning while improving system reliability and efficiency.
- **Scalability and Robustness of the Model:** By leveraging self-play and reward functions, the model is robust to challenging real-world scenarios and scalable for broader autonomous driving applications. We found that, in comparison with multiple LLaMA models and other reinforcement learning models, the trained model continues to make efficient decisions in complex and dynamic traffic environments, providing excellent performance across various contexts. For example, the model effectively handles lane changes and exit ramp scenarios, ensuring stable operation while maintaining the balance between real-time response and decision accuracy.
- **Cross-Domain Application Potential:** The success of this framework is not limited to autonomous driving scenarios but shows promising potential for cross-domain applications. The trained models can be applied to traffic situations such as lane changes and exit ramps and can also be extended to other complex environments, such as logistics scheduling and robot navigation. By improving the model's generalization ability and adaptability, we can further extend this method to other fields, addressing a wide range of collaborative and decision-making problems in different industries.

X. Conclusion and Future Work

A. Conclusion

In this paper, we have introduced a novel methodology that leverages self-play and imitation learning to train Large Language Models (LLMs) for decision-making tasks in autonomous systems. By incorporating self-play, the model can iteratively improve its decision-making capabilities in dynamic environments, enhancing its ability to balance cooperation and competition. This approach, combined with task-specific prompts, allows LLMs to achieve higher performance and better generalization across different scenarios, such as lane changes, exit ramps, and other real-world traffic situations.

The results demonstrate that our method significantly improves the safety, efficiency, and decision-making abilities of autonomous agents. Specifically, the trained models outperform traditional rule-based systems and reinforcement learning models in complex decision-making tasks. By utilizing self-play, our model becomes more robust and adaptable, capable of handling diverse and unpredictable real-world situations. This methodology not only enhances the model's performance in predefined traffic environments but also improves its generalization across a wide range of scenarios, making it a more effective solution for real-time autonomous systems.

In summary, the integration of self-play, imitation learning, and task-specific prompts in training LLMs has shown substantial improvements in both the accuracy and adaptability of decision-making in autonomous systems. The results lay the groundwork for deploying LLMs in more complex, real-world applications, demonstrating the potential for these models to enhance autonomous decision-making systems.

B. Future Work

The following areas will be explored to further enhance the robustness, scalability, and applicability of the proposed methodology:

- **Incorporating More Complex Real-World Scenarios:** To further improve system robustness, we plan to incorporate more complex and unpredictable real-world scenarios, such as dynamic traffic patterns, environmental changes, and diverse user behaviors. Training models in such scenarios will ensure that the model is capable of adapting to a wide range of traffic conditions, improving safety and decision-making across various contexts.
- **Expanding Multi-Agent Collaboration:** A key direction for future research is to extend multi-agent collaboration through self-play strategies in more diverse environments. For example, in autonomous driving, self-play can be used to simulate cooperation and competition between multiple vehicles, drones, or robots, providing a more realistic and scalable approach to multi-agent systems. This will enable better coordination, resource allocation, and decision-making in complex shared environments.
- **Generalization to Other Domains:** While this study focuses on autonomous driving, the proposed methodology has the potential to be generalized to other domains, such as logistics, robotics, and smart cities. Future work will explore how the trained LLMs can be adapted to solve coordination and decision-making tasks in these new areas, extending the impact of our approach.
- **Improving LLM Adaptability and Personalization:** Over the long term, we aim to investigate ways to improve the adaptability of LLMs for personalized services. This includes adjusting the model's responses based on specific user needs, preferences, and behaviors. Personalization will allow LLMs to better serve individual users and environments, enhancing user experience and efficiency.
- **Advancing Security and Robustness:** As autonomous systems become more interconnected, robust security mechanisms are essential. Future research will focus on enhancing security protocols for data transmission, task execution, and model integrity, ensuring that autonomous systems remain resilient to malicious attacks while maintaining high performance and reliability in real-world applications.

By pursuing these future directions, we aim to refine and extend the capabilities of LLM-based autonomous decision-making systems, ultimately achieving more resilient, scalable, and adaptable solutions that meet the needs of real-time, mission-critical applications.

REFERENCES

[1] Xiao, Wei, et al. "Rule-based optimal control for autonomous driving." Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems. 2021.

[2] Wang, Hong, et al. "Ethical decision making in autonomous vehicles: Challenges and research progress." IEEE Intelligent Transportation Systems Magazine 14.1 (2020): 6-17.

[3] Bali, Ahmed, et al. "Rule based auto-scalability of IoT services for efficient edge device resource utilization." Journal of Ambient Intelligence and Humanized Computing 11 (2020): 5895-5912.

[4] Alghodhaifi, Hesham, and Sridhar Lakshmanan. "Autonomous vehicle evaluation: A comprehensive survey on modeling and simulation approaches." IEEE Access 9 (2021): 151531-151566.

[5] Swain, Nihar Ranjan, et al. "Machine Learning Algorithms for Autonomous Vehicles." Handbook of Formal Optimization. Singapore: Springer Nature Singapore, 2024. 1-54.

[6] Dinneweth, Joris, et al. "Multi-agent reinforcement learning for autonomous vehicles: A survey." Autonomous Intelligent Systems 2.1 (2022): 27.

[7] Ladosz, Pawel, et al. "Exploration in deep reinforcement learning: A survey." Information Fusion 85 (2022): 1-22.

[8] Liao, Haicheng, et al. "Gpt-4 enhanced multimodal grounding for autonomous driving: Leveraging cross-modal attention with large language models." Communications in Transportation Research 4 (2024): 100116.

[9] Li, Xinyi, et al. "A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges." Vicinagearth 1.1 (2024): 9.

[10] Nie, Ming, et al. "Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving." European Conference on Computer Vision. Springer, Cham, 2025.

[11] Cheng, Pengyu, et al. "Self-playing Adversarial Language Game Enhances LLM Reasoning." arXiv preprint arXiv:2404.10642 (2024).