Generating Traffic Scenarios via In-Context Learning to Learn Better Motion Planner

Aizierjiang Aiersilan
University of Macau
ezharjan@outlook.com

Abstract

Motion planning is a crucial component in autonomous driving. State-of-the-art motion planners are trained on meticulously curated datasets, which are not only expensive to annotate but also insufficient in capturing rarely seen critical scenarios. Failing to account for such scenarios poses a significant risk to motion planners and may lead to incidents during testing. An intuitive solution is to manually compose such scenarios by programming and executing a simulator (e.g., CARLA). However, this approach incurs substantial human costs. Motivated by this, we propose an inexpensive method for generating diverse critical traffic scenarios to train more robust motion planners. First, we represent traffic scenarios as scripts, which are then used by the simulator to generate traffic scenarios. Next, we develop a method that accepts user-specified text descriptions, which a Large Language Model translates into scripts using in-context learning. The output scripts are sent to the simulator that produces the corresponding traffic scenarios. As our method can generate abundant safety-critical traffic scenarios, we use them as synthetic training data for motion planners. To demonstrate the value of generated scenarios, we train existing motion planners on our synthetic data, real-world datasets, and a combination of both. Our experiments show that motion planners trained with our data significantly outperform those trained solely on real-world data, showing the usefulness of our synthetic data and the effectiveness of our data generation method.

Code — https://ezharjan.github.io/AutoSceneGen

Introduction

Efficiently evaluating autonomous vehicles (AVs) under diverse real-world challenges on a limited budget is crucial for ensuring their safety and sustaining long-term growth in the AV industry. While existing works on vehicle or pedestrian trajectory prediction, such as TrafficPredict (Ma et al. 2019), Pishgu (Alinezhad Noghre et al. 2023), TraPHic (Chandra et al. 2019), and MSRL (Wu et al. 2023), rely on real-world datasets like ApolloScapes (Huang et al. 2018), NGSIM (Kovvali, Alexiadis, and Zhang PE 2007), UCY (Lerner, Chrysanthou, and Lischinski 2007), and ETH (Pellegrini et al. 2009), which may not fully capture the complexities of modern trajectory prediction scenarios involving both AVs and non-AVs, our custom virtual dataset, generated using our framework "AutoSceneGen", produces improved prediction results. Most importantly, generating and synthesizing data using our prompt-based, configurable, and AI-driven AutoSceneGen framework is more cost-effective and efficient than collecting real-world datasets such as Argoverse2 (Wilson et al. 2023), Waymo (Ettinger et al. 2021), Round (Krajewski et al. 2020), Ind (Bock et al. 2020), nuScenes (Caesar et al. 2020), Argoverse (Chang et al. 2019), Highd (Krajewski et al. 2018), and ApolloScapes. Our framework automatically generates datasets with real-world features and provides easy control over dataset heterogeneity through scenario descriptions, especially for safety-critical scenarios.

Status quo. Despite the challenges and inefficiencies in the testing phase of AVs, significant progress has been made in optimizing the testing of individual modules, such as motion planners. However, most efforts have focused on evaluating trajectory prediction models in simulation environments (Shet et al. 2023; Li et al. 2023). While simulations reduce testing costs related to budget and safety, creating realistic and logically specific scenarios often requires substantial human effort, though still less than testing AVs on real roads. Additionally, safety evaluation is typically performed module-by-module, demanding considerable resources to create safety-critical scenarios within simulators. Scenario-based evaluation methods have largely focused on coverage, unsafe scenarios, and indicator-estimation tests (Jesenski et al. 2019; Amersbach and Winner 2019). Recent advancements in scenario generation prioritize data-driven methods (Sun et al. 2024), with generative models trained on domain-specific data for traffic scenario generation (Tan et al. 2023). However, these methods struggle with low-resource scenarios, such as corner cases and accidents, due to limited data availability. There is a clear need for a universal, general, and budget-conscious framework that enhances traffic scenario heterogeneity automatically through scenario descriptions. This would expedite the simulation and testing process. While advanced AI tools, like large language models (LLMs), show promise, they often require extensive post-processing to correct inaccuracies or generate overly generic results. Our framework addresses these issues by automating complex scenario generation in

a cost-effective manner, overcoming the limitations of low-resource scenarios and optimizing the use of simulator functionalities to ensure scenarios are diverse, realistic, and reflective of challenging real-world situations, as highlighted by Bogdoll et al. (2021).

Contributions. Scenario generation has traditionally been a manual process requiring significant human effort. However, with the advancement of LLMs, there is now an opportunity to leverage AI to efficiently generate specific traffic scenarios. Building on prior research, the main contributions of this study are as follows:

- A universal, general, and cost-effective framework, "AutoSceneGen", is proposed to automatically enhance the heterogeneity of traffic scenarios through scenario descriptions, thereby accelerating the simulation and testing process.
- AutoSceneGen leverages in-context learning (ICL) of LLMs, eliminating the need for training or fine-tuning generative models for scenario generation tasks.
- The scenarios generated by AutoSceneGen were demonstrated to produce better datasets, leading to improved training results for motion planners.
- AutoSceneGen automatically categorizes the generated scenarios based on their descriptions, eliminating the need for downstream annotation and facilitating the training of motion planners in open-world scenarios.
- AutoSceneGen is a modular framework with dynamic components, allowing easy substitution of its generative model and simulation engine used for scenario generation and data collection.

Related Work

Scenario Generation and Gaps

Existing works, such as LCTGen (Tan et al. 2023), CSG (Xinxin, Fei, and Xiangbin 2020), and TrafficGen (Feng et al. 2023), focus on generating specific traffic scenarios but often rely on real-world data or are limited by the functionalities of toolkits like ScenarioNet (Li et al. 2024a). In contrast, AutoSceneGen bridges this gap by enabling users to efficiently collect AI-generated scenarios without relying on real-world data.

The concept of a "scenario" revolves around the "logic" governing objects within a scene. Go and Carroll (2004) defined a scenario as a detailed account involving actors, background details, objectives, and sequences of actions. In Autonomous Driving (AD), testing traditionally focuses on perception, motion planning, and decision-making modules (Huang et al. 2016). Simulators such as CARLA (Dosovitskiy et al. 2017) and AirSim (Shah et al. 2018) have been developed to support these tests. Goyal et al. (2023) proposed a method to configure a scenario and automatically generate similar variants, while König et al. (2024) introduced a formal model for test case generation and simulation verification. However, these approaches rely on abstract models, making them less scalable than prompt-based generation methods.

Several studies have explored augmenting traffic data and generating rare objects using ChatGPT (Alam 2020; Xinxin, Fei, and Xiangbin 2020). In this context, automatic scenario generation refers to creating scenarios from user prompts without detailed manual intervention. Zhang et al. (2023) reviewed the absence of automatic pipelines for safety-critical scenario generation in AD, highlighting challenges such as fidelity, efficiency, diversity, controllability, and transferability (Ding et al. 2023). For example, Yu et al. (2020) and Sun et al. (2021) addressed challenges in generating safe lane-changing and adversarial cases, respectively.

Table 1 compares the dataset generated by the AutoSceneGen with benchmark datasets, focusing primarily on AutoSceneGen's speed of data collection and diversity. The comparison emphasizes the framework's efficiency in data collection (measured per minute) and its ability to capture diverse data forms. This highlights AutoSceneGen's scalability, flexibility, and effectiveness in generating varied scenario types at scale. Unlike manually collected datasets, AutoSceneGen demonstrates superior adaptability and speed in producing diverse scenarios dynamically. Though our framework supports collecting various sensor data, including LiDAR, this paper focuses on collecting trajectory data to demonstrate its capabilities.

In-Context Learning

Do LLMs learn new tasks during inference, or do they simply recognize patterns from training? Mann et al. (2020) proposed that ChatGPT-3 can acquire new tasks through ICL, where task examples are embedded in prompts. They questioned whether models genuinely learn or merely recall familiar patterns, noting that synthetic tasks like word scrambling might be learned anew. However, doubts persist about whether LLMs truly understand prompts. For instance, Webson and Pavlick (2021) found that models performed similarly with irrelevant prompts, suggesting that improved performance might not equate to genuine comprehension. Min et al. (2022) argued that ICL depends more on input-label mapping, text distribution, and format than on actual ground-truth demonstrations, implying that ICL is ineffective without pre-existing input-label relationships. They also showed that ground-truth outputs may not be essential for tasks like generation. Their findings on NLP benchmarks leave unresolved questions about ICL's effectiveness in open-set tasks, its learning capacity during inference, and challenges like input-label mapping extraction. Additionally, Dai et al. (2022) suggested that ChatGPT-3's ICL might function through implicit gradient descent as a meta-optimizer.

The core basis of the AutoSceneGen leverages the ICL capability of LLMs to generate optimized scenario configurations from given templates, simulating these scenarios in batch within a physics-based simulator. The generated scenarios reflect realistic physical interactions, ensuring their relevance for AV safety evaluation. To our knowledge, there is a lack of a universal framework that generates abundant, safety-critical, and realistic traffic scenarios specifically tailored for the safety evaluation of AVs.

Count	UCY	ETH	NGSIM	KITTI	ApolloScapes	WaymoOpen	nuScenes	Argoverse2	AutoSceneGen
duration (min)	28	25	45	22	155	34200	0.34	458.34	60
frames ($\times 10^3$)	3000	3750	11.2	13.1	93.0	20k	40	27500	0.05
total ($\times 10^3$)									
pedestrian	120	0.65	0	0.09	16.2	N/A	222.164	20	0.25
bicycle	0	0	0	0.04	5.5	N/A	11.859	12.5	0.05
vehicle	0	0	2.91	0.93	60.1	N/A	1166.187	10k	0.1
average (1/f)									
pedestrian	0.04	0.18	0	1.3	1.6	N/A	N/A	N/A	250
bicycle	0	0	0	0.24	1.9	N/A	N/A	N/A	50
vehicle	0	0	845	3.4	12.9	N/A	N/A	N/A	100
device									
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
LiDAR	×	×	×	✓	✓	✓	✓	✓	✓
GPS	×	×	×	✓	✓	×	✓	×	✓
Radar	×	×	×	×	✓	×	✓	×	✓
IMU	×	×	×	×	×	×	✓	×	✓
GNSS	×	×	×	×	✓	×	×	×	✓
Lane	×	×	×	×	✓	✓	×	✓	✓
Collision	×	×	×	✓	×	×	×	×	✓

Table 1: A comparison of acquisition time, total frames, total instances, average instances per frame, and acquisition devices for the UCY, ETH, NGSIM, KITTI (Geiger et al. 2013) (with trajectories), ApolloScapes datasets, and the AutoSceneGen's. Unlike other datasets with static volume data, AutoSceneGen's values are dynamically calculated based on speed per minute. The data reflects static speed during collection and includes trajectories with 8 attributes. Data collection was performed on a machine with an NVIDIA GeForce RTX 3090 GPU, 12th Gen Intel(R) Core(TM) i7-12700, using CARLA version 0.9.13 with the "Town03" map (300×300).

Rare Objects in An Open World

Rare class objects, often overlooked in open-world data, are critical for ensuring model robustness in motion planning and visual detection. Manually creating such scenarios requires significant human effort, including writing detailed configurations, formulating scenario logic, and ensuring compatibility with the chosen simulator. These tasks demand extensive brainstorming and collaboration, particularly for rare and complex scenarios. For example, while an engineer might easily conceive a scenario involving a wrongly-parked vehicle with its doors open, envisioning affiliated events—such as a driver exiting the vehicle to chase someone or experiencing a sudden medical emergency—requires additional effort and time for ideation and planning. Such edge cases, though rare, are essential for the safety evaluation of AVs. This study addresses the challenge by leveraging LLMs' ICL capabilities to generate tailored configurations for rare scenarios, streamlining the ideation and scenario creation processes. Figure 1 shows a rare scenario generated with this approach.

Scenario Generation Efficiency

Advancements in scenario generation have primarily focused on enhancing domain-specific languages such as Scenic (Fremont et al. 2019) and ASAM OpenScenario. However, leveraging LLMs for scenario generation remains relatively unexplored. Li et al. (2024b) introduced a framework utilizing ChatGPT to generate trajectory data, but its raw outputs lack physical realism and consistency with real-world scenarios, requiring active user intervention through Chain-of-Thought prompting (Zhang et al. 2022). TARGET (Deng et al. 2023) have utilized LLMs for generating simulator scenarios by incorporating a domain-specific language to create configurations based on self-defined traffic rules. However, their approach relies on a newly introduced XML-based domain-specific language, requires continuous manual input and interpretation, and has not been systematically evaluated for efficiency or scalability. Similarly, other methods, such as abstract-to-scenario and concrete-to-scenario approaches proposed by Majzik et al. (2019), face limitations in automating scenario creation for accuracy-critical domains like AD. Our framework offers a universal, cost-effective solution to enhance traffic scenario heterogeneity and facilitate the generation of rare, safety-critical scenarios in open-world environments, thereby streamlining the simulation and testing processes.

In-Context Scenario Generation

Existing efforts in driving scenario generation have primarily focused on creating videos to assess the perceptual capabilities of AVs. Recent advancements in incorporating rare object synthesis into realistic images for driving scenario generation have shown promise, offering valuable insights for improving downstream tasks. However, the exploration of AI-driven content creation has broadened our perspective, moving beyond the traditional concept of a "video" as a scenario. The core of a "scenario" lies in the underlying logic that governs the existence and interaction of objects within

Figure 1: Images captured at four distinct timestamps and locations, corresponding to the AutoSceneGen input scenario description: "In downtown area, during a drizzly noon, there are vehicles malfunctioning windshield wipers and some of the vehicles' doors are open. Some vehicles exhibit negligent driving behavior, compromising visibility in wet conditions. There are 10 pedestrians on the road, with 50% of the pedestrian running. No one was hurt and no accident happened since all the vehicles except the malfunctioning one obeyed the traffic rules."

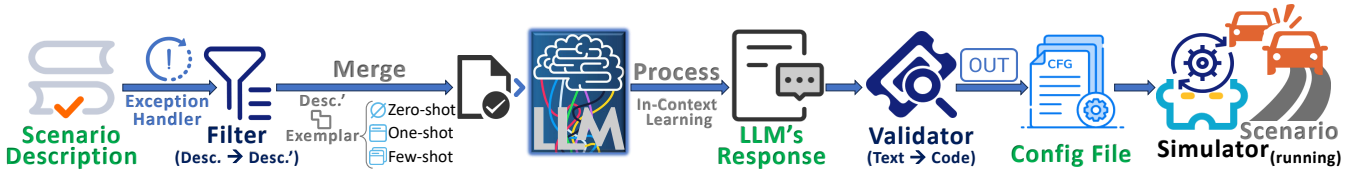


Figure 2: Architecture Overview. It begins with the user inputting a scenario description, which is managed by the *Exception Handler* to block adversarial or irrelevant inputs, ensuring the framework operates within scope and prevents downstream issues. The *Filter* processes the description, replacing simulator-incompatible terms with those aligned to the simulator's documented APIs. The filtered description (*Desc.'*) is combined with pre-constructed ICL exemplars, which can be zero-shot, one-shot, or few-shot in category, depending on the LLM's familiarity with the simulator's APIs and the complexity of the scenario. The LLM generates a response containing scenario configurations, often accompanied by explanations and comments. The *Validator* verifies each API call for compatibility, replacing unsupported terms with suitable alternatives (e.g., replacing "storm," unsupported in CARLA, with "rain") or ignoring them to prevent errors. This ensures all calls align with the simulator's capabilities, enabling execution of the final configuration file. The simulator runs the scenario, with the final step depicting the interaction between the real world and the virtual environment, while data collection can take place either inside the simulator (as is the case in this study) or externally.

it. Reflecting on past approaches, we sought a more universal method to define the logic behind object interactions for scenario generation, rather than focusing solely on the static objects themselves. This distinction sets our work apart from the scope of AI-generated content (AIGC), which typically involves optimizing workflows or synthesizing objects into images for video creation aimed at training sensors and detectors. However, AIGC requires the training of new models, a common challenge across various domains, highlighting the need for models tailored specifically to the automotive industry to ensure accuracy and control over generated content.

While training a comprehensive model may provide a reliable strategy for generating accurate, domain-specific content, sourcing the data necessary for model training can be challenging. Additionally, motion planners trained on datasets directly collected from the real world may struggle to generalize effectively to unseen tasks, impacting model robustness, especially given the long-tail distribution of open-world cases. Moreover, training models from scratch demands significant time and computational resources, presenting challenges in the context of AD. Therefore, this research aims to explore alternative scenario generation methods that bypass the need for training or fine-tuning AI models, offering a more efficient and scalable approach to generating driving scenarios.

With ICL, the need for continuous fine-tuning or training of separate models for generation tasks is largely eliminated. Instead, LLMs equipped with ICL capabilities can be leveraged effectively. These models extend beyond traditional applications, such as language translation, to encompass more novel tasks such as code generation and scenario generation. This section focuses on the methodologies employed by AutoSceneGen, which effectively integrates ICL within the domain of LLMs. The integration of LLMs is anticipated to enhance both the efficiency and controllability of AutoSceneGen, offering a more streamlined and adaptable approach to scenario generation.

System Architecture

As shown in Figure 2, the system architecture consists of key components for processing scenario descriptions, which can be provided by the user or extracted from images using a vision-language model. A filtering process ensures simulator compatibility by replacing incompatible terms with appropriate alternatives. For example, in the description "in stormy weather," the term "stormy" is identified as incompatible and replaced with "rainy." This replacement is performed using a predefined replacement dictionary derived from the simulator documentation, as detailed in Algorithm 1.

The user's scenario description is combined with few-shot

Algorithm 1: Filter for Scenario Descriptions

Input: Raw Scenario Description D
Output: Final Scenario Description D'

```
for each token t_i ∈ D do
    if t_i ∈ SimulatorDocument then
        t_i ← SimulatorDocument(t_i);
    else
        if t_i ∈ Words^incorrect ∪ Punctuations^wrong then
            Correct t_i;
```

Algorithm 2: Validator for Generated Configuration

Input: Generated configuration C_{LLM}
Output: Validated code F

```
Extract code from C_LLM to get F;
for each term t_i ∈ C_LLM do
    if t_i ∉ SimulatorDocument then
        Search for synonym t_i^syn;
        if ∃t_i^syn ∈ SimulatorDocument then
            Replace t_i with t_i^syn;
        else
            Remove t_i;
F ← C_LLM \ {comments, explanations};
```

learning examples within the framework. Developers adapting the framework to their specific domain need to create few-shot learning exemplars based on simulation needs. The process of constructing these exemplars depends on the chosen LLM: if all relevant APIs and examples are included in the LLM's train set, no additional exemplars are required. However, if the necessary documentation is not part of the train set, one-shot or few-shot exemplars need to be preconstructed to meet the simulation requirements. The LLM then generates the configuration by leveraging both the scenario description and the ICL examples to ensure an accurate simulation.

Prior to execution, the validator (as detailed in Algorithm 2) checks each API call against a documented list, replacing unsupported terms with approved synonyms (e.g., replacing "Weather.Storm" with "Weather.Rain" in CARLA). If no replacement is found, the term is ignored to prevent errors, ensuring the calls align with the simulator's capabilities. The validator also ensures that the final configuration includes only relevant code, removing any extraneous content such as explanations and comments, and verifies adherence to the syntax required by the simulator.

Finally, the simulator is launched to run the generated configuration files, ensuring seamless execution within the simulator itself or on connected digital twin platforms. Our framework automates AV evaluation by efficiently generating traffic scenarios from user-defined descriptions, ensuring cost-effective simulation across diverse open-world environments.

Input. The user provides scenario descriptions.

Process. The scenario descriptions are combined with simulator-specific examples and preprocessed to ensure clarity and compatibility. The chosen LLM then processes the input. The validator checks the generated configuration for adherence to the simulator's syntax and grammar, ensuring flawless execution.

Output. A scenario being simulated, provided by an executable configuration file, which is used by the simulator in either its purely virtual environment or on a digital twin platform connected to it.

LLMs & ICL

Since LLMs exhibit ICL, they can handle tasks they haven't been explicitly trained on without the need for retraining or fine-tuning, allowing them to perform well on specialized tasks using a few-shot learning approach.

Input. In the scenario generation process, the input consists of prompts (scenario descriptions). Additionally, to adapt the model to a specialized task, ICL examples that match the desired range of outputs must be provided alongside the scenario description as input to the LLMs.

Output. The response may include essential scenario elements directly usable by the simulator, as well as explanations that cannot be executed. To refine the response, two approaches are considered: manually scripting to isolate executable code or using the LLM to automatically extract only the code. It is generally believed that with proper ICL examples, the LLM will output only executable code without additional information. However, in the experiment, we found that even when we accentuated the requirement of "do not generate any explanations or comments except code" to the chosen LLMs, the output may still contain phrases like "Here is the result...." Additionally, considering that the code generated by LLMs undergoes grammar validation in the subsequent process, useful content extraction from its responses can also be accomplished through a manually scripted validation procedure. This approach ensures that the input prompt remains streamlined while preserving tokens for efficient processing.

Evaluations

The datasets generated using the AutoSceneGen can be extensive, with its simulator-dependency offering significant flexibility and scalability, which allows data to be collected for a wide range of scenarios as long as the simulator supports the required functions. By running the scenarios generated by the framework, diverse datasets can be efficiently created for various purposes and different applications. To evaluate AutoSceneGen, we utilized existing approaches for trajectory prediction. In the first step, we replaced their real datasets with our own; in the second step, we combined their datasets with ours. Finally, we tested exclusively with our own data. This allowed us to determine whether our data could achieve results comparable to those obtained with the original datasets. As an example, we used AutoSceneGen with GPT-4 (Achiam et al. 2023) as the selected LLM to generate 125 traffic scenarios, each based on a short prompt that described the scenario we aimed to generate. To eval-

Dataset	Method	TAE	ADEv	ADEp	ADEb	TFE	FDEv	FDEp	FDEb
A.S.	TrafficPredict	0.085	0.080	0.091	0.083	0.141	0.131	0.150	0.139
A.S. train-set + Ours	TrafficPredict	0.053	0.085	0.058	0.065	0.076	0.114	0.092	0.094
Ours	TrafficPredict	0.033	0.088	0.020	0.047	0.058	0.135	0.037	0.077
TRAF	TraPHic	5.63	N/A	N/A	N/A	9.91	N/A	N/A	N/A
A.S.(Reproduced)	TraPHic	5.10	3.62	1.02	4.49	2.81	6.73	1.88	8.44
A.S. train-set + Ours	TraPHic	1.30	1.69	0.42	0.90	2.10	2.82	0.67	1.39
Ours	TraPHic	0.27	0.14	0.19	0.44	0.40	0.21	0.30	0.62

Table 2: The comparison results of the ApolloScapes dataset, collected from AutoSceneGen (Ours), and the two datasets combined are presented. The term "ApolloScapes Dataset" is abbreviated as "A.S." in the table. Lower metrics indicate better performance. We generated 17,919 examples; the official training set of A.S. contains 94 examples. We used the original method proposed in TrafficPredict to train the planner. For TRAF and another evaluation, we used TraPHic. While the results are not as good as the results under TrafficPredict, under the method TraPHic there are still huge improvements thanks to the substitution of TRAF (Chandra et al. 2019) and ApolloScapes (Ma et al. 2019) to the dataset collected via AutoSceneGen. Suffix "v", "b" and "p" in "ADE"/"FDE" represents "vehicle", "bicycle" and "pedestrian" respectively.

uate how our work can expedite the safety evaluation process for AVs, we selected a subset of safe scenarios from the 125 scenario configurations, then loaded them into the simulator to simulate the generated scenarios. Next, we used the data collector to gather data in the required format and style, based on the comparison results that we would later use for evaluation. By directly replacing only the train set of the target dataset, we initiated the evaluation process of the scenarios collected through the AutoSceneGen framework using existing trajectory prediction approaches. Finally, after collecting the data in the required format and training the model from scratch to observe the trajectory prediction results, we directly added our dataset, collected through the AutoSceneGen, to the original training set used primarily for the evaluation of trajectory prediction methods by their proposers.

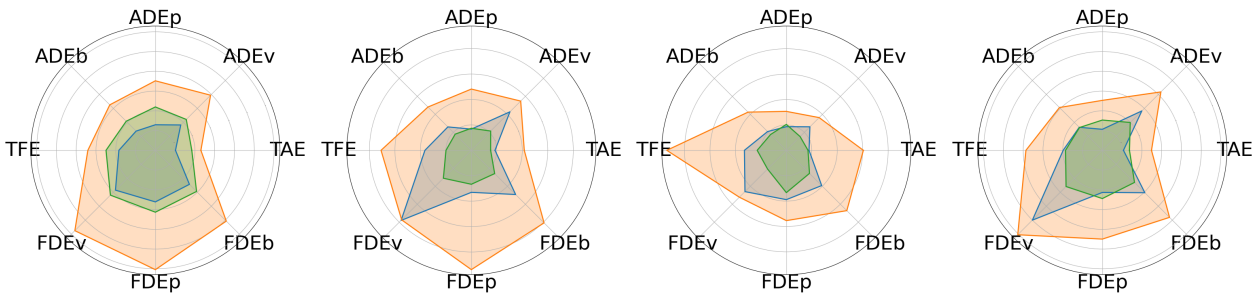
Not only can AutoSceneGen achieve the massive and efficient collection of critical scenarios, but the datasets it generates are also logically realistic and heterogeneous. In the experiment, we ran 41 AI-generated scenarios with varying numbers of vehicles and pedestrians to collect as much data as possible under different circumstances within a single scenario logic, using the official CARLA map from Town01 to Town05.

We evaluated our dataset using average displacement error (ADE) and final displacement error (FDE). We replicated TrafficPredict's results (Ma et al. 2019) with their ApolloScapes dataset, which features curated trajectory data from real road scenarios. For TRAF (Chandra et al. 2019), lacking access to their full dataset, we used ApolloScapes with the "TraPHic" method. This approach allowed us to assess the dataset collected from AutoSceneGen across different methods.

Without modifying the original trajectory prediction network, our dataset achieved superior results with reduced displacement error for each traffic participant type, as shown in Table 2. In various epochs, the dataset collected from AutoSceneGen demonstrated the highest accuracy in trajectory prediction, as illustrated in Figure 3-(a). Moreover, combining our dataset with ApolloScapes improved overall performance, enhancing all trajectory prediction metrics by incorporating diverse scenarios and extensive data, as depicted in Figures 3(b), (c), and (d).

Besides ApolloScapes, several experiments were also conducted on TraPHic, Pishgu (Alinezhad Noghre et al. 2023), and MSRL (Wu et al. 2023) using other datasets in a similar manner. Since the NGSIM dataset contains only vehicle trajectories and the ETH and UCY datasets contain only human trajectories, we excluded extra calculations on unseen participants in each comparison. For VIRAT (Oh et al. 2011) and ActEV (Awad et al. 2020), although they include three major event types (including vehicles), we focused only on the annotated pedestrian trajectories to control for variables when comparing under Pishgu. Table 3 presents a comparison of the results between our dataset and NGSIM, ETH/UCY, and VIRAT/ActEV. Our dataset, collected via AutoSceneGen using CARLA 0.9.13, includes 264 pedestrian trajectories (1,848k frames) and 770 vehicle trajectories (15,400k frames), with the vehicle trajectories replacing and augmenting NGSIM data.

Table 3 demonstrates that the dataset generated by AutoSceneGen achieves comparable or even superior results on several state-of-the-art trajectory prediction methods when compared to those using real-world road datasets. While benchmark datasets in Table 3 typically provide more diverse scenarios with manual annotations, the generated dataset is based on 125 scenario descriptions. This highlights the framework's effectiveness by revealing key trends in ADE and FDE. When combined with benchmark datasets, the generated data reduces displacement error by introducing valuable new scenarios. For instance, incorporating the generated data into NGSIM and ETH/UCY improves results compared to using either dataset alone, due to the complementary nature of the data rather than simply increasing volume. The generated dataset outperforms in most cases due to its scenario diversity. However, in VIRAT/ActEV, no significant improvement is observed, largely due to the chosen simulator's focus on vehicle simulation and limited pedestrian simulation capabilities. Nevertheless, the tendency for the generated data to outperform remains evident, suggest-



(a) Epoch 127 (b) Epoch 147 (c) Epoch 160 (d) Epoch 200

Figure 3: The comparison of all metrics between the datasets collected via AutoSceneGen (Blue), ApolloScapes (Orange), and the combination of the two datasets (Green) across different epochs is shown. While the dataset collected purely from AutoSceneGen outperforms ApolloScapes in some epochs, such as epoch 127, the combination of AutoSceneGen and ApolloScapes demonstrates better overall results. Due to the distinct distribution of traffic participants in the two datasets, Figures (b), (c), and (d) show sharper peaks for FDE-vehicle and ADE-vehicle. However, the combination of the two datasets achieves reasonable values overall. In this experiment, ApolloScapes has a total of 3,917 frames, AutoSceneGen has 17,919 frames, and the combined AutoSceneGen + ApolloScapes has 27,605 frames.

Dataset	Method	ADE	FDE
NGSIM	Pihgu	0.88	1.96
Ours	Pihgu	7.98	15.43
NGSIM train-set + Ours	Pihgu	0.84	1.87
ETH/UCY	Pihgu	1.10	2.24
Ours	Pihgu	1.48	2.70
ETH/UCY train-set + Ours	Pihgu	0.79	1.50
VIRAT/ActEV	Pihgu	14.11	27.96
Ours	Pihgu	16.05	31.09
VIRAT/ActEV + Ours	Pihgu	15.32	29.65

Table 3: The comparison results of the NGSIM dataset, the dataset collected from AutoSceneGen (ours), and the combination of the two datasets are shown. When replacing the NGSIM dataset with ours, the ADE and FDE values are much higher (worse) than when using the original NGSIM dataset. However, when we combine the two datasets—NGSIM and ours—the ADE and FDE decrease and outperform the results obtained from using the NGSIM dataset alone, indicating that the original NGSIM dataset is augmented by our dataset collected via AutoSceneGen. A similar observation was made with the ETH/UCY dataset.

ing potential for further improvement with more diverse pedestrian simulation.

Limitations

Our framework has several limitations, including performance, simulator integration, and data availability, as well as regulatory constraints due to varying global traffic laws. Developers may need to create custom virtual environments or choose different simulators if those mentioned in this paper do not meet their regulatory requirements. The framework's effectiveness depends on the capabilities of both LLMs and the simulator, particularly for tasks such as scenario generation and accident reconstruction. If the chosen LLM lacks ICL capabilities or the simulator cannot handle complex scenarios, the framework's utility is significantly reduced, making it difficult to generate corner cases. Like many AIGC toolkits, the explainability of the generated scenario logic is limited to the explainability of the scenario descriptions and ICL examples, especially at higher abstraction levels. This limitation arises from the chosen LLMs, not from the framework itself. To address this, the framework enhances explainability through detailed configuration files stored after validation, allowing users to trace the rationale behind generated scenarios by comparing input prompts with generated configurations.

Last but not least, if the example data for ICL is insufficient, some simulators may not respond effectively, necessitating post-revision of the responses. Most example data is sourced from official documentation or the open-source community of the chosen simulator, but availability and usage rights are not always guaranteed.

Conclusions

AutoSceneGen framework is designed for the generating of traffic scenarios using foundation models. The resulting scenarios can be simulated within virtual environments in simulators, within the real world, or even both concurrently. The framework achieves the improvement of the work efficiency on generating sufficient safety-critical scenarios conveniently for testing AVs in an open world, as well as the convenience of traffic accident reconstruction. As highlighted by Kalra and Paddock (2016), AVs face the challenge of needing to cover vast distances — potentially hundreds of millions to even billions of miles — to establish their reliability in terms of minimizing fatalities and injuries. AutoSceneGen is particularly advantageous for enhancing the reliability testing of AVs slated for real-world deployment. It achieves this by harnessing the power of foundation models to simulate extensive arrays of traffic scenarios, thereby providing a robust testing environment for AVs.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alam, M. S. 2020. Automatic generation of critical driving scenarios.

Alinezhad Noghre, G.; Katariya, V.; Danesh Pazho, A.; Neff, C.; and Tabkhi, H. 2023. Pishgu: Universal path prediction network architecture for real-time cyber-physical edge systems. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, 88–97.

Amersbach, C.; and Winner, H. 2019. Defining required and feasible test coverage for scenario-based validation of highly automated vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 425–430. IEEE.

Awad, G.; Butt, A. A.; Curtis, K.; Lee, Y.; Fiscus, J.; Godil, A.; Delgado, A.; et al. 2020. Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. *arXiv preprint arXiv:2009.09984*.

Bock, J.; Krajewski, R.; Moers, T.; Runde, S.; Vater, L.; and Eckstein, L. 2020. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1929–1934. IEEE.

Bogdoll, D.; Breitenstein, J.; Heidecker, F.; Bieshaar, M.; Sick, B.; Fingscheidt, T.; and Zöllner, M. 2021. Description of corner cases in automated driving: Goals and challenges. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1023–1028.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; et al. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Chandra, R.; Bhattacharya, U.; Bera, A.; and Manocha, D. 2019. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8483–8492.

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

Dai, D.; Sun, Y.; Dong, L.; Hao, Y.; Ma, S.; Sui, Z.; and Wei, F. 2022. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*.

Deng, Y.; Yao, J.; Tu, Z.; Zheng, X.; Zhang, M.; and Zhang, T. 2023. Target: Traffic rule-based test generation for autonomous driving systems. *arXiv preprint arXiv:2305.06018*.

Ding, W.; Xu, C.; Arief, M.; Lin, H.; Li, B.; and Zhao, D. 2023. A survey on safety-critical driving scenario generation—A methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 24(7): 6971–6988.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, 1–16. PMLR.

Ettinger, S.; Cheng, S.; Caine, B.; Liu, C.; Zhao, H.; Pradhan, S.; Chai, Y.; et al. 2021. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9710–9719.

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

Fremont, D. J.; Dreossi, T.; Ghosh, S.; Yue, X.; Sangiovanni-Vincentelli, A. L.; and Seshia, S. A. 2019. Scenic: a language for scenario specification and scene generation. In *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, 63–78.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.

Go, K.; and Carroll, J. M. 2004. The blind men and the elephant. *Interactions*, 11(6): 44–53.

Goyal, S.; Griggio, A.; Kimblad, J.; and Tonetta, S. 2023. Automatic Generation of Scenarios for System-level Simulation-based Verification of Autonomous Driving Systems. *arXiv preprint arXiv:2311.09784*.

Huang, W.; Wang, K.; Lv, Y.; and Zhu, F. 2016. Autonomous vehicles testing methods review. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE.

Huang, X.; Cheng, X.; Geng, Q.; Cao, B.; Zhou, D.; Wang, P.; Lin, Y.; and Yang, R. 2018. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 954–960.

Jesenski, S.; Stellet, J. E.; Schiegg, F.; and Zöllner, J. M. 2019. Generation of scenes in intersections for the validation of highly automated driving functions. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 502–509. IEEE.

Kalra, N.; and Paddock, S. M. 2016. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94: 182–193.

König, L.; Heinzemann, C.; Griggio, A.; Klauck, M.; Cimatti, A.; Henze, F.; et al. 2024. Towards Safe Autonomous Driving: Model Checking a Behavior Planner during Development. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 44–65. Springer.

Kovvali, V. G.; Alexiadis, V.; and Zhang PE, L. 2007. Video-based vehicle trajectory data collection. Technical report, _.

Krajewski, R.; Bock, J.; Kloeker, L.; and Eckstein, L. 2018. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st international conference on intelligent transportation systems (ITSC)*, 2118–2125. IEEE.

Krajewski, R.; Moers, T.; Bock, J.; Vater, L.; and Eckstein, L. 2020. The round dataset: A drone dataset of road user trajectories at roundabouts in germany. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–6. IEEE.

Lerner, A.; Chrysanthou, Y.; and Lischinski, D. 2007. Crowds by example. In *Computer graphics forum*, volume 26, 655–664. Wiley Online Library.

Li, D.; Liu, B.; Huang, Z.; Hao, Q.; Zhao, D.; and Tian, B. 2023. Safe motion planning for autonomous vehicles by quantifying uncertainties of deep learning-enabled environment perception. *IEEE Transactions on Intelligent Vehicles*.

Li, Q.; Peng, Z. M.; Feng, L.; Liu, Z.; Duan, C.; Mo, W.; and Zhou, B. 2024a. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in neural information processing systems*, 36.

Li, X.; Liu, E.; Shen, T.; Huang, J.; and Wang, F.-Y. 2024b. ChatGPT-based scenario engineer: A new framework on scenario generation for trajectory prediction. *IEEE Transactions on Intelligent Vehicles*.

Ma, Y.; Zhu, X.; Zhang, S.; Yang, R.; Wang, W.; and Manocha, D. 2019. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 6120–6127.

Majzik, I.; Semerath, O.; Hajdu, C.; Marussy, K.; Szatmari, Z.; Micskei, Z.; et al. 2019. Towards system-level testing with coverage guarantees for autonomous vehicles. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*. IEEE.

Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; and Zettlemoyer, L. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Oh, S.; Hoogs, A.; Perera, A.; Cuntoor, N.; Chen, C.-C.; Lee, J. T.; ; et al. 2011. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, 3153–3160. IEEE.

Pellegrini, S.; Ess, A.; Schindler, K.; and Van Gool, L. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, 261–268. IEEE.

Shah, S.; Dey, D.; Lovett, C.; and Kapoor, A. 2018. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 621–635. Cham: Springer International Publishing.

Shet, R.; Iyer, N. C.; Mirje, M.; Bikkannavar, K. V.; and Rokhade, S. 2023. Path Planning of Autonomous Vehicle for Real World Scenario Using CARLA. In *International Conference on Information and Communication Technology for Competitive Strategies*, 81–92. Springer.

Sun, H.; Feng, S.; Yan, X.; and Liu, H. X. 2021. Corner case generation and analysis for safety assessment of autonomous vehicles. *Transportation research record*, 2675(11): 587–600.

Sun, S.; Gu, Z.; Sun, T.; Sun, J.; Yuan, C.; Han, Y.; Li, D.; and Ang, M. H. 2024. Drivescenegen: Generating diverse and realistic driving scenarios from scratch. *IEEE Robotics and Automation Letters*.

Tan, S.; Ivanovic, B.; Weng, X.; Pavone, M.; and Kraehenbuehl, P. 2023. Language Conditioned Traffic Generation. In *Conference on Robot Learning*, 2714–2752. PMLR.

Webson, A.; and Pavlick, E. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

Wilson, B.; Qi, W.; Agarwal, T.; Lambert, J.; Singh, J.; Khandelwal, S.; Pan, B.; et al. 2023. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*.

Wu, Y.; Wang, L.; Zhou, S.; Duan, J.; Hua, G.; and Tang, W. 2023. Multi-stream representation learning for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2875–2882.

Xinxin, Z.; Fei, L.; and Xiangbin, W. 2020. Csg: Critical scenario generation from real traffic accidents. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1330–1336. IEEE.

Yu, Y.; Liu, S.; Jin, P. J.; Luo, X.; and Wang, M. 2020. Multiplayer dynamic game-based automatic lane-changing decision model under mixed autonomous vehicle and human-driven vehicle environment. *Transportation research record*, 2674(11): 165–183.

Zhang, X.; Tao, J.; Tan, K.; Törngren, M.; Sánchez, J. M. G.; et al. 2023. Finding critical scenarios for automated driving systems: A systematic mapping study. *IEEE Trans. Softw. Eng.*, 49(3): 991–1026.

Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.