iMOTION-LLM: MOTION PREDICTION INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce iMotion-LLM, a Multimodal Large Language Model (LLM) integrated with trajectory prediction, designed to guide interactive motion prediction scenarios. Unlike conventional multimodal trajectory prediction approaches, iMotion-LLM generates diverse and feasible future trajectories conditioned on textual instructions as a guidance signal. By augmenting real-world driving scenarios in the Waymo Open Motion Dataset (WOMD) with textual motion instructions, we propose InstructWaymo data augmentation. Leveraging this data augmentation, iMotion-LLM integrates a pretrained LLM, fine-tuned with LoRA, to map scene features into the LLM input space. Key results demonstrate that making the trajectory prediction model conditional improves its instruction-following capabilities. Specifically, the integration of the LLM enables a 11.07x ratio of actual-scenario feasible to infeasible recall instruction following, compared to 5.92x when using the Conditional GameFormer alone. These findings highlight the ability of iMotion-LLM to generate trajectories that not only align with feasible instructions but also reject infeasible ones, enhancing overall safety. Despite its improvements in instruction following, iMotion-LLM inherits the strong trajectory prediction performance of the baseline model, making it versatile across different driving modes. This combination of skills positions iMotion-LLM as a powerful augmentation technique for trajectory prediction models, empowering autonomous navigation systems to better interpret the motion prediction. This work lays the groundwork for future advancements in instruction-based motion prediction.

1 INTRODUCTION

In autonomous driving, accurate trajectory prediction is critical for ensuring safe and efficient navigation in dynamic environments. Given a window of observed history, the task is to predict multi-modal trajectories of multiple agents surrounding the ego vehicle in addition to the ego vehicle. A significant challenge in this field is the inherently diverse nature of driving behaviors in real-world scenarios, where an agent's future trajectory is not deterministic but can follow multiple feasible paths due to various factors such as traffic rules, interactions with other agents, and environmental conditions. Hence, developing models that can effectively predict diverse trajectories is crucial for autonomous systems to anticipate and adapt to potential hazards, make informed decisions, and ultimately achieve reliable and safe operation.

Recent challenges, *e.g.*, Waymo Open Motion Dataset (WOMD) challenges (Ettinger et al., 2021b), 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 (Huang et al., 2023a; Shi et al., 2022a; Seff et al., 2023) have been developed to tackle this challenge. Although previous models can predict multi-modality trajectories, the predicted paths are not diverse enough and mainly focus on one driving behavior (*e.g.*, the trajectory of only one feasible direction.). The reason is that previous prediction models are trained to imitate real-driving scenarios, fitting the driving behavior recorded future ground truth trajectory. Therefore, they lack comprehension of different driving behaviors in a given scenario.

To address the aforementioned challenge, we introduce a novel task called *Text-Guided Intention Trajectory Prediction* that aims to generate trajectories conditioning on a driving instruction for a selected vehicle. Additionally, the task provides a textual description predicting the feasibility of

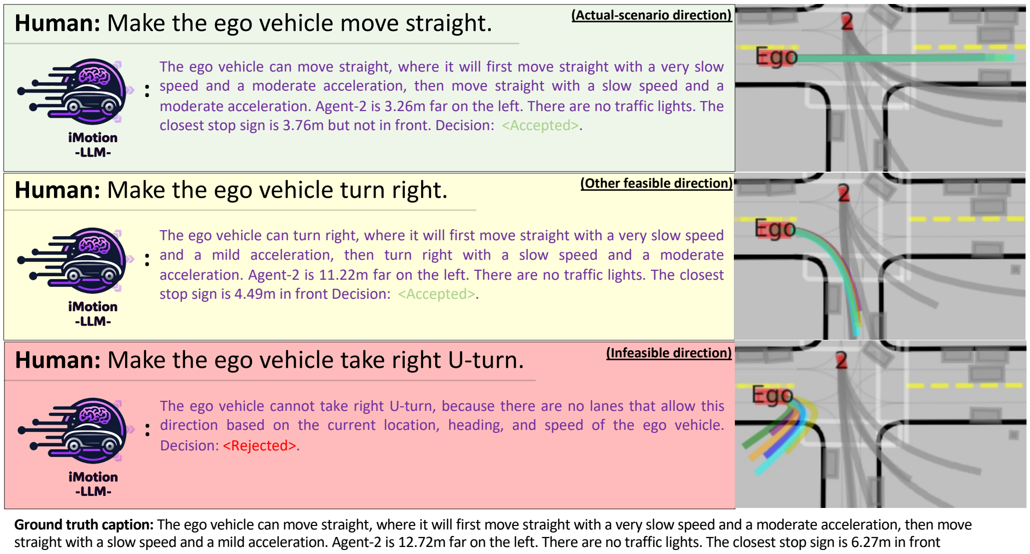


Figure 1: Our iMotion-LLM model can process three types of instructions and predict the corresponding trajectories. First, it can handle ground truth instructions that align with the direction of the recorded real-scenario trajectory (*e.g.*, Waymo Open Motion Dataset), correctly accepting the instruction and providing an explanation and trajectory. Second, iMotion-LLM can follow other feasible non-ground truth directions and predict the correct explanation and trajectory. Finally, when given an infeasible direction, iMotion-LLM correctly rejects the instruction.

a given action and explains how it would be executed in terms of different driving behaviors. We categorize driving behaviors based on two primary components: direction and acceleration, which together form the basis for diverse driving styles.

The proposed *Text-Guided Intention Trajectory Prediction task* offers several key advantages over traditional trajectory prediction models. First, it allows examining trajectory prediction modeling capability in covering different feasible driving behaviors for a given scenario. Second, by generating text-guided driving scenarios that are safety-critical or more challenging, the task can be used to train or test trajectory prediction and planning frameworks under diverse and demanding conditions, improving the robustness of autonomous systems. Furthermore, this approach enhances the interpretability of prediction models by providing explicit descriptions of driving intentions and how they translate into vehicle movements, making it easier to understand and trust the model's decisions.

To instantiate a dataset and model for this task, we augment WOMD (Ettinger et al., 2021a) with vehicle direction instructions. Additionally, we did evaluation experiments to show the generalizability on the NuPlan dataset (H. Caesar, 2021). The instruction details and statistics are explained in Section 3. For the evaluation, we propose two novel metrics. The Instruction Following Recall (IFR) measures how well the predicted trajectories adhere to the specified driving instruction, while the Direction Variety Score (DVS) captures the diversity of predicted directions. 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 integrating their encoder to map scene vector features and their decoder to decode trajectories. As shown in Figure 3, it employs an LLM Projection to project encoded scene context embeddings from the Scene Encoder into the LLM input space. The LLM generates an instruction token [I] and N [S] tokens representing the scene context embeddings. The instruction token is mapped to represent an additional intention query used by the decoder, while the scene tokens are used as keys and values. Our design of the encoder-decoder for the trajectory prediction model introduces an additional instruction query, alongside the learnable queries present in the original model design, which act as decoding seeds.

Our experiments, using GameFormer (Huang et al., 2023a) as a backbone, show that iMotion-LLM empowers autonomous navigation systems to interpret and predict the dynamics of agents, while almost matching the performance of the base model.

Our contributions can be summarized as:

- We augmented WOMD with instruction categories, enabling the motion prediction task to be instructed. This augmentation, named InstructWaymo, is easily expandable to include more detailed driving scenarios and will benefit future research in this direction.
- We enable traditional trajectory prediction modeling through the design integration of iMotion-LLM to generate text-guided instructable trajectory predictions, allowing the model to cover diverse feasible driving behaviors in a given scenario.
- We integrate LLMs with traditional trajectory prediction models to reason about predicted trajectories and determine how they should be executed in steps, while also training iMotion-LLM to accept or reject instructions based on feasibility.
- We introduced two evaluation metrics: Instruction Following Recall (IFR) and Direction Variety Score (DVS), to measure the model's ability to follow instructions and the diversity of predicted modalities across different directional categories, which cannot be captured by conventional metrics used in motion prediction.

2 Related Work

Multimodal Large Langauge Models. Large Language Models (LLMs) have significantly advanced in recent years (Radford et al., 2019; Devlin et al., 2018; Brown et al., 2020; Touvron et al., 2023b;a; Achiam et al., 2023), with models like GPT-4 (Achiam et al., 2023) demonstrating remarkable abilities in generating coherent, contextually relevant text across numerous domains. With the strong performance of LLMs, there is an emergence of multi-modal LLMs (MLLMs) (Alayrac et al., 2022), which extend the LLMs with reasoning abilities across diverse modalities. Notable works includes Flamingo (Alayrac et al., 2022), InstructBLIP (Dai et al., 2023b), MiniGPT-4 (Chen et al., 2023; Zhu et al., 2023), LLaVA (Liu et al., 2024; 2023), and Vicuna (Chiang et al., 2023). These works used visual instruction tuning to align with human intentions. There are some extensions that focus on detection and segmentation (Zhu et al., 2023; Wang et al., 2024; Lai et al., 2023; Bai et al., 2023), videos (Li et al., 2023; Zhang et al., 2023; Maaz et al., 2023), and 3D (Hong et al., 2023; Xu et al., 2023; Guo et al., 2023). Our work focuses on MLLMs for motion prediction tasks.

Trajectory Prediction Models for Driving Scenarios. The task of trajectory prediction involves analyzing the historical tracks of agents on a corresponding map to predict their joint future positions several seconds into the future. LSTMs (Alahi et al., 2016; Hochreiter & Schmidhuber, 1997) have been used to encode the historical states of agents, while CNNs (Cui et al., 2019; Gilles et al., 2021; Salzmann et al., 2020) have been employed to encode the rasterized images of the scene. Recently, GNNs (Chen et al., 2022; Huang et al., 2022b; Mo et al., 2022) have been employed to depict agent interactions effectively. The advent of Transformer-based models, like SceneTransformer (Ngiam et al., 2021) and WayFormer (Nayakanti et al., 2023), has further enhanced prediction through their efficient structure, though they primarily focus on the encoding process of driving scenarios vectorized representation. Motion Transformer (Shi et al., 2022b; 2024) and GameFormer (Huang et al., 2023b) innovates by improving the decoding stage, leading to better accuracy. MotionLM (Seff et al., 2023) used similar structures of LLM for the modeling, but still did not introduce the language reasoning ability to motion prediction task.

Multimodal Large Language Models for autonomous driving. With the emergence of Large Language Models (LLMs), there is a growing trend to adapt LLMs for autonomous driving scenarios (Chen et al., 2024; Dewangan et al., 2023; Hu et al., 2023; Huang et al., 2022a). Innovations like GPT-Driver (Mao et al., 2023) and SurrealDriver (Jin et al., 2023) exemplify the transformative impact of LLMs on motion planning and driving maneuver generation, marking significant advancements in autonomous vehicle technology. However, most existing methods primarily focus on text or image inputs, overlooking the benefits of vector representation in motion prediction. Vector representation offers an abstraction of driving scenarios, directly capturing the necessary information for motion prediction. Similar to Driving with LLMs (Chen et al., 2024), we integrate LLMs with vector-based data for motion prediction. While (Chen et al., 2024) introduced a benchmark focused mainly on QA tasks for driving scenarios, with motion only represented as a single quantized action (acceleration, braking, and steering), our work differs by focusing on motion represented as multi-modal multi-agent trajectories. This approach aligns more closely with existing trajectory prediction modules, making it more suitable for safe and reliable motion prediction.

3 InstructWaymo: Instruction Augmentation of Waymo Open Dataset

InstructWaymo offers a new perspective on the WOMD by making motion prediction instructable and language descriptive. Inspired by WOMD mAP calculation, which evaluates model performance across various driving behaviors, we designed a module that categorizes future motion into different directions, speeds, and acceleration. InstructWaymo uses future direction information as instructions, alongside future motion details—two-step direction, speed, and acceleration—as captions. Additionally instruction (direction) feasibility is calculated adding an extra layer of comprehension by identifying feasible and infeasible directions for each driving scenario. InstructWaymo will be provided as a publicly available script to augment WOMD. The script extracts additional useful information for future research, such as transcribed agents and object-relative locations to the focal agent, including neighboring agents, nearby stop signs, and traffic lights (e.g., Agent-2 is 13 meters to the right, Agent-3 is 3 meters ahead moving in the opposite direction, and there are 3 nearby stop signs, with the closest 1 meter in front). While this information exists in the dataset, the InstructWaymo script makes it easily accessible for future research requiring driving scenarios transcribed data. This data augmentation was applied to different driving scenarios. The scenarios are preprocessed similarly to GameFormer preprocessing, where each scenario includes up to 32 neighboring agents, with a total of 33 agents including the ego agent. Each agent in the scene is considered the focal agent (the ego-view agent), resulting in 4,228,499 samples. Of these, 2,011,265 samples involve the focal agent being a vehicle with valid detected instructions.

Direction. Direction is fundamental for instructing navigation, we adopted WOMD direction bucketing script to obtain eight conceivable direction conditions encompassing 8 classes listed in Table 1 with their statistics. The table shows a bias toward some behaviors like moving straight. See the details of the calculation of the directions in Section B in the appendix. We use driving directions as instructions in this work.

Table 1: Direction categories with their corresponding presence proportion in the train set.

Category	Stationary	Straight	Straight-right	Straight-left	Right	Left	Right u-turn	Left u-turn
Train	1.6%	55.8%	3.3%	3.7%	16.7%	17.5%	0.1%	1.4%

Speed and Acceleration. Following the intuition used in (Mohamed et al., 2022), we categorize trajectories of moving vehicles based on speeds and relative change in speeds. For that, we defined 5-speed categories and 9-acceleration categories; the suggested upper threshold and the categories are listed in the appendix in Table 6.

Feasibility of directions. We define the feasibility of directions into three categories: 1) actual-scenario direction (AS), which is based on the ground truth future trajectory and hence is always assumed to be a feasible direction; 2) Other feasible directions (OF), which are derivable directions but not the actual-scenario direction; 3) Infeasible directions (INF), which is the complement set of feasible directions. To assess feasibility, we consider a set of candidate destinations relative to the ego vehicle's current location and heading. These candidate destinations are possible locations on associated lanes within a range determined by the vehicle's speed (minimum range r_1, maximum range r_2). This range is calculated based on a minimum and maximum speed change of 45 km/h within 8 seconds and within a maximum range of 60 meters. Figure 2 illustrates this concept with two feasible directions. For the feasibility of staying stationary, the minimum range is considered to detect if the vehicle can slow down to stop in a range of 5 meters.

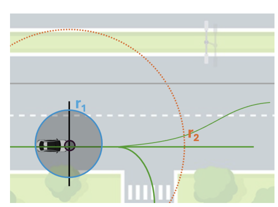


Figure 2: Illustration of feasibility detection of "move straight" and "turn right" within a range of ($r1$, $r2$).

LLM Instruction and caption. Based on the previously extracted attributes, we generate a template of input instruction and output caption that the LLM can process. The input instruction is the final direction the vehicle should arrive in. The output caption that the LLM aligns to generate auto-regressively includes the final direction, with two-step directions, speeds, and accelerations achieving the final direction as an interpretation of how an instruction is followed.

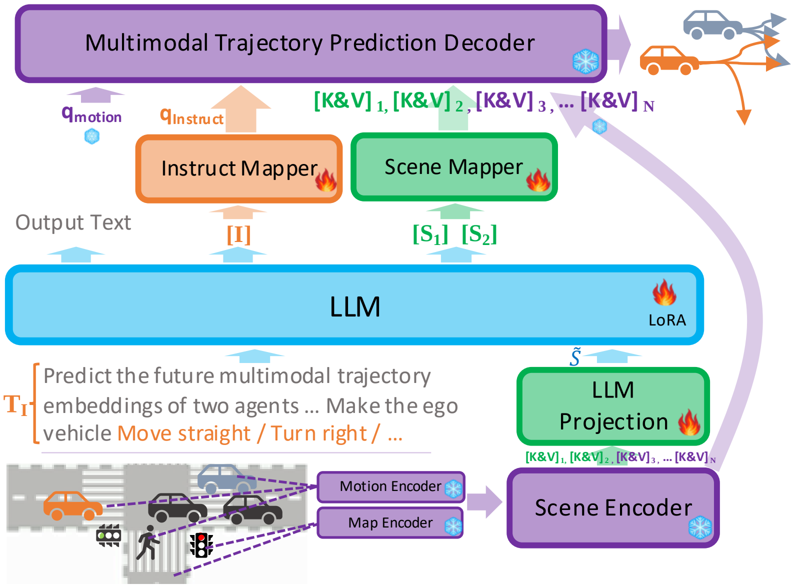


Figure 3: The proposed pipeline, referred to as iMotion-LLM, leverages the multi-modal trajectory prediction capabilities of pretrained models, employing an encoder-decoder transformer architecture. Given a textual instruction and scene context embeddings, iMotion-LLM utilizes an LLM Mapper to project 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 multi-modal trajectory prediction decoder.

4 iMotion-LLM

4.1 Revisiting existing models

Recent successful transformer-based interactive trajectory prediction models (Huang et al., 2023a; Shi et al., 2022a) commonly employ a schema comprising two main blocks. Initially, a scene encoder encodes the observed map and agent information into embeddings representing scene context information $S \in \mathbb{R}^{R \times d_{\text{scene}}}$, where d_{scene} is the embedding dimension. This context information is crucial for understanding the dynamics of the environment. The second component is a multimodal trajectory prediction decoder. This decoder employs cross-attention, using the scene context S as the keys and values (denoted as $K\&V$). The decoder also utilizes K learnable queries $q_{\text{motion}} \in \mathbb{R}^{K \times d_{\text{scene}}}$ to predict a Gaussian Mixture Model (GMM) of the potential future multi-modal trajectory of multiple agents. Based on this, the GameFormer model (Huang et al., 2023a) consists of two core blocks, the Scene Encoder and Trajectory Decoder, which are visually represented in purple in Figure 3. In the scene encoder,

- Vectorized motion data is encoded using a Long Short-Term Memory (LSTM) network.
- Map features are processed via Multi-Layer Perceptrons (MLPs) for continuous data, such as the geometric layout of center lanes, or through embedding layers for categorical data like the state of traffic lights.
- Once encoded, the scene encoder serves as a feature fusion layer, combining all the processed features to form scene representation.

After feature fusion, each token retains a specific correspondence to its map components. For example, in GameFormer's two-agent joint prediction model:

- The ego agent state is represented by two tokens, with one token being self-referential (when the ego agent is the focal agent), and the other token corresponding to the interaction with another agent (when the other agent becomes the focal agent).
- This pattern extends across all map features, where each map feature has two versions: one normalized with respect to the ego agent and another normalized with respect to the interacting agent.

4.2 Conditional Multimodal Trajectory Prediction Decoder

To generate a conditional output, cGAN (Mirza & Osindero, 2014) uses a conditioning signal in the generator model's input. Inspired by this, we fuse an additional learnable query, $q_{\text{instruction}}$, with the motion generation queries, q_{motion}. For making the base model conditional (conditional GameFormer), $q_{\text{instruction}}$ is learned using a simple embedding layer with a categorical class as input. When integrating an LLM with the base model, $q_{\text{instruction}}$ is derived from the LLM's output embeddings as described in the next subsection. The details of the conditional GameFormer training are provided in the pseudo-code in appendix E.

4.3 Integration of iMotion-LLM

In our proposed design we integrate, align, and instruct fine-tune the LLM with a pretrained GameFormer (Huang et al., 2023a) consisting of a Scene Encoder and the Multi-modal Trajectory Prediction Decoder. The LLM lies between them, and enables instructability and interpretability. To enable this integrational design, illustrated in Figure 3, five main blocks are required: 1) LLM Projection module. 2) LLM itself. 3) Scene Mapper. 4) Instruction Mapper. 5) Output Caption. The details of iMotion-LLM can be found in the pseudo-code in appendix E.

LLM Projection. Inspired by Vision-LLMs (Dai et al., 2023a; Zhu et al., 2024), 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}. R is the number of scene tokens, two of which correspond to the ego vehicle.

LLM. All projected scene embeddings $\tilde{S}$ and input instruction T_I are fed to the LLM to generate output tokens, $[I; S_1; S_2]$, where I represents instruction embedding and S_n represents the ego corresponding 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 ego tokens $[S_i] \in \mathbb{R}^{2 \times d_{LLM}}$ back to $\mathbb{R}^{2 \times d_{scene}}$, that are used with the rest of keys and values of other scene information coming directly from the scene encoder bypassing the LLM ($\mathbb{R}^{(R-2) \times d_{scene}}$), combined serving as keys and values in the Multimodal Trajectory Prediction Decoder. The scene mapping can be defined as in Eq. 1.

$$K_i \& V_i = MLP([S_i]); i \in 1, 2. \tag{1}$$

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.

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

Output Caption. Along with generating scene and instruction tokens, the LLM outputs a text that describes how the instruction is executed, and a textual decision of ("<Accept>" or "<Reject>") to indicate whether an instruction is feasible or not.

5 Instruction Following and Diversity Metrics

Our primary objective is to render current motion prediction models interactive and instructable. 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: Instruction Following Recall (IFR) and Direction Variety Score (DVS).

Instruction Following Recall (IFR). To gauge the model's ability to adhere to instructions, we compare given instructions direction $D_{intruct}$, with the directions of the generated multimodal trajectories. For each of the M modalities, we calculate its direction D_{pred_j}, using the same module used to extract the actual-scenario ground truth future direction. Based on that IFR is computed as the average recall across N samples of multimodal trajectory predictions:

$$IFR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{M} \sum_{j=1}^{M} \text{Recall} \left(D^i_{pred_j} \mid D^i_{intruct} \right), \tag{3}$$

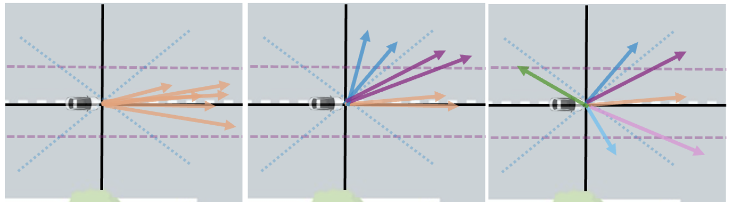


Figure 4: Illustrative examples of IFR and DVS of 6 modalities given a direction instruction of "move straight".

Where a higher IFR indicates higher adherence to a given instruction signal. For an unconditional model that takes no instruction signal, we can still measure the IRF where $D_{intruct}$ is considered the actual-scenario ground truth's future direction.

Direction Variety Score (DVS). To assess the directional diversity of predicted modalities, we measure the ratio of unique direction categories predicted over the total number of modalities M. This metric is calculated irrespective of the actual or given instruction as:

$$DVS = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{Unique}(D^i_{pred})}{M}, \tag{4}$$

Where a higher DVS indicates more diversity of predicted directions.

Figure 4 shows three illustrative examples given an input instruction or an actual-scenario instruction of “move straight.”. The left example shows the highest possible IFR, where all modalities are precisely in the "move straight" direction. The middle example has only two true positives while covering 3 unique directions, resulting in a $2/6$ IFR and $3/6$ DVS. The right example shows the highest possible directional diversity of a maximum possible number of unique directions, with only one true positive resulting in $1/6$ IFR. In our experiments, we report the values in percentages.

6 Experiments

6.1 Experimental Setup

Implementation Details. We started by reproducing the checkpoint of GameFormer (Huang et al., 2023a) using 4,228,499 training samples (using the same training setup and hyperparameters as suggested). Similarly, we trained the conditional GameFormer model with 2,011,265 training samples augmented with categorical instructions. We use this as our pretrained trajectory forecasting backbone. For iMotion-LLM, which integrates the pretrained conditional GameFormer with Llama-2-7B (Touvron et al., 2023b) and extends the vocabulary to include 3 additional tokens ([I], [S1], and [S2] tokens), the LLM mapping modules are fully fine-tuned, and LoRA weights are fine-tuned with LoRA parameters of $r = 8$ and $\alpha = 16$ for 3,510 training steps. The training involved 39 iterations per inner epoch, a batch size of 64 per GPU, using 4x A100-80GB GPUs, effectively covering 900,000 training samples over 90 epochs. We utilized the Adam optimizer with an initial learning rate (LR) of $1e$-4, incorporating a linear warmup for the first 100 steps starting from a warmup LR of $1e$-6, followed by a cosine LR scheduler. Training takes approximately 8 hours to complete 90 epochs.

Training Scenarios. The model was trained with Actual-Scenario (AS) instructions and Infeasible (INF) instructions. During training, the selection of a driving scenario (AS or INF) sample was random. For AS samples, the loss is calculated using both the LLM output text cross-entropy loss (feasibility detection text, and transcription of how the action is performed) and the trajectory negative log-likelihood loss (the same training objective as GameFormer). For INF instructions, since there are no reference ground-truth trajectories, the loss solely consists of the cross-entropy of the LLM output text for feasibility detection.

Metrics. In addition to the proposed metrics, *i.e.*, IFR and DVS, which are discussed in Section 5, we employ the conventional motion metrics; minADE and minFDE (Ettinger et al., 2021a). The minADE and minFDE are evaluated using the same examples used to evaluate the actual-scenario instructions setup.

Evaluation. Each model is evaluated with three instruction types: actual-scenario, other feasible, and infeasible. We use 2,311 evaluation examples. We compare different models with the exact set of evaluation examples, we considered using equal number of examples across each category except "right u-turn" because it is rarely presented. Evaluation takes around 40 minutes on a single A100-80GB GPU.

6.2 RESULTS & DISCUSSION

(a) **IFR Performance**: IFR performance of different models under different instruction types and the ratio of AS-to-INF IFR and OF-to-INF IFR, higher ratios indicate better performance.

Model	Instruct.	Feasibility Detect.	(AS) IFR ↑	(OF) IFR ↑	(INF) IFR ↓	(AS / INF) IFR Ratio ↑	(OF / INF) IFR Ratio ↑
GameFormer			68.60%	3.36%	1.47%	46.67	2.29
Conditional GameFormer (ours)	✓		**81.39%**	**30.13%**	13.74%	5.92	2.19
iMotion-LLM (ours)	✓		81.37%	24.53%	10.61%	7.67	**2.31**
iMotion-LLM (ours)	✓	✓	73.94%	13.72%	**6.68%**	**11.07**	2.05

(b) **DVS Results**: DVS results of the models with no feasibility detection. Lower DVS combined with high feasible instructions IFR indicates higher directional precision.

Model	Instruct.	(AS) DVS	(OF) DVS	(INF) DVS
GameFormer		15.78%	12.96%	12.98%
Conditional GameFormer (ours)	✓	8.74%	19.75%	21.80%
iMotion-LLM (ours)	✓	6.04%	15.42%	17.96%

(c) **Feasibility Detection**: accuracy of iMotion-LLM feasibility detection on all three types of input instructions.

Model	iMotion-LLM
(AS) **Acc.**	87.35%
(OF) **Acc.**	40.75%
(INF) **Acc.**	75.96%

Table 2: **Main Results**. Evaluating models with/out instruction input during inference (Instruct.), models with/out feasibility classification capability (Feasibility Detect.), and under three instruction types (AS: Actual-Scenario, OF: Other Feasible, INF: Infeasible).

GameFormer. Although GameFormer does not take any conditions and cannot classify feasibility, we evaluated it across all three categories. As expected, the model performs reasonably at generating predictions that fit the actual scenario and struggles to produce predictions that follow instructions for other directions; see Table 2a (top-row).

Conditional GameFormer. As shown in Table 2a, making GameFormer conditional on a discrete direction enhances its ability to follow actual-scenario instructions, as indicated by a 12.8% increase in IFR (AS). The recall for following other feasible and infeasible instructions also improves. The details of the conditional GameFormer can be found in Section 4.2 and in the pseudo-code in appendix E.

iMotion-LLM without feasibility detection. Even though iMotion-LLM was trained with feasibility classification capability, we show in Table 2a how the model performs, assuming all generated trajectories are valid. With this setup, iMotion-LLM without feasibility detection, despite the overall drop in IFR, outperforms the Conditional GameFormer in the feasible to infeasible (both AS/INF and OF/INF) IFR ratios.

iMotion-LLM with feasibility detection. Given iMotion-LLM's ability to detect whether an instruction should be accepted or rejected, any prediction with rejected feasibility is assigned an IFR of 0. The model did not perform the best on other feasible instructions. More notably can be seen where iMotion-LLM achieves a lower (OF/INF) ratio when considering feasibility detection (the last row).

Insignificance of other feasible instructions following. For other feasible instructions besides the actual-scenario instruction, as shown in Table 2 both the conditional GameFormer and iMotion-LLM exhibit lower IFR and higher DVS compared to the actual-scenario case. Intuitively, this behavior correlates with infeasibility of instructions rather than feasibility. Even though iMotion-LLM detects the feasibility of actual-scenario and infeasible instructions with a high rate, it does not detect other feasible instructions' true positives with such significance. We attribute this to two factors. First, driving behaviors for other feasible instructions may diverge from real scenarios, making the task more complex and requiring better generalizability. Interestingly, iMotion-LLM rejects this instruction. Figure 5 shows a successful case of accepting feasible instructions and rejecting infeasible instructions; stationary was labeled as infeasible due to the vehicle's current velocity. The feasibility detection accuracy is shown in Table 2c. We show additional results in appendix C. In the appendix n in Figure 7 we show that feasible directions might not always align with safety, laws, or convenience.

Evaluation on minADE and minFDE. Although this work primarily focuses on instruction-following ability in the proposed new metrics, we also evaluate the models using the traditional minADE and minFDE metrics in two scenarios: with and without the condition instruction during testing. In Table 3, we show the state of existing leading trajectory prediction models, as reported by the original authors on the WOMD interaction prediction challenge test set, as a reference to ensure our development does not deteriorate the performance of the base task we are building on. Table 4 demonstrates that our model's performance does not diverge significantly from the baseline in the traditional metrics. The GameFormer model used is a retrained checkpoint, and its performance on the validation set aligns closely with the reported results on the WOMD test set. The Conditional GameFormer is an additional model we trained, which incorporates a conditional direction label as input. In Table 4, the iMotion-LLM and the iMotion-LLM (Drop instruct.) represent the same model, but are evaluated in two ways: using the actual scenario instruction and without the input instruction.

Table 3: **WOMD Test Set minADE & minFDE**. Joint prediction performance reported by different SOTA models.

Model	minADE ↓	minFDE ↓
GameFormer (Huang et al., 2023b)	0.9161	1.9373
MTR (Shi et al., 2022b)	0.9181	2.0633
MotionLM (Seff et al., 2023)	0.8911	2.0067

Table 4: **WOMD Validation Subset minADE & minFDE**: Our reported Joint prediction performance.

Model	Cond.	minADE	minFDE
GameFormer (reproduced)		0.8888	1.9293
iMotion-LLM (Drop instruct.) (ours)		1.1642	2.7477
Conditional GameFormer (ours)	✓	0.8223	1.7001
iMotion-LLM (ours)	✓	0.9758	2.1257

Generalizability to NuPlan Dataset. To investigate the model capability to generalize to other datasets, we investigated three setups in Table 5. Even though the basline modules that are integrated into iMotion-LLM are pretrained on Waymo Open Dataset, we show the model generalizability by 1) zero-shot evaluation, 2) fine-tuning the LLM and mapping modules, 3) End-to-end finetuning (including the baseline modules). For finetuning, we consider 2,212 examples from the Pitssburg Train split. And evaluation was performed cross-city, where all the reported results are from the "Boston Train" split. As many of the features the GameFormer model uses from Waymo Open Dataset cannot be matched to features available from NuPlan, those features were set to default values to avoid changing the design of the pretrained GameFormer modules. Intrestingly, the model showed good IFR, combined with larger DVS that indicate some level of uncertinatiy, which is expected. The displacement errors show a lack of accuracy of the model, yet it gets improved when finetuned on a small fold of the data. These generalizability results were conducted using the actual-scenario instructions extracted from the ground truth future motion only. These experiments were done based on iMotion-LLM which generates all scene tokens, not only the ego agent tokens.

Table 5: Model Generalizability to NuPlan Dataset. Comparison of finetuning strategies based on IFR, DVS, and vehicle minADE/minFDE metrics.

Finetuning Strategy	IFR	DVS	Vehicle minADE	Vehicle minFDE
zero-shot	83.9	7.6	2.66	5.48
Finetuned	86.9	6.3	2.09	4.76
End-to-End Finetuned	85.8	7.3	1.90	4.50

7 LIMITATIONS AND FUTURE DIRECTIONS

Our study provides a key step by focusing on direction-based instructions, illustrating the potential of the LLM in executing driving tasks. By showing that the model can effectively interpret and act on these instructions, we have established a baseline that future research can build upon. As the

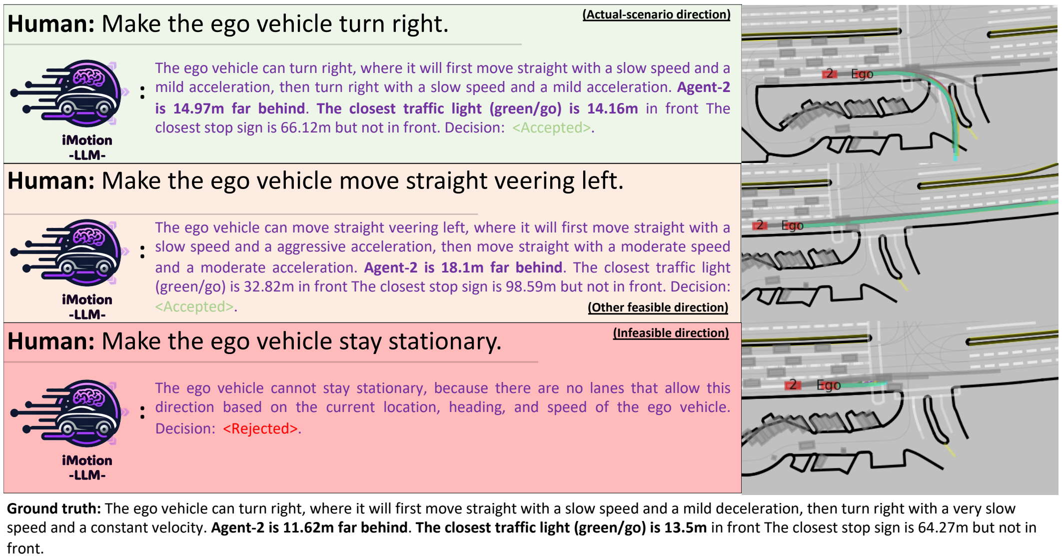


Figure 5: Qualitative result showing the model ability in following feasible instructions (top two figures), and making sense of surroundings. While also rejecting irrational scenarios like staying stationary in the bottom figure. Yet it generates a trajectory where the ego is stopping, and the interactive agent (Agent-2) is overtaking it.

baseline model by design allows multi-agent trajectory prediction, in Appendix D we show trials to instruct multiple agents at the same time. While there is a noticeable performance drop when extending instructions to multiple agents, we expect that further analysis can lead to a better design to improve this direction in the future. Exploring more complex instructions that encompass greater granularity and contextual information will further enhance the model's nuanced understanding and execution of multifaceted driving tasks. Furthermore, we employed relatively simple instructions and output captions, demonstrating the feasibility and effectiveness of this approach. Our work paves the way for incorporating more advanced and diverse input instructions and output captioning with varying levels of reasoning based on the ego state and surroundings. Although these elements were not included in this study, the attributes we extracted in InstructWaymo can facilitate their seamless integration. This presents an exciting opportunity for future research to develop more sophisticated and naturalistic implementations, extending the impact of our initial findings.

8 Conclusion

In conclusion, we introduce iMotion-LLM, a Large Multimodal Model powered by LLMs, tailored for trajectory prediction in interactive multi-agent scenarios within autonomous navigation. By leveraging textual instructions as key inputs, our model not only generates contextually relevant trajectory predictions but also showcases an enhanced ability to interpret and act upon these instructions. 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 inherits the performance of the underlying backbone model, marking a significant advancement in empowering autonomous navigation systems to anticipate the dynamics of multi-agent environments. iMotion-LLM, combined with InstructWaymo instructions and captions, provides the capability to align trajectories with feasible instructions and reject infeasible ones, thereby enhancing operational safety. This work not only advances the field of autonomous navigation by enabling systems to better anticipate and react within multi-agent environments but also sets a solid foundation for further innovations in interactive autonomous systems.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

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. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Scept: Scene-consistent, policy-based trajectory predictions for planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17103–17112, 2022.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2090–2096. IEEE, 2019.

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*, 2023a.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *CoRR*, abs/2305.06500, 2023b. doi: 10.48550/ARXIV.2305.06500.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Vikrant Dewangan, Tushar Choudhary, Shivam Chandhok, Shubham Priyadarshan, Anushka Jain, Arun K Singh, Siddharth Srivastava, Krishna Murthy Jatavallabhula, and K Madhava Krishna. Talk2bev: Language-enhanced bird's-eye view maps for autonomous driving. *arXiv preprint arXiv:2310.02251*, 2023.

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)*, pp. 9690–9699, Los Alamitos, CA, USA, oct 2021a. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00957. URL `https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00957`.

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*, pp. 9710–9719, 2021b.

Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Home: Heatmap output for future motion estimation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 500–507. IEEE, 2021.

Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, et al. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023.

K. Tan et al. H. Caesar, J. Kabzan. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *CVPR ADP3 workshop*, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *arXiv preprint arXiv:2307.12981*, 2023.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pp. 9118–9147. PMLR, 2022a.

Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2605–2611. IEEE, 2022b.

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)*, pp. 3903–3913, October 2023a.

Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. *arXiv preprint arXiv:2303.05760*, 2023b.

Ye Jin, Xiaoxi Shen, Huiling Peng, Xiaoan Liu, Jingli Qin, Jiayang Li, Jintao Xie, Peizhong Gao, Guyue Zhou, and Jiangtao Gong. Surrealdriver: Designing generative driver agent simulation framework in urban contexts based on large language model. *arXiv preprint arXiv:2309.13193*, 2023.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

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

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

Xiaoyu Mo, Zhiyu Huang, Yang Xing, and Chen Lv. Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):9554–9567, 2022.

Abduallah Mohamed, Deyao Zhu, Warren Vu, Mohamed Elhoseiny, and Christian Claudel. Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation. In *European Conference on Computer Vision*, pp. 463–479. Springer, 2022.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987. IEEE, 2023.

Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2021.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pp. 683–700. Springer, 2020.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8579–8590, 2023.

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*, 2022a.

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*, 35:6531–6543, 2022b.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

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

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

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.

Table 6: Speed and acceleration/deceleration categories and upper thresholds.

Speed category	Very slow	Slow	Moderate	Fast	Very fast
Threshold (km/h)	20	40	90	120	>120
Accel./Decel. category	Constant velocity	mild	Moderate	aggressive	extreme
Threshold (km/h increase in 8s)	6	25	46	65	>65

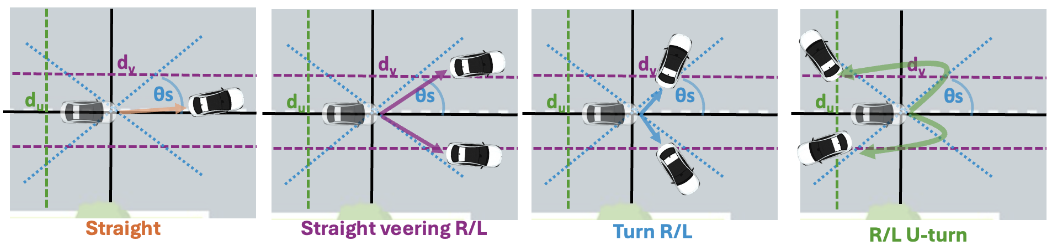


Figure 6: Illustrative examples of directions categories.

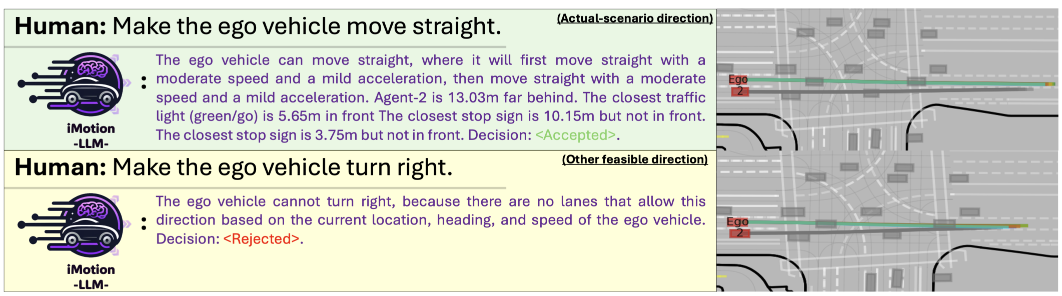


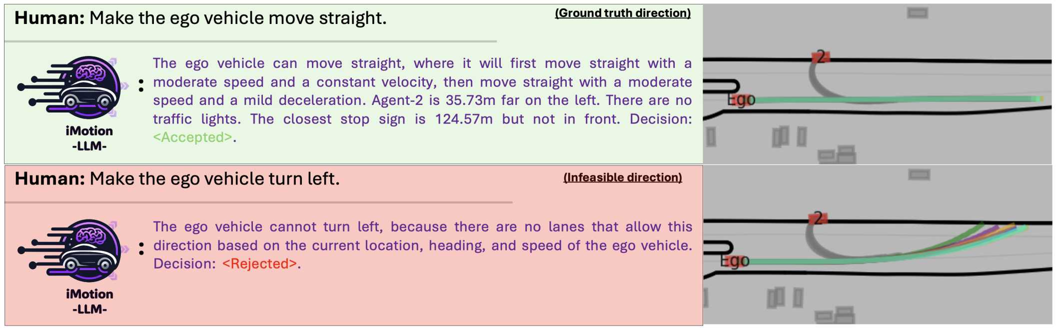
Figure 7: Qualitative result showing in the top figure how iMotion-LLM accept and follow the actual-scenario instruction of "move straight" and in the bottom figure how it rejects "turn right" even though "turn right" is labeled as a feasible direction.

A Speed and acceleration categories

The set of 5 different speed categories ranging from very slow to very fast, and the set of acceleration or deceleration ranging from mind to extreme, including a no acceleration (i.e., constant velocity). We designed these thresholds heuristically, yet they can be easily adapted. Table 6 shows the used thresholds.

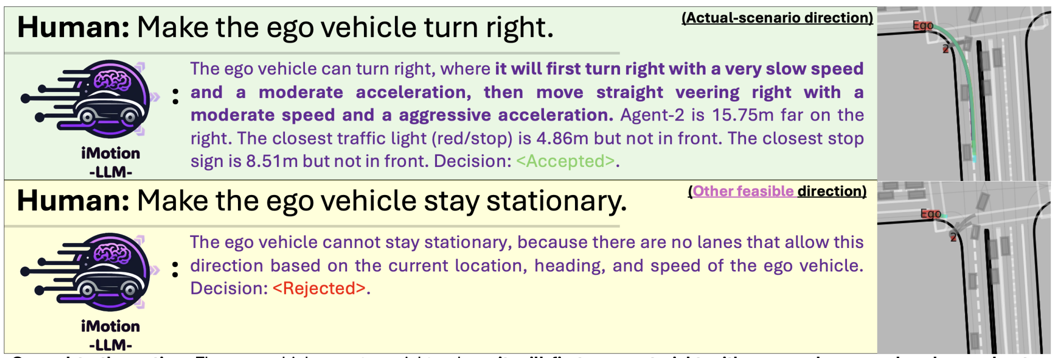
B Calculation of the directions

Following the illustration shown in Figure 6, motion direction is measured based on the relative heading angle between a time step and a future target step. We calculate direction solely based on trajectory information; the heading angle is calculated using two consecutive trajectory discrete samples. If the maximum future speed is within a threshold of $\mathrm{v}_{\mathrm{stationary}} = 2\mathrm{m/s}$, and the vehicle traveled a distance within $\mathrm{d}_{\mathrm{stationary}} = 5\mathrm{m}$, the vehicle is considered stationary. Otherwise, the vehicle is moving straight if the relative heading is within $\theta_{\mathrm{s}} = 30$ degrees. But if the longitudinal displacement is greater than $\mathrm{d}_{\mathrm{v}} = 5\mathrm{m}$, it is categorized as straight veering right/left. If the relative heading exceeds θ_{s}, and the latitudinal shift is less than $\mathrm{d}_{\mathrm{u}} = 5\mathrm{m}$ in the opposite direction, it is considered as turning right/left. Otherwise, it is a U-turn. Right and left directions are distinguished based on the sign of the relative heading. Figure 6 illustrates the different classes. Table 1 provides detailed statistics on these eight categories.



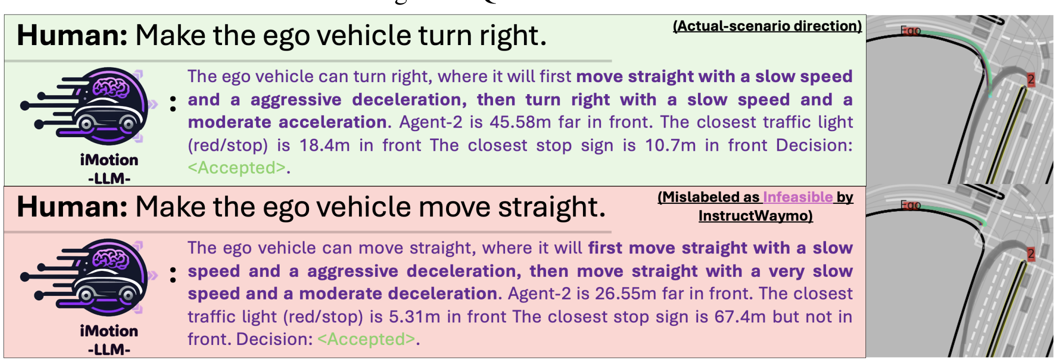
Ground truth: The ego vehicle can move straight, where it will first move straight with a moderate speed and a constant velocity, then move straight with a moderate speed and a constant velocity. Agent-2 is 28.46m on the left. There are no traffic lights. The closest stop sign is 60.52m but not in front.

Figure 8: Qualitative Results.



Ground truth caption: The ego vehicle can turn right, where **it will first move straight with a very slow speed and a moderate acceleration, then turn right with a slow speed and a aggressive acceleration**. Agent-2 is 10.4m far on the right. The closest traffic light (red/stop) is 0.79m but not in front. The closest stop sign is 88.5m but not in front.

Figure 9: Qualitative Results.



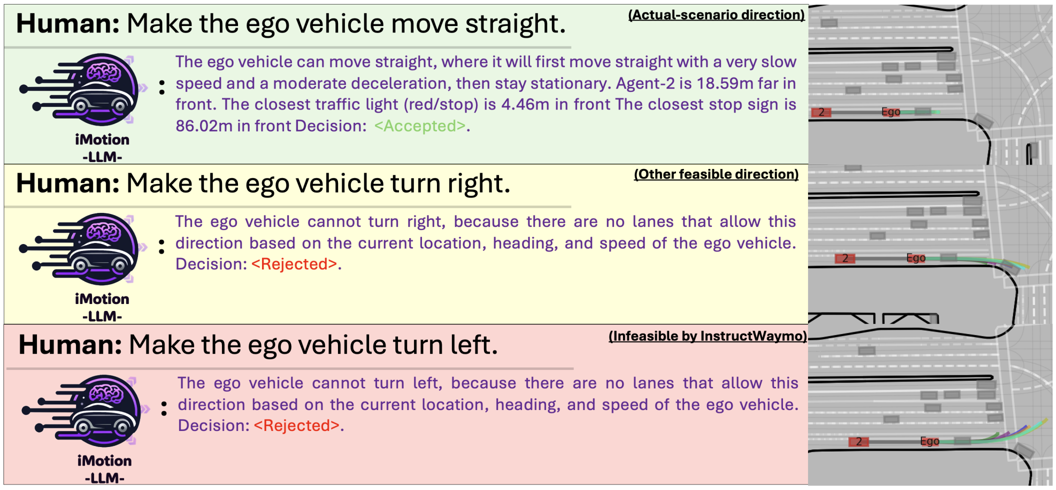
Ground truth caption: The ego vehicle can turn right, where it will first **move straight veering right with a slow speed and a aggressive deceleration, then turn right with a slow speed and a moderate acceleration**. Agent-2 is 46.15m far in front. The closest traffic light (red/stop) is 27.16m in front The closest stop sign is 14.25m but not in front.

Figure 10: Qualitative Results.

C Additional Qualitative Results

D Multi-Agent Support

As the baseline model by design allows multi-agent trajectory prediction, iMotion-LLM can consider instructing multiple agents by providing instructions for multiple agents in the scene. For 2-Agent support, during training and evaluation, we sample combinations of different types of instructions, for example feasible instruction for the ego based on actual-scenario, and infeasible instruction for Agent-2, or infeasible instruction for the ego and on of the other feasible instructions of Agent-2. For evaluation, we evaluate each agent separately while switching the combinations of instructions. Table 7 and 8 shows the performance for the Ego (Agent-1) and Agent-2 respectively. Performance for both



Ground truth caption: The ego vehicle can move straight, where it will first move straight with a very slow speed and a mild deceleration, then move straight with a very slow speed and a mild acceleration. Agent-2 is 22.29m far behind. The closest traffic light (red/stop) is 22.44m in front The closest stop sign is 94.64m in front

Figure 11: Qualitative Results.

is similar, yet shows a drop in performance compared to when instructing a single agent in both IFR and accuracy.

Table 7: Two-agent iMotion-LLM evaluating the ego agent (Agent-1)

Instruction Type	Agent-1	Agent-2	**IFR**	**DVS**	**Acc.** ↑
Actual-scenario	Actual-scenario	Actual-scenario	51.86%	3.07%	71%
Other feasible	Other feasible	Actual-scenario	15.93%	7.15%	61%
Other feasible	Other feasible	Other feasible	14.05%	6.85%	54%
Infeasible	Infeasible	Actual-scenario	5.90%	5.85%	48%
Infeasible	Infeasible	Infeasible	5.44%	5.64%	56%

Table 8: Two-agent iMotion-LLM evaluating Agent-2

Instruction Type	Agent-2	Agent-1	**IFR**	**DVS**	**Acc.** ↑
Actual-scenario	Actual-scenario	Actual-scenario	42.77%	8.18%	63%
Other feasible	Other feasible	Actual-scenario	12.90%	15.79%	50%
Other feasible	Other feasible	Other feasible	16.47%	15.22%	45%
Infeasible	Infeasible	Actual-scenario	6.81%	18.11%	49%
Infeasible	Infeasible	Infeasible	6.70%	16.43%	64%

E CONDITIONAL GAMEFORMER AND IMOTION-LLM TRAINING PSEUDO CODES

Algorithm 1: The pseudocode of Conditional-GameFormer (C-GameFormer).

Input : $C_{instruction} \in \mathbb{Z}$: Instruction category; N_a: Num. agents; d_a: Num. state features; N_m: Num. map lanes; N_p: Num. points per lane; d_m: Num. map features; d_{scene}: latent dimension; $t_{obs} = 11$: Observed time steps; $t_{pred} = 80$: To predict time steps; $t_{select} = [29, 49, 79]$: Selected time steps; N_{pred}: Two Agents to predict; M: Num. modalities (futures); $\mathbf{Agents} \in \mathbb{R}^{N_a \times t_{obs} \times d_a}$: history states ; $\mathbf{Maps} \in \mathbb{R}^{N_{pred} \times N_m \times N_p \times d_m}$; N: Num. scene embeddings;

Output : $\mathbf{Pred} \in \mathbb{R}^{M \times N_{pred} \times t_{pred} \times 4}$: prediction GMM parameters $(\mu_x, \mu_y, \sigma_x, \sigma_y)$, where (μ_x, μ_y) are the 2D trajectory centers

$queried\_agents \leftarrow [0, 1, ..., N_{pred} - 1]$; `// Target agents, [0,1] for two agents`
$queried\_modalities \leftarrow [0, 1, \ldots, M - 1]$; `// M modalities`
$S \leftarrow [\,]$; `// Initialize scene tokens empty list of embeddings`
for *each agent_state in agents_history* **do**
  agent_emb ← Motion_Encoder(agent_state); `// Encode agent state`
  $S \leftarrow S \cup \{agent\_emb\}$; `// Append agent embedding to S`
end
for *each map_feature in map_features* **do**
  map_emb ← Map_Encoder(map_feature); `// Encode map feature`
  $S \leftarrow S \cup \{map\_emb\}$; `// Append map embedding to S`
end
$S \leftarrow$ selfAttention(S); `// Apply fusion self-attention encoder (Scene Encoder)`
$K, V \leftarrow S$; `// Use S as the keys and values of the trajectory decoder`
$Q \leftarrow [\,]$; `// Initialize Q`
$q\_instruction \leftarrow$ Embedding($C_{instruction}$) ; `// Learnable instruction query (proposed)`
for *each agent_number in queried_agents* **do**
  q_agent ← Embedding(agent_number); `// agent query`
  for *each modality_number in queried_modalities* **do**
    $q\_modality \leftarrow$ Embedding(modality_number); `// Modality query`
    $q\_motion \leftarrow q\_agent + q\_modality$; `// Combine queries`
    $q\_motion \leftarrow q\_motion + q\_instruction$; `// Add instruction query (proposed)`
    $Q \leftarrow Q \cup \{q\_motion\}$; `// Append motion query to Q`
  end
end
output_features ← Multimodal_Trajectory_Decoder(Q, K, V);
Pred, Scores ← MLP(output_features), MLP(output_features); `// Get multimodal trajectories and modality scores`
NLL_loss ← NLL(Pred[best_mode, :, t_{select}], ground_truth_2D)
gmm_loss ← NLL_loss - CrossEntropy(Scores, best_mode)

Algorithm 2: The pseudocode of iMotion-LLM.

Input : Same inputs as C-GameFormer (Algorithm-1);
T_I: Text input instruction;

Output : Same output as C-GameFormer (Algorithm-1);
Output Text

$queried\_agents \leftarrow [0, 1, ..., N_{pred} - 1]$; `// Target agents, [0,1] for two agents`
$queried\_modalities \leftarrow [0, 1, \ldots, M - 1]$; `// M modalities`
$S \leftarrow$ Scene_Encoder(agents_history, map_features) `// (3-12) in Algorithm-1`
$\tilde{S} \leftarrow [\,]$
for *each* $S_{embedding}$ *in S* **do**
  $\tilde{S} \leftarrow \tilde{S} \cup$ LLM_Projection($S_{embedding}$) ; `// Projections from` $\mathbb{R}^{1xd_{scene}} \Rightarrow \mathbb{R}^{1\times d_{LLM}}$
end
emb_T$_\mathrm{I}$ $\leftarrow$ LLM_Tokenizer(T$_\mathrm{I}$) ; `// Embeddings of input text` $\Rightarrow \mathbb{R}^{N_{tokens}\times d_{LLM}}$
LLM_Input_emb $\leftarrow$ [emb_T$_\mathrm{I}$; $\tilde{S}$] ; `// concatenating text and scene embeddings`
if *Training* **then**
  hidden_states, tokens, LLM_loss $\leftarrow$ LLM(LLM_Input_emb) ; `// Autoregressive output last hidden states, corresponding tokens, and LLM cross-entropy loss`
  generation_hidden_states $\leftarrow$ select_generation_states(hidden_states) ; `// Selecting tokens that correspond to` $[I], [S_1], [S_2], \ldots [S_N]$
end
if *Inference* **then**
  while *[I] not detected* **do**
    next_token $\leftarrow$ LLM(LLM_Input_emb) ; `// Autoregressive next token generation until the first trajectory generation token [I] is found.`
    LLM_Input_emb $\leftarrow$ LLM_Input_emb $\cup$ next_token_emb ; `// Include the next token to generate the following one`
  end
  hidden_states $\leftarrow$ Masked_Generation_LLM(LLM_Input_emb) ; `// Forcing the generation of all tokens` $[I], [S_1], [S_2], \ldots [S_N]$
end
K,V $\leftarrow$ Scene_Mapper($[[S_1], [S_2], \ldots [S_N]]$) ; `// Mapping each token independently, replaces (Line 13) in Algorithm-1`
$q\_instruction \leftarrow$ Instruct_Mapper([I]) ; `// Mapping instruction token to` $q_{instruct}$`, replaces (15) in Algorithm-1`
$q_{motion} \leftarrow$ Embedding($queried\_agents, queried\_modalities$) ; `// Combined agents-modalities queries, (16-20) in Algorithm-1`
$Q \leftarrow q_{motion} + q_{instruction}$; `// Combine queries, (Line-22) in Algorithm-1`
output_features $\leftarrow$ Multimodal_Trajectory_Decoder(Q, K, V);
Pred, Scores $\leftarrow$ MLP(output_features), MLP(output_features)
NLL_loss $\leftarrow$ NLL(Pred[best_mode, :, t_{select}], ground_truth_2D)
gmm_loss $\leftarrow$ NLL_loss - CrossEntropy(Scores, best_mode)
iMotion_loss = LLM_loss + gmm_loss