

Enhancing Autonomous Vehicle Test Scenario Reasoning in Language Models

Denise Ataei [†]

Dept. of Electrical and Computer Engineering
University of California, Santa Cruz
Santa Cruz, CA
dataei@ucsc.edu

Ishaan Paranjape [†]

Dept. of Computational Media
University of California, Santa Cruz
Santa Cruz, CA
iparanja@ucsc.edu

Jim Whitehead

Dept. of Computational Media
University of California, Santa Cruz
Santa Cruz, CA
ejw@ucsc.edu

Abstract—Scenario based testing is a promising approach to evaluate autonomous vehicles for safety due to its ability to evaluate several components at once. Automated generation of these scenarios in simulation is needed to address the scale and diversity requirements in scenarios. Large Language Models (LLMs) can address this need due to their ability for world modeling. However, these models are ineffective in their reasoning thereby limiting their abilities in generating complex, dynamic vehicle interaction scenarios. In this paper, we present Cruzway Scenario Reasoner, an LLM based system that enhances reasoning capabilities of language models for complex vehicle interaction questions from the *Waymo Open Motion Dataset - Reasoning* dataset. This system consists of a suite of prompting approaches which include both Chain-of-thought prompting as well as prompting based on model based task planning in the Planning Domain Definition Language (PDDL). In addition, this system also contains *LLM as a judge* modules for the effective evaluation of generated responses and PDDL models. With this system, we are able to elevate the reasoning capabilities of the OpenAI GPT 4o-mini model. In addition, we also provide an in-depth qualitative analysis of language model responses to 15 scenarios categorized by complexity in the information provided.

Index Terms—autonomous vehicles, large language models.

I. INTRODUCTION

Autonomous vehicles (AVs) need to be rigorously tested before being deployed on public roads. Simulation testing has emerged as a cost effective approach to evaluate an AV within which scenario based testing is a promising approach [1]. However, the scale of potential test scenarios is large and approaches that can automatically generate these scenarios are needed. *Scenario generation* approaches fulfill this need. In [2], four challenges with current scenario generation approaches are highlighted; scenario fidelity, diversity and controllability along with algorithm efficiency. LLM based scenario generation can address these challenges. They are particularly well-suited to address these issues because LLMs have a world model [3], an open natural language vocabulary and have proven to be quite versatile [4]. It has an adequate familiarity with autonomous vehicle scenarios and PDDL file generation [5]. However, the introduction of LLMs also introduces new challenges the lack of an ability to reason and plan effectively [6] i.e. System 2 reasoning [7]. Several approaches

have been proposed to address this such as *LLM Modulo frameworks*, where LLMs are integrated with symbolic solvers [8] [9] [10] and *post training* of LLMs where LLMs are trained further using reinforcement learning or supervised finetuning [11] [12]. In AV research, LLMs have been used utilized either for driving the vehicle itself or for test scenario generation and evaluations [13]. LLMs have been integrated with planners to reason and navigate certain scenarios [14]. However, unlike the contributions in this paper, it is important to note that planners here are utilized for determining low level controls of the vehicle and not to reason about the vehicle behaviors and intentions.

Specifically in the scenario generation literature, current LLM based generation approaches such as *ScenicNL* lack the capability to generate complex dynamic vehicle interaction scenarios [15]. Knowledge about the reasoning capability of language models for such scenarios would precede approaches to generate the same. In scenario based testing literature, several representations have been presented. In this model, the first three layers describe the static scenarios where elements do not change with time and two layers describe dynamic scenarios where elements do change with time. Dynamic scenarios cover vehicle and pedestrian behaviors. In simulators such as CARLA, these scenarios are in the form of a library of behavior trees which are composed out of *atomic behaviors* and *atomic conditions* for these behaviors [16]. Another way to express scenarios in this simulator is by using Scenic. Here too, diverse dynamic scenarios are severely limited. Dynamic scenarios can be represented in a similar manner as model based task planning problems i.e. as a *sequence of conditional actions* such as the ones defined in a *PDDL domain* [17]. Therefore to design an effective dynamic scenario generator, we need to consider designing an effective planning problem generator. Since each planning problem has a corresponding domain model which contains possible states and actions, we need to automate the domain modeling process to happen alongside the planning problem generation. This would ensure context sensitivity of the domain.

In this paper, we present Cruzway Scenario Reasoner, an LLM based system that constitutes a suite of prompting approaches that enhance the reasoning capabilities of language models specifically for complex dynamic vehicle interaction

[†]Equal contribution

scenarios. The prompting approaches can be categorized in the following manner: (1) Chain-of-thought (CoT) prompting and (2) Model based task planning based prompting. In the former, we prepare prompts with examples to facilitate in-context learning, an effective approach to improve language model reasoning. Inspired by prompting approaches to image generators, we also enhance these CoT prompts with *negative prompts* to communicate what not to generate to the language models. These negative prompts are prepared after an in-depth analysis of language model responses to 15 scenarios. In the latter prompting approach, we use language models to generate a Planning Domain Definition Language (PDDL) based domain model and problem that corresponds with the context and question information respectively for each complex dynamic vehicle interaction scenario. This domain model and problem is fed into a symbolic planner which generates a solution for the PDDL problem. The PDDL generation follows the approach of an LLM modulo system where syntactic verifiers are integrated with LLM based generation approaches. Here, we use VAL as our syntax verifier. For semantic verification, we make use of an *LLM-as-a-judge* [18] [19] [20]. The domain model, problem and plan are given as a prompt to a language model to answer a question. To evaluate answer correctness, we make use of LLM as a judge approaches which provide us quantitative as well as qualitative responses. The latter facilitates in-depth analysis as mentioned above. Through both of these approaches we enhance reasoning capabilities and gain insights towards LLM based dynamic scenario generation.

The rest of this paper is divided into sections as follows: Section 2 reviews research in the domains of scenario based testing of AVs, use of LLMs for test scenario generation, and LLMs used for planning. Section 3 presents our approach. Section 4 details the experimental setup and the research questions that the experiments aim to answer. Section 5 presents the results of these experiment along with a qualitative and quantitative analysis. Section 6 presents conclusions and future directions. The code for this paper is available at the following URL: <https://github.com/AugmentedDesignLab/CarPlanningProblemGen>

II. RELATED WORKS

Several representations of scenarios have been suggested in the literature such as the *six layer model* where the scenario can be broadly divided into static layers such as roads and dynamic layers such as vehicles and pedestrians [21]. Here, the objects in layer 4 are *dynamic objects* where the state of the object changes with time. Several works have generated these dynamic behaviors, either as a data driven approach [22] [23] or through manual design (cite scenario runner BTs, OpenSCENARIO and Scenic). Here, the former has the advantage of scale but the disadvantage of a lack of interpretability and controllability in generation. The latter has the benefit of better interpretability but cannot be scaled easily and is also not as realistic. Scenarios in natural language are interpretable and general purpose natural language generators such as LLMs give us the ability to scale the same. In our

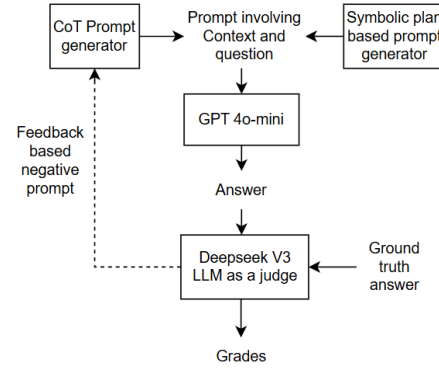


Fig. 1. Cruzway Scenario Reasoner Architecture

work, we address the shortcoming of LLMs low reasoning ability about these scenarios.

LLMs have been used in the domain of autonomous vehicles for driving functions [13]. LLMs have also been used for test scenario generation. In several works such as [24] and [15] LLMs have been utilized for generating domain specific languages such as Scenic [25] to generate test scenarios for autonomous vehicle simulators such as the CARLA simulator. A significant drawback of these methods is the absence of complex dynamic vehicle interaction scenarios. Addressing reasoning capabilities of LLMs would precede the ability to correctly generate these scenarios. This paper addresses the need to enhance reasoning capability for these scenarios.

There have been works that have used symbolic solvers to improve reasoning capabilities of LLMs [9] [10]. We explore the approach in [9] where we provide symbolic solver problems and solutions corresponding to the given question in the prompt of a language model while asking it to generate a response.

III. METHOD

The system architecture is presented in Figure 1. This architecture contains two types of prompting approaches: (1) Chain-of-thought (CoT) prompting and (2) Model based task planning (MBTP) prompting. The CoT prompting block contains several prompts based on the Chain-of-thought prompting method. The MBTP block contains a subsystem for an LLM based MBTP system which generates a PDDL model and corresponding problems and a solved plan generated by an external tool. The design details of each component in this architecture is presented in the rest of this section.

A. Dataset and Preprocessing

The natural language based question and answer (QA) data we use is from the Waymo Open Motion Dataset - Reasoning (Waymo-Reasoning dataset) [26]. This dataset is based on the Waymo Open Motion dataset [27]. It contains real-world vehicle, pedestrian and cyclist trajectory data collected for the purpose of creating motion forecasting models. The Waymo-Reasoning dataset contains QA data pertaining to diverse interaction types such as ones based on traffic rules or human

intentions. We preprocess this dataset using LLMs to organize the subgroups *environment QA* and *surrounding agent QA* as a *scenario context* and the remaining subgroup *interaction QA* as the interaction QA. The evaluation task given to the language models is to answer the interaction questions given the scenario context. We considered having an LLM rewrite the context QA subgroup as a paragraph of text but since it contained inconsistencies with respect to the original data, we add the QA data as is into the context attribute. An exemplar parsed scenario is shown in Listing 1. Here, the file size is used as a proxy measure of scenario complexity since each scenario is contained in its own file. These file sizes are categorized as A, B and C representing small, medium and large file sizes respectively.

```
"Context": "Can you describe the layout of the lanes where
the ego agent is currently located? The ego agent is on
the 1 lane from the left, out of 3 lanes. What is the
current velocity of the ego agent? The ego agent's
current speed is 12 m/s. What is the ego agent's
current action in terms of speed adjustment? The ego
agent is decelerating. Could you specify the ego agent's
current lane position? The ego agent is on the 1 lane
from the left. What type of agent is surrounding agent
#0? Surrounding agent #0 is a vehicle. How fast is
surrounding agent #0 currently moving? Its current
speed is 16 m/s. What is the motion status of
surrounding agent #0? It is moving at a constant speed.
Where is surrounding agent #0 in relation to the ego
agent? It is 7 meters in front of the ego agent and 3
meters on the right. What is the type of surrounding
agent #2? Surrounding agent #2 is a vehicle. What is
the current speed of surrounding agent #2? Its current
speed is 17 m/s. Can you describe the motion status of
surrounding agent #2? It is moving at a constant speed.
Where is surrounding agent #2 positioned relative to
the ego agent? It is 19 meters in front of the ego
agent and 3 meters on the right.",
...
"Interactions_1": { "reference_question": "How will the ego
agent and surrounding agent #2 interact on the road?",
"reference_answer": "Surrounding agent #2 will maintain
its position ahead of the ego agent as both are
moving in the same direction with surrounding agent
#2 at a constant speed and the ego agent
decelerating." ...
```

Listing 1. Example parsed scenario after preprocessing. Few details have been omitted for conciseness.

B. Prompting Using Model based Task Planning

We prompt an LLM to generate PDDL domain files for the given contextual information and a problem file for each scenario interaction in the parsed scenario data mentioned previously. We then perform a syntax verification of the LLM generated outputs using VAL [28]. In addition, we also carry out semantic verification of the generated PDDL files for coverage with respect to the facts in the given context. Once the generated PDDL file is syntax error free, we make use of the breadth first search based planner in the [29] library to create a plan that would solve the generated problem. The final prompt to the language model contains the scenario context, the interaction question, the PDDL domain model that corresponds with the context, the PDDL problem that corresponds with the interaction question, and a plan generated by a symbolic solver that solves the aforementioned PDDL problem. This process is presented in Figure 2. An example of PDDL problem generation for the parsed scenario presented

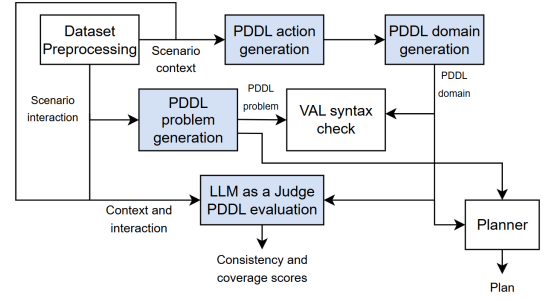


Fig. 2. Model based Task Planning (MBTP) based Prompting. The blue boxes make use of LLMs for generation.

```
(define (problem ego-vehicle-scenario-v2)
...
(:objects
  ego - agent
  lane1 lane2 lane3 - lane
  vehicle0 vehicle2 - vehicle
)
(:init
...
; Vehicle0: 7m ahead, 3m right (front-right), in lane2
(vehicle-at-fast-speed vehicle0)
(vehicle-at-constant-speed vehicle0)
(vehicle-is-in-lane vehicle0 lane2)
(vehicle-is-in-front-right-of-ego vehicle0)
(ego-near-vehicle-front-right vehicle0)
(vehicle-in-front-right vehicle0)

; Vehicle2: 19m ahead, 3m right (far front-right), in
lane2
(vehicle-at-fast-speed vehicle2)
(vehicle-at-constant-speed vehicle2)
(vehicle-is-in-lane vehicle2 lane2)
(vehicle-is-far-ahead-right-of-ego vehicle2)
(vehicle-far-in-front-right vehicle2)
)
(:goal
  (ego-is-decelerating)
)
)
```

Listing 2. PDDL problem generated by the LLM GPT 4.1 for the interaction question within the scenario parsed in Listing 1 and after the PDDL domain is generated. We can observe the states in this file correspond to the parsed interaction question and answer. Few details have been omitted for conciseness.

in Listing 1 is presented in Listing 2. The semantic verification for this PDDL problem using LLM-as-a-Judge based evaluation is presented in Listing 3.

C. Chain of Thought Prompting

The Chain of Thought (CoT) module contains a suite of n -shot prompts based on the Chain-of-thought prompting approach [30]. Here n represents the number of examples contained in these prompts. These examples are based on the interaction QA data of one scenario (identified as index 78) from the dataset after preprocessing shown above. Each of the prompts are also appended with two types of *negative prompt* (inspired by [31]) which is prepared based on a detailed analysis of *GPT 4o-mini* responses for 15 scenarios across the small, medium and large scenarios. The details of this in-depth analysis are presented in Section V. The two types of negative prompt involve long and short-form, which are differentiated

```

"Problem data provided": "How will the ego agent and
surrounding agent #2 interact on the road?",
"Score explanation": "The problem file directly models the
interaction context: it places the ego in the leftmost
lane, next to the two relevant vehicles, with all
spatial and speed predicates correctly instantiated for
vehicle0 (close, front-right) and vehicle2 (farther
front-right). The goal is set for the ego to be
decelerating, reflecting a cautious or responsive
behavior to the nearby (closer) vehicle in the adjacent
lane. While the explicit interaction with agent #2 (
vehicle2) is less emphasized in the goal than with
vehicle0, the presence and predicates for vehicle2 show
that its position is accounted for in the conditions
surrounding the ego's behavior. A minor shortcoming may
be the lack of an explicit future state following the
interaction (such as passing or merging), but for the
deceleration-focused goal and the state of the road
given, the coverage is very strong.",
"Grade": "9", ...

```

Listing 3. PDDL Problem Coverage Evaluation using GPT4.1 LLM as a judge for the problem presented in Listing 2

by the following word count per n-shot prompt: long-form contains 1561, 2067, and 1036 words, and short-form contains 359, 297, and 385 words, with each category listed with respect to two, four, and six-shot prompting. We also considered refined prompting where the examples are given to a language model for rewriting. This refined prompt is of two types - basic and lecturing. Basic utilizes *GPT4.1* refine prompting combined with short-form negative prompting. We used short form because of the near-identical impact it shared with long-form negative prompting (reference Table III), despite being drastically shorter in word count. Lecturing is inspired by the prompt formatting in [32] where concepts are explained in the form of a lecture transcript generated by the LLM. An overview of all prompting approaches used is found in Figure 3.

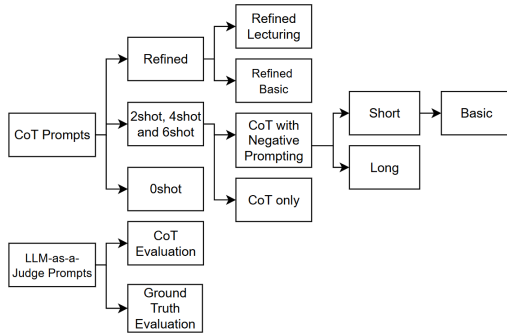


Fig. 3. All prompting types experimented with and their connections to one another.

D. Response Evaluations using LLM-as-a-Judge Approach

We consider *GPT 4o-mini* as a representative language model for reasoning enhancement in this paper. Small LMs are typically less capable in reasoning (and thereby more error prone) than LLMs. Hence, the effectiveness of our prompting approaches presented in this section will be more apparent in Small LMs. We evaluate the reasoning capability of our

```

"Correctness score": "10",
"Correctness explanation": "The AI's answer is correct.
Surrounding agent #3 is stationary and positioned to
the left and slightly behind the ego agent. Since the
ego agent is accelerating and moving forward, it will
pass by surrounding agent #3 without any interaction.
The ground truth answer incorrectly states that
surrounding agent #3 will follow the ego agent, which
is not possible as it is not moving and is not on the
same lane."

```

Listing 4. Scenario 52 evaluated solely on two-shot prompting, demonstrating ground truth invalidity.

```

"Given Question": "How will the ego agent and surrounding
agent #2 interact on the road?",
"Ground Truth Answer": "Surrounding agent #2 will maintain
its position ahead of the ego agent as both are moving
in the same direction with surrounding agent #2 at a
constant speed and the ego agent decelerating.",
"Ground Truth Grades": {"Correctness score": "8",
"Correctness explanation": "The answer is mostly correct
but lacks precision. Surrounding agent #2 is indeed
maintaining its position ahead of the ego agent, as it
is moving at a constant speed of 17 m/s, which is
faster than the ego agent's speed of 12 m/s (while
decelerating). However, the answer could be improved by
explicitly mentioning the speed difference and the
fact that the distance between them will increase over
time due to this speed difference.",
"Updated Answer": "Surrounding agent #2 will maintain its
position ahead of the ego agent and increase the
distance over time, as it is moving at a constant speed
of 17 m/s, which is faster than the ego agent's speed
of 12 m/s while decelerating.", ...

```

Listing 5. Ground truth evaluation using LLM as a judge for the scenario parsed in Listing 1. Few details have been omitted for conciseness.

prompting approaches using an LLM as a judge approach with the LLM *Deepseek V3* as the judge LLM. Here the LLM makes a comparison between the small LM response and the corresponding ground truth answer in the parsed data and generates a *correctness score* out of 10.

In addition to evaluating reasoning capability of GPT 4o-mini, we also evaluate the ground truth in the dataset itself using the judge LLM mentioned above. This validates the comparative LLM-as-a-judge evaluation above. In Listing 5, we can observe an LLM-as-a-judge evaluation where the ground truth is incorrect. In response to this, we carry out LLM-as-a-judge based evaluation of the ground truth itself using Deepseek V3. An exemplar ground truth evaluation of the scenario parsed in Listing 1 is presented in Listing 5. The evaluation is presented in Section V.

IV. EXPERIMENTAL SETUP

In this section, we explore how GPT4o-mini rationalizes with close-proximity autonomous driving interactions without the corresponding PDDLs for small, medium, and large Waymo -Reasoning files, and procedures taken to optimize this rationalization to that of an LLM's performance. Our exploration built upon our research questions outlined in subsection A. To address such questions, we experimented with a prepared dataset as described in subsection B. Using this dataset, we developed and ran experiments testing GPT4o-

mini’s reasoning performance with a suite of prompting types that are further explained in subsection C.

A. Research Questions

The following research questions were explored in our experiments: (1) How to prompt using CoT to reason about dynamic scenarios? (2) How to use model based task planning to reason about scenarios? (3) How to judge outputs and the validity of the ground truth itself using an LLM as a judge approach? (4) How do we interpret grades generated by the LLM as a judge?

B. Dataset Preparation

WOMD-R contains around 100,000 files, with each file containing one scenario. In each scenario, context information with several interactions and related vehicle-intention questions and answers are found. The data is reformatted during the preprocessing stage as described in Section III. During this stage, scenarios were categorized into small (less than 6000 bytes), medium (between 6000 and 9999 bytes inclusive), and large (greater than 9999 bytes), with each category corresponding to varying amounts of information in a scenario. Here, we identify the files by their scenario’s index in WOMD-R’s dataset. For our experiments, we selected 15 scenarios at random to examine the performance of our suite of prompting types. Our experiments are broken down as such: five scenarios per file size for a total of 15 unique scenarios, with each scenario undergoing one trial of each type of CoT prompting type to generate performance grades. We refer to these scenarios as the *base scenario set* for the rest of this paper.

C. Model based Task Planning Evaluation

In our experiments, we explore the PDDL generation process presented in Section III and in Figure 2. There are two LLM as a Judge evaluations carried out; one is for evaluating the PDDL files generated for consistency and coverage and the other for evaluating GPT 4o-mini’s reasoning ability for an interaction question within a given scenario. We explore existence proofs of this approach which we have presented in Section III.

D. Prompt Evaluations

In our experiments, we explored all the prompt types defined in section III for each scenario in the base scenario set. We used a randomly selected scenario (index 78) to prepare our in-context examples within zero, two, four, and six-shot prompting for a total of 60 experiments. We prepared prompts for ground truth validation, CoT, two types of negative prompting combined with CoT (negative long and negative short), and two types of refined prompting with and without lecturing (refined basic), where GPT4.1 carries out the refinement. For ground truth validation, we considered our base scenario set using the *ground truth validation prompt* so the judge LLM could grade the answers for each interaction per scenario that was factually consistent with respect to the corresponding

context information. The grading format has been presented in Section III. The Q1 scores for reasoning for interactions within the basic scenario set are found in Table I.

V. RESULTS

In this section we analyze and observe the outcome of our suite of prompting styles. In subsection A, we observe the outcomes of our base scenario set, which influence the design of later prompting styles. In subsection B, we observe the results of a new prompting approach to address the errors found in subsection A. In subsection C, we observe the results of two additional prompting approaches to enhance the scores of factually consistent scenarios in subsection A.

A. Model based task planning (MBTP) Prompt Evaluation

We aim for an existence proof for this prompting method. PDDL generation for an example scenario are presented in Section III. As described in Section IV, we run the evaluation for GPT 4o-mini with MBTP prompting. In addition, we record the LLM as a judge based PDDL consistency and coverage scores to obtain a measure of PDDL generation quality.

B. Chain of Thought Prompt and Ground Truth Evaluations for Scenario Reasoning

To enhance the effect of CoT prompting types per file size category, we analyzed the growth and decline made in the 25th percentile (Q1) of correctness scores received per scenario. Since Q1 is the lower quartile, its change implies the remaining distribution also changing in the same way. For the base scenario set, we observed that there was a near consistent improvement in Q1 scores for scenarios 475, 182, and 13, while scenarios 254, 136, and 41 portrayed a near consistent decline. Q1 scores are presented in Table I. Here, growth is highlighted in green and decline in red.

Scenario (file size)	GTV	CoT-0	CoT-2	CoT-4	CoT-6
239 (S)	7.5	5.50	6.0	4.5	6.0
562 (S)	8.0	8.0	8.0	8.0	7.0
999 (S)	8.0	7.25	7.75	3.5	7.75
2827 (S)	8.0	6.0	10.0	4.0	5.0
475 (S)	8.5	6.5	7.0	9.0	9.0
6 (M)	8.0	6.0	6.5	6.0	6.0
254 (M)	4.5	5.5	3.0	2.75	3.0
622 (M)	7.25	7.25	8.0	8.5	7.25
136 (M)	4.5	7.0	6.5	5.5	5.5
182 (M)	8.5	2.5	9.5	8.0	9.0
52 (L)	4.0	2.0	2.0	2.0	3.0
13 (L)	10.0	2.25	9.25	9.25	10.0
41 (L)	5.25	6.0	4.0	4.0	3.0
102 (L)	3.0	2.0	5.5	4.0	4.75
600 (L)	8.0	6.0	6.5	4.0	4.5

TABLE I
BASE SCENARIO SET Q1 SCORES FOR GROUND TRUTH VALIDATION (GTV) AND N-SHOT PROMPTING TYPES RESPECTIVELY

To understand the scoring growth and decline of these indices, we carried out an in-depth qualitative analysis of the correctness explanations generated by the judge LLM. Based on this analysis, we develop *negative prompts*. First,

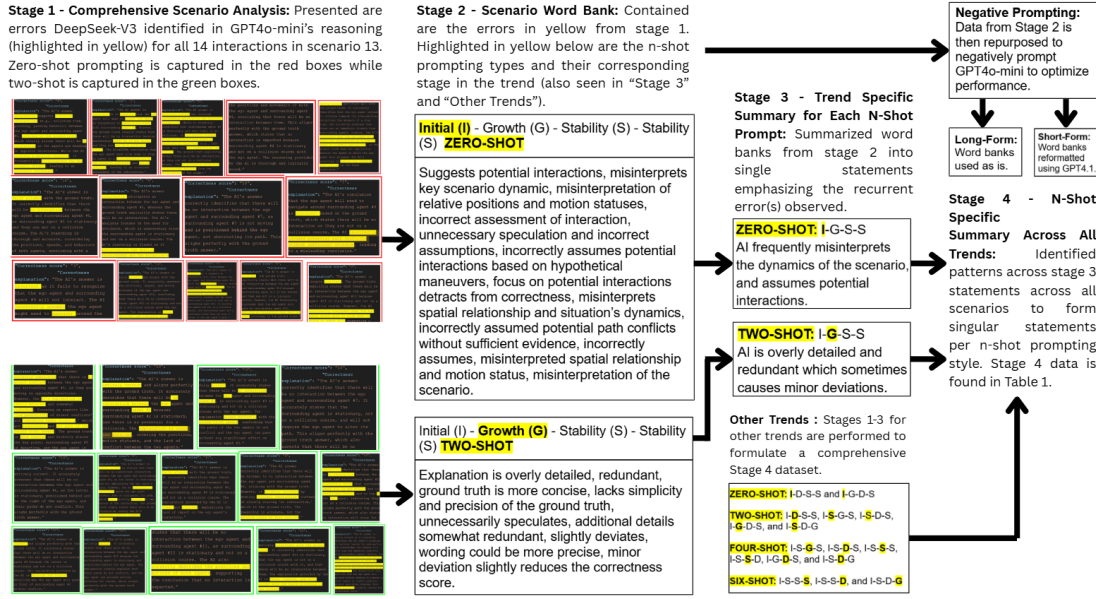


Fig. 4. Analysis of the Initial-Growth-Stability-Stability (I-G-S-S) trend across zero and two-shot prompts for scenario 13. Please refer to Table I for Q1 scores from scenario 13 portraying the I-G-S-S trend.

we categorized irregular Q1 patterns from Table I by creating trend categories consisting of four stages. These stages are separated by a dash (-), with the first stage always being zero-shot, also named "Initial (I)." The following three stages represent two, four, and six-shot prompting, ordered as listed across all trend categories. While zero-shot is always labeled as "Initial," two, four, and six-shot alternate through a metric of three outcomes: growth, decline, and stability. Growth (G) is an increase larger than 1, stability (S) is an increase or decrease between 0 and 1 inclusive, and decline (D) is a decrease larger than 1 with respect to the prior stage of the trend.

With this new metric, we grouped all 15 scenarios into eight trend categories. Letters in bold represent the analysis performed between stages to better understand a shift towards growing, declining, or stabilizing Q1 scores. Our analysis emphasizes two or three stages at a time in order to minimize repeating or insignificant outcomes. We list our trend categories and their corresponding scenarios: **I-G-S-S** (13), **I-D-S-S** (254, 41), **I-S-G-S** (475), **I-S-D-S** (600), **I-S-S-S** (562, 6, 136, 52), **I-S-S-D** (622), **I-G-D-S** (182, 2827, 102), and **I-S-D-G** (999, 239).

Based on these trend categories, we analyzed 283 paragraphs of correctness explanations generated by DeepSeek-V3 (Stage 1 of Figure 4). In this analysis, we noted the errors DeepSeek-V3 described in GPT4o-mini's reasoning, and identified recurrent trends in the errors made. We then condensed the 283 paragraphs into word banks showing the mistakes GPT4o-mini's reasoning made (Stage 2 of Figure 4). For our interpretation, we summarized the data from Stage 2 for all n-shot prompting types into singular statements (Stage 3 of Figure 4). We used the stage 3 statements in all trends to summarize error patterns made by each n-shot prompting for all scenarios. Table II shows a completed stage 4 analysis.

N-Shot Prompt	Stage 4 Summary
Zero-Shot	AI frequently overcomplicated analysis by assuming potential interactions and verbose explanations, while often misinterpreting or neglecting key motion dynamics and important details of the scenario.
Two-Shot	AI frequently overcomplicated analysis by being overly detailed, verbose, and making incorrect assumptions, which leads to neglecting key details, misinterpreting interactions, and minor deviations in accuracy.
Four-Shot	AI frequently misinterprets motion dynamics and interactions—especially yielding behavior—by making unnecessary assumptions, focusing on extraneous details, and failing to recognize or accurately address critical scenario aspects such as agent movement, future actions, and interactions with traffic elements and surrounding agents.
Six-Shot	AI frequently overlooks critical scenario details—particularly yielding and decelerating behavior—misinterprets agent movement and interactions, and often introduces unnecessary concepts or focuses on irrelevant agents instead of accurately addressing essential motion dynamics.

TABLE II
STAGE 4: STATEMENTS DESCRIBING ERROR ANALYSIS PER N-SHOT PROMPTING (APPLICABLE TO ALL SCENARIOS IN THE BASE SCENARIO SET).

We arrived at negative prompting by using word banks from stage 2 of Figure 4 to refine n-shot prompting. To identify a more effective approach for negative prompting, we compare "Long-Form" as opposed to "Short-Form." In long-form, keywords from the word banks were separated by commas to form a singular large paragraph, while short-form summarized the entire paragraph into 8-11 statements. These statements acted as guidelines for each n-shot prompting type. Both of the negative prompting types were specifically placed after the in-context example so the mini LLM would consider its prior reasoning errors after obtaining scenario information. The results of this procedure are found in Table III.

Because long and short-form negative prompting returned similar Q1 scores, we proceeded to use only short-form in

File Size	Combined Scenarios	Prompting Style	Zero-Shot	Two-Shot	Four-Shot	Six-Shot
Small	239, 562, 999, 2827, 475	CoT Only	6.00	7.00	5.25	6.25
		NL-CoT	6.00	8.00	6.50	7.00
		NS-CoT	6.00	8.00	6.00	7.25
Medium	6, 254, 622, 136, 182	CoT Only	6.00	6.00	6.00	4.50
		NL-CoT	6.00	4.00	4.00	5.00
		NS-CoT	6.00	5.00	4.00	4.00
Large	52, 13, 41, 102, 600	CoT Only	3.00	4.75	4.00	4.00
		NL-CoT	3.00	5.75	4.00	4.00
		NS-CoT	3.00	6.00	3.00	4.00

TABLE III

Q1 SCORES PER N-SHOT PROMPTING OF COMBINED SCENARIOS IN THE BASE SCENARIO SET. NL-CoT AND NS-CoT ARE NEGATIVE LONG AND SHORT-FORM PROMPTING COMBINED WITH CoT.

other variations of our experiments. Although the negative prompting improved Q1 scores in some instances, it also declined in others when compared to our base scenario set (CoT only). To mitigate these irregular outcomes, we incorporate short-form negative prompting into our refined (without lecturing) prompting strategy, also known as refined basic prompting.

Scenario Index (File Size)	GTV	CoT-2 (B)	CoT-2 (L)	CoT-4 (B)	CoT-4 (L)	CoT-6 (B)	CoT-6 (L)
239 (S)	7.5	6.5	6.5	6.0	5.0	6.5	6.5
562 (S)	8.0	8.0	6.0	8.0	8.0	8.0	7.0
999 (S)	8.0	8.5	8.5	7.0	7.5	6.5	8.5
2827 (S)	8.0	9.0	6.0	10.0	8.0	3.0	8.0
475 (S)	8.5	7.0	8.5	7.0	8.5	6.0	7.5
6 (M)	8.0	3.5	5.0	4.0	6.5	4.0	5.5
254 (M)	4.5	3.0	3.0	4.0	3.75	3.0	3.75
622 (M)	7.25	4.0	4.5	3.25	7.75	4.0	5.0
136 (M)	4.5	5.5	6.0	6.5	5.5	5.5	6.5
182 (M)	8.5	10.0	8.0	10.0	7.0	10.0	7.0
52 (L)	4.0	4.0	3.0	4.0	3.0	7.0	4.0
13 (L)	10.0	10.0	8.25	10.0	7.25	10.0	10.0
41 (L)	5.25	3.25	4.0	4.0	4.0	4.75	3.25
102 (L)	3.0	4.0	4.0	4.0	4.0	5.5	4.75
600 (L)	8.0	8.0	6.0	6.0	6.5	7.5	6.5

TABLE IV

GPT 4o-mini Q1 SCORES PER SCENARIO FOR REFINED BASIC (B) AND LECTURING (L) PROMPTING (WITH GPT 4.1) FOR 2, 4 AND 6 SHOT CoT PROMPTS WITH SHORT NEGATIVE PROMPTING. IN BOLD ARE THE SCORES THAT HAVE IMPROVED BY 1 POINT OR MORE WITH RESPECT TO THE CORRESPONDING N-SHOT PROMPTING IN TABLE I

C. Threats to Validity

In this subsection, we present doubts that might threaten the validity of the results presented in this paper and the steps to mitigate the same: (1) **Challenges with LLM as a Judge Methods:** Several works highlight issues with LLM as a judge methods such as biases [33]. We address these by ensuring that the main evaluation, that of scenario reasoning capability of GPT 4o-mini is a *comparative evaluation* given a ground truth. In addition, we make use of the LLM as a Judge itself to evaluate the ground truth answer. We have observed that very low scores in this evaluation reflect factual inaccuracies with respect to the scenario context information but higher scores might also reveal LLM preferences. (2) **Repeatability:**

Since the key contribution of this paper is a suite of prompting approaches, it is important to ensure that responses to a prompt are repeatable. LLMs do not have completely deterministic outputs. This is necessary for outputs that are needed to be well formatted e.g. in PDDL generation. To mitigate this, we ensure that the reasoning capability enhanced by prompts is evaluated over a representative set of scenarios with varying information and examples. In addition, this evaluation is carried out over several tries.

VI. CONCLUSION

In this paper, we present a suite of prompting approaches to enhance the reasoning capability of a language model for complex dynamic vehicle interaction scenarios in the *Waymo Open Motion Dataset - Reasoning* dataset. We explore both symbolic method based prompts as well as Chain of Thought Prompting. We evaluate the performance of these prompts using a suite of LLM as a Judge approaches. In the future, evaluation of a larger set of scenarios in this dataset as well as datasets such as DriveLM [34] can be explored along with a wider range of prompting approaches.

REFERENCES

- [1] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE access*, vol. 8, pp. 87 456–87 477, 2020.
- [2] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—a methodological perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 6971–6988, 2023.
- [3] Y. Gu, B. Zheng, B. Gou, *et al.*, "Is your llm secretly a world model of the internet? model-based planning for web agents," *arXiv preprint arXiv:2411.06559*, 2024.
- [4] S. Minaee, T. Mikolov, N. Nikzad, *et al.*, "Large language models: A survey," *arXiv preprint arXiv:2402.06196*, 2024.
- [5] L. Guan, K. Valmeekam, S. Sreedharan, and S. Kambhampati, "Leveraging pre-trained large language models to construct and utilize world models for model-based task planning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 79 081–79 094, 2023.
- [6] K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati, "On the planning abilities of large language models—a critical investigation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 75 993–76 005, 2023.
- [7] D. Kahneman, "Thinking, fast and slow," *Farrar, Straus and Giroux*, 2011.
- [8] S. Kambhampati, K. Valmeekam, L. Guan, *et al.*, "Llms can't plan, but can help planning in llm-modulo frameworks," *arXiv preprint arXiv:2402.01817*, 2024.

- [9] L. Pan, A. Albalak, X. Wang, and W. Y. Wang, "Logiclm: Empowering large language models with symbolic solvers for faithful logical reasoning," *arXiv preprint arXiv:2305.12295*, 2023.
- [10] S. Kirtania, P. Gupta, and A. Radhakrishna, "Logiclm++: Multi-step refinement for symbolic formulations," *arXiv preprint arXiv:2407.02514*, 2024.
- [11] M. Pternea, P. Singh, A. Chakraborty, *et al.*, "The rl/llm taxonomy tree: Reviewing synergies between reinforcement learning and large language models," *arXiv preprint arXiv:2402.01874*, 2024.
- [12] J. Hong, A. Dragan, and S. Levine, "Q-sft: Q-learning for language models via supervised fine-tuning," *arXiv preprint arXiv:2411.05193*, 2024.
- [13] Z. Yang, X. Jia, H. Li, and J. Yan, "Llm4drive: A survey of large language models for autonomous driving," in *NeurIPS 2024 Workshop on Open-World Agents*, 2023.
- [14] Y. Zheng, Z. Xing, Q. Zhang, *et al.*, "Planagent: A multi-modal large language agent for closed-loop vehicle motion planning," *arXiv preprint arXiv:2406.01587*, 2024.
- [15] K. Elmaaroufi, D. Shanker, A. Cismaru, *et al.*, "Scenicnl: Generating probabilistic scenario programs from natural language," in *First Conference on Language Modeling*, 2024.
- [16] CARLA. "ScenarioRunner for CARLA." Web page, accessed 29 November 2024. (2024).
- [17] D. M. McDermott, "The 1998 ai planning systems competition," *AI magazine*, vol. 21, no. 2, pp. 35–35, 2000.
- [18] L. Zheng, W.-L. Chiang, Y. Sheng, *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [19] J. Fu, S.-K. Ng, Z. Jiang, and P. Liu, "Gptscore: Evaluate as you desire," *arXiv preprint arXiv:2302.04166*, 2023.
- [20] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-eval: Nlg evaluation using gpt-4 with better human alignment," *arXiv preprint arXiv:2303.16634*, 2023.
- [21] M. Scholtes, L. Westhofen, L. R. Turner, *et al.*, "6-layer model for a structured description and categorization of urban traffic and environment," *IEEE Access*, vol. 9, pp. 59 131–59 147, 2021.
- [22] L. Rowe, R. Girgis, A. Gosselin, L. Paull, C. Pal, and F. Heide, "Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments," *arXiv preprint arXiv:2503.22496*, 2025.
- [23] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, "King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients," in *European Conference on Computer Vision*, Springer, 2022, pp. 335–352.
- [24] J. Zhang, C. Xu, and B. Li, "Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 459–15 469.
- [25] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: A language for scenario specification and scene generation," in *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, 2019, pp. 63–78.
- [26] Y. Li, C. Ge, C. Li, *et al.*, "Womd-reasoning: A large-scale language dataset for interaction and driving intentions reasoning," *arXiv preprint arXiv:2407.04281*, 2024.
- [27] S. Ettinger, S. Cheng, B. Caine, *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.
- [28] R. Howey, D. Long, and M. Fox, "Val: Automatic plan validation, continuous effects and mixed initiative planning using pddl," in *16th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, 2004, pp. 294–301.
- [29] E. Lejeune, *Jupyddl, an extensible python library for pddl planning and parsing*, en, 2021. DOI: 10.13140/RG.2.2.22418.89282. [Online]. Available: <http://rgdoi.net/10.13140/RG.2.2.22418.89282>.
- [30] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [31] Y. Ban, R. Wang, T. Zhou, M. Cheng, B. Gong, and C.-J. Hsieh, "Understanding the impact of negative prompts: When and how do they take effect?" In *European Conference on Computer Vision*, Springer, 2024, pp. 190–206.
- [32] P. Lu, S. Mishra, T. Xia, *et al.*, "Learn to explain: Multimodal reasoning via thought chains for science question answering," in *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [33] J. Gu, X. Jiang, Z. Shi, *et al.*, "A survey on llm-as-a-judge," *arXiv preprint arXiv:2411.15594*, 2024.
- [34] C. Sima, K. Renz, K. Chitta, *et al.*, "Drivelm: Driving with graph visual question answering," in *European Conference on Computer Vision*, Springer, 2025, pp. 256–274.