

SurrealDriver: Designing Generative Driver Agent Simulation Framework in Urban Contexts based on Large Language Model

Ye Jin

Institute for AI Industry Research,
Tsinghua University
Beijing, China

Xiaoxi Shen

Institute for AI Industry Research,
Tsinghua University
Beijing, China

Huolin Peng

Institute for AI Industry Research,
Tsinghua University
Beijing, China

Xiaoan Liu

Institute for AI Industry Research,
Tsinghua University
Beijing, China

Jiayang Li

Institute for AI Industry Research,
Tsinghua University
Beijing, China

Jintao Xie

Institute for AI Industry Research,
Tsinghua University
Beijing, China

Pengzhong Gao

Institute for AI Industry Research,
Tsinghua University
Beijing, China

Guyue Zhou

Institute for AI Industry Research,
Tsinghua University
Beijing, China

Jiangtao Gong*

Institute for AI Industry Research,
Tsinghua University
Beijing, China
gongjiangtao2@gmail.com

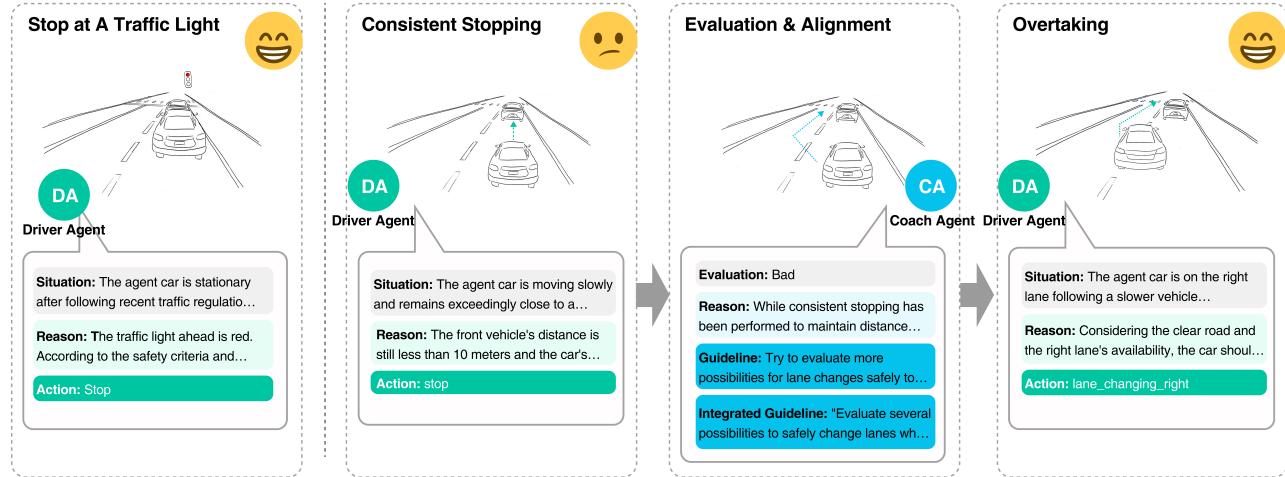


Figure 1: SurrealDriver: SurrealDriver is an LLM-powered driver agent simulation framework, which can generate human-like driving behaviors: understanding situations, reasoning, and taking actions. It also learns from expert drivers' driving advice, accumulates its own driving experience (guidelines), and continuously improves its driving skills.

*Corresponding Author

ABSTRACT

Simulation plays a critical role in the research and development of autonomous driving and intelligent transportation systems. However, the current simulation platforms exhibit limitations in the realism and diversity of agent behaviors, which impede the transfer of simulation outcomes to the real world. In this paper, we propose a generative driver agent simulation framework based on large language models (LLMs), capable of perceiving complex traffic scenarios and providing realistic driving maneuvers. Notably, we conducted interviews with 24 drivers and used their detailed descriptions of driving behavior as chain-of-thought prompts to develop a ‘coach agent’ module, which can evaluate and assist driver agents in accumulating driving experience and developing human-like driving styles. Through practical simulation experiments and user experiments, we validate the feasibility of this framework in generating reliable driver agents and analyze the roles of each module. The results show that the framework with full architect decreased the collision rate by 81.04% and increased the human-likeness by 50%. Our research proposes the first urban context driver agent simulation framework based on LLMs and provides valuable insights into the future of agent simulation for complex tasks.

CCS CONCEPTS

- Human-centered computing → Interactive systems and tools;
- Computing methodologies → Interactive simulation.

KEYWORDS

human-AI interaction, driver model, agent, generative AI, large language model, simulation framework

ACM Reference Format:

Ye Jin, Xiaoxi Shen, Huilin Peng, Xiaoan Liu, Jingli Qin, Jiayang Li, Jintao Xie, Pengzhong Gao, Guyue Zhou, and Jiangtao Gong. 2018. SurrealDriver: Designing Generative Driver Agent Simulation Framework in Urban Contexts based on Large Language Model. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 12 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Since pioneering work led by E. Dickmann [9] in the 1980s laid the foundation for the development of autonomous driving vehicles, autonomous driving technology has witnessed vigorous development in both industry and academia. However, autonomous driving is a matter of human life safety, and its safety and reliability still require extensive testing and validation. A report by the RAND Corporation [16] indicates that relying solely on real-world driving tests would necessitate hundreds of millions or even billions of miles, which could span tens or hundreds of years to demonstrate safety against accidents and injuries. This underscores the need to push for the development of highly realistic simulation platforms as an alternative approach in autonomous driving technology research and development [17]. Currently, there is also a substantial body of research [10, 24] focused on enhancing the fidelity of simulations to improve the efficiency of transferring research outcomes from simulation to reality.

Compared to unmanned mining sites and highways, urban driving scenarios are relatively complex [19, 33]. Apart from the more intricate driving conditions and traffic rules, what is even more crucial is the presence of numerous other human traffic participants. Currently, most of these human traffic participants are constructed using rule-based [1, 17, 40] or data-driven [5, 40] approaches. Rule-based agent behavior models tend to be inconsistent with human behavior [40], lack of Realism [1], and lack of dynamic interactivity [17], making it challenging to simulate the authentic reactions of humans in traffic scenarios. Data-driven, such as imitation learning-based agents rely on extensive and costly demonstration data, and their models often produce dataset-bias, overfitting, unexplainable and uncontrollable behavior [5, 40]. These issues significantly impact the simulation of human traffic participants on simulation platforms, particularly the simulation of behaviors of other human-like driving vehicles.

Recently, with the remarkable advancements of large language models (LLMs) are known for their zero-shot prompting and complex reasoning capabilities, the concept of building agents with an LLM as its core controller has gained significant attention¹. This is primarily achieved through the construction of key modules such as planning, memory, tools, actions, etc., which collectively form the brain of the agent. These frameworks have been applied in multiple domains, including interactive natural language processing [20], intelligent robots [11], social computing systems [27], virtual task [35] and more. There have also been attempts to build human-like driver agents based on LLMs in simpler scenarios like highways [13]. These efforts have showcased the potential of LLM-based agents. However, to the best of our knowledge, no one has applied LLMs to construct an agent capable of driving in urban environments just like a human.

In this paper, we introduce "SurrealDriver," an LLM-based framework that enables the safe training of human-level autonomous driving agents within high-fidelity simulation platforms. Given the complexity of driving tasks in urban scenarios and the requirements for safety and human-likeness in building driving agents within simulation platforms, we have proposed four specific design guidelines. Additionally, we have developed modular components, including atomic scene understanding and action, safety criteria, short-term operational memory, and long-term driving guidelines to address these needs. Notably, we obtained input from 24 real-world drivers regarding their considerations in driving behavior, which served as prompts for constructing the CoachAgent. The CoachAgent assists the DriverAgent in acquiring long-term driving guidelines aligning with those of human drivers. We evaluated our framework by both algorithm experiments and human evaluation. The results validated the efficiency of our framework design.

Therefore, the contributions of this paper are as follows: i) We have constructed the first LLM-based driver agent designed for urban scenarios, capable of performing complex human-like driving tasks, such as autonomous lane changing and overtaking, within high-fidelity simulation environments; ii) We have designed and implemented guidelines for the development of driver agents in urban scenarios; iii) We have empirically validated the effectiveness of our design guidelines through algorithmic ablation experiments; iv) We

¹<https://lilianweng.github.io/posts/2023-06-23-agent/>

have conducted user experiments to confirm the human-likeness of the driver agent; v) We have engaged in an in-depth discussion of the potential and limitations of constructing LLM-based driver agents, offering valuable insights for the design of future LLM-based agents for complex perception-action tasks. These contributions collectively advance the field of LLM-based agent development for complex tasks.

2 RELATED WORK

2.1 Driver Modelling

In the quest for designing robust and realistic autonomous driving systems, driver modeling serves as an essential cornerstone. While rule-based models have been traditionally employed for their deterministic nature and computational efficiency, they face challenges in emulating human-like driving behavior, particularly in complex scenarios [34, 40, 42]. These rule-based models also often lack the nuance required for realistic traffic simulations [1]. Moreover, in prominent simulators like CARLA, the behavior of vehicles is often hardcoded, leading to a static and less interactive simulation environment [17]. On the other side of the spectrum, data-driven models, particularly those anchored in Reinforcement Learning and Imitation Learning, present their own sets of limitations. Reinforcement learning models, for instance, have been shown to be less reliable in scenarios requiring collision avoidance [18]. Imitation learning also grapples with issues such as dataset bias, overfitting, and an absence of causal modeling, making it less than ideal for fully autonomous applications [5]. Both the rule-based and data-driven approaches have offered valuable contributions to the field but come with notable constraints in mimicking realistic, dynamic, and human-like driving behaviors [2, 8].

These observations have steered research toward exploring alternative paradigms like generation-based methods for driver modeling, which we posit could offer more adaptive and robust solutions.

2.2 LLM-based Agent Framework

The advent of large-scale language models (LLMs) like GPT-4 has revolutionized the design and evaluation of generative agents. These models, renowned for their zero-shot prompting and advanced reasoning abilities, have significantly advanced agent research across various domains, including natural language processing, robotics, social computing systems, and more. Recent developments can be categorized as follows:

Generative Agents: Researchers have harnessed LLMs to create generative agents that possess memory, retrieval, reflection, and interaction capabilities. For instance, Generative Agents [27] utilize LLMs to shape agents' personalities, store experiences, and mimic human learning. CAMEL [20] focuses on cooperative agent frameworks through role-playing, fostering collaboration among agents. Social Simulacra [28] aims to generate realistic social interactions for addressing challenges in social computing systems. FurChat [4] employs LLMs to create expressive conversational agents. CGMI [15] simulates human-computer interactions, enabling communication and knowledge exchange. Zhang et al. [43] construct agents proficient in cooperative reasoning and proactive collaboration. AgentVerse [3] explores multi-agent collaboration, demonstrating collective capabilities.

Human-Like Thinking Agents: LLMs are utilized to create agents that think in a human-like manner. SWIFTSAGE [22] combines behavior cloning and LLMs to shape agents with two thinking modes: SWIFT and SAGE. VOYAGER agents [35] autonomously explore, acquire skills, and engage in discovery. Croissant et al. [6] use LLMs to simulate human emotions effectively. TradingGPT [21] features layered memory for simulating human memory patterns. Expel agent [44] learns from experience, extracts knowledge, and improves competence over time.

Tool Agents: LLMs are integrated with existing technologies to enhance their capabilities. For instance, Wang et al. [37] address interactive NLP fundamentals, CHATDB [14] combines LLMs with SQL databases, Style2Fab [12] uses generative AI for personalized 3D model creation, Wei et al. [39] simulate automatic graph learning, and Da et al. [7] analyze traffic dynamics and environmental conditions for traffic signal control.

Incorporating LLMs into driver modeling offers the potential to create driving agents capable of navigating complex urban conditions. These agents bring advantages like adaptability, interpretability, zero-shot learning, common sense reasoning, and the ability to handle diverse situations. Besides, we also can see that making agents have the capability of complex tasks needs a well-designed framework.

2.3 Prompt Engineering for Agents

As LLM advancements continue, researchers have explored prompt techniques for human-like agent behavior. Wang et al. [37] introduced In-context learning (ICL), enabling machines to transition from specific tasks to more human-like problem-solving. Min et al. [25] emphasized label and input space importance for ICL. Wei et al. [38] introduced Chain-of-thought prompts, simulating human thought processes. Few-shot and zero-shot prompts have also been tested. Liu et al. [23] found few-shot prompts reduce costs and perform well, while Rubin et al. [31] discussed challenges. Romera et al. [30] focused on zero-shot prompts.

Other prompts explored include ReAct by Yao et al. [41], which introduces bidirectional reasoning-action steps, and Reflection by Shinn et al. [32], enhancing learning from past mistakes. Wang [36] proposed Instruction prompts, categorizing instructions based on example seeds.

Thus, we can see that prompting effectively is critical for building agents. In our research, we collected original interview data from real drivers as few-shot, chain-of-thought prompts, and utility reflection prompts for building driver agents.

3 SURREALDRIVER FRAMEWORK

3.1 Motivation and Design Considerations

Autonomous agents designed based on large language models have the capability to assist humans in performing relatively complex tasks rather than simply executing commands. With the recent advancements in large language models [20, 35], these agents may also simulate and interact with humans in a manner that closely resembles human behavior within simulated environments. This emerging field holds significant potential for creating intelligent and adaptable agents capable of assisting humans across various domains and tasks.

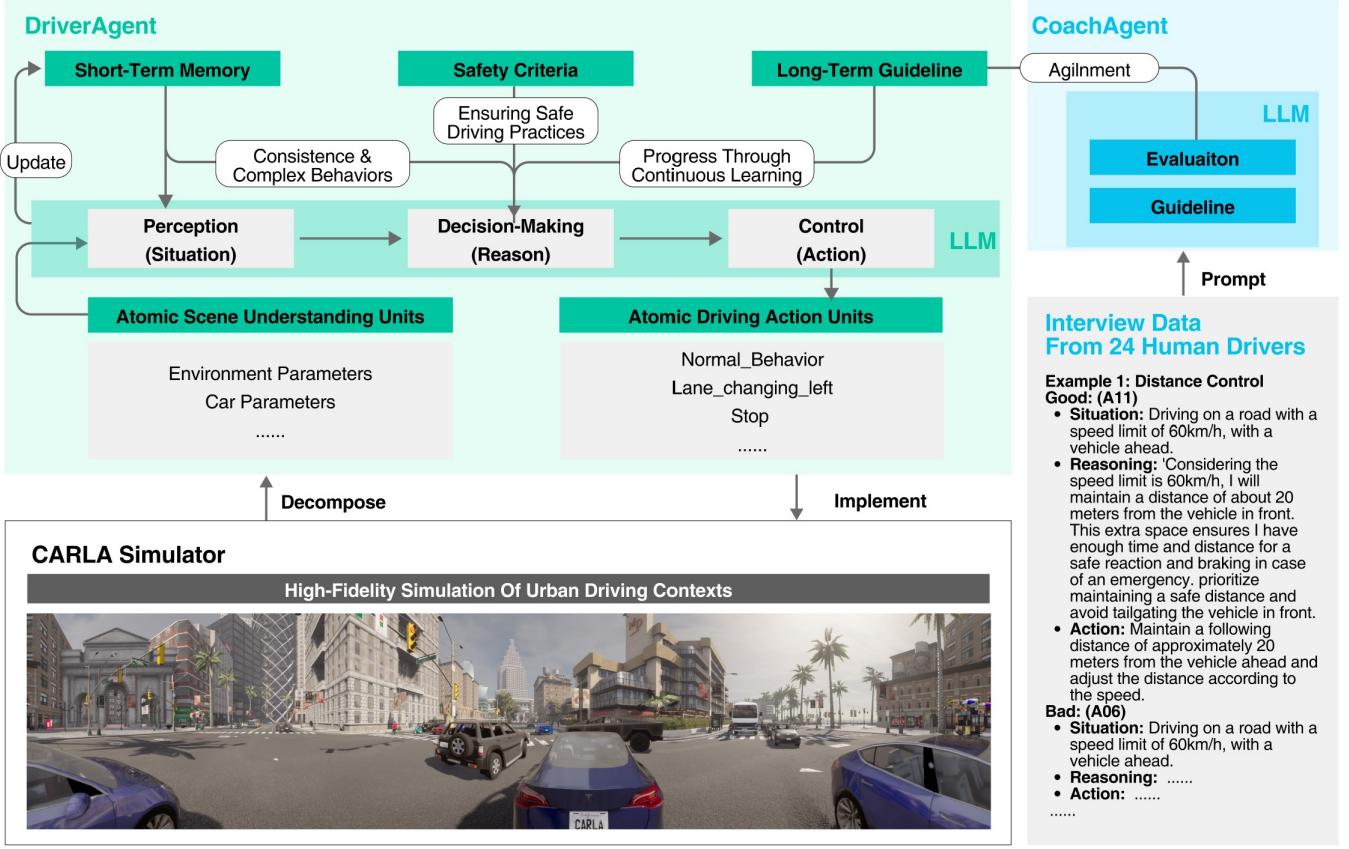


Figure 2: The framework of SurrealDriver.

However, driving behavior in real urban environments is exceptionally complex [19, 33]. Firstly, urban road scenarios are intricate, including features like intersections, roundabouts, parking lots, winding roads, and highways. Secondly, urban roadways require all vehicles to adhere to traffic laws and road regulations. Vehicles must stay in the correct lanes, maintain safe speeds within prescribed limits or below, yield to other vehicles at intersections, overtake safely when necessary, identify congestion, and execute U-turns when needed. Additionally, drivers must engage in social interactions and negotiations with unpredictable traffic participants on urban roads, such as other human drivers, pedestrians, and non-motorized vehicles. This complexity underscores the significant challenges involved in simulating and developing autonomous driving systems for urban environments.

Therefore, designing an agent capable of driving requires it to comprehend the complexity and diversity of driving environments, execute a continuous series of intricate operations, ensure safety, and harmonize with other human-driven vehicles. Based on these considerations, we have established the following design considerations for this framework:

3.1.1 Perception: Atomic Scene and Atomic Actions: Human driving scenarios encompass vast diversity, making it essential for the agent to comprehend intricate scenarios on an atomic level. Conventional

approaches to simulating driving behavior involve training with a broad spectrum of driving scenarios, incurring substantial costs.

To enable LLMs to grasp diverse elements within driving scenarios and understand complex situations, we disassemble driving scenarios into discrete parameters within our framework. These parameters are furnished to the agent, facilitating its assessment of the current situation based on common sense. Furthermore, we deconstruct driving actions within the simulator into elementary operations, empowering the agent to amalgamate these operations for executing intricate driving behaviors.

3.1.2 Execution: Short-Term Driving Memory: Effective car driving demands seamless and continuous actions, minimizing abrupt braking or sharp turns whenever feasible. Additionally, actions such as overtaking and following entail a fusion of fundamental maneuvers (e.g., acceleration, lane changing), rendering driving actions relatively intricate.

To maintain smooth driving, we capture the agent's recent driving behavior over a few steps in the Short-term driving memory module. These short-term driving memories aid the agent in sustaining consistency in decision-making. Moreover, the agent can employ these driving memories to amalgamate several basic driving operations for executing complex driving behaviors.

3.1.3 Planning: Long-Term Human-like Driving Guidelines: The agent must align its planning with human drivers. This module facilitates the agent in emulating the process by which humans learn from experienced drivers to amass expertise and continually enhance their driving skills.

To this end, we design CoachAgent to assess the DriverAgent's driving behaviors and impart guidelines that must be adhered to. These guidelines are consistently integrated, contributing to the ongoing enhancement of the DriverAgent's driving proficiency.

3.1.4 Overall Process: Strict Safety Criteria: Ensuring safety is the most critical requirement for driving behavior simulation. Any simulated driving system must prioritize safety and establish rules within its framework to ensure the agent's safety.

Thus, throughout the entire driving process, safety should be consistently ensured through safety redundancy mechanisms. The agent is provided with stringent safety criteria to ensure the fundamental safety of the driving process.

Based on the Design Considerations mentioned above, we have designed the framework (see Fig. ??) for SurrealDriver, allowing it to address the challenges associated with diverse scenarios, consistent and complex driving actions, safety, and human alignment.

3.2 Memory and Safety Mechanisms Design in DriverAgent

DriverAgent is the core component SurrealDriver framework. We built memory and safety mechanisms based on the basic driving pipeline in the CARLA simulator.

3.2.1 Basic Driving Pipeline. The basic driving pipeline (see Fig. 3) consists of three main processes: perception, decision-making, and control.

In the perception part, DriverAgent receives information from the CARLA simulator regarding the vehicle's own state and the surrounding environment. We break down this information into atomic modules and provide it as input parameters to the DriverAgent, allowing it to integrate this data. These pieces of information are provided to the DriverAgent in the form of parameters. We have already provided specific descriptions of the meanings of these parameters in the agent's system prompt to enable it to understand their significance. Upon receiving these parameters, the DriverAgent analyzes and integrates them based on common sense and the provided prompt, comprehending the current situation of the vehicle.

Once the perception phase completes the understanding of the situation, DriverAgent will make decisions for the next steps based on this situation. The decision-making criteria include the initial requirements provided to it, which are to ensure safety and efficient driving.

After the decision-making process is complete, the control phase is initiated. In this phase, DriverAgent will generate a JSON-formatted command to be sent to the CARLA program. It will select an appropriate action from a predefined set of actions based on the current scenario. The available actions it can choose from include the following: stop, normal_behavior, maintain_speed, lane_changing_left, lane_changing_right, speed_up, speed_down. These actions are

atomic driving operations, and DriverAgent can combine these actions to generate coherent and complex maneuvers.

3.2.2 Memory and Safety Mechanisms. The memory and safety mechanisms are built on top of the basic driving pipeline to store the information needed by the DriverAgent. It consists of three modules: Safety criteria, Short-term memory, and Long-term guidelines.

Safety Criteria: We have designed a strict set of safety criteria to ensure that the vehicle does not engage in dangerous operations. The safety redundancy mechanism comprises two priority levels. The first level, designated as mandatory, includes rules such as stopping the car irrespective of its speed when the distance to another vehicle or walker falls below 10 meters and halting if the traffic light signal is red. The second level, optional but advisable, involves actions like decelerating when the agent car approaches within 20 meters of a vehicle or pedestrian, slowing down upon detecting an intersection ahead, maintaining a minimum 1-meter distance from moving cars, and optimizing energy consumption by minimizing unnecessary acceleration and deceleration.

Short-term Memory: As shown in Fig. 4, DriverAgent will simultaneously output the aforementioned drivings, including situation, reasoning, and action. To ensure the continuity and complexity of driving, we will store the driving behaviors of the current agent from the past few iterations and continuously update it: replacing the oldest with the latest to maintain a certain number of stored behaviors. These behaviors will then be provided to the DriverAgent again, becoming part of its perception.

Long-term Driving Guideline: Evaluating and reflecting on the driving behavior generated by the DriverAgent and formulating them into long-term driving guidelines is a crucial step in shaping a unique driving style. Assessing the quality of the DriverAgent's driving behavior is offered by CoachAgent based on interview data of its driving behavior provided by professional and regular drivers from real-world experiments. Detailed information can be seen in section 3.3.

3.3 CoachAgent Based on Human Drivers' Interview

To better align SurrealDriver with human drivers, we collected additional real drivers' driving behavior text data as a few-shot prompt. This data served as a standard to assess the quality of initial actions and as guidelines to improve driving performance. Utilizing few-shot prompts reduces the input, memory, and storage consumption significantly while enhancing the agent's comprehension of good and bad driving behaviors.

We invited 24 drivers (10 expert drivers and 14 novice drivers, demographics in the appendix) to participate in in-depth interviews regarding driving behavior. To delve deeply into specific driving behaviors, we initially had each driver perform an urban road driving task covering 13 driving conditions, with a total length of 5.7 kilometers. Subsequently, we conducted in-depth post-task interviews (interview guidelines provided in the appendix). The entire interviews were recorded, with each interview lasting approximately between 1.5 to 2 hours. Our team of four researchers conducted data cleaning and open coding, selecting potential prompt materials from these interviews. These materials were then transformed into prompt templates for the coach agent.

Driving Framework

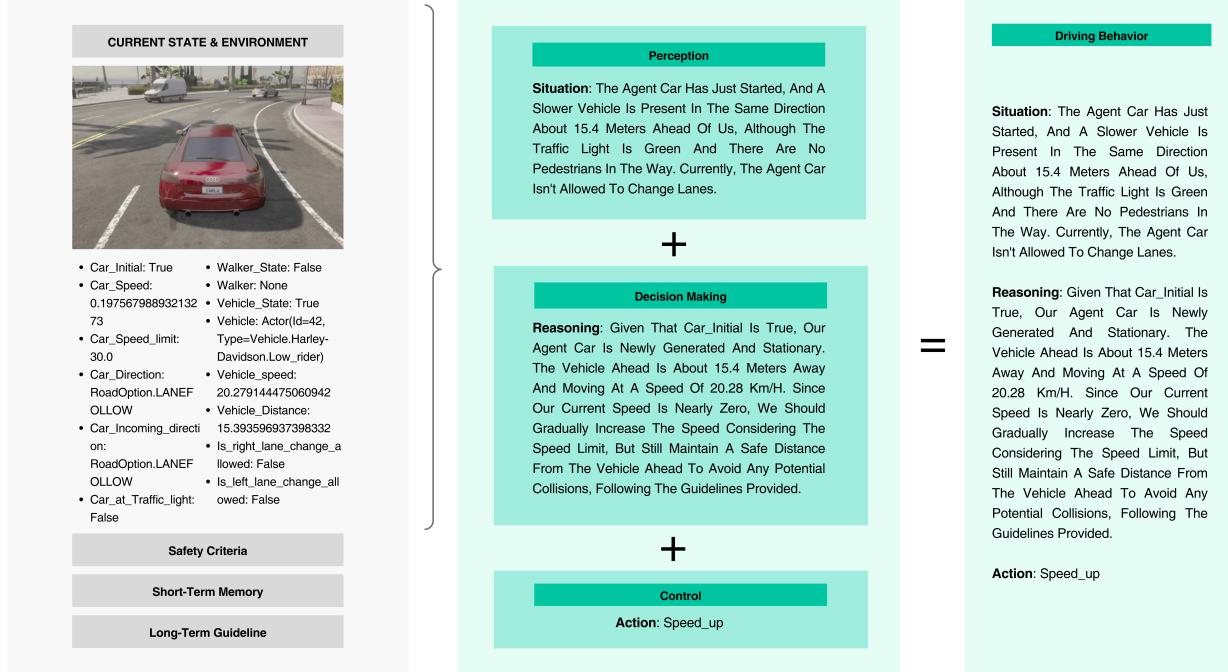


Figure 3: The Details of DriverAgent.

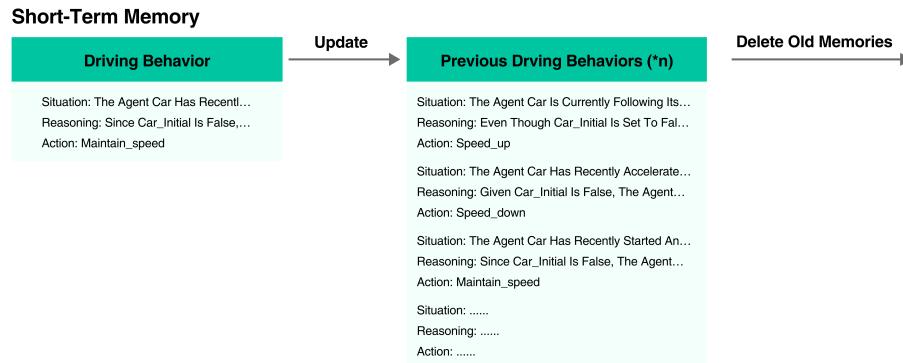


Figure 4: The short-term memory module.

This study was approved by the Institutional Review Board of the authors' institution. Before the experiment, all participants were ensured with informed consent, acknowledging potential risks and their right to discontinue the study. To preserve participant confidentiality, all personal and confidential information has been anonymized, and the research results presented below have been subjected to de-identification.

After cleaning the interview data, we identified differences in behavior between expert and novice drivers. For example, we observed that expert drivers have specific methods and standards for controlling the following distance, which they adjust based on real-time conditions:

If the speed limit is, say, 40, then I might maintain a distance of around 10 meters, which is more than two car lengths. For 60 mph, if the road isn't too crowded, I'll extend it further, maybe around 20 meters. At 60 mph,

I might keep a distance of about 20 meters because you need to allow for sufficient reaction time and braking distance. I won't tailgate the car in front; that situation never occurs. When driving, you need to stay highly focused and vigilant; you have to be ready, in case the car in front suddenly brakes. If you're too close, you won't react in time.

To improve SurrealDriver's performance, we can adopt the practices of expert drivers, enabling the agent to assess the following distance based on road conditions. In our CoachAgent's prompt, we introduced "Criteria for Evaluation and Guidelines" and provided three instances of different operations by expert and novice drivers in various scenarios. We incorporated expert drivers' driving behaviors as "good examples" and novice drivers' driving behaviors as "bad examples."

While designing examples, we followed a three-dimensional approach: situation, reasoning, and action (see Fig. 5). Situation provided specific road conditions during driver operations, and for each comparison case, we set the same road conditions, referencing the road conditions real drivers faced during their interviews. Reasoning was designed based on the content of driver interviews, with irrelevant information removed to make our examples concise and efficient in demonstrating human thinking, guiding the agent to learn human thought patterns.

4 EVALUATION

4.1 Experiment Environment Set-up

In this study, the experimental setup involved a ThundeRobot Zero desktop computer as the hardware foundation. The simulation environment was built upon the CARLA simulator, specifically, version 0.9.14 [10] and operated on Python 3.7 with Unreal Engine 4 [29]. The simulated environment was chosen to be Town10, and the Audi TT was the designated vehicle for all experiments, with fixed starting and Continuously, randomly generated ending points for its path (After a vehicle is generated at a predefined fixed point, a random endpoint is generated. Upon reaching the endpoint, another endpoint is randomly generated, and so on. This process continues until the required number of driving rounds is completed.). We leverage OpenAI's GPT-4 [26] APIs for simulating drivers' driving decisions and solving related problems in a simulated environment. However, it takes several seconds for GPT (whether 3.5 or 4) to make a decision, which is too long in a driving context for making immediate decisions. Therefore, we slowed down CARLA's simulation time based on the required token count by setting a fixed time-step of 0.0006–0.0015 seconds.

4.2 Experiment Details

We conducted driving experiments using agents from different frameworks in the same scenario, analyzing variations in their behaviors to understand how directives from different frameworks influence their driving. We evaluated the agents based on two primary dimensions: safety driving capability and human-likeness. Safety driving capability was assessed using collision rate, while human-likeness was evaluated through a user experiment. with 24 adult participants (aged 29.3 ± 4.9 , male = 17) with legal driving licenses. The participants first watched the video record of our

Table 1: Crash Rating of Ablation Experiment

Framework	Collision Rate by Distance (per meter)	Collision Rate by Time (per second)
w/o safety criteria, w/o short-term memory, w/o long-term guidelines	0.01453958	0.041315485
with safety criteria, w/o short-term memory, w/o long-term guidelines	0.00923361	0.02366976
with safety criteria, with short-term memory, w/o long-term guidelines	0.005046864	0.009530682
Full framework	0.002757353	0.005100011

experiments and then filled out a questionnaire to rate the human-likeness with a five-point Likert scale.

4.3 Results

Before this paper submission, the overall experiment lasted 108405.90s (30.11 hours), average experiment time for each condition was 7079.67s, 13730.6s, 23870.28s, and 63725.35s, respectively. We conducted statistical analyses separately for collision rates per unit distance and collision rates per unit time. The detailed results are shown in Table 1.

4.3.1 Safety Criteria plays a crucial role in ensuring safety. Collision rate data shows that the framework with safety module has a collision rate 57.46% lower than the one without it.

In the absence of Safety Criteria, when the vehicle was at a distance of 5 meters from the preceding vehicle, the DriverAgent initiated a lane change, leading to a collision with the front vehicle. However, When running a framework with Safety Criteria, the vehicle encountered a situation where the distance to the preceding vehicle was 7 meters. Based on the information provided by the safety criteria, it initiated a stop behavior, safely coming to a halt behind the lead vehicle (see Fig. 7).

It is evident that safety criteria play a significant role in ensuring fundamental safety driving behaviors of the vehicle. Moreover, such criteria are described in natural language, highlighting the convenience of incorporating rules into the LLM-Based Driver Agent.

4.3.2 Short-term Memory is essential for ensuring driving continuity and performing complex operations. Collision rate data shows that the framework with Short-term Memory has a collision rate 82.96% lower than the one without it.

We define continuity as DriverAgent encountering several different options when making decisions, where the previous actions influence its next move, resulting in more reasonable and smooth driving behavior.

In the experiment, the vehicle initially accelerated for a few steps, and when DriverAgent had to decide its next action, it had two options: to continue accelerating or to maintain its current speed. Considering its previous acceleration actions, it chose to maintain

Long-Term Guideline

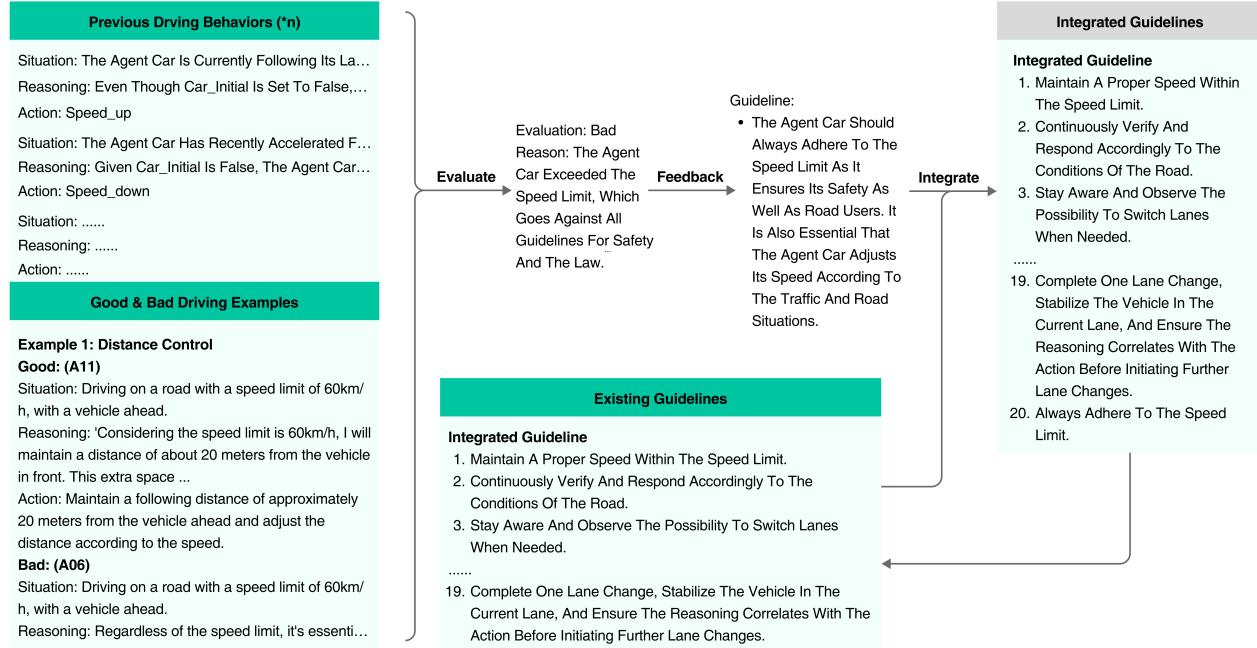


Figure 5: The CoachAgent for human alignment.



Figure 6: The examples followed a three-dimensional approach.

its current speed. The reason is: "It can continue accelerating to

reach closer to the speed limit. However, since speeding up had been the action for the past multiple sequences, it would be reasonable to maintain the car's current speed for a while to allow the driver to adjust to the new speed and thus ensure safety standards."

For complex operations, in this context, we define them as those that can combine two or more basic actions to accomplish more intricate driving tasks. In our experiment, we observed that when the DriverAgent encountered a slower vehicle in front of it with an open adjacent lane, it made the decision to perform an overtaking maneuver. Subsequently, it initiated a lane change, followed by the realization that the ultimate goal of this maneuver was overtaking. Therefore, it executed an acceleration action, successfully combining lane change and acceleration to complete the overtaking task. This achievement would not have been possible in a framework without short-term memory (see Fig. 8).

4.3.3 Long-term Guidelines. Collision rate data shows that the framework with Long-term Guidelines has a collision rate 83.03% lower than the one without them.

With long-term guidelines, the DriverAgent demonstrated an improvement in driving skills. In one experiment, the CoachAgent analyzed the initial driving behaviors and classified them as 'Bad.' The reason for this assessment was the excessive frequency of stopping.

After some time, when faced with a similar situation, it followed this guideline and performed a lane change:

- "Situation": "The agent car is on the right lane following a slower vehicle approximately 5 meters ahead. Lane change



Figure 7: Performance differences between frameworks with and without safety criteria.

- to the left is restricted but the right lane is allowable. There is no traffic light or pedestrian imposing restrictions".
- "Reasoning": "Given Car_Initial is False, the agent car isn't newly generated. It is driving behind a slow vehicle at a potentially hazardous distance. Considering the clear road and the right lane's availability, the car should change to this lane to safely distance itself from the slow vehicle".
 - Action: "lane_changing_right"

4.3.4 Human-likeness: Full Architecture exhibits the highest human-likeness. Through user experiments, it was found that an agent framework with all the mentioned modules achieved a 50% significantly higher human-likeness score compared to the basic framework without long-term guidelines, short-term memory, and safety criteria ($F = 4.353$, $p = 0.01$, in repeated-measurement ANOVA with LSD post-hoc test, $p = 0.009$).

5 DISCUSSTIONS

5.1 SurrealDriver Framework

The experimental data clearly indicate that the Full framework outperforms frameworks lacking Safety Criteria, short-term memory, or long-term guidelines in terms of safety, driving continuity and complexity, human-likeness, and completion of driving tasks. This underscores the significance of these modules in driver simulation. Safety Criteria are essential to ensure basic driving safety, while

short-term memory is necessary for maintaining driving continuity and complexity. Long-term driving guidelines provided by the CoachAgent are also indispensable for continuous improvement in driving proficiency.

This LLM-based driver agent framework offers a new approach for future driver simulations, bringing driver agent behavior closer to human-like driving and, consequently, simulating more realistic traffic environments.

5.2 Design Implications for Driver Agent Based on LLM

5.2.1 Easier to Incorporate Rules: Natural Languages. Data-based driver agents have poor interpretability, making it challenging to incorporate rules. On the other hand, rule-based agents have strong interpretability but limited adaptability to diverse scenarios. Driver agents based on large language models can utilize natural language and examples to add rules more conveniently, allowing for easier rule adjustments.

5.2.2 Better Evaluation and Enhancement: Interview Transcripts. We provide the Agent with descriptions of real drivers' behaviors obtained through interviews conducted during real vehicle experiments. The Agent uses its capabilities based on large language models to autonomously assess the quality of its driving behavior compared to real driving descriptions. It then enhances its driving



Figure 8: (Left) Frameworks with short-term memory exhibit complex driving behaviors; (Right) DriverAgent and CoachAgent based on the full framework with GPT-4, recognizing ambulances and including them as special vehicles in the consideration.

skills based on the behavior of professional drivers. This approach differs from traditional reinforcement learning and other training methods, enabling the agent to learn directly from driver transcripts, similar to humans, without the need for translation into code.

5.2.3 Automated and Human-Like Training: Long-Term Guidelines. Long-term guidelines enable the agent to engage in continuous learning, consistently improving its driving proficiency. As the program runs more extensively, it accumulates richer guidelines, mirroring the ongoing learning process of real human drivers and allowing for continuous evolution.

5.3 Limitation and Future Works

In our experiment, the results are limited by some technical issues. Firstly, we used the CARLA simulator for our experiments. However, there are occasional difficulties in recognizing or misidentifying certain environmental parameters due to Carla's internal mechanism. For instance, sometimes it fails to detect red lights. Besides, due to the time it takes for GPT models to return results, we had to slow down the world time in the driving simulator to

compensate for the delay caused by GPT responses. This inconvenience in using the agent in the driving simulator results in longer testing times. One possible solution is to limit the number of tokens returned by GPT, as the delay primarily depends on the number of tokens returned. The rate limit of GPT models, especially GPT-4, imposes restrictions on the agent's capabilities. When using the GPT-4 model, we frequently encounter issues where the rate limit is reached, preventing further usage, especially when input and output increase.

6 CONCLUSIONS

In this work, we proposed SurrealDriver, which can train an LLM-based driver agent to simulate human driving behavior. It can understand the vehicle's situation, make decisions based on this situation, perform coherent, complex, and safe operations, and improve its driving capabilities based on human driver experience. Building upon the existing LLM-Based Agent framework, we have added memory modules tailored for automotive driving scenarios: short-term memory, long-term guidelines, and safety criteria. Through

ablation experiments, we have validated the effectiveness of these modules. Short-term memory enables the agent to make more coherent and complex maneuvers, long-term memory enhances the agent's driving skills, and safety criteria ensure a minimum level of safety. We suggest that our SurrealDriver Framework can provide valuable insights for building a more realistic traffic environment for driving simulators through more authentic driving behaviors.

ACKNOWLEDGMENTS

Our sincere thanks to the participants in our work.

REFERENCES

- [1] Raunak Bhattacharyya, Soyeon Jung, Liam A Kruse, Ransalu Senanayake, and Mykel J Kochenderfer. 2021. A hybrid rule-based and data-driven approach to driver modeling through particle filtering. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021), 13055–13068.
- [2] Hua Cai, Yingzi Lin, and Bo Cheng. 2010. Mimicking human driving behaviour for realistic simulation of traffic flow. *International Journal of Simulation and Process Modelling* 6, 2 (2010), 126–136.
- [3] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors in Agents. arXiv:2308.10848 [cs.CL]
- [4] Neeraj Cherakara, Finny Varghese, Sheena Shabana, Nivan Nelson, Abhiram Karukail, Rohith Kulothungan, Mohammed Afil Farhan, Birthe Nesset, Meriam Moujahid, Tanvi Dinkar, Verena Riesen, and Oliver Lemon. 2023. FurChat: An Embodied Conversational Agent using LLMs, Combining Open and Closed-Domain Dialogue with Facial Expressions. arXiv:2308.15214 [cs.CL]
- [5] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. 2019. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9329–9338.
- [6] Maximilian Croissant, Madeleine Frister, Guy Schofield, and Cade McCall. 2023. An Appraisal-Based Chain-Of-Emotion Architecture for Affective Language Model Game Agents. arXiv:2309.05076 [cs.AI]
- [7] Longchao Da, Minchuan Gao, Hao Mei, and Hua Wei. 2023. LLM Powered Sim-to-real Transfer for Traffic Signal Control. arXiv:2308.14284 [cs.AI]
- [8] Atabak Dehban, Atena Sajedin, and Mohammad Bagher Menhaj. 2016. A cognitive based driver's steering behavior modeling. In *2016 4th International Conference on Control, Instrumentation, and Automation (ICCA)*. IEEE, 390–395.
- [9] Ernst Dieter Dickmanns and Volker Graefe. 1988. Dynamic monocular machine vision. *Machine vision and applications* 1, 4 (1988), 223–240.
- [10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.
- [11] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Azyaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378* (2023).
- [12] Faraz Faruqi, Ahmed Katary, Tarik Hasic, Amira Abdel-Rahman, Nayeeum Rahman, Leandra Tejedor, Mackenzie Leake, Megan Hofmann, and Stefanie Mueller. 2023. Style2Fab: Functionality-Aware Segmentation for Fabricating Personalized 3D Models with Generative AI. arXiv:2309.06379 [cs.HC]
- [13] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. 2023. Drive Like a Human: Rethinking Autonomous Driving with Large Language Models. *arXiv preprint arXiv:2307.07162* (2023).
- [14] Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. ChatDB: Augmenting LLMs with Databases as Their Symbolic Memory. arXiv:2306.03901 [cs.AI]
- [15] Shi Jinjin, Zhao Jiaobai, Wang Yilei, Wu Xingjiao, Li Jiawen, and He Liang. 2023. CGMI: Configurable General Multi-Agent Interaction Framework. arXiv:2308.12503 [cs.AI]
- [16] Nidhi Kalra and Susan M Paddock. 2016. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice* 94 (2016), 182–193.
- [17] Prabhjot Kaur, Samira Taghavi, Zhaofeng Tian, and Weisong Shi. 2021. A survey on simulators for testing self-driving cars. In *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*. IEEE, 62–70.
- [18] Jong-Hoon Kim, Jun-Ho Huh, Se-Hoon Jung, and Chun-Bo Sim. 2021. A study on an enhanced autonomous driving simulation model based on reinforcement learning using a collision prevention model. *Electronics* 10, 18 (2021), 2271.
- [19] Yoshiaki Kuwata, Justin Teo, Gaston Fiore, Sertac Karaman, Emilio Frazzoli, and Jonathan P How. 2009. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on control systems technology* 17, 5 (2009), 1105–1118.
- [20] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbulin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760* (2023).
- [21] Yang Li, Yangyang Yu, Haohang Li, Zhi Chen, and Khaldoun Khashanah. 2023. TradingGPT: Multi-Agent System with Layered Memory and Distinct Characters for Enhanced Financial Trading Performance. arXiv:2309.03736 [q-fin.PM]
- [22] Bill Yuchen Lin, Yicheng Fu, Karina Yang, Prithviraj Ammanabrolu, Faez Brahman, Shiyu Huang, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2023. Swift-Sage: A Generative Agent with Fast and Slow Thinking for Complex Interactive Tasks. arXiv:2305.17390 [cs.CL]
- [23] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems* 35 (2022), 1950–1965.
- [24] Lin Ma, Longrui Chen, Yan Zhang, Mengdi Chu, Wenjie Jiang, Jiahao Shen, Chuxuan Li, Yifeng Shi, Nairui Luo, Jirui Yuan, Yilun Chen, Guyue Zhou, and Jiangtao Gong. 2023. Evaluation of Pedestrian Safety in a High-Fidelity Simulation Environment Framework. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 901–908.
- [25] Sewon Min, Xinxix Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837* (2022).
- [26] OpenAI. 2023. GPT-4 technical report. *arXiv* (2023), 2303–08774.
- [27] Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442* (2023).
- [28] Joon Sung Park, Lindsay Popowski, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. arXiv:2208.04024 [cs.HC]
- [29] Weichao Qiu and Alan Yuille. 2016. UnreaveL: Connecting computer vision to unreal engine. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part III* 14. Springer, 909–916.
- [30] Bernardino Romera-Paredes and Philip Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning*. PMLR, 2152–2161.
- [31] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633* (2021).
- [32] Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366* (2023).
- [33] Liting Sun, Wei Zhan, Ching-Yao Chan, and Masayoshi Tomizuka. 2019. Behavior planning of autonomous cars with social perception. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 207–213.
- [34] Jorge Villagra and Antonio Artuñedo. 2023. Chapter 3 - Behavior planning. In *Decision-Making Techniques for Autonomous Vehicles*, Jorge Villagra and Felipe Jiménez (Eds.). Elsevier, 39–59. <https://doi.org/10.1016/B978-0-323-98339-6.00010-5>
- [35] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlikar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* (2023).
- [36] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560* (2022).
- [37] Zekun Wang, Ge Zhang, Kexin Yang, Ning Shi, Wangchunshu Zhou, Shaochun Hao, Guangzheng Xiong, Yizhi Li, Mong Yuan Sim, Xiuying Chen, et al. 2023. Interactive natural language processing. *arXiv preprint arXiv:2305.13246* (2023).
- [38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.
- [39] Lanning Wei, Zhiqiang He, Huan Zhao, and Quanming Yao. 2023. Unleashing the Power of Graph Learning through LLM-based Autonomous Agents. arXiv:2309.04565 [cs.LG]
- [40] Zheng Xu, Yihai Fang, Nan Zheng, and Hai L Vu. 2022. Analyzing the inconsistency in driving patterns between manual and autonomous modes under complex driving scenarios with a VR-enabled simulation platform. *Journal of Intelligent and Connected Vehicles* 5, 3 (2022), 215–234.
- [41] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).
- [42] Hongtao Yu, H Eri Tseng, and Reza Langari. 2018. A human-like game theory-based controller for automatic lane changing. *Transportation Research Part C: Emerging Technologies* 88 (2018), 140–158.
- [43] Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, Xiaojun Chang, Junge Zhang,

- Feng Yin, Yitao Liang, and Yaodong Yang. 2023. ProAgent: Building Proactive Cooperative AI with Large Language Models. arXiv:2308.11339 [cs.AI]
- [44] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2023. ExpeL: LLM Agents Are Experiential Learners.

arXiv:2308.10144 [cs.LG]

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009