TRAJECTORY-LLM: A LANGUAGE-BASED DATA GENERATOR FOR TRAJECTORY PREDICTION IN AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

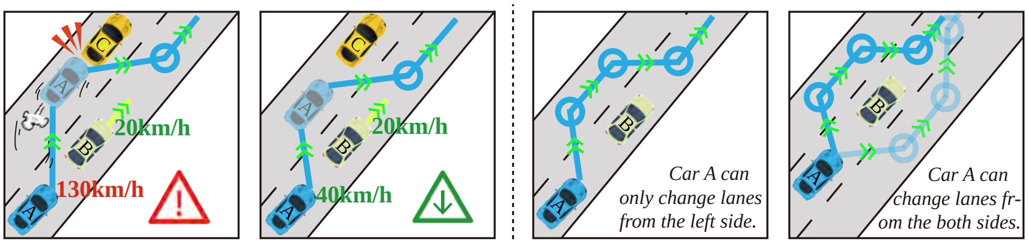
Vehicle trajectory prediction is a crucial aspect of autonomous driving, which requires extensive trajectory data to train prediction models to understand the complex, varied, and unpredictable patterns of vehicular interactions. However, acquiring real-world data is expensive, so we advocate using Large Language Models (LLMs) to generate abundant and realistic trajectories of interacting vehicles efficiently. These models rely on textual descriptions of vehicle-to-vehicle interactions on a map to produce the trajectories. We introduce Trajectory-LLM (Traj-LLM), a new approach that takes brief descriptions of vehicular interactions as input and generates corresponding trajectories. Unlike language-based approaches that translate text directly to trajectories, Traj-LLM uses reasonable driving behaviors to align the vehicle trajectories with the text. This results in an "interaction-behavior-trajectory" translation process. We have also created a new dataset, Language-to-Trajectory (L2T), which includes 240K textual descriptions of vehicle interactions and behaviors, each paired with corresponding map topologies and vehicle trajectory segments. By leveraging the L2T dataset, Traj-LLM can adapt interactive trajectories to diverse map topologies. Furthermore, Traj-LLM generates additional data that enhances downstream prediction models, leading to consistent performance improvements across public benchmarks.

1 INTRODUCTION

Accurately predicting the trajectory of vehicles is a crucial aspect of autonomous driving systems. A vast amount of data regarding vehicle trajectories is required to train the trajectory prediction models. However, collecting such data from real-world scenarios requires considerable manual effort and resources. This results in significant costs associated with capturing vehicle trajectories that exhibit intense interactions like overtaking, yielding, and bypassing.

The autonomous driving industry has developed a solution of using traffic-flow generators to synthesize trajectories and enhance training and testing data. An excellent generator should generate much synthetic data with minimal human operation. Previous works (Li et al., 2024; Feng et al., 2024) have shown the effectiveness of combining real-world and synthetic data for training prediction models. Popular generators like Apollo (Baidu., 2020) and Lgsvl (Rong et al., 2020) have ***graphical interfaces*** that enable users to create and control vehicle trajectories and interaction relationships on a map through drag-and-drop operations. However, even a single vehicle's precise editing often requires tens of drag-and-drop operations. Other generators like Scenic (Fremont et al., 2022) and OpenSCENARIO (ASAM.) provide ***programmable interfaces*** that allow users to control vehicle motion using functional parameters such as position, velocity, and yaw. However, they require users to have excellent programming skills to utilize codes to depict complex vehicle interactions.

The recent literature advocates a more user-friendly generator with ***language interfaces*** built on top of the large language models (LLMs) (Zhong et al., 2023a;b; Tan et al., 2023; Ding et al., 2023; Mao et al., 2023; Zhao et al., 2024), which translates the brief text of vehicle interactions to the trajectories in the map (a.k.a., “interaction-trajectory” translation). The brevity of the text precludes the specific parameters regarding vehicle motions, rendering it a highly abstract description of vehicle interactions. Therefore, users can efficiently generate many vehicle trajectories that align with


C
A
20km/h
B
130km/h
A
C
A
20km/h
B
40km/h
A
B
A
Car A can only change lanes from the left side.
B
A
Car A can change lanes fr-om the both sides.

(a) Dangerous Speed (b) Safe Speed (c) Narrow Road (d) Wide Road

Figure 1: (a) Dangerous speeding of Car A for overtaking a slow Car B is unusual, which may lead to a collision with Car C. (b) A reasonable speed of Car A for overtaking a slow Car B follows a safer driving logic. Without the guidance of driving logic, a language interface that has only seen trajectories of left turns in narrow road scenes like (c) may find it difficult to generate trajectories such as left and right turns in wide road scenes like (d).

the interaction described by the text. **This efficiency motivates us to study the vehicle trajectory generator based on the large language model for saving data collection efforts.** The vehicle trajectories can be viewed as a collection of waypoints, with each waypoint associated with the specific motion parameters of the vehicle as it passes through. These trajectories concretely describe the vehicle interaction. Language interfaces utilize the map's environmental information and vehicles' partial trajectories as guidance to generate the complete trajectories aligned with the text. This approach is similar to trajectory prediction models, but these two tasks fundamentally differ. While trajectory prediction models aim to predict relatively deterministic vehicle paths, using language interfaces may generate many trajectories that align with the interaction described in the short text.

However, when delving deeper into the vehicle trajectories generated by the language interface, many of them show unreasonable driving behaviors (see Figure 1). In contrast to the brief text of vehicle interaction and the concrete parameters of vehicle motion, the driving behaviors (e.g., change lane, cruise, accelerate/decelerate, and stop) reflect the inherent logic of human beings or autonomous driving systems that dynamically adjust the vehicle motions based on the changing environment. In Figure 1(a–b), given the text description of "Car A overtakes a very slow Car B", the trajectory of Car A in (a) represents a dangerous behavior of speeding, while in (b) Car A overtakes B at a safe speed. This leads to a lack of realism in the generated data. Furthermore, without reasonable driving behaviors, the language interface may become accustomed to generating trajectories like those in the known scenarios rather than the novel ones in the unseen scenarios, thus lowering the data diversity. In Figure 1(c–d), given the text of "Car A bypasses a stationary Car B", Car A in (c) can only change lanes from the left side to bypass B on a narrow road after extensive training, but failing to generate a legal right-hand bypassing on an unseen wide road in (d). **Thus, the primary objective of this paper is to involve the human driving logic to guide the translation between the text description of vehicle interactions and trajectories.**

This paper reformulates the "interaction-trajectory" translation by the language interface to the "interaction-behavior-trajectory" translation by adding the driving behaviors in-between to guide the generation of vehicle trajectories. We propose a Trajectory-LLM (Traj-LLM), a language-based generator for producing vehicle trajectories. As illustrated in Figure 2, Traj-LLM divides the "interaction-behavior-trajectory" translation into two stages. At the first stage (see Figure 2(a)), Traj-LLM takes input as the text of interaction between multiple vehicles and the map with lanes, outputting the driving behaviors of each vehicle. Chronologically, we organize the driving behaviors of each vehicle into a text sequence. Each driving behavior is followed by a text that describes the logic behind this behavior, in the format like "Change Lane: There are approximately 5 meters of road width on the left, while the road width on the right is insufficient, therefore choosing to change to the left lane" and "Change Lane: Both the left and right sides have ample road widths, allowing for a lane change to either the left or the right". At the second stage (see Figure 2(b)), Traj-LLM fuses the vehicle interactions and behaviors in the text with the map information to output specific motion parameters for each vehicle's trajectory.

We introduce a dataset, named ***Language-to-Trajectory*** (L2T), to aid in the training of Traj-LLM and associated research efforts. L2T comprises vehicle trajectories derived from 240K traffic scenarios, encompassing six diverse road topologies such as straightway, bend, roundabout, cross/T-shaped/Y-shaped intersection. Skilled drivers carefully craft these intricate trajectories and utilize textual segments to capture the vehicle interactions and behaviors within each scenario. We engage our drivers to provide annotations regarding vehicle interactions and behaviors for these scenarios.

Traj-LLM can generate realistic, diverse, and interactive vehicle trajectories based on brief texts. These generated trajectories can be utilized for training trajectory prediction models, significantly improving their performances. We summarize our contributions as follows:

- We advocate a new paradigm of the language interface with the "interaction-behavior-trajectory" translation to generate vehicle trajectories coherent with the human driving logic.
- We propose a novel language interface named Traj-LLM, a vehicle trajectory generator based on the large language model. Furthermore, we collect a new L2T dataset containing 240K traffic scenarios with vehicles' interactive trajectories. This dataset also contains rich text descriptions of vehicle behaviors and interactions for training Traj-LLM's "interaction-behavior-trajectory" translation.
- Traj-LLM trained on the L2T dataset can generate vehicle trajectories as additional data for training trajectory prediction models, whose performances are effectively improved on the public Waymo and Argoverse datasets. These results can further inspire the relevant research on the language-based trajectory generator.

2 RELATED WORK

We survey three groups of generators, which are equipped with the graphical, programmable, and language interfaces to control the vehicle interactions and generate the trajectories.

Graphical Interface Generators with graphical interfaces can be divided into Logsim and Worldsim. Logsim (Baidu., 2020) uses the real scenes to provide complex traffic situations. The scenes provided by Logsim are unchangeable, thus it fails to test the object interactions specified by different users. Worldsim allows users to design the scenes for testing the autonomous vehicles. These generators (Baidu., 2020; NVIDIA., 2020; Shah et al., 2018; Samak et al., 2023; Dosovitskiy et al., 2017; Rong et al., 2020; Silvera et al., 2022) employ the game engines like Unreal (Games.) and Unity 3D (Technologies.) that have the graphic renders of the realistic scenes obeying the real-world rules. Nevertheless, Worldsim costs expensive labor to operate the graphical interface to control every vehicle to create interactions. In contrast, the language interface relies on the short text to efficiently create the object interactions.

Programmable Interface Compared to the graphical interface that partially omits the details of vehicle motions to simplify the generator, the programmable interfaces (Zhao et al., 2024; Yang et al., 2023) offer an extensive collection of controlling functions of the vehicle interactions. Scenic (Fremont et al., 2022) and OpenSCENARIO (ASAM.) are the domain-specific languages for describing the vehicle behaviors. Yet, they need a significant amount of code to build a road scene with complex vehicle interactions. The Reinforcement learning (Rempe et al., 2023; Janner et al., 2022; Li et al., 2019; Shah et al., 2018; Dosovitskiy et al., 2017; Zhang et al., 2021; Rong et al., 2020; Amini et al., 2020; Zhong et al., 2023b) allows the traffic rules, which also belong to the formal language, to guide the generation of the specific vehicle interactions without requiring intricate coding. The traffic rules help to construct the reward function that drives reinforcement learning. The limited number of traffic rules not only constrain the driving behavior of each vehicle, but fail to express the logic of making any behavior. This logic is important for teaching vehicles to navigate different maps, executing maneuvers such as lane changes and merges safely.

In this work, we introduce driving logic applicable to different map topologies through text. These logics can guide LLM to generate reasonable vehicle trajectories in various map topologies, thereby increasing the diversity of generated data. Diverse data aids in training downstream trajectory prediction models, enhancing prediction accuracy in complex scenarios.

Language Interface The recent LLMs (Kenton & Toutanova, 2019; Stiennon et al., 2020; Brown et al., 2020; Touvron et al., 2023; Bai et al., 2023) enable the language interface to create virtual scenes with complicated vehicle interactions. Though people can use the text-to-video methods (Blattmann et al., 2023; Wu et al., 2023; Ho et al., 2022; Voleti et al., 2022; Research., 2023) and traffic generation methods (Park et al., 2023; Zhong et al., 2023a;b; OpenAI, 2023; Vemprala

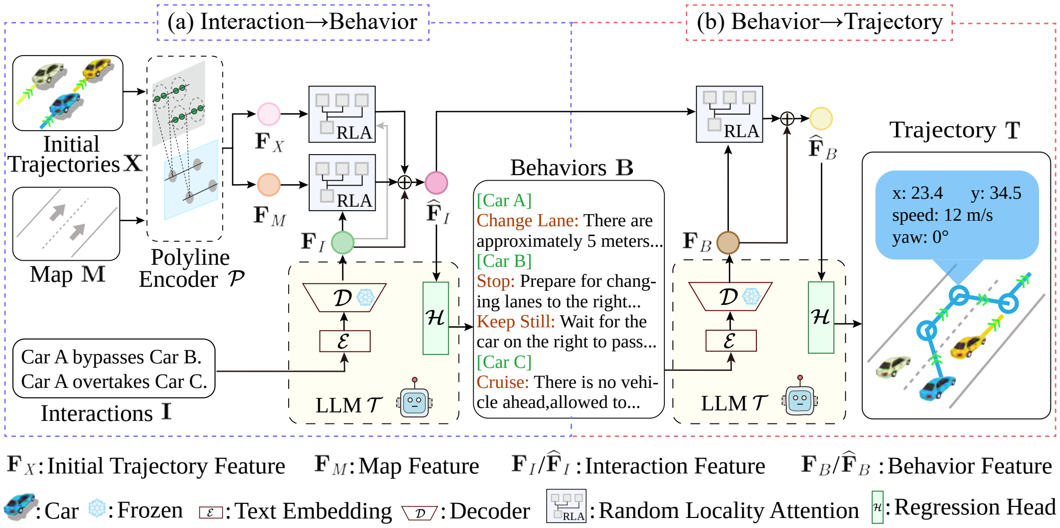

(a) Interaction→Behavior
(b) Behavior→Trajectory
Initial Trajectories X
Map M
Polyline Encoder $\mathcal{P}$
$\mathbf{F}_X$
$\mathbf{F}_M$
$\mathbf{F}_I$
$\widehat{\mathbf{F}}_I$
RLA
RLA
RLA
$\mathcal{D}$
$\mathcal{E}$
$\mathcal{H}$
LLM $\mathcal{T}$
Car A bypasses Car B.
Car A overtakes Car C.
Interactions I
Behaviors B
[Car A]
Change Lane: There are approximately 5 meters...
[Car B]
Stop: Prepare for changing lanes to the right...
Keep Still: Wait for the car on the right to pass...
[Car C]
Cruise: There is no vehicle ahead,allowed to...
$\mathbf{F}_B$
$\widehat{\mathbf{F}}_B$
$\mathcal{D}$
$\mathcal{E}$
$\mathcal{H}$
LLM $\mathcal{T}$
Trajectory T
x: 23.4 y: 34.5
speed: 12 m/s
yaw: 0°
$\mathbf{F}_X$: Initial Trajectory Feature
$\mathbf{F}_M$: Map Feature
$\mathbf{F}_I/\widehat{\mathbf{F}}_I$: Interaction Feature
$\mathbf{F}_B/\widehat{\mathbf{F}}_B$: Behavior Feature
: Car
: Frozen
$\mathcal{E}$: Text Embedding
$\mathcal{D}$: Decoder
RLA: Random Locality Attention
$\mathcal{H}$: Regression Head

Figure 2: The two stages of the "interaction-behavior-trajectory" translation. (a) We employ LLM with the random locality attention to translate the textual description of vehicle interactions into the behavior of each vehicle. Each behavior is associated with the underlying logic. (b) Given the vehicle interactions and behaviors, LLM translates them to the sequential motion parameters that represent the trajectory of each vehicle. We illustrate the random locality attention in Figure 3.

et al., 2023; Cui et al., 2023; Wen et al., 2023; Chen et al., 2023; Mao et al., 2023; Sha et al., 2023; Jin et al., 2023) to design the object interactions, they may fail to produce the usual vehicle motions like those in the real world. CTG (Zhong et al., 2023b) and CTG++ (Zhong et al., 2023a) depend on the traffic rules to generate realistic motions for the individual objects, yet lack a practical scheme for producing the complex vehicle interactions.

In contrast to the existing language interfaces, we propose a two-stage translation from vehicle interactions and behaviors, to trajectories. Real driving logic guides the entire translation process, resulting in a high realism of the generated vehicle trajectories. Additionally, we propose random locality attention that effectively utilizes information from vehicle interactions to guide the trajectory generation, thus enabling the generated trajectories to exhibit complex interactions.

3 TRAJECTORY-LLM

We present the pipeline of using Traj-LLM to conduct the "interaction-behavior-trajectory" translation. Figure 2 provides an overview of Traj-LLM, which is divided into two stages (see **Interaction-Behavior Translation** in Figure 2(a) and **Behavior-Trajectory Translation** in Figure 2(b)).

Interaction-Behavior Translation At the first stage illustrated in Figure 2(a), Traj-LLM translates the text of vehicle interactions to behaviors. We denote the text of vehicle interactions as $\mathbf{I}$, which records O pairs of vehicle-to-vehicle interaction between N vehicles. Each pair of interactions is documented in text like "Car A overtakes Car B" and "Car B bypasses Car C". We keep these texts short to enhance the convenience of using Traj-LLM. We feed K text segments of $\mathbf{I}$ into LLM $\mathcal{T}$, which outputs the text of the behaviors $\mathbf{B}$ for N vehicles. We document the behaviors of each vehicle like "[Car A] Change Lane: Both the left and right sides have ample road widths...; Keep Still:...".

We feed the text interactions $\mathbf{I}$ into the text embedding layer $\mathcal{E}$ of LLM $\mathcal{T}$ to compute the interaction feature $\mathbf{F}_I \in \mathbb{R}^{O\times C}$ in Eq. (1). C counts the feature channels. We input a map $\mathbf{M}$ represented by polylines and the initial trajectories $\mathbf{X} \in \mathbb{R}^{N\times(S\times 4)}$ of N vehicles to a polyline encoder $\mathcal{P}$ in (Shi et al., 2022; Gao et al., 2020) to compute the initial trajectories $\mathbf{F}_X \in \mathbb{R}^{N\times C}$ and the map feature $\mathbf{F}_M \in \mathbb{R}^{M\times C}$ as:

$$\mathbf{F}_I = \mathcal{E}(\mathbf{I}), \quad \mathbf{F}_X = \mathcal{P}(\mathbf{X}), \quad \mathbf{F}_M = \mathcal{P}(\mathbf{M}). \tag{1}$$

Each initial trajectory includes the waypoints at S moments, and each waypoint is associated with four motion parameters (i.e., x/y-coordinates, heading, and speed). M is the number of polylines in the map $\mathbf{M}$. The initial trajectories and the map together represent the spatial configuration of vehicles on the map.

To ensure the consistency between vehicle interaction and map for preventing invalid situations such as vehicles moving beyond the map boundaries, we propose a random locality attention between the interaction $\mathbf{I}$ (see Figure 3), the initial trajectories $\mathbf{X}$, and the specific map $\mathbf{M}$ as:

$$\begin{aligned}
&\mathbf{S}^{o}_{I\leftrightarrow X}{=}softmax\left(\frac{\mathbf{F}^{o}_{I}{\cdot}\mathbf{F}^{1}_{X}}{\sqrt{C}},\ldots,\frac{\mathbf{F}^{o}_{I}{\cdot}\mathbf{F}^{N}_{X}}{\sqrt{C}}\right),\ \left\{\mathbf{S}^{o,1}_{I\leftrightarrow X},\ldots,\mathbf{S}^{o,K}_{I\leftrightarrow X}\right\}{=}top_{K}\left(\mathbf{S}^{o,1}_{I\leftrightarrow X},\ldots,\mathbf{S}^{o,N}_{I\leftrightarrow X}\right),\\
&\mathbf{S}^{o}_{I\leftrightarrow M}{=}softmax\left(\frac{\mathbf{F}^{o}_{I}{\cdot}\mathbf{F}^{1}_{M}}{\sqrt{C}},\ldots,\frac{\mathbf{F}^{o}_{I}{\cdot}\mathbf{F}^{M}_{M}}{\sqrt{C}}\right),\ \left\{\mathbf{S}^{o,1}_{I\leftrightarrow M},\ldots,\mathbf{S}^{o,K}_{I\leftrightarrow M}\right\}{=}top_{K}\left(\mathbf{S}^{o,1}_{I\leftrightarrow M},\ldots,\mathbf{S}^{o,M}_{I\leftrightarrow M}\right),
\end{aligned} \tag{2}$$

$\mathbf{S}_{I\leftrightarrow X} \in \mathbb{R}^{O\times N}$ and $\mathbf{S}_{I\leftrightarrow M} \in \mathbb{R}^{O\times M}$ represent the correlation scores between O pairs of vehicle interaction, N initial trajectories and M polylines in the map $\mathbf{M}$. In Eq. (2), top_K operation selects the top-K largest scores from the set of correlation scores. The selected scores are used to weight the initial trajectory feature $\mathbf{F}_X$ and map features $\mathbf{F}_M$, which are added to the feature $\mathbf{F}_I$ as:

$$\widehat{\mathbf{F}}^{o}_{I} = \mathbf{F}^{o}_{I} + \sum_{k=1}^{K}(\alpha^{k}_{I\leftrightarrow X}\mathbf{S}^{o,k}_{I\leftrightarrow X}\cdot\mathbf{F}^{k}_{X} + \alpha^{k}_{I\leftrightarrow M}\mathbf{S}^{o,k}_{I\leftrightarrow M}\cdot\mathbf{F}^{k}_{M}),\quad \alpha^{k}_{I\leftrightarrow X},\alpha^{k}_{I\leftrightarrow M}\sim\mathcal{N}(0,1). \tag{3}$$

By setting $K < N$ and M, we let the feature $\widehat{\mathbf{F}}_I \in \mathbb{R}^{O\times C}$ capture the locality correlation between the interaction and the spatial configuration of the map. This locality enables each pair of vehicle interactions to occur at the most reasonable locations on the map. Besides, $\alpha^{k}_{I\leftrightarrow X}, \alpha^{k}_{I\leftrightarrow M}$ are random variables obeying the normal distribution, which jitters the correlation scores $\mathbf{S}^{o,k}_{I\leftrightarrow X}, \mathbf{S}^{o,k}_{I\leftrightarrow M} \in \mathbb{R}$. They let the feature $\widehat{\mathbf{F}}^{o}_{I} \in \mathbb{R}^{C}$ represent the o^{th} interaction, whose location has some jitter within a range.

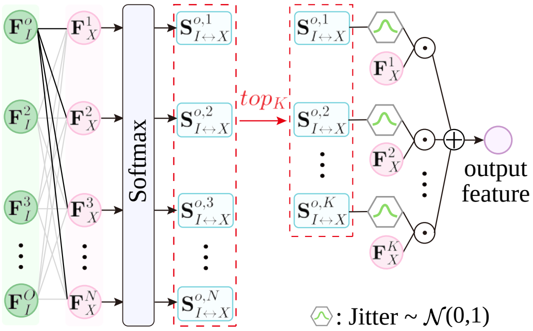

$\mathbf{F}^{o}_{I}$
$\mathbf{F}^{2}_{I}$
$\mathbf{F}^{3}_{I}$
$\mathbf{F}^{O}_{I}$
$\mathbf{F}^{1}_{X}$
$\mathbf{F}^{2}_{X}$
$\mathbf{F}^{3}_{X}$
$\mathbf{F}^{N}_{X}$
Softmax
$\mathbf{S}^{o,1}_{I\leftrightarrow X}$
$\mathbf{S}^{o,2}_{I\leftrightarrow X}$
$\mathbf{S}^{o,3}_{I\leftrightarrow X}$
$\mathbf{S}^{o,N}_{I\leftrightarrow X}$
top_K
$\mathbf{S}^{o,K}_{I\leftrightarrow X}$
output feature
: Jitter ~ $\mathcal{N}(0,1)$

Figure 3: Illustration of the random locality attention. Here, we use the attention between O interaction features $\{\mathbf{F}^{o}_{I} \mid o = 1, ..., O\}$ and N initial trajectory features $\{\mathbf{F}^{n}_{X} \mid n = 1, ..., N\}$ as an example. This illustration applies to the attention formulated in Eqs. (2) and (6).

Given the interaction feature $\widehat{\mathbf{F}}_I$, LLM $\mathcal{T}$ uses the regression head to output N text segments in $\mathbf{B}$. We denote this regression head as $\mathcal{H}(query, key, value)$, which has a cross-attention architecture with tunable parameters denoted as:

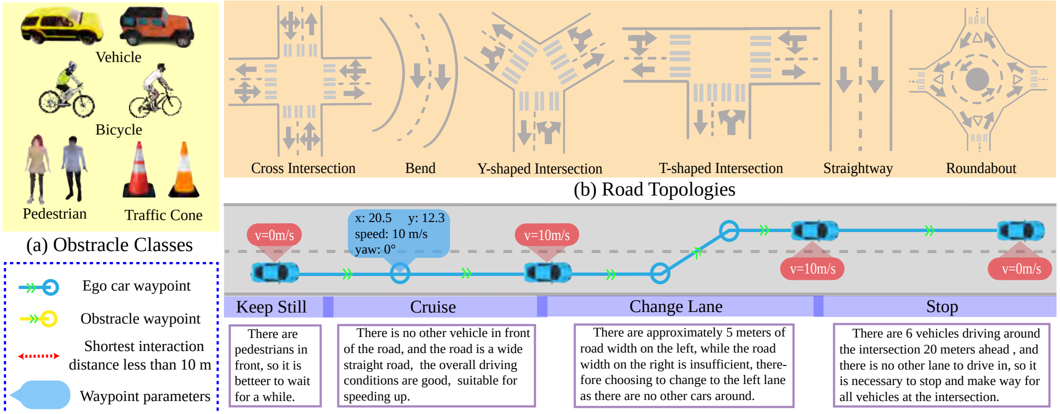
$$\mathbf{B} = \mathcal{H}(\overline{\mathbf{B}}, \sigma(\widehat{\mathbf{F}}_I), \sigma(\widehat{\mathbf{F}}_I)), \quad \mathbf{L}_B = \mathcal{L}_{CE}(\mathbf{B}, \widehat{\mathbf{B}}). \tag{4}$$

In Eq. (4), σ means the fully-connected layers with non-linear activation. The regression head follows the autoregressive style, utilizing the interaction feature $\widehat{\mathbf{F}}_I$ and the already output behavioral text $\overline{\mathbf{B}}$ to update the final behavioral text $\mathbf{B}$. Here, the already output text $\overline{\mathbf{B}}$ queries the contextual information within the interaction feature $\widehat{\mathbf{F}}_I$, which is conducive to outputting vehicle behaviors aligned with the interactions. During the training phase, we use the cross-entropy loss $\mathbf{L}_B$ to penalize the difference between the regressed segments in $\mathbf{B}$ and the ground-truth segments in $\widehat{\mathbf{B}}$, as formulated in Eq. (4).

Behavior-Trajectory Translation At the second stage illustrated in Figure 2(b), Traj-LLM translates the textual description of vehicle behaviors in $\mathbf{B}$ to trajectories $\mathbf{T} \in \mathbb{R}^{N\times(S\times 4)}$ for N vehicles. We convert numerical data of trajectories into text for training and inference of LLM. Again, each trajectory includes S waypoints with four motion parameters.

Figure 2(b) illustrates the behavior-trajectory translation. Again, we input the text segment of the behaviors $\mathbf{B}$ into LLM $\mathcal{T}$, and text embedding layer $\mathcal{E}$ computes the behavior feature $\mathbf{F}_B \in \mathbb{R}^{N\times C}$ as:

$$\mathbf{F}_B = \mathcal{E}(\mathbf{B}). \tag{5}$$



(a) Obstracle Classes

(b) Road Topologies

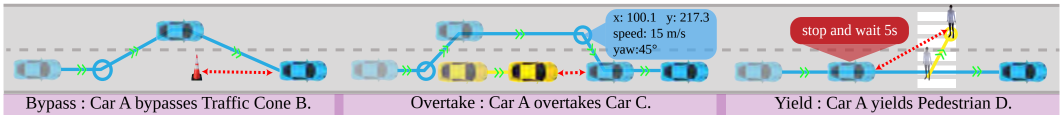
There are pedestrians in front, so it is betteer to wait for a while.

There is no other vehicle in front of the road, and the road is a wide straight road, the overall driving conditions are good, suitable for speeding up.

There are approximately 5 meters of road width on the left, while the road width on the right is insufficient, therefore choosing to change to the left lane as there are no other cars around.

There are 6 vehicles driving around the intersection 20 meters ahead , and there is no other lane to drive in, so it is necessary to stop and make way for all vehicles at the intersection.

(c) Vehicle Behaviors


x: 100.1 y: 217.3
speed: 15 m/s
yaw:45°
stop and wait 5s
Bypass : Car A bypasses Traffic Cone B.
Overtake : Car A overtakes Car C.
Yield : Car A yields Pedestrian D.

(d) Vehicle Interactions

Figure 4: (a) We prepare four kinds of objects (i.e., vehicle, pedestrian, bicycle, and traffic cone). (b) There are six kinds of road topologies, where the road shapes are different. (c) Each object can take the behaviors of changing lanes, cruising, stopping, and keeping still. (d) Bypassing the static object, overtaking and yielding the dynamic object are the interactions in the L2T dataset.

We employ the random locality attention to capture the correlation between the behavior feature $\mathbf{F}_B$ and the interaction feature $\widehat{\mathbf{F}}_I$, computing the correlation scores $\mathbf{S}_{B\leftrightarrow I} \in \mathbb{R}^{N\times O}$ as:

$$\mathbf{S}^n_{B\leftrightarrow I} = softmax\left(\frac{\mathbf{F}^n_B \cdot \widehat{\mathbf{F}}^1_I}{\sqrt{C}}, \dots, \frac{\mathbf{F}^n_B \cdot \widehat{\mathbf{F}}^O_I}{\sqrt{C}}\right), \ \left\{\mathbf{S}^{n,1}_{B\leftrightarrow I}, \dots, \mathbf{S}^{n,K}_{B\leftrightarrow I}\right\} = top_K\left(\mathbf{S}^{n,1}_{B\leftrightarrow I}, \dots, \mathbf{S}^{n,O}_{B\leftrightarrow I}\right), \quad (6)$$

where $K < O$. We use the correlation scores $\mathbf{S}_{B\leftrightarrow I}$ to weight the interaction feature $\widehat{\mathbf{F}}_I$ as:

$$\widehat{\mathbf{F}}^n_B = \mathbf{F}^n_B + \sum_{k=1}^{K} \alpha^k_{B\leftrightarrow I} \mathbf{S}^{n,k}_{B\leftrightarrow I} \cdot \widehat{\mathbf{F}}^k_I, \quad \alpha^k_{B\leftrightarrow I} \sim \mathcal{N}(0,1), \quad (7)$$

where we sample the random variable $\alpha^k_{B\leftrightarrow I}$ from the normal distribution. In Eq. (7), we compute the behavior feature $\widehat{\mathbf{F}}^n_B \in \mathbb{R}^C$ for the n^{th} vehicle. The random locality attention focuses the behavior feature $\widehat{\mathbf{F}}^n_B$ on the most relevant interaction features and the map information contained within them. The random variable $\alpha^k_{B\leftrightarrow I}$ also jitters the behavior feature $\widehat{\mathbf{F}}^n_B$, allowing Traj-LLM to generate more diverse trajectories aligned with the interactions and the map.

We feed the behavior feature $\widehat{\mathbf{F}}_B \in \mathbb{R}^{N\times C}$ of N vehicles to the tunable head $\mathcal{H}(qurey, key, value)$ of LLM $\mathcal{T}$ to generate the trajectories in $\mathbf{T}$ as:

$$\mathbf{T} = \mathcal{H}(\overline{\mathbf{T}}, \sigma(\widehat{\mathbf{F}}_B), \sigma(\widehat{\mathbf{F}}_B)), \quad \mathbf{L}_T = \mathcal{L}_{CE}(\mathbf{T}, \widehat{\mathbf{T}}), \quad (8)$$

where $\overline{\mathbf{T}}$ is the already-outputted trajectories. It queries the behavior information from $\widehat{\mathbf{F}}_B$ during the generation of the trajectories in $\mathbf{T}$. We minimize the cross-entropy loss $\mathbf{L}_T$ to penalize the difference between the generated trajectories $\mathbf{T}$ and the ground-truth trajectories $\widehat{\mathbf{T}}$.

4 LANGUAGE-TO-TRAJECTORY DATASET

The L2T dataset contains 240K road scenes, where the object classes, road topologies, and interactions are illustrated in Figure 10. We divide the L2T dataset into training and testing sets, which contain 200K and 40K scenes, respectively. Below we briefly provide the basic information of the L2T dataset. More information can be found in the App A.

Object Classes Each object can be taken from the classes of vehicle, pedestrian, bicycle, and traffic cone (see Figure 10(a)), which are widely seen in reality. Expect traffic cones to always be static; other objects can be stationary or moving. These objects may be of different sizes in road scenes. In this paper, we only use the bird's-eye view projection of objects to determine their sizes.

Road Topologies We prepare six typical kinds of road topology, including straightway, bend, roundabout, cross/T-shaped/Y-shaped intersection (see Figure 10(b)). Each kind of road topology has variant shapes and lanes in the training and testing sets. A set of 2D coordinates represents the boundary of each lane. We use the XODR file to store each road topology with multiple lanes.

Vehicle Behaviors Each vehicle has the behaviors of changing lanes, cruising, keeping still, and stopping. Cruising or changing lanes means moving along the same lane or across lanes. We limit the velocity during vehicle cruising to 120km/h. Keeping still/stopping is temporary/permanent. As illustrated in Figure 10(c), we document a vehicle's behaviors sequentially. We recruit a group of drivers with rich experience to explain and document the logic behind each behavior.

Vehicle Interactions A vehicle may interact with 1∼5 static/moving objects. In contrast to the behavior of a single object, the interaction defined here captures the relationship between a vehicle and an object. The possible interactions present in the L2T dataset are illustrated in Figure 10(d). The vehicle can bypass a static object to avoid collision. It can also overtake or yield the moving object. During overtaking or yielding, the vehicle occupies a lane sooner or later than the interactive object. We focus on the short-range interactions between objects. These high-intensity interactions should be completed in a short range of less than 10 meters, where the vehicle must take prompt action to avoid collision with the objects. The recruited drivers document each interaction in text.

Vehicle Trajectories Every moving object has a trajectory, which consists of a series of waypoints arranged chronologically. As illustrated in Figure 10(c–d), we associate each waypoint with four motion parameters (i.e., x/y-coordinates, heading, and speed). The duration of all the trajectories we provide is within 20 seconds and the sampling frequency is 20 Hz.

5 EXPERIMENTAL RESULTS

5.1 REALISM, DIVERSITY, AND CONTROLLABILITY

For implementation details of Traj-LLM, see App C. Below, we evaluate the realism and diversity of the trajectories generated by various methods. We also compare the control capability of different methods for trajectories to verify if these methods can generate the expected trajectories according to the interactions specified in the text.

The compared methods differ in their control conditions for generating vehicle trajectories. For a fair comparison, all methods utilize the data of L2T for training and testing. SNet (Bergamini et al., 2021) and TSim (Suo et al., 2021) use the BEV road topologies to generate trajectories. We train BITS (Xu et al., 2023) and CTG (Zhong et al., 2023b) on the trajectories from L2T, relying on rule-based conditions for trajectory generation as in (Xu et al., 2023; Zhong et al., 2023b). CTG++ (Zhong et al., 2023a), LGen (Tan et al., 2023), and Traj-LLM utilize text of interaction as the condition.

Realism In Table 1 "Realism", we report the agent- and scene-level (Zhong et al., 2023b;a) realism of the generated trajectories. The agent-level realism measures the distribution distances of longitudinal acceleration magnitude (**LO**), lateral acceleration magnitude (**LA**), jerk (**JE**), and yaw rate (**YR**), and their average (**AVG**), between each agent's generated and ground-truth trajectories. The scene-level realism is assessed by the relative scores of **LO**, **LA**, **JE**, **YR**, and **AVG**. More details about calculating these metrics can be found in App B. To compute a relative score, we gauge the distribution distance between each pair of vehicles' generated/ground-truth trajectories. Then, we evaluate the difference between the above two distances as the relative score.

Traj-LLM surpasses other methods in terms of realism at both the agent and scene levels. Traj-LLM introduces the interaction-behavior translation. Through learning from the real data, the short text of the vehicle interactions is translated into the behaviors with driving logic. It makes each vehicle's trajectory consistent with the human driving habits, generating more natural trajectories. Since each scene contains multiple vehicles, the entire scene is also made more realistic.

Table 1: We compare Traj-LLM with state-of-the-art methods on the L2T dataset. The results measure the realism and diversity of the trajectories generated by various methods. The realism score has been multiplied by 100. ↓/↑ means a smaller/larger value of the metric represents a better performance.

Method		**Realism**										**Diversity**			
		Agent					**Scene**					**Agent**			**Scene**
		LO↓	**LA↓**	**JE↓**	**YR↓**	**AVG↓**	**LO↓**	**LA↓**	**JE↓**	**YR↓**	**AVG↓**	**MD↑**	**AD↑**	**FD↑**	**WD↑**
Image	**SNet**	14.20	9.84	8.38	7.92	10.09	11.28	4.78	2.76	8.72	6.88	0.00	0.00	0.00	0.00
	TSim	25.36	21.78	5.33	3.65	14.03	15.94	4.60	10.67	7.91	9.78	0.52	0.25	0.30	2.92
Rule	**BITS**	10.16	5.82	9.23	4.30	7.38	10.57	**1.70**	3.54	4.75	5.14	6.80	3.22	6.98	7.27
	CTG	8.26	7.33	19.92	1.68	9.30	10.40	6.14	10.10	3.64	7.57	6.49	3.44	6.03	5.70
Text	**LGen**	7.55	9.73	13.09	1.91	8.07	9.59	7.89	10.21	4.10	7.95	3.38	1.74	2.19	4.02
	CTG++	6.13	7.13	21.19	1.13	8.90	7.94	4.16	10.37	2.22	6.17	6.38	3.55	6.08	5.79
	Traj-LLM	**2.56**	**3.17**	**0.74**	**0.28**	**1.69**	**0.73**	2.68	**0.52**	**0.54**	**1.12**	**14.91**	**6.81**	**8.71**	**13.81**

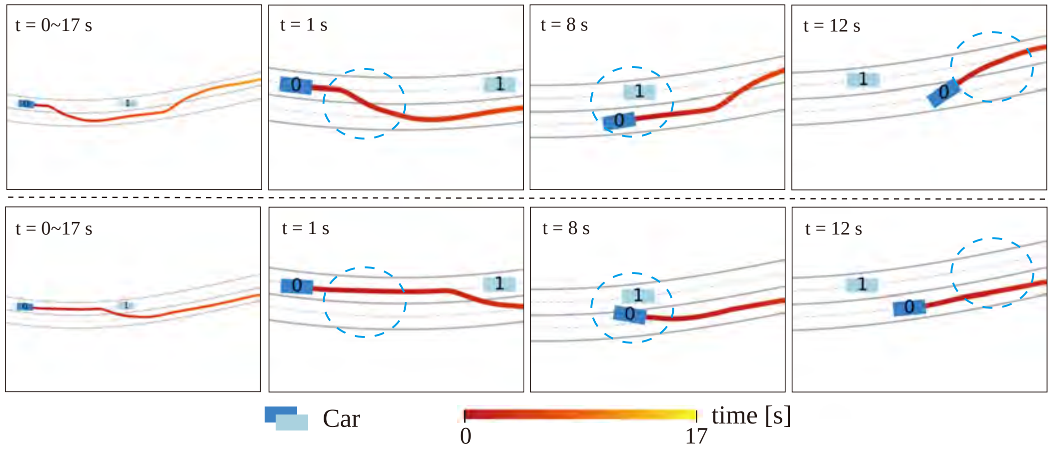


Figure 5: Diverse trajectories generated by Traj-LLM. Here, the initial positions of Car 0 in the first and second rows are the same.

Diversity In Table 1 "Diversity", we measure the agent-level diversity (Zhang et al., 2022) by reporting the map-aware average self distance (**MD**), average self distance (**AD**), and final self distance (**FD**). We measure the scene-level diversity (Xu et al., 2023) by reporting Wasserstein distance (**WD**). To evaluate the agent-level diversity of trajectories, we conduct multiple experiments with the same initial conditions (history trajectories, rules, images) and evaluate metrics for the same vehicle in every two experiments. **MD/AD** denotes the average L2 distance between the most divergent/similar trajectories. **FD** assesses the average L2 distance among the final position of the most similar trajectories. To evaluate the scene-level diversity, we compute the generated trajectory distribution on the map. Then, we calculate the density histogram of the distribution as the density profile. We run multiple experiments and measure the average **WD** between the pairwise density profile of each generated scene.

Traj-LLM achieves better trajectory diversity, as it employs the random locality attention to the behavior-trajectory translation. It dynamically focuses on the road topology, the driving status of neighbor vehicles, and the history trajectories. We jitter the parameters of the driving trajectories, further enriching the diversity of generated trajectories. We present the visualization of the generated trajectories in Figure 5, demonstrating that Traj-LLM generates more diverse trajectories.

Table 2: Comparison of interaction success rates(%) with text-based generation methods.

Method	**CTG++**	**LCTGen**	**Traj-LLM**
Overtake	65.2	36.9	**80.1**
Bypass	**91.8**	54.5	81.3
Yield	77.7	45.5	**83.5**
Avg	78.2	45.6	**81.6**

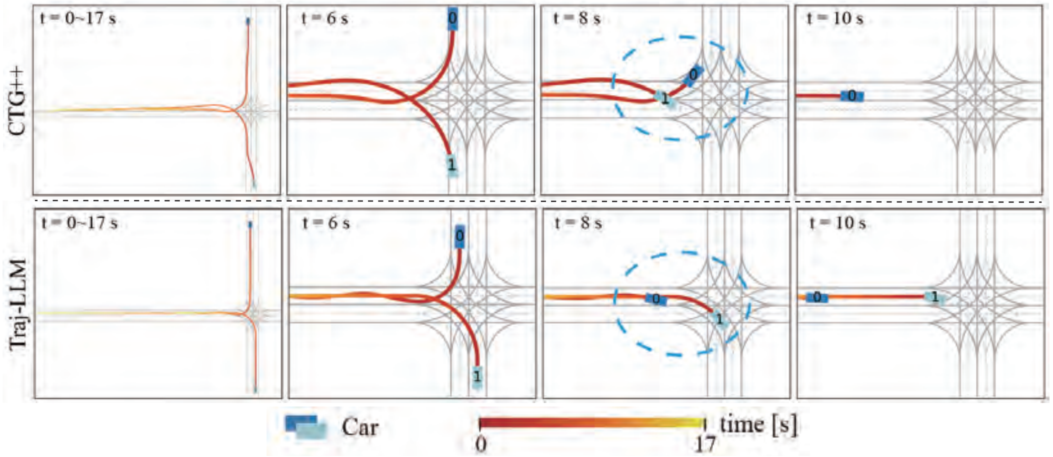


Figure 6: Controllability of CTG++ and Traj-LLM. In this example, Traj-LLM successfully control Car 0 to overtake Car 1, while the state-of-the-art CTG++ fails in this case.

Controllability In Table 2, we compare Traj-LLM with state-of-the-art methods that also rely on text to generate interactive vehicle trajectories, in terms of the scene-level controllability. We conduct multiple experiments for each vehicle, controlling its interactions with the designated vehicle. We record the success rate of achieving the specified interaction and adhering to traffic rules. We calculate the average success rate of different vehicles having various interactions over multiple experiments.

Traj-LLM achieves a higher success rate because it introduces the two-stage translation that generates realistic driving logic. This translation incorporates the random locality attention to enhance the realism and diversity of the generated trajectories. Solely relying on the short text, Traj-LLM presents a stronger control over the interactions than other methods (see Figure 6). Traj-LLM hopefully shows stronger controllability with more detailed control instructions.

5.2 ABLATION STUDY

In Table 3, we conduct the ablation study to measure the realism and diversity of the trajectories and the controllability of LLM. We denote **I-T/I-B-T** as “Interaction-Trajectory"/“Interaction-Behavior-Trajectory" translation. We use the **AVG**, **WD** and **success rate** for measuring the scene-level realism, diversity, and controllability. Without the help of driving behaviors that reflect the inherent logic of human driving, **I-T** produces inferior results (see the first row of Table 3) compared to **I-B-T**.

Table 3: **I-T/I-B-T** means “Interaction-Trajectory"/“Interaction-Behavior-Trajectory" translation. The results measure the realism, diversity and controllability (%).

Translation	Logic	Random	Locality	Realism↓	Diversity↑	Control↑
I-T				14.52	14.39	36.7
I-B-T		✓	✓	3.08	**22.64**	48.2
	✓			4.81	9.66	62.1
	✓	✓		1.45	17.26	58.3
	✓		✓	2.60	12.55	67.9
	✓	✓	✓	**1.12**	13.81	**81.6**

In Table 3 “**I-B-T**”, we list the critical modules of **I-B-T** translation, i.e., the driving-logic learning (**Logic**), random jitter (**Random**), and locality attention (**Locality**). Without **Logic** in the second row, we use LLM to generate the temporal sequence of behaviors only for each vehicle. This degrades realism, diversity, and controllability, demonstrating the importance of driving-logic learning. Based on **Logic**, we remove **Random** or **Locality** (see the third to fifth rows). Especially, the alternative without **Random** and **Locality** degrades the random locality attention to a vanilla cross-attention (the third row). Compared to these alternatives, the complete Traj-LLM shows better performance. This is because **Locality** enables the vehicles to generate reasonable trajectories aligned with the interactions and the map, while **Random** makes the generated trajectories more diverse. Note that **Locality** also offers efficient constraints and affects diversity. The fifth row removes **Random** but diversity doesn't drop much compared with the last row, while in the third and fourth rows, when **Locality** is removed, **Random** significantly impacts diversity.

5.3 GENERATED DATA FOR TRAJECTORY PREDICTION

In Table 4, we evaluate the effectiveness of Traj-LLM, by using the generated vehicle trajectories to train downstream trajectory prediction models, i.e., MTR(Shi et al., 2022) and HDGT(Jia et al., 2022). Apart from the model training and validating on the same dataset, we follow the protocol of the cross-dataset validation in (Torralba & Efros, 2011), where we train and test each prediction model on different datasets. The cross-dataset validation helps to evaluate the robustness of the prediction model trained and tested on various data distributions. We utilize two parts of trajectory data from the training set of WOMD (**WOMD (train)**) and the trajectories generated by Traj-LLM (**Traj-LLM (L2T)**), where Traj-LLM is trained on 200K scenarios of the L2T dataset. Each part of the training data contains 400K scenes. Each scene includes 2-8 vehicles with various interactions. The trained models are evaluated on the validation sets of WOMD (**WOMD (val)**) and L2T (**L2T (val)**), each containing about 40K scenes. We report the result of trajectory prediction in terms of the mean average precision (**mAP**).

By training MTR/HDGT on **Traj-LLM (L2T)** and validating it across on **WOMD (val)** (see blue cells in the second row), we find a performance degradation compared to the prediction models trained on **WOMD (Train)** (see the blue cells in the first row). This is because different trajectory distributions exit in **WOMD (Train)** and the generated data of **Traj-LLM (L2T)**. A large portion of the trajectories in **Traj-LLM (L2T)** represent the dense interactions between vehicles, whereas **WOMD (Train)** provides many vehicles without interaction. We also find a similar degradation in the purple cells of the first and second rows when training the prediction models on **WOMD (Train)** and validating it on **L2T (test)**.

Table 4: Results of trajectory prediction.

Train		**Test (mAP ↑)**			
WOMD (train)	**Traj-LLM (L2T)**	**WOMD (val)**		**L2T (val)**	
		MTR	HDGT	MTR	HDGT
✓		0.405	0.272	0.211	0.134
	✓	0.106	0.149	0.340	0.196
✓	✓	**0.416**	**0.294**	**0.383**	**0.243**

By integrating the data of **WOMD (Train)** and the generated trajectories in **Traj-LLM (L2T)**, MTR and HDGT outperform (see the last row of Table 4) their counterparts trained on a single dataset. The quantitative and qualitative results demonstrate that the trajectories generated by Traj-LLM exhibit remarkable realism and diversity, thus benefiting downstream prediction models. Furthermore, we conduct ablation study of generated data and L2T under fixed / non-fixed total dataset size in App D.

6 CONCLUSION

The precision prediction of vehicle trajectories holds significance in autonomous driving. To train accurate trajectory prediction models, a substantial quantity of data of vehicle trajectories is imperative. The recent development of generative artificial intelligence makes it possible to produce a large amount of trajectory data for training the trajectory prediction models at the cost of less human effort. This paper proposes a trajectory generator, Traj-LLM, which relies on the short text description of vehicle interaction to produce the interactive trajectories efficiently. We utilize the large language model to build Traj-LLM. In contrast to the language-based generators for “interaction-trajectory" translation, Traj-LLM conducts the “interaction-behaviour-trajectory", where it possesses a powerful capability in understanding the interactions between vehicles and outputting their behaviors and driving logics. This capability allows Traj-LLM to produce realistic and diverse trajectory data for training downstream trajectory prediction models. We also contribute the L2T dataset, which contains the interactive trajectories associated with the text descriptions of vehicle interactions, behaviors, and driving logic, to train Traj-LLM and promote relevant research in the future.

REFERENCES

Alexander Amini, Igor Gilitschenski, Jacob Phillips, Julia Moseyko, Rohan Banerjee, Sertac Karaman, and Daniela Rus. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. *IEEE Robotics and Automation Letters*, 2020.

ASAM. ASAM OpenSCENARIO: User Guide. *https://releases.asam.net/OpenSCENARIO/.*

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*, 2023.

Baidu. Apollo Auto. *https://github.com/ApolloAuto/apollo.*, 2020.

Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Błażej Osiński, Hugo Grimmett, and Peter Ondruska. Simnet: Learning reactive self-driving simulations from real-world observations. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5119–5125. IEEE, 2021.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align Your Latents: High-Resolution Video Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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*, 2020.

Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with LLMs: Fusing Object-Level Vector Modality for Explainable Autonomous Driving. *arXiv preprint arXiv:2310.01957*, 2023.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. Drive as You Speak: Enabling Human-Like Interaction with Large Language Models in Autonomous Vehicles. *arXiv preprint arXiv:2309.10228*, 2023.

Wenhao Ding, Yulong Cao, Ding Zhao, Chaowei Xiao, and Marco Pavone. RealGen: Retrieval Augmented Generation for Controllable Traffic Scenarios. *arXiv preprint arXiv:2312.13303*, 2023.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning*, 2017.

Lan Feng, Mohammadhossein Bahari, Kaouther Messaoud Ben Amor, Éloi Zablocki, Matthieu Cord, and Alexandre Alahi. Unitraj: A unified framework for scalable vehicle trajectory prediction. *arXiv preprint arXiv:2403.15098*, 2024.

Daniel J Fremont, Edward Kim, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L Sangiovanni-Vincentelli, and Sanjit A Seshia. Scenic: A language for scenario specification and data generation. *Machine Learning*, 2022.

Epic Games. Unreal Engine 4. *https://www.unrealengine.com.*

Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2021.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Xiaosong Jia, Penghao Wu, Li Chen, Hongyang Li, Yu Liu, and Junchi Yan. HDGT: Heterogeneous Driving Graph Transformer for Multi-Agent Trajectory Prediction via Scene Encoding. Apr 2022.

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.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, 2019.

Quanyi Li, Zhenghao Mark Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in neural information processing systems*, 36, 2024.

Weizi Li, David Wolinski, and Ming C Lin. ADAPS: Autonomous driving via principled simulations. In *International Conference on Robotics and Automation (ICRA)*, 2019.

Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. GPT-Driver: Learning to Drive with GPT. *arXiv preprint arXiv:2310.01415*, 2023.

NVIDIA. NVIDIA Drive - autonomous vehicle development platform. *https://developer.nvidia.com/drive.*, 2020.

OpenAI. GPT-4 Technical Report. *ArXiv*, 2023.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and Pace: Controllable Pedestrian Animation via Guided Trajectory Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Runway Research. Gen-2: The next step forward for generative ai. *https://research.runwayml.com/gen2.*, 2023.

Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *IEEE International conference on intelligent transportation systems (ITSC)*, pp. 1–6. IEEE, 2020.

Tanmay Samak, Chinmay Samak, Sivanathan Kandhasamy, Venkat Krovi, and Ming Xie. AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin Ecosystem for Autonomous Driving Research & Education. *Robotics*, 2023.

Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. LanguageMPC: Large Language Models as Decision Makers for Autonomous Driving. *arXiv preprint arXiv:2310.03026*, 2023.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics: Results of the 11th International Conference*, 2018.

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

Gustavo Silvera, Abhijat Biswas, and Henny Admoni. DReye VR: Democratizing Virtual Reality Driving Simulation for Behavioural & Interaction Research. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2022.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 2020.

Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10400–10409, 2021.

Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation. *arXiv preprint arXiv:2307.07947*, 2023.

Unity Technologies. Unity. *https://unity.com/.*

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

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*, 2023.

Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities. *Microsoft Auton. Syst. Robot. Res*, 2023.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in Neural Information Processing Systems*, 2022.

Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. DiLu: A Knowledge-Driven Approach to Autonomous Driving with Large Language Models. *arXiv preprint arXiv:2309.16292*, 2023.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. BITS: Bi-level Imitation for Traffic Simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2929–2936. IEEE, 2023.

Kairui Yang, Zihao Guo, Gengjie Lin, Haotian Dong, Die Zuo, Jibin Peng, Zhao Huang, Zhecheng Xu, Fupeng Li, Ziyun Bai, et al. Natural-language-driven Simulation Benchmark and Copilot for Efficient Production of Object Interactions in Virtual Road Scenes. *arXiv preprint arXiv:2312.04008*, 2023.

Qichao Zhang, Yinfeng Gao, Yikang Zhang, Youtian Guo, Dawei Ding, Yunpeng Wang, Peng Sun, and Dongbin Zhao. Trajgen: Generating realistic and diverse trajectories with reactive and feasible agent behaviors for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24474–24487, 2022.

Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. DriveDreamer-2: LLM-Enhanced World Models for Diverse Driving Video Generation. *arXiv preprint arXiv:2403.06845*, 2024.

Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. In *Conference on Robot Learning*, 2023a.

Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *IEEE International Conference on Robotics and Automation*, 2023b.

More demos and data are available at https://github.com/anonymous-github-Traj-LLM/Traj-LLM.

A DETAILS OF L2T

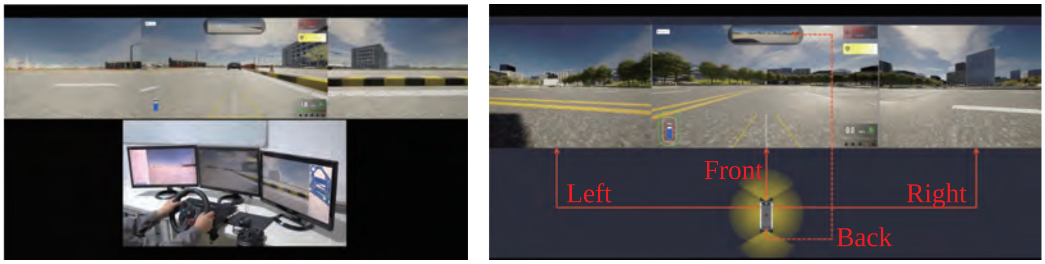
In this section, we provide details of the L2T dataset. Each scenario in L2T is organized into individual JSON format files containing the following components.

Interaction Section The interaction section is a summary of the map topology, the number of the agents, and their interactions. The interaction section of the example below serves as the context for understanding the interaction label.

Behavior Section In the behavior section, we provide the initial position (x, y), heading, and speed of each agent. Each agent is associated with a sequence of behaviors and their logic descriptions. Each behavior has a tag of KEEPSTILL, CRUISE, STOP, or CHANGELANE.

Trajectory Section In the trajectory section, we provide the sequence of motion parameters for each agent for representing its trajectory. The motion parameters are x/y-coordinates, x/y-accelerations, yaw, and rest time. We organize the sequence of motion parameters in a frame-by-frame fashion.

```
{
 "scenario_label": {
    "map": "straight road",
    "interaction": "egocar0 bypasses npccar3821",
    "agent_nums": 2
 },
 "agent_behavior": {
    "egocar0": {
        "init": {
            "x": 401.938,
            "y": -303.069,
            "heading": 1.7837963267948966,
            "speed": 0
        },
        "behaviors": [
            {
                "behavior": "KEEPSTILL",
                "description": "wait for 1 seconds before the scenario start"
            },
            {
                "behavior": "CRUISE",
                "description": "To avoid collision, the egocar bypasses the npccar3821 from
                    the left side."
            }
        ]
    },
    "npccar3821": {
        "init": {
            "x": 394.742,
            "y": -253.616,
            "heading": 2.0107963267948965,
            "speed": 0
        },
        "behaviors": [
            {
                "behavior": "CRUISE",
                "description": "The npccar3821 is driving straight ahead."
            }
        ]
    }
 },
 "agent_trajectory": {
    "frame0": {
        "egocar0": {
                "agenttype": "AgentType.EGO",
                "position_x": 401.938,
                "position_y": -303.069,
                "heading": 1.7837963267948966,
                "speed": 0.0
            },
            "npccar3821": {
                "agenttype": "AgentType.NPC",
                "position_x": 394.742,
                "position_y": -253.616,
                "heading": 2.0107963267948965,
                "speed": 0.0
            }
        },
        "frame1": {
            "egocar0": {
                "agenttype": "AgentType.EGO",
                "position_x": 401.93798828125,
                "position_y": -303.069000244141,
                "heading": 1.7837963267948966,
                "speed": 0.0
            },
            "npccar3821": {
                "agenttype": "AgentType.NPC",
                "position_x": 394.742004394531,
                "position_y": -253.615997314453,
                "heading": 2.0107963267948965,
                "speed": 0.0
            }
        },
        ...
    }
}
```



(a) (b)

Figure 7: (a) The hardware component of our simulator. (b) The first-person view of the virtual road environment.

A.1 DETAILS ABOUT THE COLLECTION AND ANNOTATION OF L2T

Our simulator integrates the re-developed hardware and software to efficiently collect and annotate the vehicle trajectories in the L2T dataset.

The hardware component of our simulator mainly includes a steering wheel, a gear shifter, gas and brake pedals, and several displays, as shown in Figure 7(a) of the supplementary file. The displays visualize the first-person view of the road environments from the left, right, front, and back sides of the car, as shown in Figure 7(b). Here, the road environments are simulated by LGSVL, a simulation engine for autonomous driving. Our hardware simulates a realistic car cockpit, helping users to drive on the virtual road to produce vehicle trajectories as real as possible.

The software component is mainly built on top of the LGSVL engine. LGSVL can record all trajectories of vehicles on the virtual road. We re-develop this engine to enable the connection among

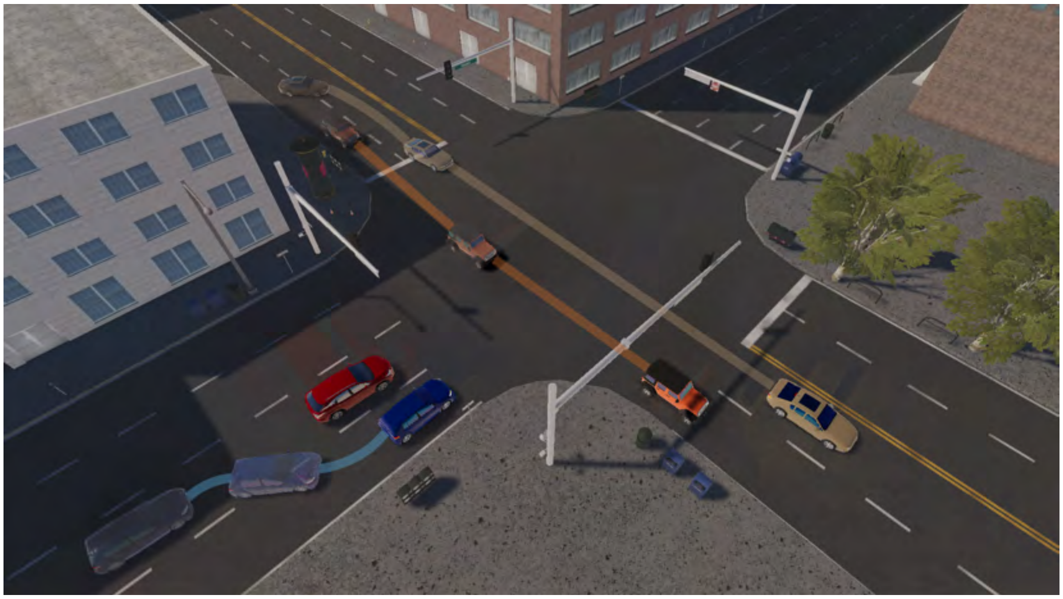


Figure 8: Multiple users control different cars on the same virtual road.

multiple sets of the above hardware, allowing multiple users to control different cars on the same virtual road. This software facilitates the simulation of vehicle interactions like those in the real world. In Figure 8, we show the interactive cars in the virtual road from the third-person view.

We select drivers as the primary participants to create the L2T dataset. Based on the manually designed map, the drivers obtain trajectories by operating the above simulators. We recruit 30 drivers with rich experience to explain and document vehicle interactions and behaviors with driving logic. Here, we conduct a two-level check to evaluate the annotation quality. We develop an automated Python script at the first check level to verify whether the trajectories align with the text interactions and behaviors. These scripts incorporate some basic checking rules, such as the rule that the overtaking between two vehicles must involve acceleration, and the behavior parameters within the acceleration must be equivalent to the motion parameters of the trajectories.

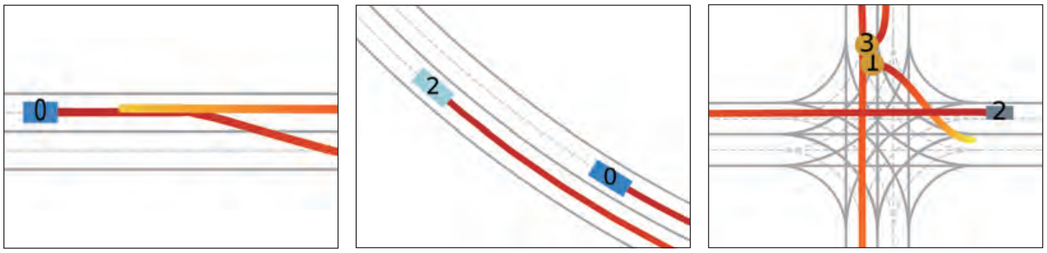
At the second check level, we divide the scenarios that pass the first check level into 10 groups. Three drivers check the annotations of each group of scenarios. We randomly select 20% of scenarios from each group. If half of the selected scenarios have false annotations, all scenarios in this group must be re-annotated; otherwise, only the scenarios with false annotations are re-annotated.

Since our work concerns the driving logic of drivers in the real world, the trajectories collected through our method closely resemble the real-world data.

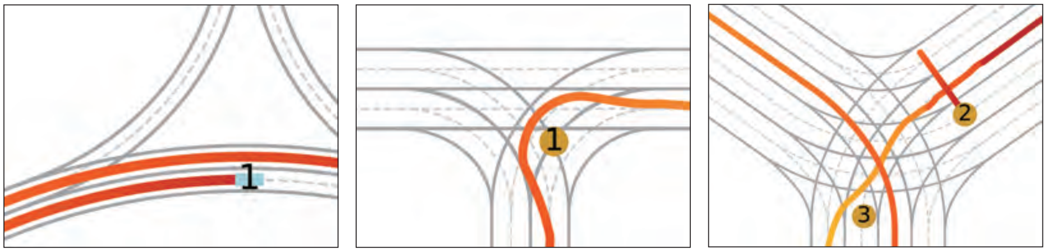
A.2 MAPS AND DATA DISTRIBUTIONS

The L2T dataset contains six kinds of road topologies, including straightway, bend, roundabout, cross/T-shaped/Y-shaped intersection. We provide the examples of maps in Figure 9. In Figure 10(a), we report the proportion of each type of map. Due to the complexity of roundabout maps and the high cost of annotations required, the proportion of roundabout data in the dataset is relatively low. The proportion of interactions and combinations under each type of map is relatively uniform.

In Figure 10(b), we further report the proportion of different types of behaviors that occur along with each type of interactions and combinations. Since most vehicles in the dataset are in driving mode at most of the time, the proportion of cruising is the highest among the optional behaviors.

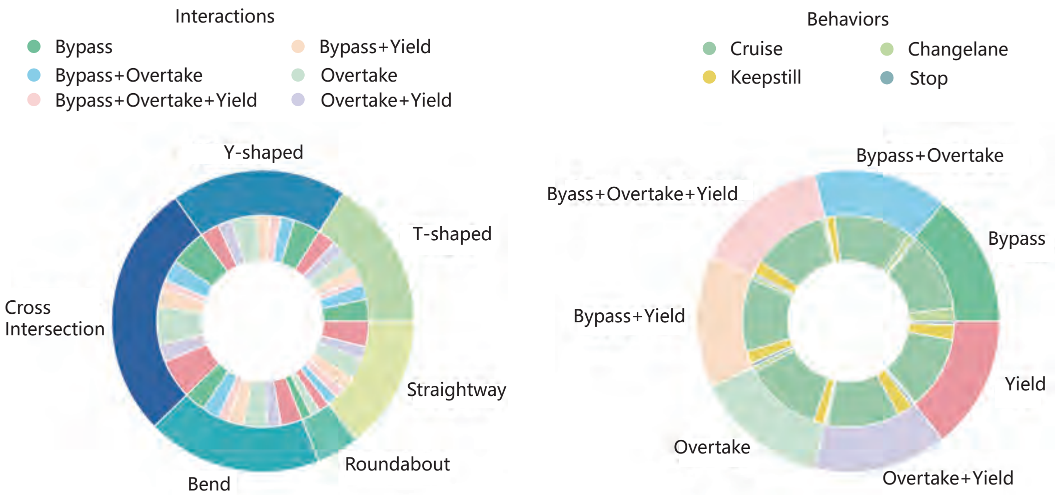


Straightway Bend Cross Intersection



Roundabout T-shaped Intersection Y-shaped Intersection

Figure 9: Examples of maps.



(a) The proportion of interactions and combinations under each type of map.

(b) The proportion of different types of behaviors that occur along with each type of interactions and combinations

Figure 10: Statistics of the L2T dataset.

A.3 COMPARISONS BETWEEN L2T AND OHTER DATASETS

To validate the characteristics of L2T in comparison to real-world datasets, we conduct an in-depth comparison between L2T and other datasets like Waymo. Figure 11 illustrates the similarity between L2T and other datasets regarding vehicle trajectory shapes and the number of objects of interest within the scene. It should be noted that due to the removal of traffic cones in the WOMD dataset. L2T retains them as stationary objects, L2T exhibits a higher number of stationary in overall trajectory shapes.

In Figure 12, we have compared the data distribution of our L2T and other datasets with trajectories in the real world. Specifically, we compare the distributions of trajectories' shapes represented by the waypoints' coordinates. The distributions are similar, showing that the vehicle trajectories in the L2T dataset are realistic, like those in the Argoverse and WOMD datasets.

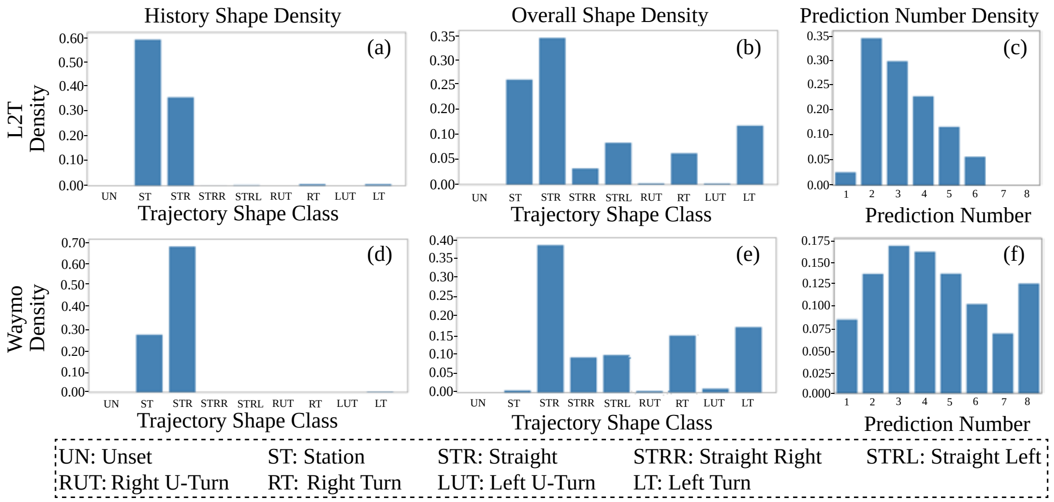
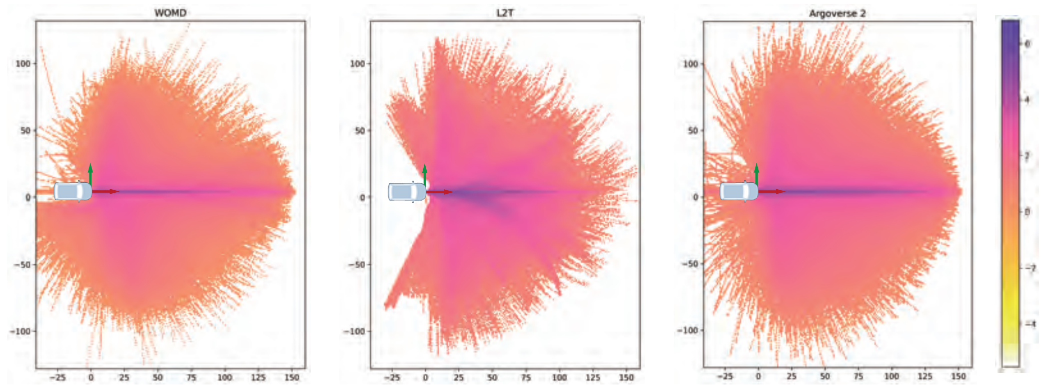


Figure 11: Comparison of the L2T and WOMD datasets in terms of trajectory shapes and prediction numbers.



(a) (b) (c)

Figure 12: Comparison of trajectories' shape distributions of (a) Argoverse, (b) Waymo, and (c) L2T datasets.

A.4 CORE IMPACT IN L2T AND TRAJ-LLM

Our method can generate strong interaction trajectory data similar to real scenes through interactive text and can improve real data's performance, proving our effectiveness. To better quantify it, we propose a novel metric: the scene-level closest interaction distance (CID), which is defined as the average of the closest distances between the ego and other vehicles/obstacles in the current scene during the entire movement process. Figure 13 shows the density map (green and red regions) of the CID in the L2T dataset and the WOMD dataset. It can be seen that the closest interaction distance of our data is closer and the interactivity is stronger. Meanwhile, the scatter plot shows that after adding our data, the improvement of WOMD's performance is mainly concentrated in the interaction scenes with closer distances, further validating the effectiveness of our method.

B EVALUATION METRICS

We use the metrics below to evaluate the realism, diversity, and controllability.

Realism To evaluate the realism of the generated trajectories, we compare them with the ground truth trajectories. This comparison is computed via the Wasserstein distance between the normalized

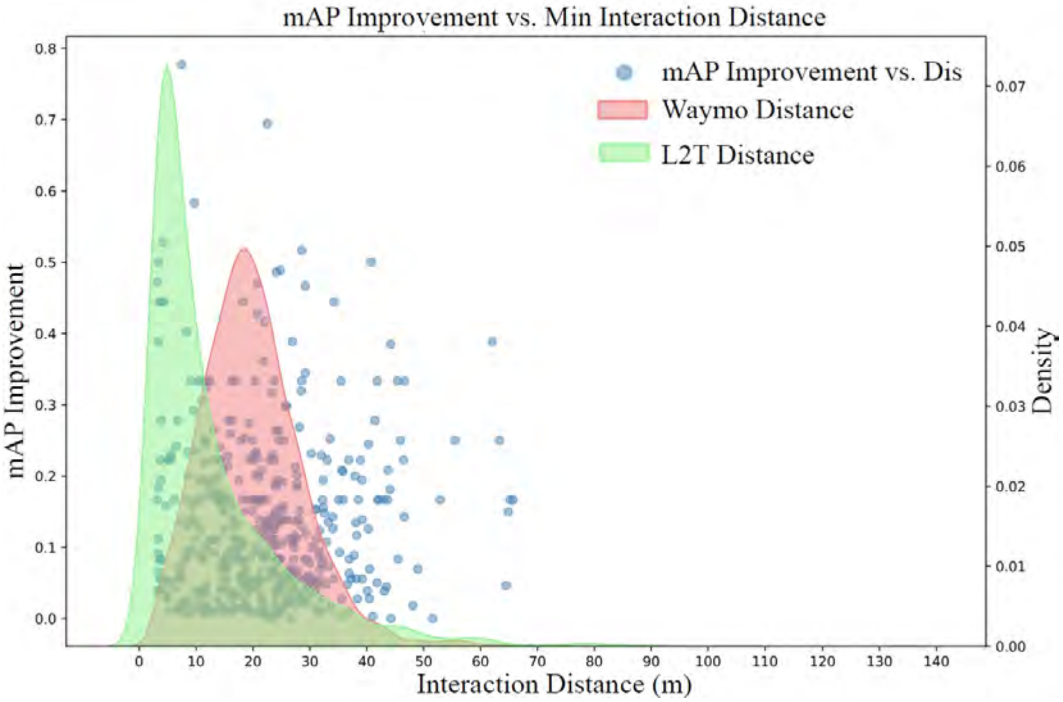


Figure 13: The distribution of interaction distances and the improvement in mAP when adding L2Tdata into the training set for MTR, as evaluated on the WOMD (val).

histograms of the driving profiles for the generated and ground truth trajectories as:

$$\mathbf{Realism}_X = \mathcal{W}(\mathcal{N}orm(Hist(\mathbf{X}_{gen})), \mathcal{N}orm(Hist(\mathbf{X}_{gt}))). \tag{9}$$

In Eq (9), the symbol $\mathcal{W}$ represents the Wasserstein distance, while X is the driving profile being compared. Realism is measured at both the agent and scene levels. Agent-level realism involves comparing four driving profiles: longitudinal acceleration magnitude (**LO**), latitudinal acceleration magnitude (**LA**), jerk (**JE**), and yaw rate (**YR**).

Scene-level realism, on the other hand, compares four relative driving profiles between agents: relative longitudinal acceleration magnitude, relative lateral acceleration magnitude, relative jerk, and relative yaw rate.

Diversity We evaluate the agent- and scene-level diversity. We use the average self distance(AD), final self distance(FD), and map-aware average self distance (MD) to measure agent-level diversity of trajectories generated multiple times.

In Eq. (10), **AD** denotes the average L2 distance of all waypoint positions between the two closest generated trajectories for each agent.

$$\mathbf{AD} = \frac{1}{NDS} \sum_{n=1}^{N} \sum_{d=1}^{D} \min_{d' \in \mathcal{D} \wedge d' \neq d} \sum_{s=1}^{S} \left\| \mathbf{T}_s^{n,d} - \mathbf{T}_s^{n,d'} \right\|^2 . \tag{10}$$

In Eq. (11), **FD** denotes the average L2 distance of the final position between the two closest generated trajectories for each agent.

$$\mathbf{FD} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{d=1}^{D} \min_{d' \in \mathcal{D} \wedge d' \neq d} \left\| \mathbf{T}_S^{n,d} - \mathbf{T}_S^{n,d'} \right\|^2 . \tag{11}$$

In Eq. (12), **MD** denotes the average L2 distance of all agents' trajectories between the two most distinct generated scenarios.

$$\mathbf{MD} = \max_{d,d' \in \mathcal{D}} \frac{1}{NS} \sum_{n=1}^{N} \sum_{s=1}^{S} \left\| \mathbf{T}_s^{n,d} - \mathbf{T}_s^{n,d'} \right\|^2 . \tag{12}$$

To measure scene-level diversity, we use Kernel Density Estimation to calculate trajectory density profile ρ for each sample scenario. We calculate the average Wasserstein distance ($\mathbf{WD}$ in Eq. (13)) of density profiles for pairwise sample scenarios, as the scene-level diversity score.

$$\mathbf{WD} = \frac{2}{D(D-1)} \sum_{i=1}^{D-1} \sum_{j=i+1}^{D} \mathrm{W}\left(\rho_i, \rho_j\right). \tag{13}$$

In Eqs. (10)-(13), N is the number of agents in each traffic scenario. $\mathcal{D}$ is the set of generated samples for each scenario and D is the number of samples in $\mathcal{D}$. S is the length of generated trajectories.

Controllability For text-controlled methods, we evaluate the controllability of generated trajectories via the success rate (**SR** in Eq. (14)) of achieving text-specified interaction (overtake, bypass, and yield), which is the portion of interactive samples (IS) to total samples (TS).

$$\mathbf{SR} = \frac{IS}{TS}. \tag{14}$$

B.1 THE CALCULATION OF CONTROLLABILITY

We utilize a rule-based automatic behavior detection method to calculate **SR**. This method determines behaviors (such as overtake, bypass, yield, etc.) based on the interaction distance, speed, trajectory shape, etc. of the vehicle during movement, and assigns a confidence level reflecting the certainty of the results. Notably, we apply the same judging method for all methods, and only incorporate high-confidence results in our controllability calculations.

C IMPLEMENTATION DETAILS OF TRAJ-LLM

Llama-7B (Touvron et al., 2023) is the language model used in Traj-LLM. We train Llama with 8×A800 GPUs. We utilize Low-Rank Adaptation (Hu et al., 2021) (LoRA) with a rank of 8 to fine-tune Llama. The LoRA alpha value is set to 32 to control the relative strength of the update compared to the original model's weights. A dropout rate of 0.1 is applied in LoRA to prevent overfitting. The training process span two epochs with a batch size of 16 for each iteration. We set the learning rate to 2e-5. The training data comprises 200K samples from the L2T training set. The entire training takes about 14 hours.

D ABLATION STUDY OF GENERATED DATA AND L2T

In Table 5, we combine the original data, i.e. L2T, with the training data to train downstream trajectory prediction models MTR(Shi et al., 2022).

By adding all the data from **WOMD (Train)**, the trajectories of **L2T (train)** and the generated trajectories in **Traj-LLM (L2T)** (the last row), MTR performs better than without **L2T (train)** (the fourth row) or **Traj-LLM (L2T)** (the first row). This is because the L2T dataset is collected from real-world scenarios, which reflects the complexity and diversity of the real world. The real-world nature of the data enables the trajectory prediction models to learn from the real-world variations. Furthermore, Traj-LLM can generate numerous trajectories to enrich the training data. By combining the original data in **L2T (train)** with the generated trajectories in **Traj-LLM (L2T)**, we leverage the strengths of both, enhancing the accuracy and robustness of the trajectory prediction models.

Given the training data from **WOMD (Train)**, MTR achieves a better performance when it is trained on the original trajectories in **L2T (train)** and the generated trajectories in **Traj-LLM (L2T)** (the last row), compared to the training on the generated trajectories only (the fourth row). Though the generated trajectories may be diverse, they may still fail to capture the real-world trajectories' natural pattern fully. Combining generated data with original data can achieve a better balance in the training set. We find that this improvement is marginal. To some extent, this demonstrates the high quality of the generated trajectories that serve as training examples to strengthen the model's generalization of unseen scenarios.

In the second, third, and fourth rows of Table 5, the performance of MTR improves with increasing amounts of generated trajectories. This enhancement is attributed to using richer trajectory data from the Traj-LLM to train the trajectory prediction model.

Table 5: Results of trajectory prediction on MTR.

Train			**Test (mAP ↑)**
WOMD (train)	**L2T (train)**	**Traj-LLM (L2T)**	**WOMD (val)**
100%	**100%**	**0**	0.410
100%	**0**	**30%**	0.397
100%	**0**	**50%**	0.412
100%	**0**	**100%**	0.416
100%	**100%**	**100%**	**0.417**

In order to separate the utility of Traj-LLM data from just the additional data volume, we conduct another version of Table 5 with fixed total dataset size, the results are reported in Table 6. we define 100% as the same size as WOMD(train). It can be seen that the performance on the WOMD(val) drops accordingly as the proportion of WOMD (train) decreases. This is because WOMD and Traj-LLM (L2T) have different distributions (see A.3,A.4). Specifically, Traj-LLM (L2T) focuses on strong interaction scenarios, whereas WOMD distributes on weak interaction scenarios. When combining Traj-LLM (L2T) and WOMD (train) as the training set, the test set WOMD (val) has a different distribution from the training set. This will lead to a decrease in test performance. Furthermore, the generated data, Traj-LLM (L2T), can improve the trajectory prediction performance in strong-interaction scenarios (see A.4), as evidenced in Table 4, thereby demonstrating the effectiveness of our dataset.

Table 6: Results of trajectory prediction on MTR under fixed total dataset size.

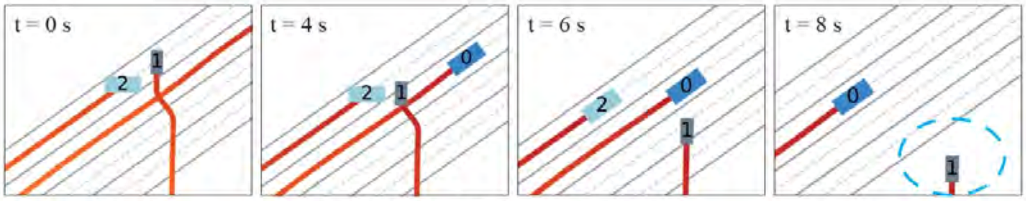
Train		**Test (mAP ↑)**
WOMD (train)	**Traj-LLM (L2T)**	**WOMD (val)**
100%	**0**	0.405
50%	**50%**	0.383
30%	**70%**	0.353

E LIMITATION ANALYSIS

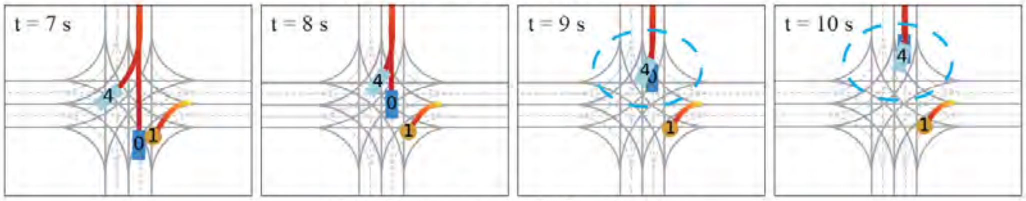
In Table 7, we report the proportion of different failure types within failure cases of Traj-LLM. For instance, when Traj-LLM fails to generate an Overtake interaction in the Straightway map, we classify the failure into three typical types: collision (**Collision**), off-road (**Off-road**), and the miss of the specific interaction (**No-Interaction**). We then calculate the proportion of each type for different combinations of Interaction and Map. Figure 14 shows failure cases.

Table 7: Proportion of different types within failure cases of Traj-LLM.

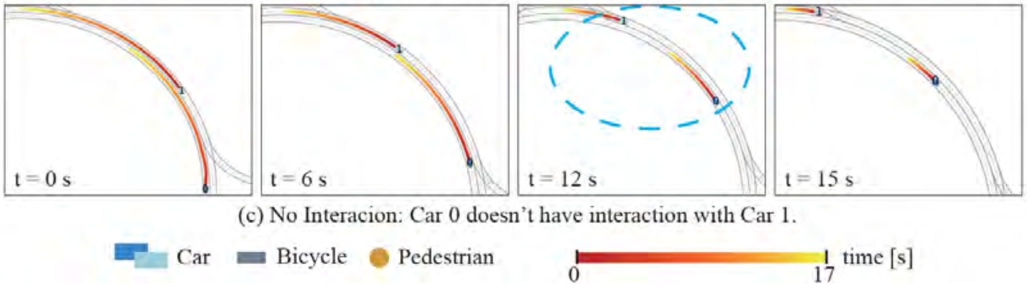
Interaction		**Straightway**	**Bend**	**Roundabout**	**Cross**	**Y-shaped**	**T-shaped**
Overtake	**Collision**	0.795	0.516	0.719	0.311	0.688	0.660
	Off-Road	0.045	0.042	0.094	0.070	0.020	0.181
	No-Interaction	0.160	0.442	0.187	0.619	0.292	0.159
Bypass	**Collision**	0.693	0.642	0.300	0.581	0.685	0.961
	Off-Road	0.132	0.074	0.200	0.245	0.201	0.029
	No-Interaction	0.175	0.284	0.500	0.174	0.114	0.010
Yield	**Collision**	0.695	0.369	0.222	0.656	0.752	0.789
	Off-Road	0.024	0.331	0.334	0.315	0.067	0.132
	No-Interaction	0.281	0.300	0.444	0.029	0.181	0.079



(a) Off-Road: Bicycle 1 goes off road.



(b) Collision: Car 0 collides with Car 4.



(c) No Interacion: Car 0 doesn't have interaction with Car 1.

Figure 14: The visualization of trajectories generated by Traj-LLM that resulted from different failure reasons.

The failure rate of **No-Interaction** is relatively lower than the other two types, highlighting the controllability of Traj-LLM. **Collision** and **Off-Road** occur more frequently.

We conjecture that the above problem mainly stems from the polyline encoder employed by Traj-LLM's failure to interpret complex maps accurately. In our implementation, Traj-LLM heavily relies on a standalone polyline encoder to embed the map information into the hidden feature, which is concatenated with the interaction and behavior features for generating the vehicle trajectories. It means that Traj-LLM does not comprehensively parse the map information. This differs from the text of vehicle interactions and behaviors that are well parsed by Traj-LLM and embedded into features. In the future, we will follow two directions to solve the above problem:

- We can investigate how to transform the map's polylines into text properly. Traj-LLM, which has been pre-trained on a large amount of textual data, may be better able to parse the textual map.
- We can also study how to make Traj-LLM a large visual-language model. The model can regard the map as an image representing not only the lane lines but also other environmental factors like trees, buildings, etc. The model trained on the map images can parse the maps with complex environmental factors, generating the vehicles that fit the maps.

F ADDITIONAL VISUAL CASES

In Figures 15 and 16, we visualize the generated trajectories with the interaction of yielding and bypassing. We provide more visualization results of trajectories generated by Traj-LLM in Figure 17. The Traj-LLM method can generate interactions between vehicles and pedestrians, bicycles, and traffic cones (see Figure 18).

In Figure 19, we show the positive influence of adding the trajectories generated by Traj-LLM for training the trajectory prediction model, MTR. In each scenario, we focus on visualizations of each agent's ground-truth trajectory in gray and the one with the highest confidence value in color

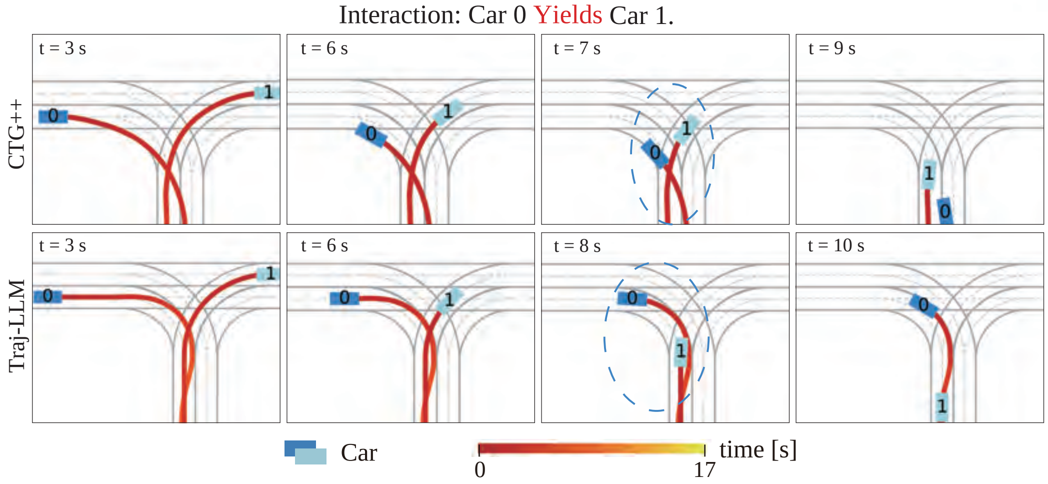


Figure 15: Controllability of CTG++ and Traj-LLM. In this example, Traj-LLM successfully control Car 0 to yield Car 1, while the state-of-the-art CTG++ fails in this case.

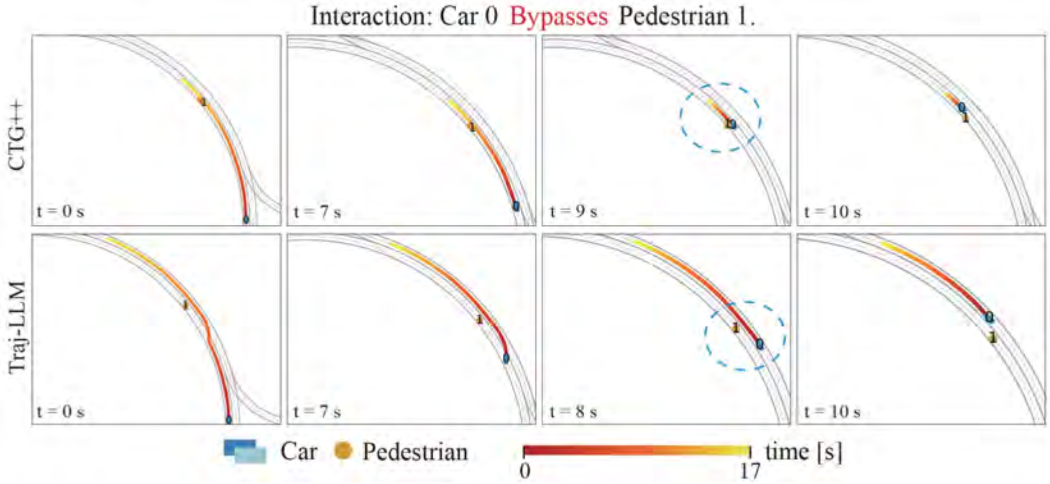
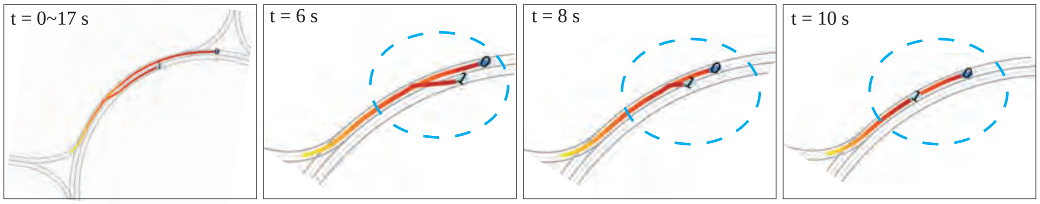


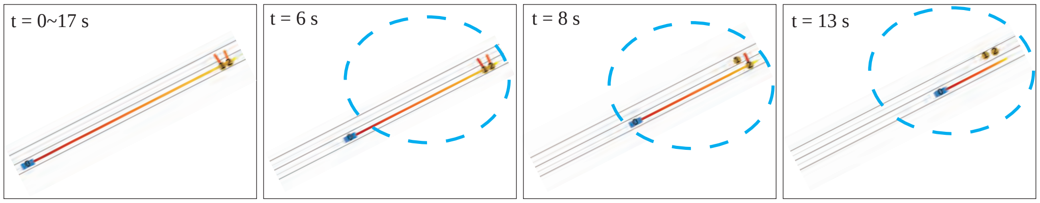
Figure 16: Controllability of CTG++ and Traj-LLM. In this example, Traj-LLM successfully control Car 0 to bypass Car 1, while the state-of-the-art CTG++ fails in this case.

predicted by MTR. By adding the generated trajectories to the training data, MTR reduces the cases of collision and off-road while predicting the interactive trajectories better than the model without training on the generated data.

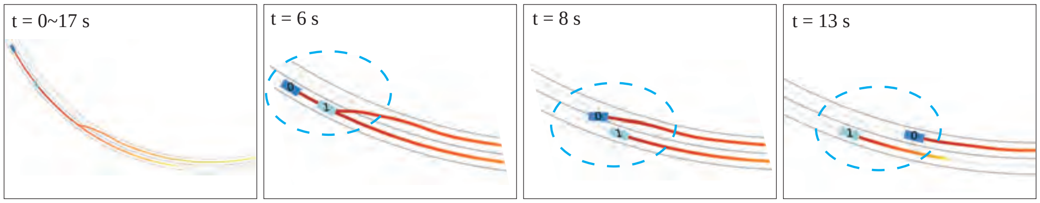
Interaction: Car 0 Yields Car 1.



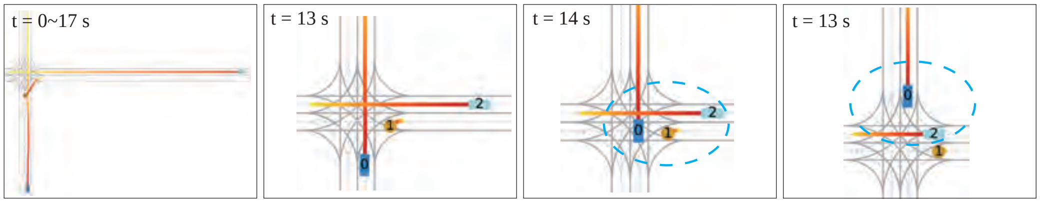
Interaction: Car 0 Yields Perdestrian 1, 2.



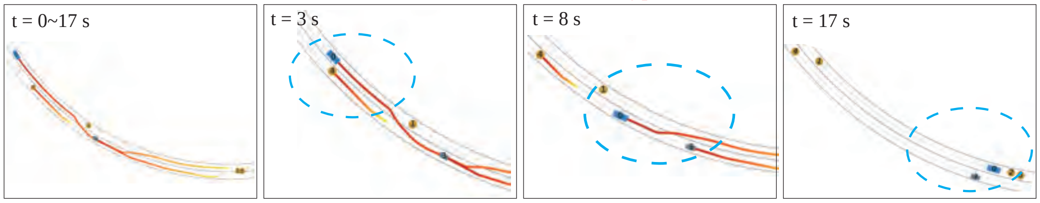
Interaction: Car 0 Overtakes Car 1.



Interaction: Car 0 Overtakes Car 2.



Interaction: Car 0 Overtakes Pedestrians 4, Bypasses Pedestrians 1.



Interaction: Car 0 Yields Car 1, 2, 5.

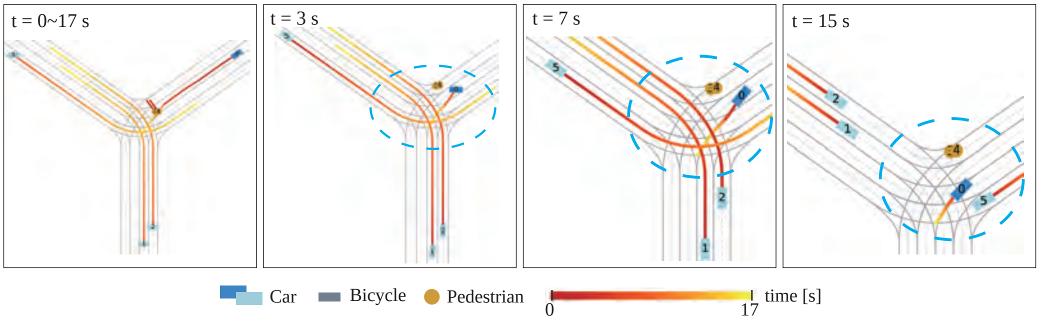
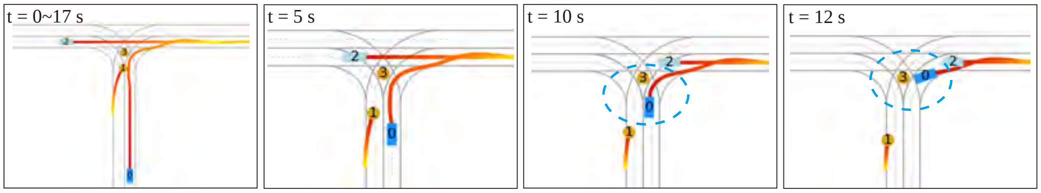
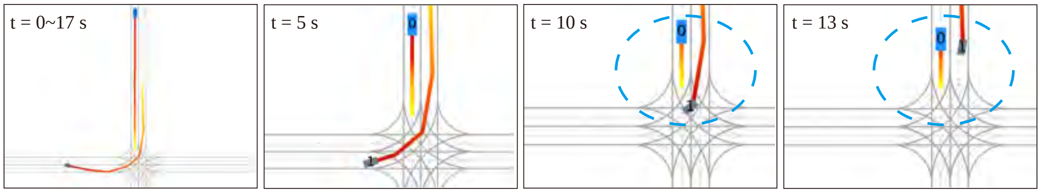


Figure 17: Visualization of trajectories generated by Traj-LLM in various road topologies with different interactions.

Interaction: Car 0 Bypasses Pedestrian 3.



Interaction: Car 0 Yields Bicycle 1.



Interaction: Car 0 Bypasses Traffic Cone 3,Traffic Cone 5.

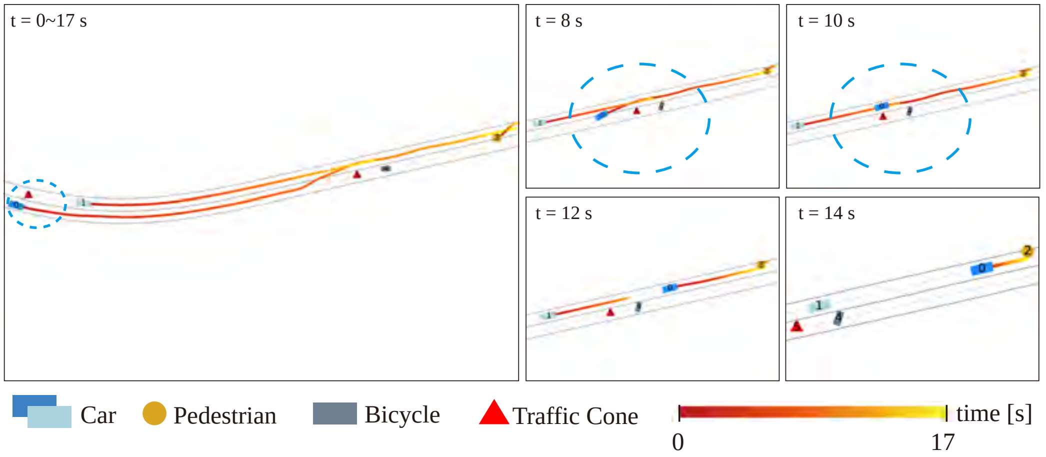


Figure 18: Visualization of trajectories generated by Traj-LLM, which demonstrates its ability to generalize well to scenarios involving traffic cones, bicycles, and pedestrians.

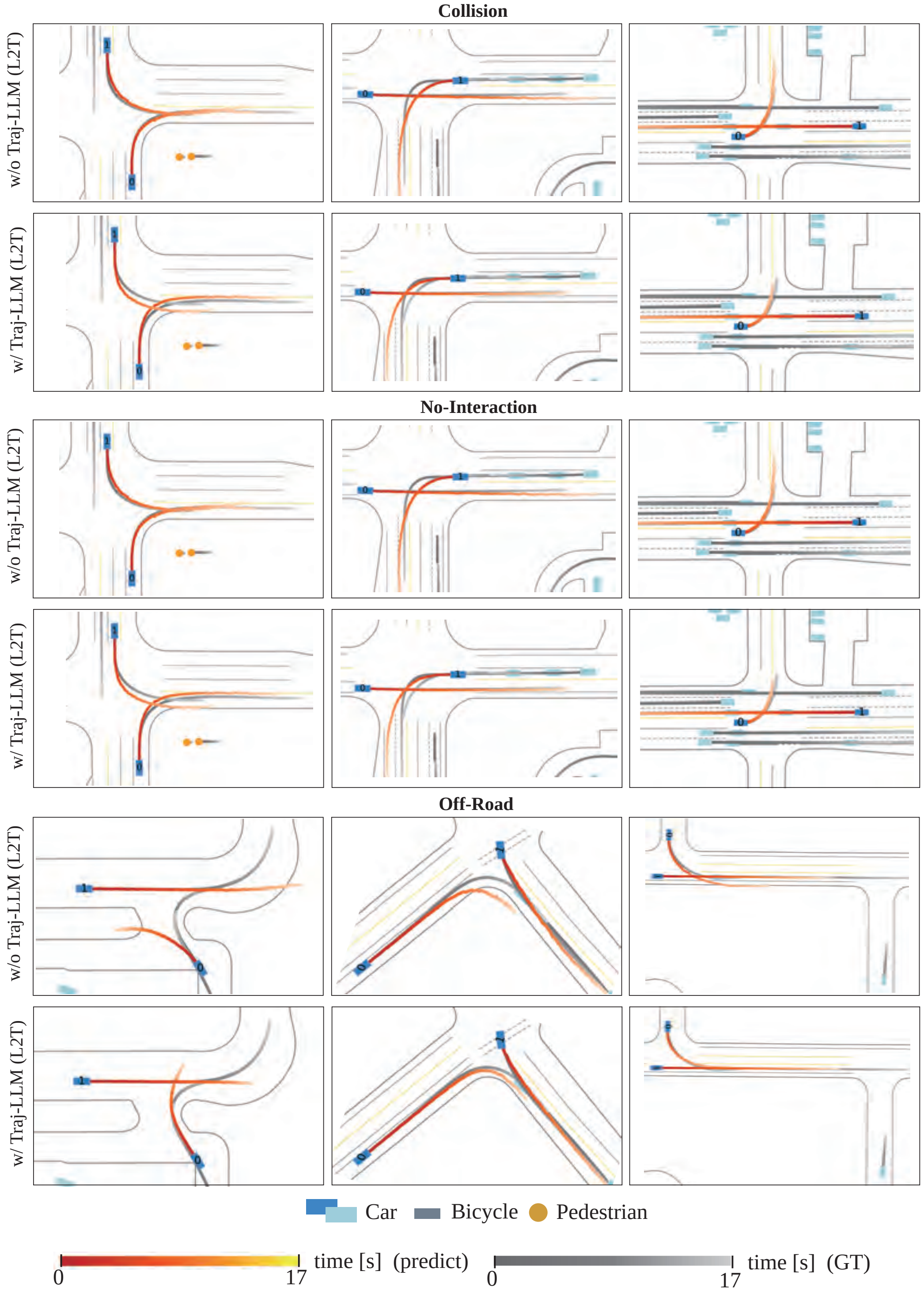


Figure 19: We add trajectories generated by Traj-LLM to train MTR, which reduces **Collision**, **No-Interaction**, and **Off-Road** in the trajectory prediction task.