Traj-LLM: A New Exploration for Empowering Trajectory Prediction with Pre-trained Large Language Models

Zhengxing Lan, Lingshan Liu, Bo Fan, Yisheng Lv, Senior Member, IEEE, Yilong Ren, Member, IEEE, Zhiyong Cui, Member, IEEE

***Abstract*—Predicting the future trajectories of dynamic traffic actors is a cornerstone task in autonomous driving. Though existing notable efforts have resulted in impressive performance improvements, a gap persists in scene cognitive and understanding of complex traffic semantics. This paper proposes Traj-LLM, the first to investigate the potential of using pre-trained Large Language Models (LLMs) without explicit prompt engineering to generate future motions from vehicular past trajectories and traffic scene semantics. Traj-LLM starts with sparse context joint encoding to dissect the agent and scene features into a form that LLMs understand. On this basis, we creatively explore LLMs' strong understanding capability to capture a spectrum of high-level scene knowledge and interactive information. To emulate the human-like lane focus cognitive function and enhance Traj-LLM's scene comprehension, we introduce lane-aware probabilistic learning powered by the Mamba module. Finally, a multimodal Laplace decoder is designed to achieve scene-compliant predictions. Extensive experiments manifest that Traj-LLM, fueled by prior knowledge and understanding prowess of LLMs, together with lane-aware probability learning, transcends the state-of-the-art methods across most evaluation metrics. Moreover, the few-shot analysis serves to substantiate Traj-LLM's performance, as even with merely 50% of the dataset, it surpasses the majority of benchmarks relying on complete data utilization. This study explores endowing the trajectory prediction task with advanced capabilities inherent in LLMs, furnishing a more universal and adaptable solution for forecasting agent movements in a new way.**

Manuscript received xxx; revised xxx; accepted xxx. This work was supported by the National Key Research and Development Project of China under Grant 2022YFB4300400, the National Natural Science Foundation of China under Grant 52202378, the Open Research Project Program of the State Key Laboratory of Internet of Things for Smart City under Grant SKL-IoTSC(UM)-2021-2023/ORP/GA08/2022, the Chunhui Collaboration Project Program of the Ministry of Education of China under Grant 202200650, and the Youth Talent Support Program of Beihang University under Grant YWF-22-L-1239. (Corresponding author: Yilong Ren, Zhiyong Cui.)

Zhengxing Lan and Lingshan Liu are with the School of Transportation Science and Engineering, Beihang University, Beijing 100191, China, and also with the State Key Lab of Intelligent Transportation System, Beijing 100191, China.

Bo Fan is with the Beijing Key Laboratory of Traffic Engineering, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China.

Yisheng Lv is with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049.

Yilong Ren and Zhiyong Cui are with the School of Transportation Science and Engineering, Beihang University, Beijing 100191, China, also with the State Key Lab of Intelligent Transportation System, Beijing 100191, China, also with the Zhongguancun Laboratory, Beijing 100194, China, and also with the Beihang Hangzhou Innovation Institute Yuhang, Hangzhou 310023, China.

***Index Terms*—Trajectory prediction, Large language models, Mamba, Autonomous vehicles.**

I. Introduction

PREDICTING the trajectories of agents primarily reasons how the future might unfold based on observed behaviors of road users. Serving as an essential bridge connecting the upstream perception system and the downstream planning system, it assumes a pivotal role in the autonomous driving realm. However, owing to the intrinsic stochasticity of agent motion, endowing autonomous vehicles (AVs) with the capacity to anticipate surrounding trajectories as proficiently as humans remains a formidable challenge.

Recent works have focused on novel deep network architectures, offering promising avenues for enhancing prediction efficacy. These endeavors, characterized by modeling temporal dependencies and spatial interactions, reframe the trajectory prediction task as a time-series forecasting problem. Most of them apply distinct modules to mine temporal dynamics along with spatial relationships, and further integrate trajectories into composite models possessing spatial and temporal components, allowing for an understanding of spatio-temporal characteristics [1]. Specifically, in encoding temporal interdependencies, a variety of classical networks, exemplified by RNN [2] and their derivatives (e.g., LSTM [3] [4], GRU [5] [6], and Transformer [7]) commonly play pivotal roles. Furthermore, in capturing spatial interactions, numerous emerging techniques such as the pooling mechanism [8] and GCN [9] [10], integrate the influential information from surrounding vehicles. Although these advancements have brought impressive performance improvements, a notable gap persists in scene cognition and understanding. The complex real driving conditions call for prediction schemes with more powerful modeling to capture a spectrum of high-level scene knowledge and interactive information, just as skilled drivers would.

The latest developments in Large Language Models (LLMs) have paved the way for addressing challenges related to trajectory prediction. These LLMs exhibit a remarkable ability to mimic human-like comprehension and reasoning, as evidenced by various tasks [11], [12]. Through exploiting this potential, researchers have extended LLMs to the autonomous driving domain, encompassing tasks such as motion planning, perception, and decision-making [13]–[15]. Notably, some

pioneering efforts have explored the application of LLMs in pedestrian trajectory prediction and lane change prediction [16]–[18]. Their approaches carefully craft tailored textual prompts, thus shifting the paradigm toward prompt-based predictions. However, designing clear, structured, and effective prompts is a non-trivial task, necessitating meticulous attention to diverse facets like query and response formulation [19]. Improper prompt engineering will produce suboptimal results because the performance of LLMs can vary significantly based on the prompt input. Indeed, another beauty of LLMs is to provide high-level connotations and extensive knowledge embedded within large pre-trained models [20]. The emergent phenomenon of LLMs has expanded their functionality beyond language processing alone. Given the difficulty of designing proper textual prompts and the emergent capacity of LLMs, an intriguing question arises: can we directly utilize LLMs to improve trajectory prediction without explicit prompt engineering?

To accomplish this objective, we put forward Traj-LLM, a novel trajectory prediction framework, aiming to investigate the feasibility of LLMs in inferring agent future trajectories devoid of explicit prompt engineering. We attempt to inject the advanced capabilities of current language models into the trajectory prediction task, providing a general and easily adaptable solution for forecasting in complex driving scenarios based on LLMs. Traj-LLM starts with sparse context joint encoding, responsible for parsing the features of agents and scenes into a form that LLMs understand. We then guide LLMs to learn wealthy high-level knowledge inherent in trajectory prediction tasks, such as scene context and social interactions. By employing the Parameter-Efficient Fine-Tuning (PEFT) technique, we cost-effectively fine-tune pre-trained LLMs to mitigate the potential differences between trajectory tokens and natural language texts. To imitate the human-like lane focus cognitive function and further enhance the scene understanding of Traj-LLM, we present lane-aware probabilistic learning, driven by the Mamba module. Formulated as selective State Space Models (SSMs), Mamba excels in refining and summarizing relevant information via input-dependent adaptations and embedding hidden state representations. This gated selection mechanism is similar to the sophisticated decision-making process of human drivers, who prudently weigh crucial lanes to inform their driving choices. Eventually, we introduce the multi-modal Laplace decoder to achieve scene-compliant predictions. Extensive experiments on the nuScenes dataset demonstrate that Traj-LLM, empowered by prior knowledge and inference ability of LLMs, along with human-like lane-aware probability learning, surpasses the state-of-the-art methods across most evaluation metrics. Furthermore, the few-shot study further validates Traj-LLM's performance, as it exceeds the majority of baselines that rely on exhaustive data utilization, despite utilizing only half of the data volume.

The major contributions of our research are summarized below:

- We propose Traj-LLM, the first trajectory prediction approach powered by pre-trained LLMs without explicit prompt engineering. By integrating the advanced capabilities of LLMs into the trajectory prediction task, we offer a versatile and easily adaptable solution for agent motion forecasting in a new way.
- We present a novel lane-aware probability learning method powered by the Mamba module, which is initially applied in the trajectory prediction realm, to emulate the humanoid lane-focus cognitive function during the driving decision-making process. It not only enhances the scene understanding skill of Traj-LLM, but also explicitly guides motion states to align with potential lane segments.
- Extensive experiments on real-world datasets consistently demonstrate the effectiveness of Traj-LLM. Experimental results manifest that Traj-LLM outperforms the state-of-the-art methods in most evaluation metrics. This performance superiority extends to both overall predictions and the capacity to handle few-shot challenges, affirming the strength of our framework in addressing complex trajectory prediction issues.

II. Related Work

A. Trajectory Prediction

Deep learning-based methodologies have propelled the predictive accuracy of trajectory prediction tasks to unprecedented heights. CS-LSTM, marking a milestone in the sphere of trajectory prediction, pioneered the incorporation of convolutional social pooling layers to augment vehicle-vehicle interdependencies [21]. In its wake, an abundant number of approaches have emerged to capture intricate spatio-temporal interactions [22], [23]. Advanced graph-based proposals, like GCN [24]–[26] and GAT [27]–[29], harness two pivotal elements, namely, vertexes and edges, to model dynamic spatio-temporal interactions between the target vehicle and its neighbors. Furthermore, Transformer has emerged as a prominent approach for addressing long sequence trajectory prediction problems, thanks to its unique attention mechanism [30], [31]. In response to data uncertainty and sample multi-modality, a series of research leverages generative models, such as GAN [32], [33], VAE [34], [35] and CVAE [36], to produce multi-modal predictions. Recently, Mamba has exhibited striking performance in long-sequence modeling, drawing widespread attention from both academia and industry [37]. Fueled by SSMs, Mamba excels at refining and summarizing information, offering a potential solution to improve trajectory prediction performance [38], [39].

Additionally, a multitude of techniques has been invented to help predict how vehicles will move via lane-based scene information. For example, goal-based models forecast achievable objectives positioned within credible lanes, subsequently constructing entire trajectories [40], [41], [42]. Anchor-based methodologies employ a predetermined collection of anchors aligned with trajectory distribution modes, which facilitates the regression of predicted multi-modal trajectories [43]–[45]. Despite impressive performance enhancements brought by these methodologies, a discernible gap remains in scene cognition and comprehension. The complex driving scene calls for prediction schemes with more powerful modeling to capture a range of scene high-level knowledge and interactive information.

B. Large Language Models

Recently, LLMs have attracted considerable attention due to their extraordinary comprehension and reasoning abilities in tackling diverse tasks, such as time-series prediction [46], [47], classification [48], few-shot learning [49], and zero-shot learning [50]. Scholars have recognized powerful reasoning and understanding skills natural in LLMs, prompting their integration into the territory of autonomous driving, where they have undergone extensive research. For instance, the combination of vectorized numeric modalities with pre-trained LLMs in LLM-Driver substantially facilitates the comprehension of complex traffic scenarios [13]. Human-centric autonomous systems based on LLMs are competent to meet user demands by inferring natural language commands, whereby rational prompt designs can enhance LLMs' performance [51]. The model of Drive as You Speak perfectly incorporated LLMs into the decision-making process of autonomous driving, enabling personalized travel experiences and humanoid decision-making, thus propelling autonomous driving towards greater innovation and efficacy [52].

Meanwhile, a few researchers have initially applied LLMs with prompts to the sphere of trajectory prediction. LG-Traj exploits LLMs to guide pedestrian trajectory prediction, integrating motion cues to increase the understanding of pedestrian behavioral dynamics, and adopts a Transformer-based architecture to capture social interactions and learn model representations [17]. LMTraj, a language-based multi-modal pedestrian trajectory predictor, treats the trajectory forecasting issue as a question-answer task, utilizing LLMs to understand scene contexts and social relationships [16]. Additionally, LC-LLM employs LLMs for explainable lane change prediction [18]. These few efforts share a focus on designing relevant prompts, shifting the paradigm towards prompt-based predictions. However, formulating clear, structured, and effective prompts is not a trivial task, which requires consideration of many elements, such as question-answer template [19]. It potentially introduces modeling bias and integration hurdles. In contrast to these pioneering efforts, we propose to explore the advanced capabilities of LLMs without explicit prompt engineering, thus offering a more general and easily adaptable solution for trajectory forecasting in complex driving scenarios.

III. Problem Formulation

The trajectory prediction problem is to infer the time-series coordinates of the target agent over a future time horizon t_f. Formally, let $\mathcal{X}_i$ denote x and y positions of agent i within a designated time horizon $\{-t_h+1, \ldots, 0\}$, where $i=0$ signifies the target vehicle and $i=1:N$ indicates its surrounding vehicles. Simultaneously, it is assumed that our model has access to the high-definition (HD) map $\mathcal{M}$ with rich scene information. Both historical trajectories and lane centerlines are structured as vectorized entities, similar to previous research paradigms [53], [54]. Specifically, for agent i, its historical trajectory $\mathcal{X}_i$ is segmented into an ordered sequence of sparse trajectory vectors $\mathcal{V}_i = \{v_i^{-t_h+2}, \ldots, v_i^0\}$, spanning historical temporal steps t_h. Each trajectory vector v_i^t comprises coordinates of the start and end points, denoted as $p_i^{t,s}$, and $p_i^{t,e}$ respectively, alongside attribute features a_i (e.g., object type and timestamps). Furthermore, to ensure the invariance of input features to the agent position, the coordinates of all vectors are normalized around the latest position of the target agent. To capture sophisticated lane information, lane centerlines abstracted by polylines are divided into predefined segments. In this way, lanes containing a variable number of vectors can be denoted as $\mathcal{M}_i^{1:L} = \{v_i^1, \ldots, v_i^L\}$, where L signifies the total vector length. Each lane vector v_i^l is annotated with sampled points $p_i^{l,s}$ and $p_i^{l,e}$, attribute characteristics a_i, and the indicator $p_i^{l,pre}$ representing the predecessor of the start point.

Given the HD map and agent states, our goal is to forecast the trajectory conditional distribution $P(\mathcal{Y}|\mathcal{X}, \mathcal{M})$ for subsequent intervals t_f, where $\mathcal{Y} = \left\{ y_0^1, y_0^2, \ldots, y_0^{t_f} \in \mathbb{R}^{t_f \times 2} \right\}$. It is hypothesized that $\mathcal{Y}$ follows Laplace distributions. In this work, we are dedicated to generating K future trajectories for the target agent and allocating a probability score for each prediction.

IV. Proposed Model

The overall architecture of Traj-LLM is illustrated in Fig. 1, including four components: sparse context joint encoding, high-level interaction modeling, lane-aware probability learning, and multi-modal Laplace decoder. Our approach stands as a pioneer in exploring the capabilities of LLMs for trajectory prediction tasks, dispensing with explicit prompt engineering. The sparse context joint encoding initially parses the features of agents and scenes into a form that LLMs understand. Subsequently, the resulting representations are fed into pre-trained LLMs to address high-level interactions. To mimic the human-like lane focus cognitive function and further enhance the scene understanding of Traj-LLM, lane-aware probability learning is presented based on the well-designed Mamba module. Finally, the multi-modal Laplace decoder is used to generate reliable predictions. Below, we describe each module of Traj-LLM in detail.

A. Sparse Context Joint Encoding

Traj-LLM first encodes the spatial-temporal scene input, such as agent states and lanes. For each of them, we employ an embedding network that consists of a GRU layer and an MLP to extract the high-dimensional features. Thereafter, the resulting tensors h_i and f_l are sent into the fusion submodule, facilitating the complex information exchange between agent states and lanes within localized regions. This process is conducted in a token embedding-like manner that matches how LLMs work.

More specifically, the fusion process entails the utilization of the multi-head self-attention mechanism ($MultiSelfAtt$) for agent-agent feature fusion, followed by Gated Linear Units [55] (GLU). Furthermore, the lane-agent and agent-lane features are fused via updating the agent and lane representations through the multi-head cross-attention mechanism

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3418522

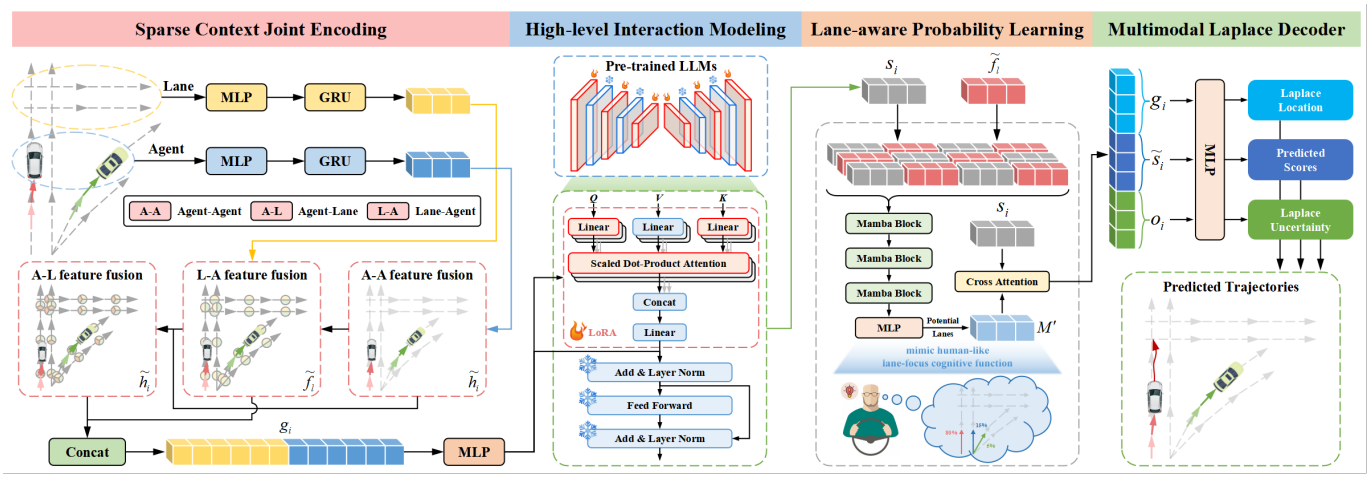

Fig. 1: The framework of Traj-LLM. Firstly, the sparse context joint encoding transforms the features of agents and scenes into a format that LLMs can understand. These representations are then input into high-level interaction modeling, which learns various dependencies through pre-trained LLMs. To emulate human-like cognitive function related to lane focus and enhance scene understanding, we introduce lane-aware probability learning, leveraging the well-designed Mamba module. Finally, the multi-modal Laplace decoder is employed to generate reliable predictions.

($MultiCrossAtt$) with skip connections. This process can be formally expressed as follows:

$$\widetilde{h}_i = MultiSelfAtt(h_i, h_i), i \in \{0, \ldots, N\}, \quad (1)$$

$$\widetilde{h}_i = GLU(h_i, \widetilde{h}_i), i \in \{0, \ldots, N\}, \quad (2)$$

$$\widetilde{f}_l = f_l + MultiCrossAtt(f_l, \widetilde{h}_i), l \in \{0, \ldots, L\}, \quad (3)$$

$$\widetilde{h}_i = \widetilde{h}_i + MultiCrossAtt(\widetilde{h}_i, \widetilde{f}_l), i \in \{0, \ldots, N\}. \quad (4)$$

Eventually, we concatenate $\widetilde{f}_l$ and $\widetilde{h}_i$ to produce sparse context joint encodings g_i, which intuitively carry dependencies related to local receptive fields among the vectorized entities. The sparse context joint encoding is designed to make LLMs understand trajectory data, thereby activating the advanced abilities of LLMs.

B. High-level Interaction Modeling

Trajectory transitions adhere to a pattern governed by high-level constraints produced by various elements in the scene. To learn these high-level interactions, we explore the capacity of LLMs to model a range of dependencies inherent in trajectory prediction tasks. Despite the similarity between trajectory data and natural language texts, it is deemed unsuitable to directly utilize LLMs to handle sparse context joint encodings, given that generic pre-trained LLMs are primarily tailored for textual data processing. One alternative proposal is to undergo a comprehensive retraining of the entire LLMs, a process that demands lots of computational resources, thus rendering it somewhat impracticable. Another more efficient solution lies in the application of PEFT to fine-tune pre-trained LLMs. By adjusting or introducing trainable parameters, PEFT showcases outstanding capability to optimize pre-trained LLMs for downstream tasks by a large margin [56], [57].

In this research, we utilize parameters from NLP pre-trained transformer architectures, particularly focusing on the GPT2 model [58] as shown in Fig. 2, for high-level interaction modeling. We opt to freeze all pre-trained parameters and inject new trainable ones by implementing the Low-Rank Adaptation (LoRA) technique [59]. LoRA is applied to the attention layers in LLMs. Let denote the rank of LoRA as r, the inputs of j-th attention layer θ_j as a_j with size d, and the output as $\widetilde{a_j}$ with size k. For a pre-trained weight matrix $W \in \mathbb{R}^{d \times k}$ in the network θ_j, LoRA approximately updates it using:

$$\widetilde{W} \approx W + BA, \quad (5)$$

where the rank decomposition matrix $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, with $r \ll min(d, k)$. During the training process, $W \in \mathbb{R}^{d \times k}$ keeps frozen, while B and A are treated as trainable parameters and initialized to zero and Gaussian distributions, respectively. Therefore, the forward transfer function of LoRA can be concisely expressed as:

$$\widetilde{a_j} = W \cdot a_j + BA \cdot a_j. \quad (6)$$

On this basis, we pass the given sparse context joint encodings g_i into pre-trained LLMs, which own a series of pre-trained transformer blocks equipped with LoRA. This procedure ends up with the generation of high-level interaction representations z_i:

$$z_i = LLMs(g_i). \quad (7)$$

After being processed by pre-trained LLMs, the output representations z_i are transformed via an MLP layer to match the dimensions of g_i, thus yielding the ultimate high-level interaction states s_i.

C. Lane-aware Probability Learning

The overwhelming majority of experienced drivers will directly select potential lane segments based on their driving experience, rather than traversing all cues indiscriminately. To

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3418522

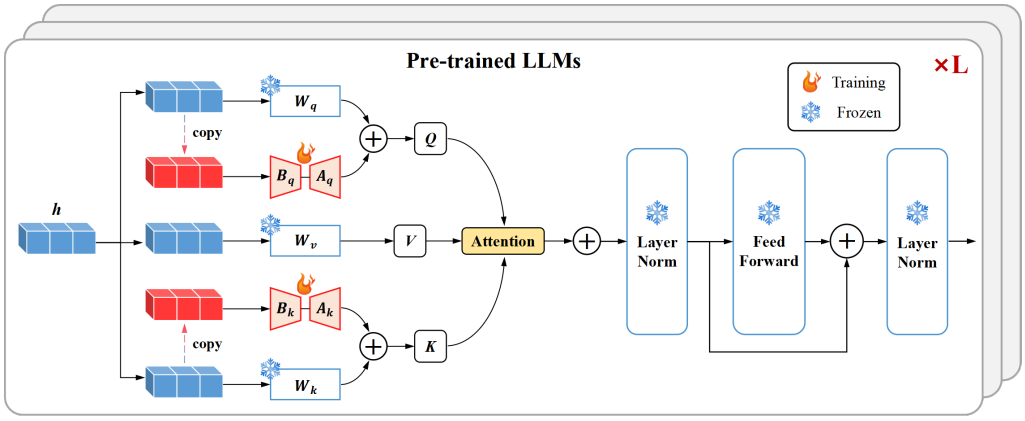

Fig. 2: The overview of Pre-trained LLMs. All pre-trained parameters of LLMs are frozen, and the multi-head attention layers are injected with new trainable parameters by implementing the PEFT technique.

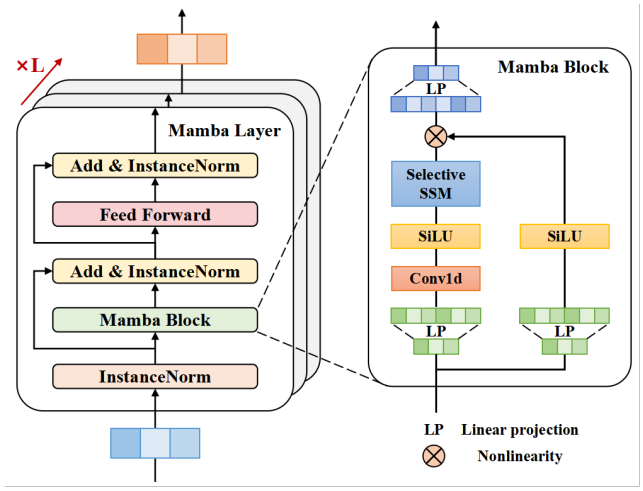

Fig. 3: The proposed Mamba layer for lane-aware probability learning, involves a Mamba block, three layers of normalization, a position-wise feed-forward network, and residual connections. The Mamba block, as structured SSMs, excels at selectively distilling and summarizing scene features.

mimic this human-like cognitive function and further enhance the scene understanding of Traj-LLM, we employ lane-aware probabilistic learning to continuously estimate the likelihood of motion states aligning with lane segments, as shown in Algorithm 1. Precisely, we synchronize the target agent's motion with lane information at each time step $t \in \{1, \ldots, t_f\}$ with the introduced Mamba layer. Functioning as the SSMs, Mamba excels in refining and summarizing relevant information via input-dependent adaptation and embedding hidden state representations. This gated selection mechanism is analogous to the sophisticated decision-making process of human drivers, which is employed to advance lane-aware probability learning.

Fig. 3 depicts the Mamba layer, which involves a Mamba block, three layers of normalization, a position-wise feed-forward network, and residual connections. The Mamba layer first normalizes the input $F = concat[s_i, \widetilde{f}_l] \in \mathbb{R}^{B\times L\times D}$, whose dimensions are determined by the batch size B, length of lane segments L, and hidden dimension D. Then the Mamba block maps the normalized $F \in \mathbb{R}^{B\times L\times D}$ to the output $Q \in \mathbb{R}^{B\times L\times D}$:

$$Q = Mamba(InstanceNorm(F)). \tag{8}$$

Concretely, as detailed in Algorithm 1, the Mamba block first expands the input dimension with dilatation coefficient E through linear projections, yielding distinct representations for two parallel processing branches, designated as m and n. Subsequently, one branch undergoes a 1D convolution and a SiLU activation [60], to capture lane-aware dependencies m'. The core of the Mamba block involves the selective SSMs with parameters discretized based on the input. Thereafter, m' is linearly projected to B, C, Δ, respectively. Δ is then used to transform $\overline{A}, \overline{B}$. The SSMs accept m' and $\overline{A}, \overline{B}, C$ as input, generating refined lane-aware features q. Simultaneously, the other branch introduces a SiLU activation to produce gating signals n', intended for filtering irrelevant information. Finally, q is multiplied with n', followed by a linear projection to deliver the ultimate output Q.

To enhance robustness, we further incorporate the instance normalization and residual connection to get implicit states $\widetilde{Q}$:

$$\widetilde{Q} = InstanceNorm(Dropout(Q)) + F. \tag{9}$$

Subsequently, we leverage a position-wise feed-forward network to improve the modeling of lane-aware estimation in the hidden dimension:

$$\widetilde{Q'} = ReLU(\widetilde{Q}W^{(0)} + b^{(0)})W^{(1)} + b^{(1)}, \tag{10}$$

where $W^{(0)}, W^{(1)}, b^{(0)}, b^{(1)}$ are trainable parameters. Once again, the instance normalization and residual connection are executed to acquire the lane-aware learning vectors S:

$$S = InstanceNorm(Dropout(\widetilde{Q'})) + \widetilde{Q}, \tag{11}$$

which are then fed into an MLP layer, resulting in the predicted score of the l-th lane segment at t:

$$p_{l,t} = \frac{\exp(MLP(S_l))}{\sum_{j=1}^{L}\exp(MLP(S_j))}. \tag{12}$$

As mentioned above, skilled drivers exhibit a keen focus on multiple potential lane segments to facilitate effective decision-making. To this end, we carefully curate the top c lane segments $\{v_1, v_2, \ldots, v_c\}$ with the top c scores $\{p_1, p_2, \ldots, p_c\}$ as the candidate lane segments, which are further concatenated to form $\mathcal{M}'$.

The lane-aware probability learning is modeled as a classification problem, wherein a binary cross-entropy loss $\mathcal{L}_{lane}$ is applied to optimize the probability estimation:

$$\mathcal{L}_{lane} = \sum_{t=1}^{t_f} \mathcal{L}_{CE}(p_t, \widetilde{p_t}). \tag{13}$$

Here, the ground truth score $\widetilde{p_t}$ is set to 1 for the lane segment closest to the truth trajectory position, while 0 for others. The distance $d(l, y_0)$ between a lane segment l and the ground truth position is discerned through Euclidean distance calculation:

$$d(l, y_0) = ||l - y_0||^2. \tag{14}$$

Algorithm 1: Lane-aware probability learning

Input: $F : (B, L, D)$
Output: $\mathcal{M}' : \{v_1, v_2, \ldots, v_c\}$
$m : (B, L, ED) \leftarrow \text{Linear}^m(F)\{\text{Linear projection}\}$;
$n : (B, L, ED) \leftarrow \text{Linear}^n(F)\{\text{Linear projection}\}$;
$m' : (B, L, ED) \leftarrow \text{SiLU}(\text{Conv1d}(m))$;
$A : (D, D') \leftarrow \text{Parameter}\{\text{Structured state matrix}\}$;
$B : (B, L, D') \leftarrow \text{Linear}^B(m')\{\text{Linear projection}\}$;
$C : (B, L, D') \leftarrow \text{Linear}^C(m')\{\text{Linear projection}\}$;
$\Delta : (B, L, D) \leftarrow$
$\text{Softplus}(\text{Parameter} + \text{Broadcast}(\text{Linear}^{\Delta}(m')))$;
$\overline{A}, \overline{B} : (B, L, D, D') \leftarrow \text{discretize}(\Delta, A, B)$;
$q : (B, L, ED) \leftarrow \text{Selective SSMs}(\overline{A}, \overline{B}, C)(m')$;
$n' : (B, L, ED) \leftarrow \text{SiLU}(n)$;
$Q : (B, L, D) \leftarrow \text{Linear}(q \otimes n')\{\text{Linear projection}\}$;
$\widetilde{Q} : (B, L, D) \leftarrow \text{InstanceNorm}(\text{Dropout}(Q)) + F$;
$\widetilde{Q'} : (B, L, D) \leftarrow \text{ReLU}(\widetilde{Q}W^{(0)} + b^{(0)})W^{(1)} + b^{(1)}$;
$S : (B, L, D) \leftarrow \text{InstanceNorm}(\text{Dropout}(\widetilde{Q'})) + \widetilde{Q}$;
$p_{l,t} : (B, L) \leftarrow \frac{\exp(MLP(S_l))}{\sum_{j=1}^{L} \exp(MLP(S_j))}$;
$\mathcal{M}' \leftarrow \text{choose}\{v_1, v_2, \ldots, v_c\}$;
return $\mathcal{M}'$

D. Multi-modal Laplace Decoder

The anticipated movements of traffic agents are inherently multi-modal. Hence, we adopt a mixture model framework to parameterize the prediction distribution, where each mixture component follows a Laplace distribution, in line with established methodologies [44], [61]. For each predicted instance, the multi-modal Laplace decoder takes the representations e_i as inputs and outputs a set of trajectories $\sum_{k=1}^{K} \pi_{i,k} \prod_{t=1}^{t_f} \text{Laplace}(\mu_i^t, b_i^t)$. Here, $\{\pi_{i,k}\}_{k=1}^{K}$ are the mixing coefficients, and the Laplace density of k-th mixture component is parameterized by the location $\mu_i^t \in \mathbb{R}^2$ and its associated uncertainty $b_i^t \in \mathbb{R}^2$. The representations e_i are made of the sparse context joint encodings g_i, the lane-aware guided high-level interaction feature $\widetilde{s_i}$, together with a latent vector o sampled by a multivariate normal distribution. $\widetilde{s_i}$ is generated via cross-attention between s_i and $\mathcal{M}'$, to guide the target agent towards candidate lane segments like a skilled driver. To predict mixing coefficients, an MLP followed by a softmax function is employed, while two side-by-side MLPs are utilized to generate μ_i^t and b_i^t. Then, we use a regression loss and a classification loss to train the multi-modal Laplacian decoder. The regression loss is computed using the Winner-Takes-All strategy [54], [61], defined as:

$$\mathcal{L}_{reg} = -\frac{1}{t_f} \sum_{t=1}^{t_f} \log P\left(y_0^t | \mu_{i,k^*}^t, b_{i,k^*}^t\right), \tag{15}$$

where y_0^t represents the ground truth position, and k^* signifies the mode with the minimum L_2 error among K predictions. On the other hand, the cross-entropy loss is adapted as the classification loss $\mathcal{L}_{cls}$ to adjust the mixing coefficients.

Consequently, the overall loss function $\mathcal{L}$ of Traj-LLM can be written as:

$$\mathcal{L} = \lambda \mathcal{L}_{lane} + \mathcal{L}_{reg} + \mathcal{L}_{cls}, \tag{16}$$

where λ serves as a hyperparameter, controlling the relative importance of $\mathcal{L}_{lane}$.

V. Experiment

This section describes a series of comprehensive experiments aimed at proving the effectiveness of Traj-LLM. We commenced by showing the experimental setup, followed by a detailed comparison of results with various cutting-edge approaches. To ascertain the impact of each module, we conducted ablation experiments on Traj-LLM architectures. We then compared various representative large language model backbones to gain insights into selecting suitable pre-trained LLMs for trajectory prediction. Furthermore, we embarked on the few-short study to show the generalization capability of Traj-LLM. Finally, we performed a qualitative analysis of predictions.

A. Experimental Setup

Evaluation Datasets. The assessment of Traj-LLM is conducted on a widely used benchmark, the nuScenes dataset [62], for trajectory prediction. This dataset covers 1,000 driving scenarios spanning different cities (e.g., Boston and Singapore), which were collected via vehicle-mounted cameras and lidar sensors, providing an exhaustive depiction of urban traffic dynamics. Adhering to the official predictions scheme, Traj-LLM adopts 2-second segments of sequences to forecast subsequent 6-second trajectories. Furthermore, the predicted modality K is set to 5 and 10, which also aligns harmoniously with the recommendation of the nuScenes dataset.

Evaluation Metrics. We evaluate our model using standard metrics for motion prediction, including minimum Average Displacement Error (minADE), minimum Final Displacement Error (minFDE), and Miss Rate (MR). minADE calculates the

TABLE I: The comparison results of the proposed Traj-LLM and state-of-the-art methods for $K = 5$. The best/second best values are highlighted in boldface/underlined.

Method	minADE_5	minFDE_5	MR_5
Trajectron++ [63]	1.88	-	0.70
ALAN [64]	1.87	3.54	0.60
GATraj [65]	1.87	4.08	-
SG-Net [66]	1.85	3.87	-
WIMP [67]	1.84	-	0.55
MHA-JAM [68]	1.81	3.72	0.59
AgentFormer [69]	1.59	3.14	-
LaPred [70]	1.47	-	0.53
P2T [71]	1.45	-	0.64
GOHOME [72]	1.42	-	0.57
CASPNet [73]	1.41	-	0.60
MUSE-VAE [74]	1.38	<u>2.90</u>	-
Autobot [75]	1.37	-	0.62
THOMAS [76]	1.33	-	0.55
HLSTrajForecast [77]	1.33	2.92	-
PGP [78]	1.27	-	0.52
LAformer [54]	<u>1.19</u>	-	<u>0.48</u>
FRM [79]	**1.18**	-	<u>0.48</u>
Traj-LLM	1.24	**2.46**	**0.41**

TABLE II: The comparison result of the proposed Traj-LLM and state-of-art methods for $K = 10$. The best/second best values are highlighted in boldface/underlined.

Method	minADE_{10}	minFDE_{10}	MR_{10}
Trajectron++ [63]	1.51	-	0.57
GATraj [65]	1.46	2.97	-
SG-Net [66]	1.32	2.50	-
AgentFormer [69]	1.31	2.48	-
MHA-JAM [68]	1.24	2.21	0.45
ALAN [64]	1.22	<u>1.87</u>	0.49
CASPNet [73]	1.19	-	0.43
P2T [71]	1.16	-	0.46
GOHOME [72]	1.15	-	0.47
LaPred [70]	1.12	-	0.46
WIMP [67]	1.11	-	0.43
MUSE-VAE [74]	1.09	2.10	-
THOMAS [76]	1.04	-	0.42
HLSTrajForecast [77]	1.04	2.15	-
Autobot [75]	1.03	-	0.44
PGP [78]	0.94	-	0.34
LAformer [54]	<u>0.93</u>	-	0.33
FRM [79]	**0.88**	-	<u>0.30</u>
Traj-LLM	0.99	**1.73**	**0.23**

average ℓ_2 distance between the optimal predicted trajectory and its corresponding ground truth, while minFDE describes their space interval at the terminal position. And MR indicates the proportion of instances, whose predicted endpoints are farther than 2.0 meters of real ones. In all cases, reduced values signify improved model performance.

Implementation Details. Our model is trained on 6 NVIDIA GeForce RTX4090 GPUs with Adam optimizer, whose batch size and initial learning rate are set to 132 and 0.001, respectively. The architecture of Traj-LLM is carefully devised, beginning with a layer of sparse context joint encoding module, followed by a high-level interaction modeling module, and ultimately incorporating three layers of lane-aware probability learning module. The hidden dimensions of all feature vectors are uniformly configured to 128.

B. Results and Computational Performance

Comparison with State-of-the-art Methods. We benchmarked Traj-LLM against numerous state-of-the-art models with diverse architectural paradigms. The quantitative results are illustrated in Table I for $K = 5$ and Table II for $K = 10$. It is discernible that Traj-LLM displays excellent performance, thanks to the advanced capabilities of LLMs and well-designed lane-aware probability learning. In particular, when assessing the metrics minFDE_5 and MR_5, Traj-LLM demonstrates substantial improvements over MUSE-VAE and LAformer, elevating performance by 15.17% and 14.58%, respectively. Similarly, Traj-LLM exhibits notable enhancements in predictive accuracy on the metrics minFDE_{10} and MR_{10}, surpassing the suboptimal approaches, ALAN and FRM, by a large margin of 7.49% and 23.33%, respectively. Furthermore, GOHOME stands out as the rasterized method in the table, yet it conspicuously lags behind most vectorized counterparts in performance metrics. Compared with other lane-based methodologies like LaPred, Traj-LLM maintains a clear superiority, underscoring the effectiveness of our lane-aware probability learning in mimicking human-like lane-focus cognitive function and enhancing scene understanding. It is worth highlighting that in contrast to models based on GCN to extract interaction dependencies, such as GATraj, Traj-LLM reduces $\text{minADE}_5/\text{minFDE}_5$ $(\text{minADE}_{10}/\text{minFDE}_{10})$ from 1.87/4.08 (1.46/2.97) to 1.24/2.46 (0.99/1.73). It indicates the pivotal role of pre-trained LLMs in facilitating complex feature comprehension and augmenting predictive accuracy. The findings advocate the integration of LLMs into trajectory prediction tasks, an emerging domain for further exploration.

Computational Performance. To ascertain the efficiency of Traj-LLM, we carried out a comparative analysis of model parameters and inference speed, as shown in Table III and Table IV. The results demonstrate that rasterized methods like P2T exhibit slower inference speed compared to vectorized methods such as PGP, LAformer, and Traj-LLM. This discrepancy arises because rasterized methods process large, dense pixel images, which demand significant computational resources. In contrast, vectorized methods handle simpler vectorized points, resulting in faster inference speed. Our Traj-LLM, being a vectorized method, therefore outperforms rasterized methods like P2T in terms of speed. Although Traj-LLM, PGP, and LAformer all belong to the vectorized category, they differ in algorithms and the number of model parameters. PGP, a GCN-based model, is smaller and faster than Transformer-based LAformer. Despite having more parameters, Traj-LLM achieves an inference time of approximately 98ms, in a

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3418522

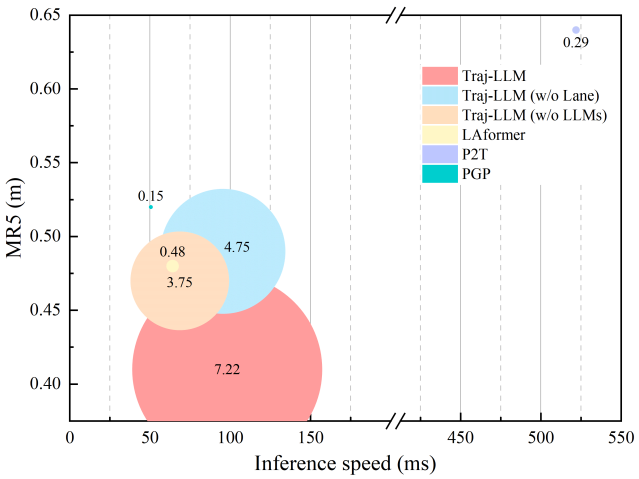

Fig. 4: Comparison of Traj-LLM with baseline models across three key metrics: trainable parameters, inference speed, and MR_5. The circle size in the figure corresponds to the trainable parameter number in each model.

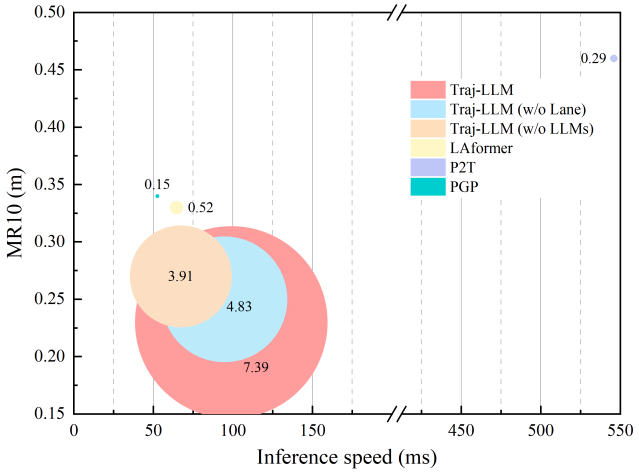

Fig. 5: Comparison of Traj-LLM with baseline models across three key metrics: trainable parameters, inference speed, and MR_{10}. The circle size in the figure corresponds to the trainable parameter number in each model.

scenario featuring an average of 12 agents. While this is not the primary strength compared to PGP and LAformer, it remains competitive and better than the 10Hz (100ms) threshold typically required for real-world deployments. Fig. 4 and Fig. 5 further emphasize the prominent equilibrium between prediction accuracy and inference efficiency of our methodology.

C. Ablation Studies

We undertook thorough ablation experiments to evaluate the contribution of each component in our proposed method. These experiments entail designing and assessing several variants of Traj-LLM to check the impact of each modification on the overall performance of our model:

1) w/o LLMs: This variant directly removes the entire high-level interaction module, allowing the model to make predictions independent of LLMs.
2) w/o LoRA: This variant retains the high-level interaction module, but employs LLMs void of LoRA, intended to explore the function of PEFT technique.
3) w/o Laplace: This variant assumes that the trajectory follows Gaussian distributions rather than Laplace distributions.
4) w/o Lane: This variant eliminates the lane-aware learning module, thus ignoring the humanoid decision-making process of selecting multiple candidate lanes to guide motions.

TABLE III: The results of the computational performance for $K = 5$.

Model	Params	Batch size	Inference speed
PGP [78]	0.15M	12	50.49ms
P2T [71]	0.29M	12	521.90ms
LAformer [54]	0.48M	12	64.03ms
Traj-LLM (w/o Lane)[1]	129.19M	12	95.45ms
Traj-LLM (w/o LLMs)[2]	3.75M	12	68.54ms
Traj-LLM	131.66M	12	98.07ms

[1] Traj-LLM (w/o Lane): the lane-aware probability learning is removed.
[2] Traj-LLM (w/o LLMs): the high-level interaction module is removed.

TABLE IV: The results of the computational performance for $K = 10$.

Model	Params	Batch size	Inference speed
PGP [78]	0.15M	12	52.50ms
P2T [71]	0.29M	12	545.99ms
LAformer [54]	0.52M	12	64.54ms
Traj-LLM (w/o Lane)[1]	129.27M	12	94.60ms
Traj-LLM (w/o LLMs)[2]	3.91M	12	67.27ms
Traj-LLM	131.83M	12	99.00ms

[1] Traj-LLM (w/o Lane): the lane-aware probability learning is removed.
[2] Traj-LLM (w/o LLMs): the high-level interaction module is removed.

The ablation results, shown in Table V and Table VI, yield enlightening insights. Firstly, the variant devoid of LLMs exhibits a weaker performance, manifesting a significant 7.52% and 9.42% reduction in accuracy for minFDE_5 and minFDE_{10}, separately, under both $K = 5$ and $K = 10$ conditions. This confirms the potential capability of LLMs in boosting trajectory prediction tasks, without explicit prompt engineering. Secondly, it becomes evident that fine-tuning LLMs is necessary to effectively transfer cross-modal inference capability to comprehension of complex traffic scenes. In addition, modeling trajectories with Laplace distributions proves to be more effective than using Gaussian distributions, as evidenced by the results. Last but not least, the humanoid lane-selection mechanism tremendously elevates goal-directed agent navigation accuracy, as reflected by the MR metric. Specifically, the variant without the lane-aware learning module displays a remarkable 12.77% and 14.81% accuracy decrease in MR_5 and MR_{10}, respectively, compared to Traj-LLM. Experimental results bear out that LLMs and the human-like lane-aware learning mechanism play indispensable roles in improving model performance.

TABLE V: The results of the Traj-LLM and its variants in the ablation experiment for $K = 5$.

Model	LLM	LoRA	Lane	Laplace	minADE	minFDE	MR
w/o LLMs	-	✓	✓	✓	1.30	2.66	0.47
w/o LoRA	✓	-	✓	✓	1.27	2.54	0.43
w/o Lane	✓	✓	-	✓	1.32	2.64	0.49
w/o Laplace	✓	✓	✓	-	1.43	2.87	0.53
Traj-LLM	✓	✓	✓	✓	1.24	2.46	0.41

TABLE VI: The results of the Traj-LLM and its variants in the ablation experiment for $K = 10$.

Model	LLM	LoRA	Lane	Laplace	minADE	minFDE	MR
w/o LLMs	-	✓	✓	✓	1.05	1.91	0.27
w/o LoRA	✓	-	✓	✓	1.02	1.82	0.26
w/o Lane	✓	✓	-	✓	1.01	1.75	0.25
w/o Laplace	✓	✓	✓	-	1.06	1.93	0.28
Traj-LLM	✓	✓	✓	✓	0.99	1.73	0.23

D. Large Model Variants

Traj-LLM offers a general and adaptable solution for trajectory prediction, allowing for the easy incorporation of different LLMs backbones. In this section, we compare three representative LLMs backbones with different capacities, as shown in Table VII and Table VIII. The results indicate that the prediction model utilizing GPT-2 outperforms the one using BERT in evaluation metrics. The slight inferiority of the predictive model utilizing Llama-2 as the backbone, in comparison to the model employing GPT-2, can plausibly be attributed to the disparity in their architectural design and model complexity. Llama-2 has a more complex structure and a larger number of parameters than GPT-2. While this generally enhances the model's capabilities, it renders fine-tuning more challenging. The increased complexity potentially hinders the efficient transfer of learned knowledge to the specific trajectory prediction task. Combined with the ablation studies, it is evident that all three backbones of LLMs enhance prediction performance compared to the model without LLMs. However, the large number of model parameters in Llama-2 results in the prediction model having an inference speed that is six times slower than GPT-2. Since BERT and GPT-2 have the same number of transformer blocks, they exhibit similar inference speed.

TABLE VII: The comparison results of Traj-LLM with different representative LLMs for $K = 5$.

Model	minADE	minFDE	MR	Params	Inference speed
w BERT	1.28	2.55	0.44	116.74M	95.83ms
w Llama-2	1.25	2.50	0.42	6649.17M	665.87ms
Traj-LLM	1.24	2.46	0.41	131.66M	98.07ms

E. Few-shot Study

In this section, we conduct a few-shot study to evaluate the performance of Traj-LLM with a limited amount of data.

TABLE VIII: The comparison results of Traj-LLM with different representative LLMs for $K = 10$.

Model	minADE	minFDE	MR	Params	Inference speed
w BERT	1.01	1.79	0.25	116.90M	96.65ms
w Llama-2	0.99	1.74	0.24	6649.34M	674.75ms
Traj-LLM	0.99	1.73	0.23	131.83M	99.00ms

TABLE IX: The results of the few-shot study for $K = 5$.

Few-shot ratio	minADE	minFDE	MR
10%	3.26	10.07	0.72
20%	1.69	4.47	0.58
30%	1.42	3.18	0.51
40%	1.36	2.94	0.49
50%	1.30	2.69	0.44

In these few-shot trials, we maintained consistent validation sets in line with the full-sample experiments, deliberately limiting the percentage of training data. Table IX and Table X present the outcomes of using only 10% to 50% of the training data. In addition, Fig. 6 and Fig. 7 illustrate various metrics across training data percentages ranging from 10% to 20%. Noteworthy is the revelation that Traj-LLM, even with only half of the data, outperforms a majority of baselines reliant on full data utilization across all evaluated metrics. Thanks to the innate representation learning capability of LLMs and well-designed lane-aware probability learning, Traj-LLM consistently keeps commendable performance even with extremely sparse data (i.e., 10% to 30% training data). These compelling revelations stress Traj-LLM's remarkable capacity for generalization and adaptation, even when confronted with insufficient samples.

F. Qualitative Analysis

1) Visualization of motion prediction under full-sample training: Illustrated in Fig. 8, are the visualizations for predicted trajectories across different driving scenes with exhaustive training dataset, enabling intuitive analysis of trajectory precision and diversity. Fig.8 (a)-(c) exhibit the instances with prediction modalities $K = 5$, and Fig.8 (d)-(f) show the instances under the conditions of $K = 10$. From these visual insights, we observe that whether the target agent maintains a straight path or navigates intersections for left or right turns, the ground truth closely corresponds with one of the multi-modal trajectories predicted by Traj-LLM in terms of trends. Furthermore, we find that the trajectories generated by Traj-LLM display sound rationality and compliance, as they do not

TABLE X: The results of the few-shot study for $K = 10$.

Few-shot ratio	minADE	minFDE	MR
10%	3.02	9.65	0.62
20%	1.39	3.54	0.39
30%	1.12	2.26	0.31
40%	1.05	1.99	0.27
50%	1.02	1.85	0.25

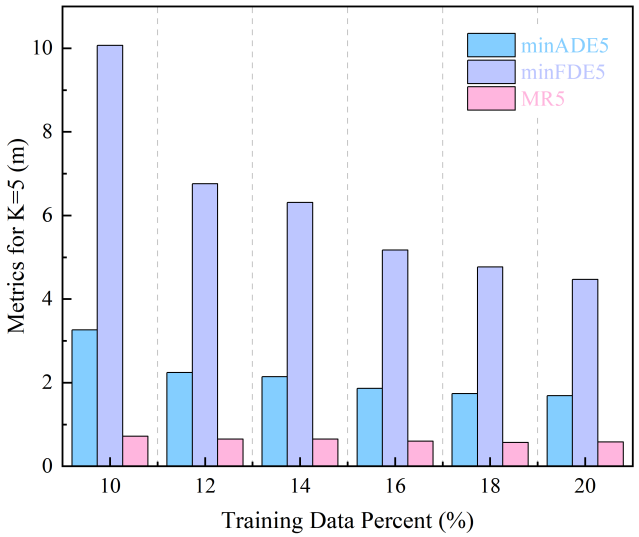

Fig. 6: The results of the few-shot study for various metrics across training data percentages ranging from 10% to 20% under $K = 5$.

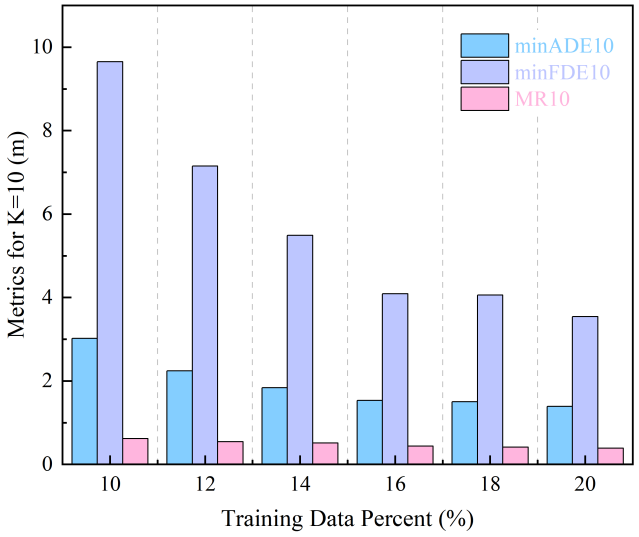

Fig. 7: The results of the few-shot study for various metrics across training data percentages ranging from 10% to 20% under $K = 10$.

exceed the boundaries of the roadway. These comprehensive predictions concerning both lateral and longitudinal vehicular behaviors robustly prove the strong predictive power of Traj-LLM.

2) Visualization of Motion Prediction under few-shot settings with 50% data samples: Fig. 9 showcases qualitative results of Traj-LLM in few-shot scenarios with $K = 5$ and $K = 10$. Despite being trained with only half of the data, Traj-LLM consistently produces plausible predictions for the target agent across diverse scenarios, including intersections and straight driving occasions. Furthermore, Traj-LLM accurately captures detailed behaviors such as acceleration and deceleration. The uncertain feature of agent motions is also depicted with robust multi-modal predictions. These few-shot predictions further underscore the potent predictive capability of Traj-LLM, which takes full advantage of the exceptional scene understanding capacity of LLMs and well-crafted lane-aware probability learning.

VI. Conclusions

In this study, we propose Traj-LLM, a novel trajectory prediction framework, which seeks to investigate the feasibility of LLMs in inferring agent future trajectories without explicit prompt engineering. Traj-LLM starts with sparse context joint encoding, responsible for parsing the features of agents and scenes into a form that LLMs understand. We then guide LLMs to learn a spectrum of high-level knowledge inherent in trajectory prediction tasks, such as scene contexts and social interactions. For the purpose of imitating the human-like lane focus cognitive function and further enhancing the scene understanding of Traj-LLM, we propose lane-aware probabilistic learning powered by the Mamba module, which is first utilized in the trajectory prediction domain. To achieve scene-compliant heterogeneous predictions, this study introduces the multi-modal Laplace decoder. Extensive experiments demonstrate that Traj-LLM outperforms the state-of-the-art methods in most metrics, and the few-shot study further validates the outstanding capabilities of Traj-LLM. Our study underscores the potential prowess of LLMs in reshaping the trajectory prediction task.

References

[1] Y. Wu, L. Wang, S. Zhou, J. Duan, G. Hua, and W. Tang, "Multi-stream representation learning for pedestrian trajectory prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 2875–2882.

[2] X. Mo, Y. Xing, and C. Lv, "Graph and recurrent neural network-based vehicle trajectory prediction for highway driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1934–1939.

[3] P. Cong, Y. Xiao, X. Wan, M. Deng, J. Li, and X. Zhang, "Dacr-amtp: adaptive multi-modal vehicle trajectory prediction for dynamic drivable areas based on collision risk," *IEEE Transactions on Intelligent Vehicles*, 2023.

[4] H. Liao, Z. Li, H. Shen, W. Zeng, D. Liao, G. Li, and C. Xu, "Bat: Behavior-aware human-like trajectory prediction for autonomous driving," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 9, 2024, pp. 10 332–10 340.

[5] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 654–17 665, 2022.

[6] H. Liao, C. Wang, Z. Li, Y. Li, B. Wang, G. Li, and C. Xu, "Physics-informed trajectory prediction for autonomous driving under missing observation," *Available at SSRN 4809575*, 2024.

[7] M. Geng, Y. Chen, Y. Xia, and X. M. Chen, "Dynamic-learning spatial-temporal transformer network for vehicular trajectory prediction at urban intersections," *Transportation research part C: emerging technologies*, vol. 156, p. 104330, 2023.

[8] Z. Zuo, X. Wang, S. Guo, Z. Liu, Z. Li, and Y. Wang, "Trajectory prediction network of autonomous vehicles with fusion of historical interactive features," *IEEE Transactions on Intelligent Vehicles*, 2023.

[9] Y. Ren, Z. Lan, L. Liu, and H. Yu, "Emsin: Enhanced multi-stream interaction network for vehicle trajectory prediction," *IEEE Transactions on Fuzzy Systems*, 2024.

[10] Y. Han, Q. Liu, H. Liu, B. Wang, Z. Zang, and H. Chen, "Tp-frl: An efficient and adaptive trajectory prediction method based on the rule and learning-based frameworks fusion," *IEEE Transactions on Intelligent Vehicles*, 2023.

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3418522

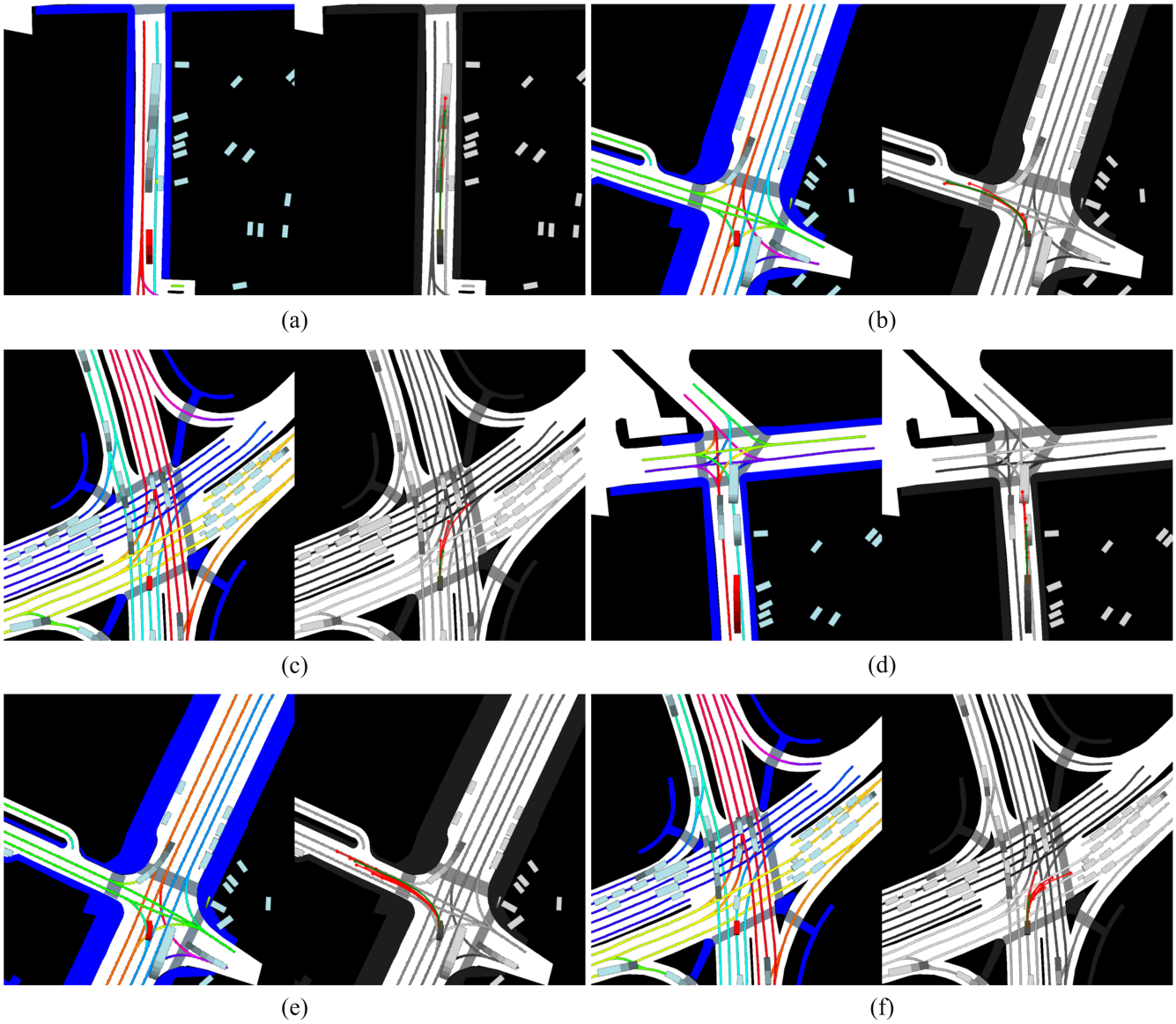

Fig. 8: Qualitative results of Traj-LLM on various scenarios under full-sample training. In each subfigure, left: HD map, right: predictions and ground truth. Ground truth future trajectory is depicted in green lines, the predicted trajectories are in red lines. Fig. (a)-(c) exhibit the instances with prediction modalities $K = 5$, and Fig. (d)-(f) show the instances with prediction modalities $K = 10$.

[11] Y. Liu, Y. Wan, L. He, H. Peng, and S. Y. Philip, "Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 7, 2021, pp. 6418–6425.

[12] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback, 2022," *URL https://arxiv.org/abs/2203.02155*, vol. 13, p. 1, 2022.

[13] L. Chen, O. Sinavski, J. Hünermann, A. Karnsund, A. J. Willmott, D. Birch, D. Maund, and J. Shotton, "Driving with llms: Fusing object-level vector modality for explainable autonomous driving," *arXiv preprint arXiv:2310.01957*, 2023.

[14] D. Wu, W. Han, T. Wang, Y. Liu, X. Zhang, and J. Shen, "Language prompt for autonomous driving," *arXiv preprint arXiv:2309.04379*, 2023.

[15] L. Wen, D. Fu, X. Li, X. Cai, T. Ma, P. Cai, M. Dou, B. Shi, L. He, and Y. Qiao, "Dilu: A knowledge-driven approach to autonomous driving with large language models," *arXiv preprint arXiv:2309.16292*, 2023.

[16] I. Bae, J. Lee, and H.-G. Jeon, "Can language beat numerical regression? language-based multimodal trajectory prediction," *arXiv preprint arXiv:2403.18447*, 2024.

[17] P. S. Chib and P. Singh, "Lg-traj: Llm guided pedestrian trajectory prediction," *arXiv preprint arXiv:2403.08032*, 2024.

[18] M. Peng, X. Guo, X. Chen, M. Zhu, K. Chen, X. Wang, Y. Wang *et al.*, "Lc-llm: Explainable lane-change intention and trajectory predictions with large language models," *arXiv preprint arXiv:2403.18344*, 2024.

[19] H. Xue and F. D. Salim, "Promptcast: A new prompt-based learning paradigm for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[20] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[21] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1468–1476.

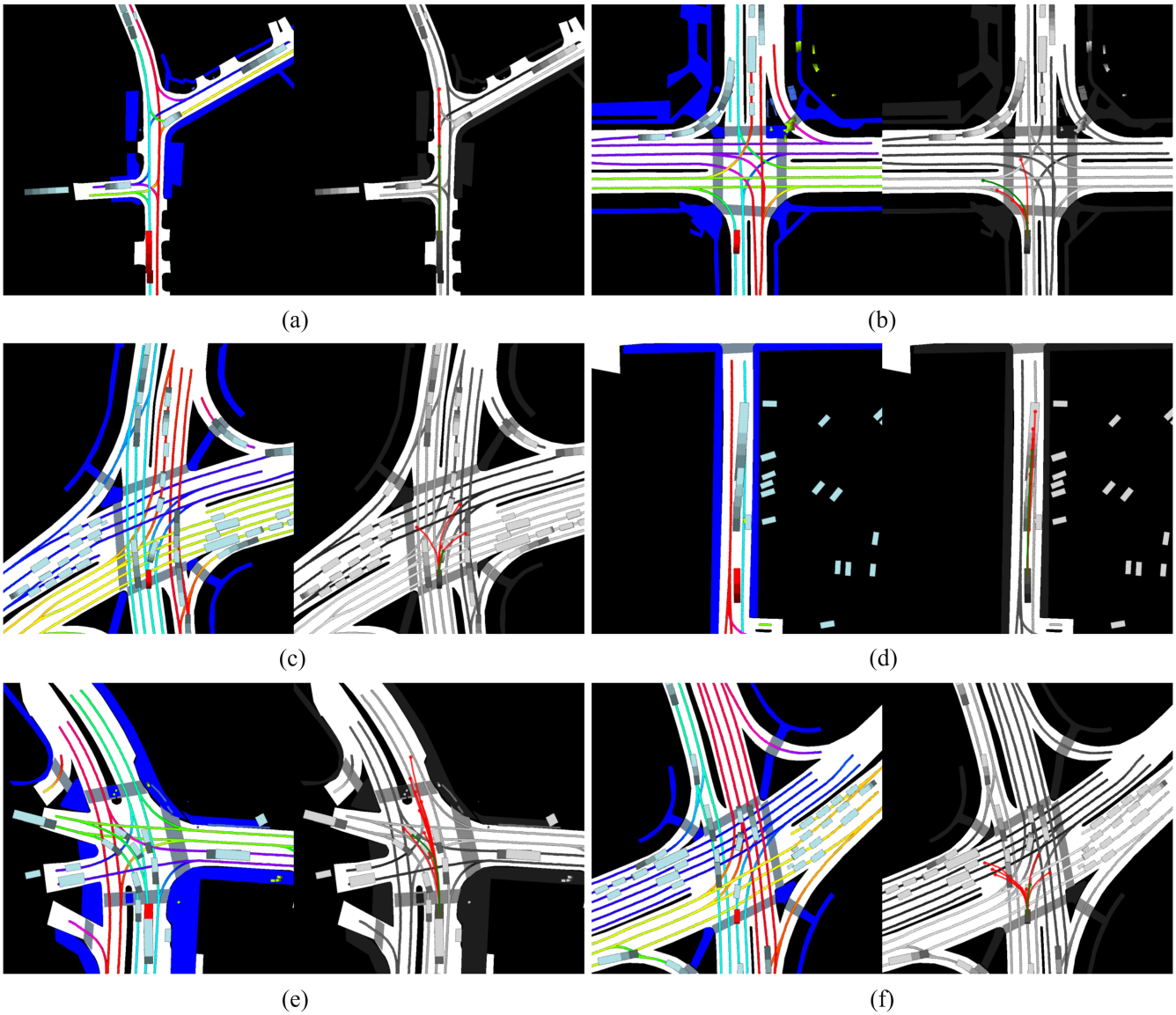

Fig. 9: Qualitative results of Traj-LLM on various scenarios under few-shot settings with 50% data samples. In each subfigure, left: HD map, right: predictions and ground truth. Ground truth future trajectory is shown in green lines, the predicted trajectories are in red lines. Fig. (a)-(c) exhibit the instances with prediction modalities $K = 5$, and Fig. (d)-(f) show the instances with prediction modalities $K = 10$.

[22] Y. Liang and Z. Zhao, "Nettraj: A network-based vehicle trajectory prediction model with directional representation and spatiotemporal attention mechanisms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 470–14 481, 2021.

[23] Z. Wang, J. Zhang, J. Chen, and H. Zhang, "Spatio-temporal context graph transformer design for map-free multi-agent trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, 2023.

[24] Q. Du, X. Wang, S. Yin, L. Li, and H. Ning, "Social force embedded mixed graph convolutional network for multi-class trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, 2024.

[25] K. Wu, Y. Zhou, H. Shi, X. Li, and B. Ran, "Graph-based interaction-aware multimodal 2d vehicle trajectory prediction using diffusion graph convolutional networks," *IEEE Transactions on Intelligent Vehicles*, 2023.

[26] L. Guo, C. Shan, T. Shi, X. Li, and F.-Y. Wang, "A vectorized representation model for trajectory prediction of intelligent vehicles in challenging scenarios," *IEEE Transactions on Intelligent Vehicles*, 2023.

[27] K. Zhang, L. Zhao, C. Dong, L. Wu, and L. Zheng, "Ai-tp: Attention-based interaction-aware trajectory prediction for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 73–83, 2022.

[28] X. Mo, Y. Xing, H. Liu, and C. Lv, "Map-adaptive multimodal trajectory prediction using hierarchical graph neural networks," *IEEE Robotics and Automation Letters*, 2023.

[29] H. Liao, Y. Li, Z. Li, C. Wang, Z. Cui, S. E. Li, and C. Xu, "A cognitive-based trajectory prediction approach for autonomous driving," *arXiv preprint arXiv:2402.19251*, 2024.

[30] M. Geng, J. Li, Y. Xia, and X. M. Chen, "A physics-informed transformer model for vehicle trajectory prediction on highways," *Transportation research part C: emerging technologies*, vol. 154, p. 104272, 2023.

[31] H. Hu, Q. Wang, Z. Zhang, Z. Li, and Z. Gao, "Holistic transformer: A joint neural network for trajectory prediction and decision-making of autonomous vehicles," *Pattern Recognition*, vol. 141, p. 109592, 2023.

[32] D. Roy, T. Ishizaka, C. K. Mohan, and A. Fukuda, "Vehicle trajectory prediction at intersections using interaction based generative adversarial networks," in *2019 IEEE Intelligent transportation systems conference (ITSC)*. IEEE, 2019, pp. 2318–2323.

[33] X. Li, G. Rosman, I. Gilitschenski, C.-I. Vasile, J. A. DeCastro, S. Karaman, and D. Rus, "Vehicle trajectory prediction using generative

adversarial network with temporal logic syntax tree features," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3459–3466, 2021.

[34] M. Neumeier, M. Botsch, A. Tollkühn, and T. Berberich, "Variational autoencoder-based vehicle trajectory prediction with an interpretable latent space," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 820–827.

[35] X. Chen, J. Xu, R. Zhou, W. Chen, J. Fang, and C. Liu, "Trajvae: A variational autoencoder model for trajectory generation," *Neurocomputing*, vol. 428, pp. 332–339, 2021.

[36] X. Feng, Z. Cen, J. Hu, and Y. Zhang, "Vehicle trajectory prediction using intention-based conditional variational autoencoder," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3514–3519.

[37] R. Xu, S. Yang, Y. Wang, B. Du, and H. Chen, "A survey on vision mamba: Models, applications and challenges," *arXiv preprint arXiv:2404.18861*, 2024.

[38] M. Pióro, K. Ciebiera, K. Król, J. Ludziejewski, and S. Jaszczur, "Moe-mamba: Efficient selective state space models with mixture of experts," *arXiv preprint arXiv:2401.04081*, 2024.

[39] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[40] A. Ghoul, K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "A lightweight goal-based model for trajectory prediction," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 4209–4214.

[41] Z. Yao, X. Li, B. Lang, and M. C. Chuah, "Goal-lbp: Goal-based local behavior guided trajectory prediction for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[42] D. Li, Q. Zhang, S. Lu, Y. Pan, and D. Zhao, "Conditional goal-oriented trajectory prediction for interacting vehicles," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[43] B. Dong, H. Liu, Y. Bai, J. Lin, Z. Xu, X. Xu, and Q. Kong, "Multi-modal trajectory prediction for autonomous driving with semantic map and dynamic graph attention network," *arXiv preprint arXiv:2103.16273*, 2021.

[44] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 863–17 873.

[45] D. Li, Q. Zhang, Z. Xia, Y. Zheng, K. Zhang, M. Yi, W. Jin, and D. Zhao, "Planning-inspired hierarchical trajectory prediction via lateral-longitudinal decomposition for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2023.

[46] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan *et al.*, "Time-llm: Time series forecasting by reprogramming large language models," *arXiv preprint arXiv:2310.01728*, 2023.

[47] C. Chang, W.-C. Peng, and T.-F. Chen, "Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms," *arXiv preprint arXiv:2308.08469*, 2023.

[48] S. Gao, T. Koker, O. Queen, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik, "Units: Building a unified time series model," *arXiv preprint arXiv:2403.00131*, 2024.

[49] Y. Bian, X. Ju, J. Li, Z. Xu, D. Cheng, and Q. Xu, "Multi-patch prediction: Adapting llms for time series representation learning," *arXiv preprint arXiv:2402.04852*, 2024.

[50] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[51] Y. Yang, Q. Zhang, C. Li, D. S. Marta, N. Batool, and J. Folkesson, "Human-centric autonomous systems with llms for user command reasoning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 988–994.

[52] C. Cui, Y. Ma, X. Cao, W. Ye, and Z. 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*, 2024, pp. 902–909.

[53] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.

[54] M. Liu, H. Cheng, L. Chen, H. Broszio, J. Li, R. Zhao, M. Sester, and M. Y. Yang, "Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints," *arXiv preprint arXiv:2302.13933*, 2023.

[55] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[56] R. Zhang, J. Han, C. Liu, A. Zhou, P. Lu, H. Li, P. Gao, and Y. Qiao, "Llama-adapter: Efficient fine-tuning of large language models with zero-initialized attention," in *The Twelfth International Conference on Learning Representations*, 2023.

[57] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, "Tip-adapter: Training-free adaption of clip for few-shot classification," in *European conference on computer vision*. Springer, 2022, pp. 493–510.

[58] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[59] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[60] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural networks*, vol. 107, pp. 3–11, 2018.

[61] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8823–8833.

[62] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[63] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. 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*. Springer, 2020, pp. 683–700.

[64] S. Narayanan, R. Moslemi, F. Pittaluga, B. Liu, and M. Chandraker, "Divide-and-conquer for lane-aware diverse trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 799–15 808.

[65] H. Cheng, M. Liu, L. Chen, H. Broszio, M. Sester, and M. Y. Yang, "Gatraj: A graph-and attention-based multi-agent trajectory prediction model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 205, pp. 163–175, 2023.

[66] C. Wang, Y. Wang, M. Xu, and D. J. Crandall, "Stepwise goal-driven networks for trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2716–2723, 2022.

[67] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, "What-if motion prediction for autonomous driving," *arXiv preprint arXiv:2008.10587*, 2020.

[68] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, "Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 165–170.

[69] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9813–9823.

[70] B. Kim, S. H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi, "Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 636–14 645.

[71] N. Deo and M. M. Trivedi, "Trajectory forecasts in unknown environments conditioned on grid-based plans," *arXiv preprint arXiv:2001.00735*, 2020.

[72] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Gohome: Graph-oriented heatmap output for future motion estimation," in *2022 international conference on robotics and automation (ICRA)*. IEEE, 2022, pp. 9107–9114.

[73] M. Schäfer, K. Zhao, M. Bühren, and A. Kummert, "Context-aware scene prediction network (caspnet)," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 3970–3977.

[74] M. Lee, S. S. Sohn, S. Moon, S. Yoon, M. Kapadia, and V. Pavlovic, "Muse-vae: Multi-scale vae for environment-aware long term trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2221–2230.

[75] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide, and C. Pal, "Latent variable sequential set transformers for joint multi-agent motion prediction," *arXiv preprint arXiv:2104.00563*, 2021.

[76] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Thomas: Trajectory heatmap output with learned multi-agent sampling," *arXiv preprint arXiv:2110.06607*, 2021.
[77] D. Choi and K. Min, "Hierarchical latent structure for multi-modal vehicle trajectory forecasting," in *European Conference on Computer Vision*. Springer, 2022, pp. 129–145.
[78] N. Deo, E. Wolff, and O. Beijbom, "Multimodal trajectory prediction conditioned on lane-graph traversals," in *Conference on Robot Learning*. PMLR, 2022, pp. 203–212.
[79] D. Park, H. Ryu, Y. Yang, J. Cho, J. Kim, and K.-J. Yoon, "Leveraging future relationship reasoning for vehicle trajectory prediction," *arXiv preprint arXiv:2305.14715*, 2023.

Zhengxing Lan is currently working toward the Ph.D. degree at the School of Transportation Science and Engineering, Beihang University. His research interests include autonomous driving and intelligent transportation.

Lingshan Liu is currently working toward the M.Sc. degree at the School of Transportation Science and Engineering, Beihang University. Her research interests include automated decision and artificial intelligence.

Bo Fan received the Ph.D. degree from the School of Information and Communications Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2018. From 2016 to 2017, he was a Visiting Scholar with the Software Group, ETH Zurich. He is currently working as an Associate Professor with the College of Metropolitan Transportation, Beijing University of Technology. His research interests include vehicular communications, intelligent transportation systems, and wireless resource optimizations.

Yisheng Lv (Senior Member, IEEE) received the B.E. and M.E. degrees from Harbin Institute of Technology, Harbin, China, in 2005 and 2007, respectively, and the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2010. He is currently a Professor with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences. His current research interests include traffic data analysis, dynamic traffic modeling, and parallel traffic management and control systems.

Yilong Ren (Member, IEEE) received B.S. and Ph.D. degrees from Beihang University in 2010 and 2017, respectively. He is currently an Associate Professor at the School of Transportation Science and Engineering, Beihang University. His research interests include urban traffic operations and traffic control and simulation.

Zhiyong Cui (Member, IEEE) is a Professor in the School of Transportation Science and Engineering at Beihang University. He was selected into the National Natural Science Foundation of China Outstanding Youth (Overseas) Fund Project. He was a Data Science Postdoctoral Fellow at the University of Washington. He received Ph.D. degree in Civil Engineering in 2021 at UW. He received M.S. and B.S. degrees both in software engineering from Peking University in 2015 and Beihang Univerity in 2012, respectively. His research mainly focuses on transportation data science, artificial intelligence, traffic prediction and control. He serves as a committee member of the TRB Intelligent Transportation Systems Committee and the ASCE T&DI Artificial Intelligence Committee. He has published over 30 journal papers, one English textbook, and one Chinese textbook. He has received the IEEE ITSS Best Dissertation Award, HKSTS Outstanding Dissertation Award, etc.