iMotion-LLM: Motion Prediction Instruction Tuning

Abdulwahab Felemban[1], Eslam Mohamed Bakr[1], Jian Ding[1],
Xiaoqian Shen[1], Abduallah Mohamed[2], Mohamed Elhoseiny[1]
[1]KAUST [2]Meta Reality Labs

Abstract

In the domain of autonomous navigation, we introduce iMotion-LLM: a fusion of Large Language Models (LLMs) with trajectory prediction, tailored to guide interactive multi-agent scenarios. Diverging from conventional motion prediction approaches, iMotion-LLM capitalizes on textual instructions as key inputs for generating contextually relevant predictions. Leveraging real-world driving scenarios in the Waymo Open Dataset enriched with textual motion instructions, iMotion-LLM integrates a pretrained LLM, fine-tuned with LoRA, to translate scene features into the LLM input space. A standout feature of iMotion-LLM is its ability to generate trajectories aligned with provided instructions while matching the performance of the underlying motion prediction model. These findings act as milestones in empowering autonomous navigation systems to interpret and predict the dynamics of multi-agent environments, laying the groundwork for future advancements in this field.

1. Introduction

Motion and trajectory prediction is a crucial component in autonomous driving. Forecasting future trajectories of surrounding entities based on historical data is vital for safety and motion planning. Recent challenges, e.g., Waymo Open Dataset challenges [5], introduce a track specifically designed to concentrate on motion prediction where 1.1 seconds of the past motion is observed, and 8 seconds to be predicted into the future. Various methodologies [6, 7] have been developed to tackle this challenge; however, they lack support for human-vehicle interaction through interactive instructions.

With the advent of large language models (LLMs), significant advancements have been made in applying LLMs to the autonomous driving context [2, 8]. Integrating LLMs into autonomous driving systems markedly improves decision-making and vehicle adaptability. This advancement results in more intelligent, reactive vehicles and promotes more intuitive human-machine interactions, making autonomous driving more effective and user-friendly.

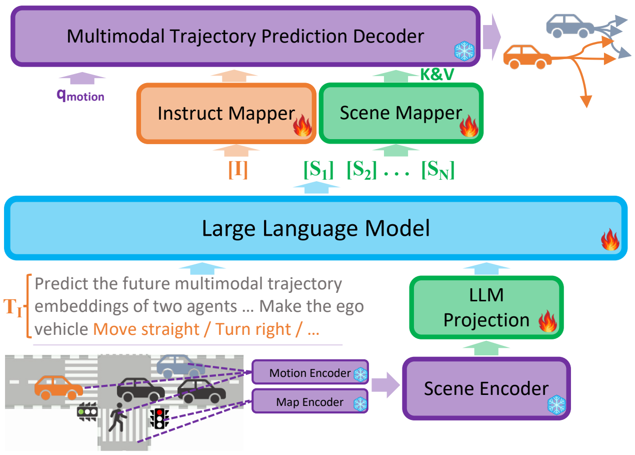
Multimodal Trajectory Prediction Decoder
q_{motion}
K&V
Instruct Mapper
Scene Mapper
[I]
$[S_1]$ $[S_2]$ . . . $[S_N]$
Large Language Model
T_I
Predict the future multimodal trajectory embeddings of two agents ... Make the ego vehicle Move straight / Turn right / ...
LLM Projection
Motion Encoder
Map Encoder
Scene Encoder

Figure 1. The proposed pipeline, referred to as iMotion-LLM, leverages the multimodal trajectory prediction capabilities of pretrained models, employing an encoder-decoder transformer architecture. Given a textual instruction and scene context embeddings, iMotion-LLM utilizes an LLM Mapper to project the encoded scene context embeddings from the Scene Encoder into the LLM input space. Subsequently, the LLM generates an instruction token [I] and a sequence of [S] tokens representing the scene context embeddings. The [I] token is projected to a query, and the scene context-generated tokens are projected to be the keys and values utilized by the multimodal trajectory prediction decoder.

However, the majority of existing LLM applications in autonomous driving [2, 8] concentrate on text or image inputs, neglecting the potential of vectorized motion prediction data. Vectorized motion data offers an abstract view of driving scenarios, providing essential information for motion prediction, such as the historical states of agents; they usually offer other vectorized map features like the locations of traffic lights or lane centers. The investigation into LLM applications for vector data is still underexplored [1], indicating an opportunity to improve autonomous driving technology by exploiting the benefits of vector data.

To this end, we integrate LLMs with vectorized motion prediction data and introduce the instructing motion prediction task that utilizes human instructions and scene data as inputs. The model outputs are the trajectory forecasts, as illustrated in Figure 1. To support this task, we augment the Waymo Open dataset [4] with ego vehicle direction instructions. The instruction details and statistics are explained in Section 2.2. Subsequently, we introduce the iMotion-LLM:

an instructable motion prediction model based on Large Language Models (LLMs). iMotion-LLM, harnesses pretrained models' multi-modal trajectory prediction capabilities through an encoder-decoder transformer architecture. As shown in Figure 1, it employs an LLM Projection to project encoded scene context embeddings from the Scene Encoder into the LLM input space. The LLM generates instruction token [I] and N [S] tokens representing scene context embeddings. These are combined into a single query by the Instruct Mapper. The resulting keys and values, derived and projected by the Scene Mapper, are used by the Multimodal Trajectory Prediction Decoder. Our experiments, using GameFormer [6] as a baseline, show that iMotion-LLM empowers autonomous navigation systems to interpret and predict the dynamics of multi-agent environments, while matching the performance of existing models.

Our contributions can be summarized as:

- We propose the instructing motion prediction task, which involves processing human instructions and scene data as inputs and outputs trajectory predictions. The proposed task enables the interactions between humans and vehicles, which is missed in traditional motion prediction.
- We augmented the Waymo Open Dataset with instruction categories enabling instructing motion prediction task. This augmentation is easily expandable to include more driving scenarios information and of higher granularity and will benefit future research in this direction.
- We introduce the iMotion-LLM: an instructing motion prediction model based on Large Language Models (LLMs). Different from earlier motion prediction models, iMotion-LLM leverages instructions, scene and vectorized motion data as an input to generate contextually relevant predictions.

2. iMotion-LLM

Firstly, in Section 2.1, we introduce our iMotion-LLM framework for highlighting the seamless integration into existing transformer-based motion prediction models, as depicted in Figure 1. Later, we delve into the generation of motion instructions in Section 2.2. To evaluate the agent's adherence to input interactions, we propose two metrics for assessing diversity and instruction-following capabilities in Section 2.3.

2.1. Instructable Motion LLM

2.1.1 Revisiting Existing Models

Recent successful transformer-based interactive trajectory prediction models [6, 7] commonly employ a schema comprising two main blocks. Initially, a scene encoder encodes the observed map and agents information into embeddings representing scene context information $S \in \mathbb{R}^{R\times d_{scene}}$, where d_{scene} is the embedding dimension. Subsequently, the multimodal trajectory prediction decoder utilizes cross-attention with S as keys and values, employing K learnable queries $q_{motion} \in \mathbb{R}^{K\times d_{scene}}$ to predict a Gaussian Mixture Model (GMM) of future trajectories for interactive agents. Both the Scene Encoder and Trajectory Decoder are depicted in Figure 1. The vectorized motion data is encoded through an LSTM, while the map features are processed using Multi-Layer Perceptrons (MLPs) for continuous features such as center lanes, or embedding layers for categorical features like the state of traffic lights. Subsequently, the Scene Encoder functions as a feature fusion layer.

2.1.2 Intigration of iMotion-LLM

In our proposed design we integrate, align, and instruct fine-tune the LLM with a pretrained Scene Encoder and the Multi-modal Trajectory Prediction Decoder. The LLM lies between them, and enables instructability. To enable this integrational design, illustrated in Figure 1, four main blocks are required: 1) LLM Projection module. 2) LLM itself. 3) Scene Mapper. 4) Instruction Mapper.

LLM Projection. Inspired by Vision-LLMs [3, 9], we employ a simple MLP-based projection layer to map input scene embeddings $S \in \mathbb{R}^{R\times d_{scene}}$ to $\tilde{S} \in \mathbb{R}^{R\times d_{LLM}}$, aligning with the LLM embeddings dimension d_{LLM}.

LLM. Projected scene embeddings $\tilde{S}$ and input instruction T_I are fed to the LLM to generate new tokens, $[I; S_1; S_2; \ldots; S_N]$, where I represents instruction embedding and S_n represents scene embeddings after grounding the instruction T_I.

Scene Mapper. To ensure seamless integration, we freeze the motion prediction model's encoder and decoder. Consequently, we map instruction-grounded tokens $[S_i] \in \mathbb{R}^{d_{LLM}}$ back to $\mathbb{R}^{d_{scene}}$, serving as keys and values in the Multimodal Trajectory Prediction Decoder, defined in Eq. 1.

$$K_i \& V_i = MLP([S_i]); i \in 1, ..., N. \qquad (1)$$

Instruct Mapper. Following the Scene Mapper, we project instruction token I back to the motion prediction model's embedding space (d_{scene}), which is fused with q_{motion} through a simple addition operation, as shown in Eq. 2.

$$Q = q_{motion} + MLP([I]). \qquad (2)$$

2.2. Trajectory Direction Instructions

As a simple attempt to make the motion prediction instructable, we employ a discrete instruction, e.g., eight possible motion categories. To derive plausible instructions, we devise a straightforward yet effective module that leverages eight seconds' worth of future observed agents' trajectories. Drawing inspiration from the mean Average Precision (mAP) metric utilized in Waymo motion prediction challenges [4], where they compute mAP across various motion

Table 1. The percentage of ground-truth future trajectory direction categories in the train and test data used. The used train data consists of 203,087 driving scenarios, and the test data includes 40,745 driving scenarios.

Class	0	1	2	3	4	5	6	7
Train	1.24%	49.12%	2.84%	3.26%	21.51%	20.44%	11.6%	1.47%
Test	1.18%	49.92%	2.65%	2.84%	21.32%	20.71%	0.11%	1.27%

ranges to comprehensively assess performance across diverse driving behaviors. We reuse their definition of driving behaviors, obtaining eight conceivable conditions encompass: 1) Stay stationary. 2) Move straight. 3) Move straight while veering to the right. 4) Move straight while veering to the left. 5) Turn right. 6) Turn left. 7) Take a right u-turn. 8) Take a left u-turn. Table 1 provides detailed statistics on these eight categories.

2.3. Instruction Following and Diversity Metrics

Our primary objective is to render current motion prediction models interactive and controllable. Hence, conventional metrics like Average Displacement Error (ADE) and Final Displacement Error (FDE) alone may not suffice to adequately evaluate the instruction-following capabilities of the proposed model. To address this, we introduce two metrics: Recall-Instruction Following (RIF) and Diversity.

Instruction Following. To gauge the model's ability to adhere to instructions, we consider the direction category of the ground-truth future trajectory D_{GT}, where C denotes the number of classes. For each prediction sample k, obtained modalities M yield direction category predictions $D^k_{pred_j}$. RIF is computed as the average recall across samples of different classes on predictions of different modalities:

$$RIF = \frac{1}{CM} \sum_{c=0}^{C-1} \frac{1}{K(c)} \sum_{i=1}^{K(c)} \sum_{j=1}^{M} RIF \left(D^i_{pred_j} \mid c \right), \quad (3)$$

where $K(c)$ denotes the number of test samples where $D_{GT} = c$.

Diversity. To assess the diversity of predicted modalities across different categories, we measure the ratio of unique direction categories predicted over the total number of modalities M. This is calculated as the average across categories:

$$Diversity = \frac{1}{C} \sum_{c=0}^{C-1} \frac{1}{K(c)} \sum_{i=1}^{K(c)} \frac{Unique(D^i_{pred})}{M}, \quad (4)$$

where $Unique(D^i_{pred})$ calculates the total number of unique categories across M modalities.

3. Experiments

3.1. Experimental Setup

Table 2. Instruction following and Diversity results comparing the original GameFormer, the GameFormer with instruction as a learnable query, the LLM integration with and without additional instruction token I. Using LLM both increased the RIF and reduced the diversity, making our proposed model follow instructions more precisely. L is the discrete labels disucssed in Section 2.2.

Model	**Instruction**	**RIF ↑**	**Diverstiry ↓**	**minADE ↓**	**minFDE ↓**
GameFormer	-	0.61	23.28%	1.18	2.68
C-GameFormer	L	0.72	23.38%	1.10	2.43
iMotion-LLM	T_I	0.86	18.46%	1.24	2.55
iMotion-LLM	$T_I + I$	0.85	18.28%	1.21	2.51

Implementation Details. We adopt the GameFormer [6] as our baseline, and use around 200K training samples. The LLM projection layers and LLM LoRA weights are fine-tuned for only two epochs with a batch size of 40. Adam optimizer is used with an initial learning rate (LR) of $1e-4$ with a linear warmup for 100 steps, and a cosine LR scheduler.

Metrics. In addition to the proposed metrics, i.e., Recall-Instruction Following (RIF) and Diversity, which are discussed in Section 2.3, we employ the conventional motion metrics; minADE and minFDE [4].

3.2. Experimental Results

Before integrating the LLM into GameFormer, we adapted it to include simple categorical conditions, labeled as Conditional-GameFormer (C-GameFormer). To ensure a fair comparison, all models were retrained using a sampled subset of the Waymo dataset [4], comprising 200K samples. In evaluating the models, we considered two motion metrics, minADE, and minFDE, alongside our proposed instruction-following metrics, RIF and Diversity.

For unconditional models, high diversity is desirable, indicating diverse plausible directions. Conversely, conditional models should exhibit low diversity, as all generated trajectories should converge in the same direction when given an instruction. However, diversity alone does not indicate good instruction-following skills, necessitating complementation with our RIF metric.

C-GameFormer. In the second row of Table 2, the conditional variant of GameFormer (C-GameFormer) achieves a higher RIF score (from 0.61 to 0.72) while maintaining the same diversity level (23.28%). Despite the improvement in RIF, the model maintains its uncertainty, as indicated by the unchanged diversity compared to the unconditional GameFormer. This could be interpreted as the model not paying enough attention to the instruction signal.

iMotion-LLM. Our proposed architecture, iMotion-LLM, demonstrates superior instruction-following capabilities, evidenced by higher RIF scores (0.85), coupled with a reduction in the diversity of generated multi-modal trajectories from nearly 23% to 18%. Improvement of both metrics infers precision of instruction following. In conclusion, an ideal instructable model should achieve high RIF with low diversity simultaneously while maintaining motion prediction performance close to the unconditional variant, as indicated by minADE and minFDE scores. Figure 2 illustrates a qualitative example of the same scenario. (a) displays the output of the baseline, unconditional GameFormer, while parts (b) and (c) showcase the generated trajectories in response to right and left instructions, respec-

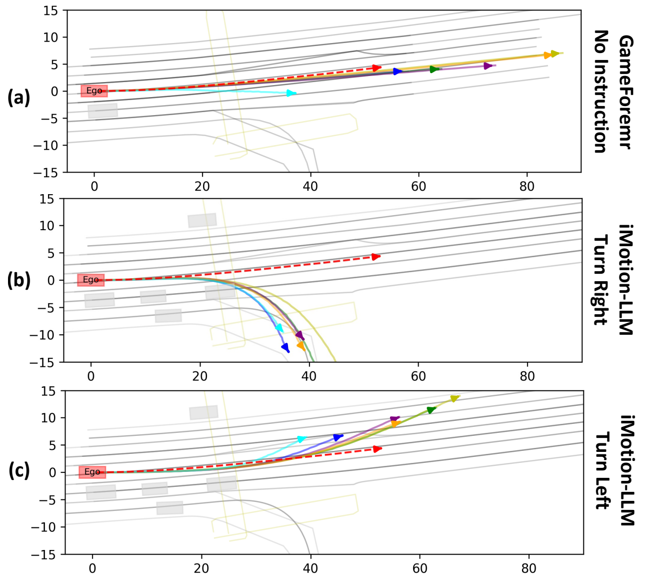

Figure 2. Qualitative results showing (a) the baseline GameFormer 6 modalities predictions each of different color, and the dashed red line represents the ground-truth real data measurement. (b) iMotion-LLM predictions with the instruction input ”turn right”. (c) iMotion predictions with the instruction input ”turn left”. The ground-truth instruction for this sample is ”move straight”.

tively, on the same scene using iMotion-LLM. This highlights the model's ability to follow instructions.

4. Limitations and Future Directions

Plausible Instructions. One limitation of our current study is the lack of examination of the model's performance in following both plausible and implausible instructions. Assessing the model's adherence to road rules and contextual information under varied conditions is essential for a comprehensive evaluation of its capabilities.

Complex Instruction Sets. Our study primarily focused on direction-based instructions. To unlock the full capabilities of the LLM, it is imperative to explore more complex instructions encompassing greater granularity and contextual information. This would enable a more nuanced understanding of the model's comprehension and execution of multifaceted driving tasks.

5. Conclusion

In conclusion, we introduce iMotion-LLM, a Large Multimodal Model powered by LLMs with trajectory prediction tailored for interactive multi-agent scenarios in autonomous navigation. By leveraging textual instructions as key inputs, our model generates contextually relevant trajectory predictions. Through integration with a pretrained LLM fine-tuned with LoRA, iMotion-LLM effectively translates scene features into the LLM input space, enabling accurate multimodal trajectory forecasts. Notably, our model's ability to generate trajectories aligned with provided instructions matches existing models' performance, marking a significant advancement in empowering autonomous navigation systems to anticipate multi-agent environments' dynamics. This work lays the foundation for future advancements in interactive autonomous navigation technology.

References

[1] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. *arXiv preprint arXiv:2310.01957*, 2023. 1

[2] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 902–909, 2024. 1

[3] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[4] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9690–9699, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 1, 2, 3

[5] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. 1

[6] Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3903–3913, 2023. 1, 2, 3

[7] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems*, 2022. 1, 2

[8] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023. 1

[9] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*, 2024. 2