Exploring Large Language Models for Trajectory Prediction: A Technical Perspective

Farzeen Munir
farzeen.munir@aalto.fi
Aalto University, Finland

Tsvetomila Mihaylova
tsvetomila.mihaylova@aalto.fi
Aalto University, Finland

Shoaib Azam
shoaib.azam@aalto.fi
Aalto University, Finland

Tomasz Piotr Kucner
tomasz.kucner@aalto.fi
Aalto University, Finland

Ville Kyrki
ville.kyrki@aalto.fi
Aalto University, Finland

ABSTRACT

Large Language Models (LLMs) have been recently proposed for trajectory prediction in autonomous driving, where they potentially can provide explainable reasoning capability about driving situations. Most studies use versions of the OpenAI GPT, while there are open-source alternatives which have not been evaluated in this context. In this report[1], we study their trajectory prediction performance as well as their ability to reason about the situation. Our results indicate that open-source alternatives are feasible for trajectory prediction. However, their ability to describe situations and reason about potential consequences of actions appears limited, and warrants future research.

CCS CONCEPTS

• **Computing methodologies** → *Motion path planning*; **Motion path planning**; • **Human-centered computing** → **Natural language interfaces**.

KEYWORDS

Autonomous Driving, Large Language Models, Trajectory Prediction

ACM Reference Format:
Farzeen Munir, Tsvetomila Mihaylova, Shoaib Azam, Tomasz Piotr Kucner, and Ville Kyrki. 2024. Exploring Large Language Models for Trajectory Prediction: A Technical Perspective. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24 Companion), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3610978.3640625

1 INTRODUCTION

Human-robot interaction (HRI) is pivotal in integrating advanced robotic technologies into our daily lives, where the explainability of these systems plays a crucial role in ensuring that interactions are intuitive, safe, and beneficial for all. This is particularly evident in autonomous vehicles, where understanding human behavior, such as predicting pedestrian movements and interpreting other drivers' actions, is fundamental. Trajectory prediction for autonomous vehicles involves encoding information gathered from the surroundings for generating safe and feasible trajectories. Rule-based methods, while offering interpretability, often struggle to handle the diversity of real-world driving scenarios. Conversely, data-driven models excel by learning from extensive human-driving behavior datasets but are criticized for their 'black box' nature, compromising their interpretability [16]. Both the rule-based and learning-based methods lack the inherent common sense reasoning of human driving, limiting their effectiveness in addressing rare and complex driving situations. This highlights the necessity for models that not only imbue common sense reasoning but also strike a balance between explainability and adaptability in trajectory prediction.

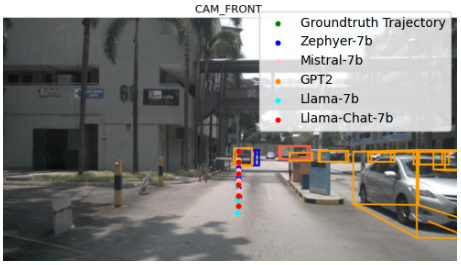

Figure 1: Qualitative results of LLMs for ego-vehicle trajectory prediction.

Recent literature shows efforts [8] to infuse human-like reasoning into autonomous vehicles, drawing inspiration from the capabilities of LLMs. One such strategy is re-imagining trajectory prediction as a language modeling problem. This method converts motion planner inputs, like detection and prediction outcomes, into unified language tokens. LLMs then process these tokens, articulating future driving trajectory waypoints as natural language descriptions and fine-tuning these models for specific tasks. Another strategy hierarchically employs LLMs within closed-loop environments, where the system generates queries influenced by current observations and past experiences. These queries then direct the decision-making process, with the system continually assessing and learning from its decisions, enhancing its ability to respond appropriately in future scenarios [2, 14].

Modeling trajectory prediction using LLMs is key in human-robot interaction for understanding and anticipating human behavior, ensuring efficient, safe, and intuitive collaboration between humans

[1]The code for the experiments is available on: https://github.com/aalto-intelligent-robotics/llm-trajectory-prediction/

HRI '24 Companion, March 11–14, 2024, Boulder, CO, USA

ACM ISBN 979-8-4007-0323-2/24/03
https://doi.org/10.1145/3610978.3640625

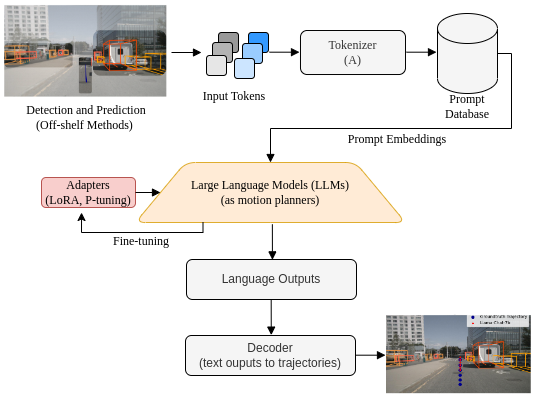

Figure 2: Illustrates the pipeline for deploying LLMs to predict the trajectory for autonomous vehicles. It includes fine-tuning the LLMs with adapters on prompt database generated by offshelf detection and prediction algorithms. The output language output is converted back to planned trajectories using decoder. Note: the decoder is a simple regressive extraction of trajectories from language outputs.

and robots. Existing LLMs methods use versions of OpenAI's GPT, for trajectory prediction. However, there are several open-source alternatives available that haven't been assessed for trajectory prediction. These alternatives, potentially offering diverse approaches and methodologies, remain unexplored and untested for their efficacy and applicability for trajectory prediction. To this end, in this report, we explore the LLMs for ego-vehicle trajectory prediction problem. We explore some open-source alternatives of OpenAI's GPT for trajectory predictions and how to finetune them by using adapters for trajectory prediction tasks. Figure 2 illustrates the framework for using the LLMs for the trajectory prediction task.

In our experimental analysis, we aim to answer the following research questions:

(1) Can open-source models that run on a single GPU achieve similar results for trajectory prediction as querying the OpenAI API?
(2) When using LLMs for trajectory prediction, can we benefit from the general knowledge acquired by the model and provide meaningful explanations of the situation for a human?

2 PROBLEM FORMULATION

Motion planning in the context of autonomous driving aims to devise a future trajectory, denoted as τ, that ensures safety and comfort. The trajectory, represented by τ, is defined as a sequence of waypoints corresponding to distinct t timestamps: $\tau \in R^{t\times 2}$, denoted as

$$\tau = \{(x_1, y_1), ..., (x_t, y_t)\}, \quad (1)$$

Here,(x_t, y_t) represents the two-dimensional waypoints corresponding to the vehicle's location at the timestamp t. The trajectory prediction inputs encompass the historical waypoints, along with the outputs from perception and prediction systems. These outputs include, for example, detected object bounding boxes and projected trajectories indicating their future movements. To conceptualize

Ground-truth assistant message:
Thoughts:
- Notable Objects from Perception: None
Potential Effects from Prediction: None
Meta Action: MOVE FORWARD WITH A DECELERATION
Trajectory: **[(0.02,2.80), (0.03,5.27), (0.06,7.46), (0.08,9.48), (0.10,11.22), (0.11,12.79)]**

GPT-2
Autonomous Driving Planner
Role: You are the brain of an autonomous vehicle.
Context
- Coordinates: X-axis is perpendicular, and Y-axis is parallel to the direction you're facing. You're at point (0,0).
- Objective: MOVE FORWARD WITH A QUICK DECELERATION
Trajectory: **[(0.02,2.78), (0.04,5.12), (0.05,7.15), (0.06,8.84), (0.07,10.25), (0.08,11.30)]**

Mistral-7B
[THOUGHTS]
Notable Objects from Perception: - car at (-3.03,-15.08) Potential Effects from Prediction: - car at (-3.03,-15.08) is predicted to be at (-3.03,10.16) in 3.0 seconds [/THOUGHTS]
[META ACTION]
Develop Meta Action based on thoughts: MOVE FORWARD WITH A CONSTANT SPEED [/META ACTION]
[TRAJECTORY]
Develop a safe and feasible 3-second Trajectory: **[(0.01,2.89), (0.03,5.72), (0.05,8.50), (0.08,11.22), (0.11,13.90), (0.14,16.51)]**
[/TRAJECTORY]

Zephyr-7B
Thoughts:
- Notable Objects from Perception: car at (-3.03,-15.08)
Potential Effects from Prediction: within the safe zone of the ego-vehicle at the 2.0-second timestep
Meta Action: MOVE FORWARD WITH A DECELERATION
Trajectory: **[(0.01,2.92), (0.03,5.65), (0.05,8.19), (0.07,10.51), (0.09,12.65), (0.11,14.57)]**
Thoughts:
- Notable Objects from Perception: car at (-6.06,14.94)
Potential Effects from Prediction: within the safe zone of the ego-vehicle at the 0.5-second timestep
- Notable Objects from Perception: car at (-6.25,8.38)
Potential Effects from Prediction: within the safe zone of the ego-veh

Llama-7B
Thoughts:
- Notable Objects from Perception: None
Potential Effects from Prediction: None
Meta Action: MOVE FORWARD WITH A DECELERATION
Trajectory: **[(0.03,2.83), (0.05,5.43), (0.07,7.85), (0.09,10.16), (0.12,12.39), (0.14,14.49)]**

Llama-7B-Chat
Thoughts: - Notable Objects from Perception: None
Potential Effects from Prediction: None
Meta Action: MOVE FORWARD WITH A CONSTANT SPEED
Trajectory: **[(0.03,2.89), (0.04,5.62), (0.06,8.20), (0.09,10.80), (0.13,13.34), (0.17,16.04)]**

Figure 3: Example for assistant message output from all tested models, fine-tuned with Low-Rank Adaptation (LoRA).

trajectory prediction as a problem within the domain of large language modeling, the trajectory τ can be represented as a sequence of words that concisely describe it.

$$\tau = A(\{(x_1, y_1), ..., (x_t, y_t)\}) = \{w_1, w_2, ..., w_n\}, \quad (2)$$

Here, w_i represents the i-th word in the sequence, obtained through the application of a large language tokenizer, represented by A. By adopting this linguistic representation, the trajectory prediction problem can subsequently be reformulated as a language modeling problem:

$$\mathcal{L}_{LLM} = -\sum_{i=1}^{N} logP(\hat{w}_i | w_1, w_2, .., w_{i-1}) \quad (3)$$

Here, $\hat{w}$ and w correspond to the words from the predicted trajectory $\hat{\tau}$ for ego-vehicle and the human driving trajectory τ, respectively. LLMs can effectively generate trajectories by maximizing the probability P associated with the occurrence of words w derived from the human driving trajectory τ.

3 EXPERIMENTS

The application of zero-shot prompting in LLMs for trajectory prediction yields sub-optimal outcomes [9, 13]. To address this, our experiments involve fine-tuning LLMs specifically for the downstream task of trajectory prediction. However, fine-tuning is computationally intensive and time-consuming because of the large model size. A more efficient strategy involves the use of adapters to train the model on domain-specific data while maintaining the LLMs in a frozen state, which represents an advantageous design choice adopted in this study.

3.1 Experimental Setup

We conduct Parameter-Efficient Fine-Tuning (PEFT) [7] using a combination of five models and two adapters. Our implementation incorporates the following open-source models from the HuggingFace Transformers library [15]. We chose OpenAI's GPT-2, as well as four recently proposed models with 7B parameters that can be trained on a single GPU.

- GPT-2[2] [10]
- Llama-7B[3] [11]
- Llama-7B-Chat[4] [11]
- Zephyr-7B[5] [12]
- Mistral-7B[6] [4]

In our experiments with adapters, we explored the use of LoRA [3] and Prompt Tuning (P-tuning) [5]. Training was conducted on the training split, while evaluation was carried out on the validation split of the dataset from GPT-Driver [8], which is derived from the nuScenes dataset [1]. The training split comprises 23,388 instances, and the validation split includes 5,119 instances. All training processes were executed on a system equipped with a single RTX 3080Ti GPU, boasting 16GB of memory. The input for all the models is the prompt used by GPT-Driver [8]: a system message provides context for the driving task, and a user message describes the observations and ego-states specific to each instance.

$L2$ metric is opted as an evaluation metric. The average L2 error is determined by calculating the distance between corresponding waypoints in the predicted and ground-truth trajectories. This metric effectively reflects the extent to which a predicted trajectory approximates a human-driving trajectory. The input prompt requests the generation of a waypoint (x, y) each 0.5 seconds. Therefore, we evaluate $L2$ for 2, 4, 6 waypoints from the predicted trajectory for the 1, 2, and 3 seconds measures.

4 RESULTS

4.1 PEFT with LoRA

For fine-tuning with LoRA, for both Llama-7B and Llama-7B-Chat models, satisfactory results were observed after just three training epochs. However, extending the training duration led to a decrease in the quality of results produced by the Llama-7B model due to overfitting. The outcomes of fine-tuning with LoRA are detailed in Table 1. Among the models tested, the application of LoRA fine-tuning techniques yielded the most accurate results for the Llama-7B and Llama-7B-Chat models. These two models outperform the results reported in [8], based on the L2 metric. The other three models also achieve good results for the L2 metric for the predicted trajectories. The results for GPT-Driver are the ones reported in their paper [8].

Table 1: Results from PEFT fine-tuning with LoRA.

Model	L2			Average	Empty traj.
	1s	2s	3s		
GPT-Driver [8]	0.21	0.43	0.79	0.48	-
GPT-2	0.19	0.40	0.73	0.44	224
Mistral-7B	0.31	0.59	0.97	0.62	2185
Zephyr-7B	0.27	0.60	1.07	0.65	1545
Llama-7B	**0.17**	**0.37**	0.70	**0.41**	53
Llama-7B-Chat	**0.17**	**0.37**	**0.69**	**0.41**	38

In Figure 3, we provide examples of the model output for all five models, fine-tuned with LoRA. Although the predicted trajectories are not too far from the ground-truth, the reasoning about the situation and understanding of the environment are not always consistent. We also experimented with asking follow-up questions to the model, such as *"Are there any other vehicles on the road?"* or *"What would be the effects of a different meta action?"*, but the responses did not contain precise information.

Figure 1 shows an example scene from the nuScenes dataset [1] with plotted the ground-truth trajectory and the trajectories predicted from our fine-tuned models, as well as the bounding boxes of the detected vehicles for this scene.

4.2 PEFT with P-tuning

In the context of P-tuning, the results for most models and data instances did not yield meaningful outputs. These were characterized

[2] https://huggingface.co/gpt2
[3] https://huggingface.co/meta-llama/Llama-2-7b-hf
[4] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[5] https://huggingface.co/HuggingFaceH4/zephyr-7b-beta
[6] https://huggingface.co/mistralai/Mistral-7B-v0.1

either by text in an inconsistent format, or by formats that were nearly correct but lacked numerical values for the trajectory, as illustrated in Figure 4(b) and Figure 4(e). The findings for Llama-7B and Llama-7B-Chat, when fine-tuned with P-tuning, are presented in Table 2. Notably, a high incidence of empty trajectories resulted in subpar performance in the L2 metric. Outputs from the other three models were not meaningful, and hence their results are not included. A single-word alteration in the prompt can dramatically affect the model's performance [6]. P-tuning incorporates trainable continuous prompt embeddings with discrete prompts. With a given discrete prompt, P-tuning appends continuous prompt embeddings to the discrete tokens, feeding them into the language model. The P-tuning approach was unsuccessful in our case for trajectory prediction, as specific prompts are crucial for achieving accurate outcomes. Altering certain words alters the states and environmental observations, leading to incorrect outputs. The GPT-2 model predictions offer insights into this issue in Figure 4(a).

(a) The U.S. Department of Defense is developing a new system to detect and track the movement of a U.S. military plane, the Pentagon said Tuesday.
The system, called the Joint-Missioned Tracking System (JMT), will be deployed to the U.S. military's Joint Expeditionary Force (JEF), which is planning to deploy to the region in the next two years. The JMT is a system that can detect and track a plane's flight path, and can also track the plane's trajectory.

(b))), (4 9 -second route using-second bus data00),35-second route using 5 -second route using
33. -second route using 5 seconds (.-second route using0-second route using-second route using3 5)0
3)-second route using)0.3

(c) Thoughts:
- Notable Objects
Barrier at (6.00,11.23) may cause an obstacle at (5.99,11.23) in 2.5 seconds
Barrier at (-7.66,1.63) may cause an obstacle at (-7.66,1.61) in 0.5 seconds
Adult at (-3.21,0.94) may cause an obstacle at (-3.21,0.94) in 0.5 seconds
Adult at (-3.12,-0.08) may cause an obstacle at (-3.12,-0.08) in 0.5 seconds
Adult at (-3.08,0.34) may cause an obstacle at (-3.08,0.34) in 0.5 seconds

(d) [THOUGHT] Notable Objects from Perception: None Potential Effects from Prediction: [/THOUGHT]
[META]Meta Action: MOVE FORWARD [/META]
[TRAJ] Trajectory (next 3 seconds): [(0.02,5.11), (0.05,10.15), (0.08,15.19), (0.11,20.23), (0.14,25.27)] [/TRAJ]

Figure 4: Failure cases of output generated from our models. (a) P-tuning changed one too many words from the input prompt. (*GPT-2 with P-tuning*); (b) Message not following the expected format. (*Llama-7B with P-tuning*); (c) Message in the correct format, but with missing trajectory. (*Zephyr-7B with LoRA*); (d) Trajectory with less than 6 predicted waypoints. (*Mistral-7B with LoRA*);

Table 2: Results from PEFT fine-tuning of the Llama2 models with P-tuning.

Model	L2			Average	Empty traj.
	1s	2s	3s		
Llama-7B	1.71	3.51	5.41	3.54	3227
Llama-7B-Chat	2.00	3.31	4.72	3.34	4519

4.3 Failure Analysis

For all models, a certain number of predictions resulted in empty trajectories. This issue was particularly observed with the Mistral-7B model, where the majority of instances failed to yield trajectories in the correct format. The models were missing a correct trajectory prediction due to several reasons:

- Empty output message.
- Messages that deviate from the prescribed output format. An illustration of this can be found in Figure 4(b).
- These messages adhere to the output format yet fail to include a trajectory. Instead, they might provide information about the environment or other scene participants. An example is detailed in Figure 4(c).
- Messages containing a complete trajectory but in a format that does not align with the expected standard are excluded from consideration as they do not constitute a valid output in general cases.
- This involves trajectories that contain fewer than six predictions. In such scenarios, we attempt to evaluate the predictions based on the available data and compare them for the corresponding number of steps. An example is provided in Figure 4(d).

5 CONCLUSIONS AND FUTURE WORK

This research highlighted the effective use of open-source LLMs in the field of trajectory prediction. Through detailed experimental analysis, it was shown that when these open-source LLMs were fine-tuned for specific downstream tasks, they yielded results comparable with their counterparts.

In the context of HRI, the use of LLMs for driving tasks would potentially allow the models to reason and provide explanations about the driving situation. This work serves as a pioneering step in employing open-source LLMs for trajectory prediction. While it does not introduce a novel learning method for training adapters in LLMs, it paves the way for future research in this direction, potentially exploring innovative training techniques and applications in trajectory prediction and beyond.

ACKNOWLEDGMENTS

This project has been partially supported by Wallenberg AI, Autonomous Systems and Software Program, WASP and Saab AB.

REFERENCES

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*.

[2] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. 2023. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. *arXiv preprint arXiv:2310.01957* (2023).

[3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[4] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[5] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602* (2021).

[6] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. GPT understands, too. *AI Open* (2023).

[7] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.

[8] Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. 2023. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415* (2023).

[9] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[10] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[11] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[12] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944* (2023).

[13] 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.

[14] Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. 2023. Dilu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292* (2023).

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 38–45.

[16] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. 2023. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412* (2023).