Controllable Traffic Simulation through LLM-Guided Hierarchical Reasoning and Refinement

Zhiyuan Liu[1], Leheng Li[2], Yuning Wang[1], Haotian Lin[3],
Hao Cheng[1], Zhizhe Liu[4], Lei He[1], Jianqiang Wang[1,†]

Abstract— Evaluating autonomous driving systems in complex and diverse traffic scenarios through controllable simulation is essential to ensure their safety and reliability. However, existing traffic simulation methods face challenges in their controllability. To address this, we propose a novel diffusion-based and LLM-enhanced traffic simulation framework. Our approach incorporates a high-level understanding module and a low-level refinement module, which systematically examines the hierarchical structure of traffic elements, guides LLMs to thoroughly analyze traffic scenario descriptions step by step, and refines the generation by self-reflection, enhancing their understanding of complex situations. Furthermore, we propose a Frenet-frame-based cost function framework that provides LLMs with geometrically meaningful quantities, improving their grasp of spatial relationships in a scenario and enabling more accurate cost function generation. Experiments on the Waymo Open Motion Dataset (WOMD) demonstrate that our method can handle more intricate descriptions and generate a broader range of scenarios in a controllable manner.

I. INTRODUCTION

Autonomous driving has developed rapidly in recent years. To ensure the reliability of these systems, testing on complex, diverse, and long-tailed traffic scenarios through traffic simulation is essential and critical. Early simulation methods, such as rule-based approaches [1]–[3], offered good interpretability but lacked diversity and complexity. In contrast, deep learning techniques [4]–[7] introduced more complexity by learning from large datasets, but they often lack controllability. Without precise control over generated scenarios, it becomes challenging to systematically evaluate how autonomous driving systems respond to specific, critical situations.

To address this problem, recent work has focused on utilizing generative models to achieve controllable traffic simulation. One approach [8]–[10] involves using condition tokens to guide the model with basic conditions, such as goal points or velocity. However, this method requires additional training with the condition tokens and struggles to handle more complex conditions. Another approach leverages conditional sampling in diffusion models, which does not require extra training and offers more flexibility by optimizing cost functions during the sampling process. The cost functions

can be either human-designed [11]–[14] or generated by large language models (LLMs) [15], [16]. Human-designed cost functions typically include predefined rules, such as speed limits or avoiding collisions, but are limited in their ability to address conditions outside of predefined functions. On the other hand, methods based on LLM-generated cost functions mark a significant progression to deal with a wide range of conditions. They take user descriptions of a scenario as input and create corresponding cost functions by LLMs, which is more adaptable than predefined functions. However, current LLMs still face challenges in either processing intricate descriptions effectively or generating outputs that are easily guidable, resulting in a low success rate for producing correct traffic simulation.

Our goal is to overcome the limitations of simulation methods based on diffusion models and LLM-generated cost functions, further utilizing the powerful generative capabilities of LLMs and leveraging the flexibility of diffusion conditional sampling. To achieve this, we propose a novel LLM-based framework consisting of two key modules: a high-level understanding module, and a low-level refinement module.

The high-level module is designed to guide LLMs in processing complex conditions and generating appropriate cost functions. Here, we systematically design a novel chain-of-thought (CoT) mechanism [17] for traffic scene analysis, which enables LLMs to systematically break down the traffic events into hierarchical structures and generate corresponding cost functions step-by-step. By reasoning the interaction graphs in the traffic scene and breaking them down into manageable components, we demonstrate the significant advantages in understanding and processing complex traffic scenes. However, the generated cost functions may occasionally be suboptimal or challenging to control via diffusion guidance. To address this, we introduce a low-level refinement module, which takes the low-level trajectory coordinates as inputs and enables the LLM to perform self-reflection and iterative improvement. This module further refines the generated cost functions, enhancing the model's ability to comprehend complex scenes and improving the performance of guided simulation.

Moreover, the formulation of appropriate cost functions is crucial. Existing approaches often rely on target points [16] or project trajectories onto surrounding lanes and utilize the projection points [15] to construct cost functions. However, these methods present significant limitations. The use of target points overly restricts the feasible trajectory space, as

[1] School of Vehicle and Mobility, Tsinghua University.

[2] Artificial Intelligence Thrust, The Hong Kong University of Science and Technology (Guangzhou).

[3] School of Computer Science, Carnegie Mellon University.

[4] Department of Computer Sciences, University of Wisconsin-Madison.

[†] Corresponding author: wjqlws@tsinghua.edu.cn

a single description may correspond to multiple valid target points rather than a single one. Additionally, the geometric interpretation of projection points is inherently difficult for LLMs to comprehend, and the process of translating these projections into cost functions further complicates the task. To address these issues, we propose a novel Frenet-frame-based cost function framework. This approach leverages lateral and longitudinal information to construct cost functions, providing clear geometric interpretations and more reasonable constraints. We demonstrate that this Frenet-based method not only enhances the LLM's understanding of the scenario but also significantly improves the success rate of diffusion guidance.

Regarding the model architecture, we integrate the state-of-the-art trajectory prediction frameworks [18]–[20] with a relative coordinate system to ensure translational and rotational invariance for each agent in the scene. Furthermore, we formulate the agent's known trajectories as a cost function that serves as guidance during diffusion test-time sampling. This design provides a flexible framework that can accommodate various applications, including unconditional generation, and conditional simulation, without relying on extra training and complex models.

We evaluate our method on the Waymo Open Motion Dataset (WOMD) [21]. Experimental results demonstrate that the proposed LLM framework and Frenet-based cost functions can effectively handle a wider range of driving scenarios compared to existing methods. This includes not only typical driving behaviors, such as cut-ins, but also abnormal behaviors, such as going off-road. Additionally, our method achieves zero-shot ability on accident events despite being trained solely on a nearly collision-free dataset. Extensive experiments further validate that our approach delivers competitive performance across multiple tasks.

Our contributions are summarized as follows:

- We propose a novel LLM-based framework designed to interpret complex traffic scenarios and generate precise cost functions, enabling more accurate and reliable traffic simulation.
- We innovatively address the effectiveness of Frenet coordinate system in traffic simulation domain, significantly enhancing the generative capabilities of LLMs and improving the success rate of diffusion-guided scenario generation.
- Through extensive experiments, we demonstrate that our method can handle more intricate scenario descriptions and generate a wider variety of traffic scenarios in a controllable manner, including complex interactions and accident simulations.

II. RELATED WORK

A. Traffic simulation

Rule-based methods. Early methods utilize heuristic controllers or replayed logs to generate scenarios and use human-designed rules to control the generation [1]–[3]. These methods have good controllability and interpretability. However,

they lacked the realism and complexity to capture the full range of real-world situations.

Learning-based methods. Learning-based simulation methods leverage large-scale real-world datasets to simulate driving behaviors [21]–[24]. Early models in this domain [4]–[7] were designed similarly to trajectory prediction models [25], [26] but operated in a closed-loop manner. While these methods provide improved realism, diversity, and complexity, they often lack sufficient controllability. To address this, some approaches incorporate conditional models [8]–[15] to balance both realism and controllability.

Diffusion-based methods. More recent work leverages diffusion models for controllable simulation, benefiting from strong generative capabilities and test-time adaptability. Most of these models [11]–[14], [27] rely on pre-defined functions, such as goal point specification and collision avoidance, to guide the sampling process. While effective, their ability to control conditions beyond these predefined functions is constrained. CTG++ [15] and SceneDiffuser [16] marks a significant progression by utilizing large language models (LLMs) to translate user-specified descriptions into cost functions, which expands the range of controllable conditions. However, it still faces challenges in understanding and processing complex descriptions.

B. LLMs for robotics

The application of LLMs in robotics has garnered significant attention due to their ability to bridge the gap between natural language instructions and robotic control. Recent approaches leverage LLMs to generate both actions [28], [29] and reward functions [30]–[32], achieving notable progress in this area. Building on these advancements, CTG++ [15] and SceneDiffuser [16] explored the use of LLMs in traffic simulation, employing them to generate cost functions based on natural language descriptions of traffic scenarios. In this work, we aim to extend the generative capabilities of LLMs to construct more comprehensive and precise cost functions, enhancing the controllability of traffic simulations.

III. PRELIMINARY

A. Problem formulation

A traffic scenario can be defined as a combination of road topology m and agent states a. We use a set of N_m polylines for road topology to describe lanes [26], characterized by their coordinates and attributes (such as lane type). In addition, we denote the connectivity of the polylines (predecessor, successor, left neighbor, and right neighbor) as graph edges $E = \{e_{ij}\}$, to represent the road topology. For agent states, we focus primarily on the position and heading of each agent over a sequence of timesteps. An agent's state history and future are denoted as $a_i = \{s_{-T_{\text{hist}}}, ..., s_0, ..., s_{T_{\text{future}}}\}$, where $s_t = (x_t, y_t, \theta_t)$. Here, T_{hist} and T_{future} represent the lengths of the history and future trajectories, respectively.

Our goal is to simulate N_a agent future trajectories given (1) the road topology m, (2) the current or historical states of N_a agents, and (3) the language description specified by
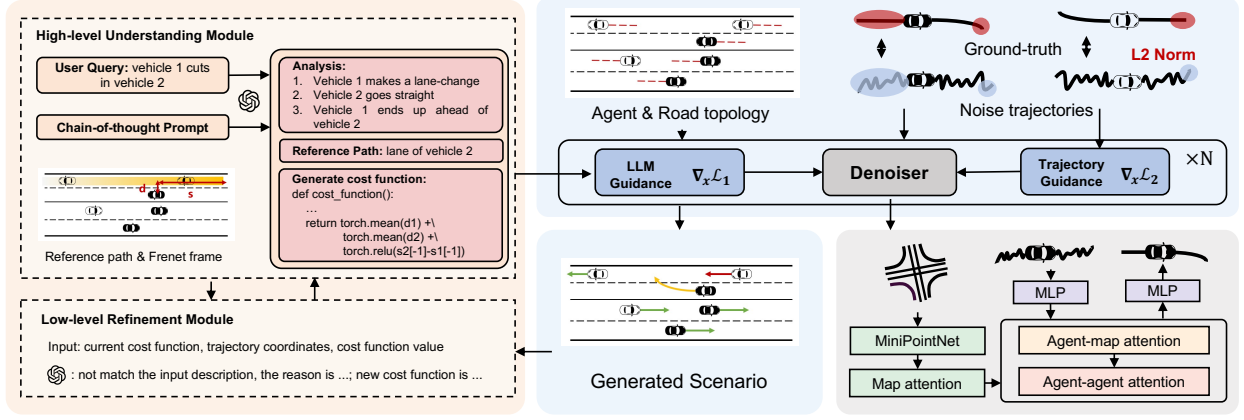
Fig. 1: **Method Overview**. We employ LLM to generate guidance and utilize a diffusion model to produce trajectories. The high-level module generates a cost function, which guides the transformer-based diffusion model to generate trajectories. If the trajectory generation fails, the coordinates are fed into the low-level module to further refine the cost function.

the user. These generated trajectories must be consistent with the scenario's language description.

B. Diffusion Models

Diffusion models [33] have emerged as a powerful framework to represent complex data distribution. In general, diffusion models parameterize data distribution as p_θ and learn to model it through score-matching, transforming a simple distribution (e.g., Gaussian noise) into a complex data distribution via multiple small, reversible steps. This process is modeled by stochastic differential equations (SDEs) [34]:

$$d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\omega \qquad (1)$$

$$d\boldsymbol{x} = [f(\boldsymbol{x}, t) - g^2(t)\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})]dt + g(t)d\omega \qquad (2)$$

where ω is the standard Wiener process. $\nabla_x \log p_t(x)$ is the time-dependent score, which can be approximated by a neural network. In our case, diffusion models are used to represent the distribution of trajectory features x conditioned on map topology $p_\theta(x; m)$.

Following EDM [35], the sampling process in this work is described by the ODE:

$$d\boldsymbol{x} = -\dot{\sigma}(t)\sigma(t)\nabla_x\log p(\boldsymbol{x}; m, \sigma(t))dt \qquad (3)$$

The denoiser model $D_\theta(\boldsymbol{x}; m, \sigma)$ is trained to recover the noised sample $\boldsymbol{x}$ from its noisy version. The relationship between the score function and the denoiser model is expressed as follows:

$$\nabla_{\boldsymbol{x}}\log p_\theta(\boldsymbol{x}; m, \sigma) = \frac{D_\theta(\boldsymbol{x}; m, \sigma) - \boldsymbol{x}}{\sigma^2} \qquad (4)$$

During training, the model is optimized by minimizing the following objective:

$$\mathbb{E}_{y\sim p_0, \boldsymbol{n}\sim\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})} \|D_\theta(\boldsymbol{y} + \boldsymbol{n}; m, \sigma) - \boldsymbol{y}\|^2, \qquad (5)$$

where p_0 is the data distribution and $\boldsymbol{n}$ represents the Gaussian noise. After training, data can be sampled from noise using the relationship in equation (4) and the ODE described in equation (3).

C. Conditional sampling

For various problems, we have extra condition c which can be described by a cost function $\mathcal{L}$. In such cases, we aim to sample data from the conditional distribution $p(\boldsymbol{x}; c)$. Under our specific problem settings, this translates to sampling from $p(\boldsymbol{x}; m, \sigma, c)$. Consequently, the score function should be modified from $\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}; m, \sigma)$ to $\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}; m, \sigma, c)$. According to Bayes' rule, the modified score function $\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}; c, m, \sigma)$ can be expressed as:

$$\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}; m, \sigma) + \nabla_{\boldsymbol{x}}\log p(c; \boldsymbol{x}, m, \sigma) \qquad (6)$$

The first term can be calculated by equation (4), and the second term can be approximated as:

$$\nabla_{\boldsymbol{x}}\log p(c; \boldsymbol{x}, m, \sigma) \approx \lambda\frac{\partial}{\partial\boldsymbol{x}}\mathcal{L}(D_\theta(\boldsymbol{x}; m, \sigma)), \qquad (7)$$

where λ is a scaling factor. By utilizing equations (3), (4) and (7), we can sample data that meets the condition c. In practice, to ensure the stability of the sampling process, a score threshold of $(-1, 1)$ is applied to equation (7).

IV. METHOD

The overall framework of our proposed method is illustrated in Figure 1. In this section, we first introduce the **Transformer-based Denoiser**, which models the complex interactions within traffic scenarios (see Section IV-A). Following this, we detail the LLM-based **Guided Generation** module in Section IV-B.

A. Transformer-based denoiser

Relative coordinate system. In alignment with recent trajectory prediction methods [18], [36], we adopt a relative coordinate system. Initially, road polylines and trajectories are rotated and normalized to the central or current position. To further capture spatial relationships among traffic elements, the feature vector $\boldsymbol{d}_{ij} = (\Delta x_{ij}, \Delta y_{ij}, \cos\Delta\theta_{ij}, \sin\Delta\theta_{ij})$ is extracted, where $(\Delta x_{ij}, \Delta y_{ij})$ represents the relative positions within the reference frame of agent i, and $\Delta\theta_{ij}$ denotes the relative heading angle. Note that the feature vectors between static elements, such as road polylines,

remain consistent regardless of the agents' movements. Consequently, this approach eliminates the need to re-encode the map topology during simulation rollout, thereby enhancing inference speed.

Trajectory conditional guidance. Our approach treats the known trajectory states A_{known} as an additional condition rather than embedding it directly into the model. Consequently, our model learns the distribution $p(x; m)$ and employs the sampling method introduced in Section III-C to generate data conditioned on historical states. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ denote the set of timesteps where condition states are provided. The predicted trajectory states at these timesteps are represented as $\boldsymbol{y}_{\text{known}} = \{\boldsymbol{y}_{t_i} \mid t_i \in \mathcal{T}\}$, and the corresponding ground-truth states are $\hat{\boldsymbol{y}}_{\text{known}} = \{\hat{\boldsymbol{y}}_{t_i} \mid t_i \in \mathcal{T}\}$. The condition is expressed using the L2 norm:

$$\mathcal{L}_{\text{known}} = \|\boldsymbol{y}_{\text{known}} - \hat{\boldsymbol{y}}_{\text{known}}\|^2 \tag{8}$$

By incorporating this loss term, we effectively integrate trajectory conditions without requiring separate encoders and additional training. This design enables our method to flexibly handle various tasks, such as unconditional prediction and conditional simulation.

Model architecture. The model architecture is shown in the bottom right of Figure 1. We employ a MiniPointNet [37] which is primarily composed of max pooling and MLPs to obtain polyline-level features from the road topology feature M. The encoding process can be expressed as:

$$X_m = \text{MiniPointNet}(M), X_m \in \mathbb{R}^{N_m \times d} \tag{9}$$

where X_m is the polyline-level feature and d denotes the hidden size of the model. For agent features $X_a \in \mathbb{R}^{N_a \times d}$, we embed the trajectories into PCA space following [12] and then utilize an MLP to encode the noise trajectories.

After feature encoding, we employ three distinct attention modules to extract the lane-lane, agent-lane, and agent-agent interactions. Let $\boldsymbol{x}_i$ denote the feature of an object, the attention mechanism can be expressed as:

$$\tilde{\boldsymbol{x}}_i = \sum_j \text{Softmax}(s_{ij})(\boldsymbol{x}_j \boldsymbol{W}_v + \text{MLP}(\boldsymbol{r}_{ij})), \tag{10}$$

$$s_{ij} = \frac{1}{\sqrt{d}}(\boldsymbol{x}_i \boldsymbol{W}_q)(\boldsymbol{x}_j \boldsymbol{W}_k + \text{MLP}(\boldsymbol{r}_{ij})), \tag{11}$$

where $\boldsymbol{W}$ represents a Linear layer. For agent-agent and agent-lane attentions, $\boldsymbol{r}_{ij}$ is the spatial relativity feature $\boldsymbol{d}_{ij}$. For lane-lane attentions, $\boldsymbol{r}_{ij} = \boldsymbol{d}_{ij} + \boldsymbol{e}_{ij}$ is the sum of the spatial relativity feature and the connectivity feature.

B. Guided generation

Our goal is to convert user descriptions into cost functions using LLMs and guide the generation through the method outlined in III-C. Existing approaches often face challenges in processing and comprehending complex descriptions. To address this issue, we propose an LLM-based framework comprising a high-level understanding module and a low-level refinement module.

High-Level Understanding Module. The high-level module employs a systematically designed Chain-of-Thought
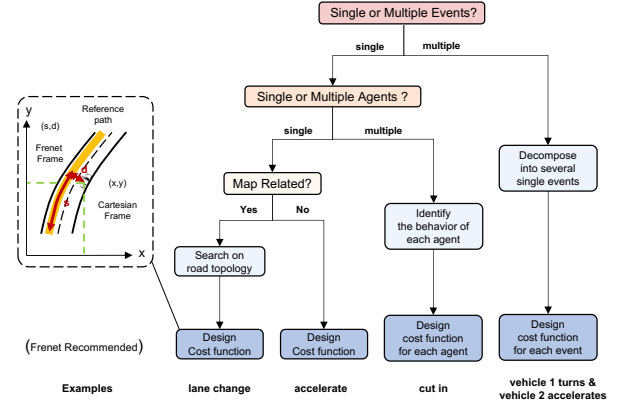

Fig. 2: We systematically design the chain-of-thought mechanism to thoroughly analyze and interpret the hierarchical behaviors in traffic scenes for controllable generation.

(CoT) mechanism to reason about the interaction graph and decompose complex events into manageable components. Figure 2 illustrates the detailed CoT process. First, the module identifies the events. For multiple events, it breaks them down into individual single events. Next, it determines the number of agents involved. If multiple agents are present, it identifies each agent's behavior. Subsequently, it evaluates whether each behavior is map-related. If so, it designs an algorithm to explore the road topology $E = \{e_{ij}\}$. Finally, it generates cost functions. For map-related descriptions, Frenet-based cost functions are recommended. We demonstrate that this structured reasoning approach enables the LLM to effectively understand and process a broader range of complex descriptions.

Low-Level Refinement Module. While the high-level module excels at generating accurate cost functions for complex descriptions, the resulting cost functions may occasionally be suboptimal or challenging to control through diffusion guidance. To address this, we introduce the low-level refinement module to further refine the cost functions. This module takes the low-level generated trajectory coordinates as input, enabling the LLM to directly analyze the generated trajectory data. Additionally, the corresponding cost function values from the trajectory optimization process are provided, offering further insights for the LLM's analysis. Using these inputs, the LLM evaluates whether the trajectory aligns with the given description, whether the cost function accurately reflects the desired behavior, and whether each loss term has appropriate hyperparameters. For example, the LLM can directly assess the correctness of the behavior from the trajectory coordinates and identify potential issues from unusually high cost values, enabling precise error analysis and correction. Through this design, we enhance the framework's ability to understand complex traffic scenarios and dynamically adapt to varying traffic contexts.

Frenet-based cost functions. Inspired by the success of the Frenet frame in motion planning, we utilize it to construct cost functions for descriptions related to map information. The Frenet frame is shown on the left side of Figure 2. It describes the relative position of trajectories concerning

Method	ADE (m) ↓	FDE (m) ↓	JSD ($\times 10^{-2}$) ↓	Collision (%) ↓	Off Road (%) ↓
Actions-Only [38]	4.81	11.89	10.4	19.9	27.6
DT [39]	1.56	3.07	8.4	<u>5.3</u>	**11.0**
CtRL-Sim [9]	<u>1.25</u>	<u>2.04</u>	<u>7.9</u>	<u>5.3</u>	**11.0**
CTG++ [15]	1.73	4.02	**7.4**	5.9	15.0
Ours	**1.17**	**1.93**	8.2	**5.1**	<u>12.8</u>

TABLE I: Results of conditional simulation. We **bolded** the optimal result and <u>underlined</u> the second best result.

the reference path by the longitudinal distance s and the lateral offset d. These parameters have clear geometric interpretations, making them intuitive for LLMs to understand and utilize. Furthermore, unlike methods that rely on target points, which can overly constrain the feasible trajectory space, our Frenet-based cost functions provide more reasonable constraints by naturally decomposing trajectories into longitudinal and lateral components. LLMs will generate cost functions based on s and d, enabling the control of both longitudinal and lateral directions. We will demonstrate that these Frenet-based cost functions significantly improve the LLMs' understanding of spatial relationships in traffic scenarios, leading to more accurate cost function generation and higher guidance success rates.

V. EXPERIMENTS

A. Experiment Setup

Dataset. Our experiments are conducted on large-scale real-world traffic scenario dataset: the Waymo Open Motion Dataset (WOMD) [21]. WOMD comprises over 570 hours of 10hz scenario data. We train our model on the entire training split, which contains 69,500 scenarios and evaluate it on a subset of 1,000 scenarios from validation split.

Evaluation. We begin by evaluating our model design through experiments on two tasks. The first task is **conditional simulation**, where we follow the experimental setup of CtRL-Sim [9]. In this setup, the model is provided with a 1.1-second history and an endpoint, and it simulates future trajectories over an 8-second horizon. We use the Final Displacement Error (**FDE**) and Average Displacement Error (**ADE**) to measure reconstruction error. We employ a distributional metric (**JSD**) to evaluate realism, defined as the mean of the Jensen-Shannon Distances computed on linear speed, angular speed, acceleration, and distance to the nearest vehicle between real and simulated scenes. Additionally, common-sense metrics like **Collision** rate and **Offroad** rate are used.

The second task is **unconditional generation**, where only the current states are provided. We evaluate the model using scene collision rate (**SCR**) and the Maximum Mean Discrepancy (**MMD**) metrics for both distance to the nearest object features (MMD_o) and distance to road edge features (MMD_r).

Finally, we conduct experiments on **guided simulation**. First, we select a range of scenario descriptions and assess whether the generated cost functions accurately reflect these descriptions. Next, we further choose four types of events, select 20 scenarios for each type, and evaluate the generation

Method	SCR(%)↓	MMD_o ↓	MMD_r ↓
TrafficGen [40]	9.7	0.31	0.23
LCTGen [41]	11.2	0.27	0.33
Ours	**6.3**	**0.24**	**0.17**

TABLE II: Results of unconditional scenario generation.

success rate. We visualize the results to determine whether the model correctly adheres to the guidance provided by the generated cost functions in each scenario.

Baselines. For conditional simulation, we compare our method against recent auto-regressive methods, including Actions-Only (a modified version of Trajeglish [38]), Decision Transformer [39], and CtRL-Sim [9]. We also include the popular diffusion-based method CTG++ [15]. For unconditional generation, we compare our approach to TrafficGen [40] and LCTGen [41]. In the guided generation experiments, we compare our proposed LLM-based framework in comparison to CTG++ [15] and SceneDiffuser [16].

Implementation Details. We set the number of layers for all attention modules to 3. The number of agents is set to $N_a = 24$, and the number of lane polylines to $N_m = 200$. We use 32 sampling steps for diffusion sampling, with other hyperparameters following EDM [35]. For closed-loop simulation, we set the re-plan rate to 2 Hz. For guided simulation, we use GPT-4o [42] for cost function generation.

B. Traffic simulation results

The conditional simulation results are presented in Table I. Our model achieves the best reconstruction performance, as measured by FDE and ADE, and ranks second in terms of common-sense metrics such as Collision Rate and Offroad Rate. However, the Jensen-Shannon Distance (JSD) metric is slightly inferior to the current state-of-the-art methods. A potential reason for this is that our method directly outputs future states, whereas other methods generate actions, which may lead to better dynamic feasibility.

The results of unconditional generation, shown in Table II, further demonstrate the effectiveness of our approach. Our model outperforms all baselines across metrics such as Scene Collision Rate (SCR) and Maximum Mean Discrepancy (MMD), confirming its ability to generate realistic traffic scenarios. These two experiments show that our flexible model design can handle various tasks and achieve competitive performance.

C. Guided simulation results

Evaluation on generated cost functions. Firstly, we evaluate our method using a diverse set of scenario descriptions, including basic rules, common behaviors and interactions, and abnormal behaviors, as detailed in Table III. We analyze the generated cost functions to verify their alignment with the input descriptions. To validate the contribution of each component in our method, we conduct an ablation study with the following variations: (1) our method without the high-level understanding module, and (2) our method without the Frenet coordinate system. Additionally, to further demonstrate the

Category	Description	CTG++	Scene Diffuser	Ours w/o understanding	Ours zero-shot CoT	Ours w/o Frenet	Ours
Basic rules	vehicle drives at a speed limit 10m/s	✓	✓	✓	✓	✓	✓
	vehicle reaches target point (25, 25)	✓	✓	✓	✓	✓	✓
	no collisions between all vehicles	✓	✓	✓	✓	✓	✓
Behavior and interactions	vehicle makes a left-lane-change	✓		✓	✓	✓	✓
	vehicle moves to the rightmost lane					✓	✓
	vehicle 1 cuts in vehicle 2		✓		✓		✓
Abnormal behaviors	vehicle drives in reverse			✓			✓
	vehicle drives out of the road						✓
	vehicle drives left and right within the lane		✓				✓

TABLE III: Results of controllable simulation.

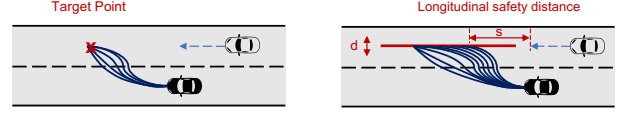| Scenario Type	Scene Diffuser	Ours w/o Refine	Ours w/ Refine
Cut in	0.45	0.65	0.85
Out of road	-	0.70	0.90
Yield	0.40	0.70	0.85
Rightmost	-	0.75	0.95

TABLE IV: Results of guidance success rate.

Fig. 3: Visualization of constraints. Compared to reaching the same endpoint, our method provides a more reasonable constraint, making it easier for the diffusion to follow.

$|d_1| + |d_2| + \mathrm{relu}(1.0 - (s_1 - s_2))$ $|d_1| + |d_2| + \mathrm{relu}(3.0 - (s_1 - s_2))$
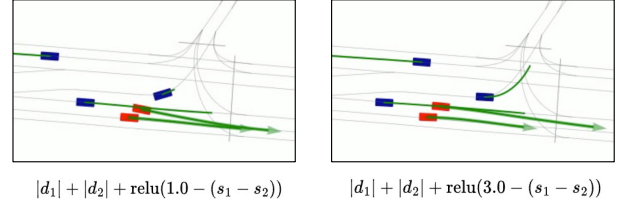
Fig. 4: A cut-in example of refinement: Two vehicles collide with each other and the longitudinal loss term is not effective. We should increase the safety distance in the cost function.

effectiveness of our design, we replace our CoT with zero-shot CoT [43], where the LLM is instructed to "think step by step" and reason about the traffic event independently, and compare the results.

In terms of basic rules, all tested methods demonstrated effective performance by employing functions like Norm for target point and ReLU for speed limit. These results suggest that LLMs do not confine the generation of cost functions to recommended forms (e.g., Frenet), but instead allow for diverse, adaptable cost function formulations.

When addressing behaviors such as lane changes, all methods performed well. However, for more complex road topologies such as navigating the rightmost lane, methods with our designed CoT demonstrated superior performance due to their enhanced map exploration capabilities. In agent interactions like cut-ins, our understanding module effectively understands and processes them by recognizing that one vehicle continues straight while another performs a lane change into its path, while CTG++ struggles to understand them. Although correctly understanding these behaviors, methods without Frenet coordinates cannot sufficiently understand the spatial relationships thus generating wrong functions. In contrast, the Frenet-based approach performs accurately, as its clear geometric interpretations further enhance the LLM's understanding of the scenario.

Regarding abnormal behaviors, our approach demonstrates impressive performance. The LLM generates $sin(d)$ to simulate the vehicle swaying left and right within the lane, and controls d to be positive, ensuring the vehicle stays on the left side of the road edge. The Frenet-based approach allows the LLM to express these abnormal behaviors with concise function forms, highlighting the expressiveness of the Frenet frame and its ability to enhance the LLM's understanding of complex traffic scenarios.

Through this experiment, we demonstrate that our method can effectively handle a broader range of scene descriptions compared to current approaches. Furthermore, when compared to zero-shot CoT, our method exhibits signifi-

cantly stronger reasoning capabilities, highlighting that understanding complex traffic scenes requires more than just the inherent knowledge of LLMs. Instead, it necessitates a systematically designed framework to interpret and reason about traffic interaction graphs. Additionally, the integration of the Frenet coordinate system proves to be a critical component, enabling more precise spatial reasoning and accurate generation.

Evaluation on success rate of guided simulation. To further validate the correctness of our LLM generation, we conduct a guidance success rate experiment across four types of scenarios, comparing our approach with a baseline method based on target points. The results, as shown in Table IV, demonstrate that our method achieves a significantly higher success rate than the target point-based baseline. This improvement can be attributed to the design of our Frenet-based cost functions, which provide reasonable and flexible constraints, as illustrated in Figure 3.

While our method demonstrates a high success rate, we observe that in some cases, the generated cost functions may contain errors or prove challenging for the diffusion model to follow due to different contexts in traffic scenarios. To address this, our refinement module plays a crucial role in improving the success rate, as evidenced by the results in Table IV. This module allows the LLM to adaptively adjust the cost functions, ensuring they better align with the intended guidance. Visualization examples in Figure 4 further illustrate this process, showing how the refinement module
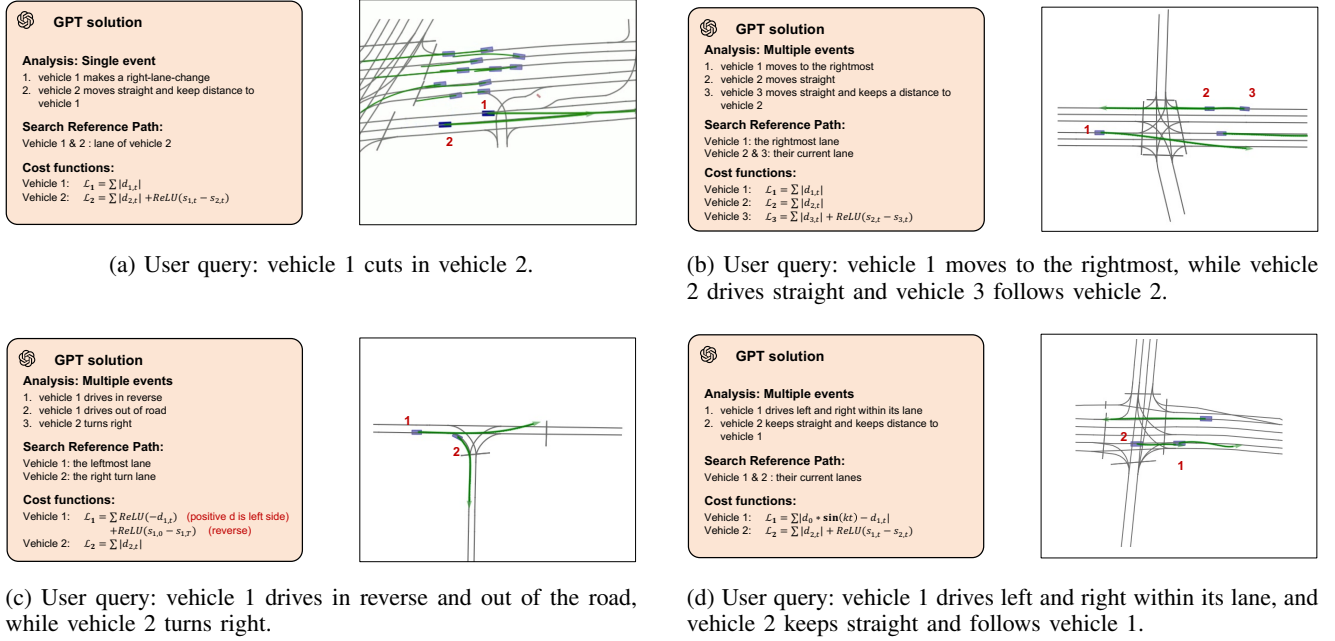
(a) User query: vehicle 1 cuts in vehicle 2.

(b) User query: vehicle 1 moves to the rightmost, while vehicle 2 drives straight and vehicle 3 follows vehicle 2.

(c) User query: vehicle 1 drives in reverse and out of the road, while vehicle 2 turns right.

(d) User query: vehicle 1 drives left and right within its lane, and vehicle 2 keeps straight and follows vehicle 1.

Fig. 5: GPT solutions and visualizations for typical cases.
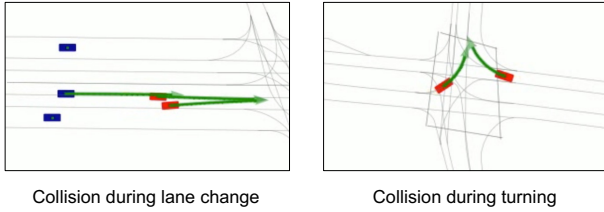

Fig. 6: Visualizations of our generated accidents.

effectively corrects and optimizes the cost functions, leading to a higher success rate and more accurate guidance.

D. Visualizations

We further combine some descriptions then query the LLM, and visualize the generation results, as shown in Fig. 5. The LLM effectively decomposed the complex descriptions and generated the correct forms of cost functions. The visualizations further demonstrate that the model accurately follows the guidance of the generated functions.

Furthermore, our method demonstrates zero-shot capability in handling accident events, despite being trained exclusively on a nearly collision-free dataset, as illustrated in Fig. 6. This ability is particularly valuable for traffic simulation, as it enables the model to generalize to rare but critical scenarios, such as accidents, without requiring explicit training data.

VI. CONCLUSION

In this paper, we proposed a novel LLM-enhanced controllable traffic simulation model. Our method is capable of taking users' descriptions as input, and generating a wide range of scenarios, including complex interactions and long-tail events. Several examples, including complex interactions and traffic accidents, have validated the effectiveness of our method. Finally, our numerous experiments and visualization demonstrate our framework achieves high-quality trajectory generation and precise instruction alignments across various benchmarks and settings.

Limitations. Although the refinement module effectively adjusts cost functions, it struggles to correct them when the LLM's initial understanding of the scenario is fundamentally incorrect. Future work could focus on enhancing the refinement module by incorporating more robust error detection mechanisms or leveraging external knowledge sources to improve its ability to correct fundamentally flawed initial assumptions.

REFERENCES

[1] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.

[2] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, p. 1805–1824, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.62.1805

[3] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," *arXiv preprint arXiv:2005.03778*, 2020.

[4] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "Trafficsim: Learning to simulate realistic multi-agent behaviors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 400–10 409.

[5] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone, "Bits: Bi-level imitation for traffic simulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2929–2936.

[6] L. Bergamini, Y. Ye, O. Scheel, L. Chen, C. Hu, L. Del Pero, B. Osiński, H. Grimmett, and P. Ondruska, "Simnet: Learning reactive self-driving simulations from real-world observations," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5119–5125.

[7] A. Kamenev, L. Wang, O. B. Bohan, I. Kulkarni, B. Kartal, A. Molchanov, S. Birchfield, D. Nistér, and N. Smolyanskiy, "Predictionnet: Real-time joint probabilistic traffic prediction for planning,

control, and simulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8936–8942.

[8] E. Pronovost, M. R. Ganesina, N. Hendy, Z. Wang, A. Morales, K. Wang, and N. Roy, "Scenario diffusion: Controllable driving scenario generation with diffusion," *Advances in Neural Information Processing Systems*, vol. 36, pp. 68 873–68 894, 2023.

[9] L. Rowe, R. Girgis, A. Gosselin, B. Carrez, F. Golemo, F. Heide, L. Paull, and C. Pal, "Ctrl-sim: Reactive and controllable driving agents with offline reinforcement learning," 2024. [Online]. Available: https://arxiv.org/abs/2403.19918

[10] S. Wang, G. Sun, F. Ma, T. Hu, Y. Song, L. Zhu, and M. Liu, "Dragtraffic: A non-expert interactive and point-based controllable traffic scene generation framework," 2024. [Online]. Available: https://arxiv.org/abs/2404.12624

[11] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, and M. Pavone, "Guided conditional diffusion for controllable traffic simulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3560–3566.

[12] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, D. Anguelov, *et al.*, "Motiondiffuser: Controllable multi-agent motion prediction using diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9644–9653.

[13] W.-J. Chang, F. Pittaluga, M. Tomizuka, W. Zhan, and M. Chandraker, "Controllable safety-critical closed-loop traffic simulation via guided diffusion," *arXiv preprint arXiv:2401.00391*, 2023.

[14] C. Xu, D. Zhao, A. Sangiovanni-Vincentelli, and B. Li, "Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles," in *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=hclEbdHida

[15] Z. Zhong, D. Rempe, Y. Chen, B. Ivanovic, Y. Cao, D. Xu, M. Pavone, and B. Ray, "Language-guided traffic simulation via scene-level diffusion," in *Conference on Robot Learning*. PMLR, 2023, pp. 144–177.

[16] C. M. Jiang, Y. Bai, A. Cornman, C. Davis, X. Huang, H. Jeon, S. Kulshrestha, J. W. Lambert, S. Li, X. Zhou, C. Fuertes, C. Yuan, M. Tan, Y. Zhou, and D. Anguelov, "Scenediffuser: Efficient and controllable driving simulation initialization and rollout," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=a4qT29Levh

[17] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023. [Online]. Available: https://arxiv.org/abs/2201.11903

[18] 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.

[19] Z. Lan, Y. Jiang, Y. Mu, C. Chen, and S. E. Li, "SEPT: Towards efficient scene representation learning for motion prediction," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=efeBC1sQj9

[20] H. Lin, Y. Wang, M. Huo, C. Peng, Z. Liu, and M. Tomizuka, "Joint pedestrian trajectory prediction through posterior sampling," *arXiv preprint arXiv:2404.00237*, 2024.

[21] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9710–9719.

[22] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8748–8757.

[23] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," *arXiv preprint arXiv:2301.00493*, 2023.

[24] 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," *arXiv preprint arXiv:1903.11027*, 2019.

[25] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, *et al.*, "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7814–7821.

[26] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. 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, pp. 11 525–11 533.

[27] Z. Huang, Z. Zhang, A. Vaidya, Y. Chen, C. Lv, and J. F. Fisac, "Versatile scene-consistent traffic scenario generation as optimization with diffusion," 2024.

[28] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 576–11 582.

[29] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters*, pp. 1–8, 2023.

[30] J. Lin, D. Fried, D. Klein, and A. Dragan, "Inferring rewards from language in context," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8546–8560. [Online]. Available: https://aclanthology.org/2022.acl-long.585

[31] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. Gonzalez Arenas, H.-T. Lewis Chiang, T. Erez, L. Hasenclever, J. Humplik, B. Ichter, T. Xiao, P. Xu, A. Zeng, T. Zhang, N. Heess, D. Sadigh, J. Tan, Y. Tassa, and F. Xia, "Language to rewards for robotic skill synthesis," *Conference of Robot Learning 2023*, 2023.

[32] W. K. Kim, Y. Lee, J. Kim, and H. Woo, "LLM-based skill diffusion for zero-shot policy adaptation," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=UGlDVc0GTU

[33] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[34] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[35] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 565–26 577, 2022.

[36] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying," *arXiv preprint arXiv:2306.17770*, 2023.

[37] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.

[38] J. Philion, X. B. Peng, and S. Fidler, "Trajeglish: Traffic modeling as next-token prediction," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=Z59Rb5bPPP

[39] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 15 084–15 097. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/7f489f642a0ddb10272b5c31057f0663-Paper.pdf

[40] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "Trafficgen: Learning to generate diverse and realistic traffic scenarios," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3567–3575.

[41] S. Tan, B. Ivanovic, X. Weng, M. Pavone, and P. Kraehenbuehl, "Language conditioned traffic generation," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=PK2debCKaG

[42] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[43] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 22 199–22 213.