A Survey on Large Model-based Safety-critical Scenario Generation for Autonomous Driving

1st Lingzhong Meng
Institute of Software
Chinese Academy of Sciences
Beijing, China
lingzhong@iscas.ac.cn

2nd Cancan Jiang
Institute of Software
Chinese Academy of Sciences
Beijing, China
jiangcancan24@mails.ucas.ac.cn

3rd Zhidong Wang
Qiyuan Lab
Beijing, China
18742040912@163.com

4th Guang Yang
Institute of Software
Chinese Academy of Sciences
Beijing, China
yangguang@iscas.ac.cn

5th Hongyun Yu
Institute of Software
Chinese Academy of Sciences
Beijing, China
hongyun@iscas.ac.cn

6th Yuxi Ma*
Institute of Software
Chinese Academy of Sciences
Beijing, China
mayuxi@iscas.ac.cn

***Abstract*—Safety-critical scenario generation for autonomous driving aims to efficiently and authentically construct long-tail, high-risk traffic scenarios for simulation testing. The core of this task lies in leveraging generative technologies to address the scarcity and high acquisition costs of extreme accident data in real-world road testing. This paper reviews existing LLM-based safety-critical scenario generation technologies.**

First, we establish a systematic classification framework for these technologies. We categorize existing methods into three types: Prompt Engineering, RAG, and AI Agent-based approaches. Second, we summarize their core mechanisms, strengths, and limitations using multi-dimensional metrics. Finally, building on the analysis of current technical constraints, we discuss the challenges facing the field and propose potential research directions, offering insights for further innovation.

I. Introduction

Autonomous driving has evolved from driver assistance to open-environment operation [1]. Despite maturity in routine traffic, real-world deployment remains challenging [2] due to the "long-tail effect" [3], where accidents typically occur in low-probability, high-complexity "safety-critical scenarios" [4]. Since real-vehicle testing is prohibitively expensive [5], simulation-based generation is an industry consensus [6].

Traditional generation methods, such as statistical sampling, adversarial optimization, and specific generation networks, often suffer from limited diversity, poor semantic control, and weak generalization [7]. Recently, Large Language Models (LLMs) introduced powerful semantic understanding and logical reasoning, overcoming these limitations.Fig. 1 illustrates this technical evolution. However, existing surveys rarely focus specifically on LLM-based safety-critical scenario generation.

To fill this gap, this paper presents the first classification and summary of this field. The main contributions of this paper are summarized as follows:

*Corresponding author.

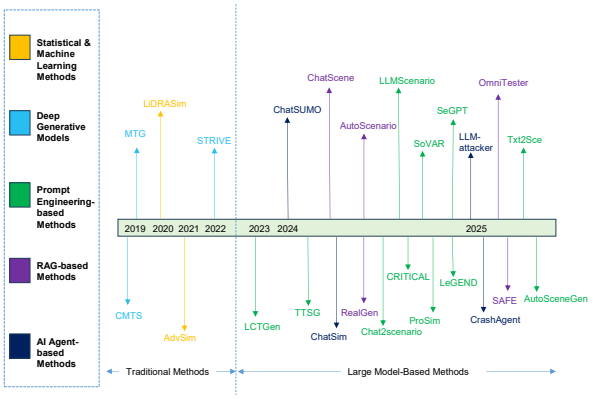

Fig. 1. The evolution of various methods over time.

1) **Method Classification:** We categorize existing methods into Prompt Engineering, RAG, and AI Agent-based approaches.
2) **Analysis of Pros and Cons:** Multi-dimensional Analysis: We summarize core mechanisms and analyze their advantages and limitations.
3) **In-depth Analysis of Future Directions:** We discuss current technical constraints and propose potential research directions to foster further innovation in this field.

II. Overview

A. Definition of Safety-Critical Scenarios

Currently, a unified definition of safety-critical scenarios remains elusive. Early definitions were predominantly based on the NHTSA accident classification [8], focusing on critical events preceding a collision.

In recent years, the research focus has shifted toward a data distribution perspective. Bogdoll et al. [9] characterize them as extreme cases encompassing long-tail and Out-Of-Distribution (OOD) samples, while Sun et al. [10] emphasize their distributional characteristics of "low probability and high risk."

With the advancement of interactive generation technologies, the definition has further evolved toward adversarial behaviors. Ding et al. [2] formalized this as a constrained optimization problem: finding the subset of parameters that induces the maximum failure response of the system, subject to constraints on physical feasibility and data likelihood.

We formally define safety-critical scenarios as a subset satisfying:

$$\mathcal{S}_{critical} = \{s \in \Omega \mid P(s) < \epsilon, \mathcal{C}(s) > \tau, \mathcal{L}(f_{ADS}(s)) > \lambda\} \tag{1}$$

where Ω represents the universal scenario space; $P(s) < \epsilon$ denotes an extremely low occurrence probability; $\mathcal{C}(s) > \tau$ indicates that the environmental or adversarial complexity exceeds the threshold τ; and $\mathcal{L}(f_{ADS}(s)) > \lambda$ signifies that the performance loss of the Autonomous Driving System (ADS) exceeds the safety limit λ .

B. Scenario Representation

Scenario representation maps the physical world into a digital model with logical relationships, directly determining the architecture and performance of the generative model. Building upon the classic PEGASUS five-layer model [11], Scholtes et al. [12] extended it to six layers, covering road geometry, infrastructure, temporary modifications, object trajectories, environmental conditions, and digital communication. However, XML-based standards' reliance on exact coordinates limits their flexibility.

To address this, Fremont et al. [13] proposed Domain Specific Languages (DSLs) such as SCENIC. Unlike verbose XML, SCENIC utilizes probabilistic programming and compact syntax to define spatial distributions via descriptive language rather than precise coordinates. This characteristic effectively circumvents "hallucination" issues in LLMs when processing numerical values: the model need only generate high-level semantic descriptions, which are then converted into precise instances by the compiler, thereby enabling the automated generation of safety-critical scenarios.

C. Metrics and Annotation

Scenario evaluation requires a synthesis of multi-dimensional metrics, primarily covering four aspects: criticality, realism, diversity, and model effectiveness.

- **Criticality:** This is the core dimension of safety assessment. It primarily relies on kinematic metrics for quantification, such as Time-to-Collision (TTC) [14], Post-Encroachment Time (PET) [15], and the deceleration required to avoid a collision [16]. Furthermore, the Responsibility-Sensitive Safety (RSS) model proposed by Mobileye defines the boundaries of safe behavior through formal mathematical formulas [17].
- **Realism:** This dimension aims to ensure that generated scenarios conform to physical laws and real-world driving distributions. A common practice involves using statistical distances, such as Jensen-Shannon (JS) [18] divergence and Wasserstein Distance (WD) [19], to measure the distributional discrepancies between generated trajectories and real-world datasets. Alternatively, the Fréchet Inception Distance (FID) metric is used to evaluate the perceptual fidelity of sensor-based generated data [20].
- **Diversity:** This metric is used to avoid the generation of a single pattern. Evaluation is typically achieved by calculating the coverage of the parameter space or by utilizing clustering algorithms to analyze the richness of the generated scenarios [21].
- **Model Effectiveness:** Since large models primarily drive simulations by generating intermediate code, this serves as a crucial complementary dimension for measuring the performance of different methods. This dimension focuses on two main aspects: executability rate and semantic consistency [22] [23].

D. Datasets

High-quality datasets underpin scenario understanding and are categorized into three modality types:

- Traffic accident text datasets provide critical semantic support. Sources like NHTSA's NMVCCS/CRSS and DMV Disengagement Reports offer real-world accident priors and failure details, facilitating parameterized reconstruction and adversarial generation.
- Naturalistic driving trajectory datasets offer physical constraints and retrieval repositories. Datasets such as HighD [24], nuScenes [25], and Waymo [26] cover diverse operational conditions, serving as foundations for retrieval databases, in-context examples, and physical plausibility evaluation.
- Multimodal datasets are pivotal for Agent-based approaches. DriveLM [27] enhances reasoning via Chain-of-Thought (CoT) annotations, while NuPrompt and Talk2Car [28] provide "language-trajectory" mappings to support end-to-end generation and cross-modal alignment.

E. Classification Framework

The evolution of LLMs from Prompt Engineering [29] [30] and RAG [31] to AI Agents [32] serves as the theoretical foundation for the taxonomy presented in this paper. Accordingly, existing methodologies are categorized into three distinct classes: Prompt Engineering-based methods focus on converting natural language inputs into static structured descriptions; RAG-based methods mitigate model hallucinations and enhance scenario realism by incorporating external knowledge bases; and AI Agent-based methods integrate memory mechanisms and tool invocation, achieving dynamic and adversarial

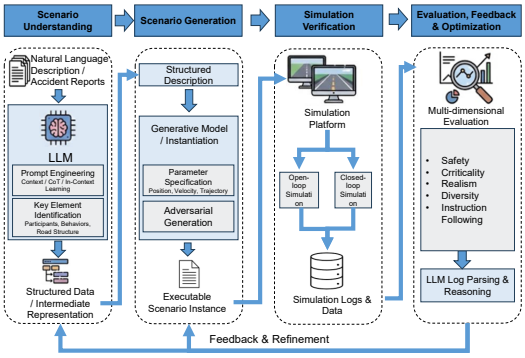

Fig. 2. The general workflow of LLM-based safety-critical scenario generation, comprising understanding, generation, execution, and feedback.

scenario construction through multi-agent collaboration and closed-loop interactions.

As illustrated in Fig. 2, the workflow of LLM-based safety-critical scenario generation typically comprises four core stages:

1) **Scenario Understanding**: Decomposing user inputs or accident reports to extract key elements and constraints;
2) **Scenario Generation**: Utilizing internal commonsense knowledge, reference cases, or external tools to precisely map ambiguous requirements into standardized simulation formats;
3) **Scenario Instantiation and Validation**: Instantiating abstract descriptions into concrete cases via simulation, logical deduction, or field tests to verify physical feasibility and filter invalid samples;
4) **Feedback Iteration**: Calculating quantitative metrics (e.g., TTC) and adjusting strategies to achieve closed-loop optimization.

III. SAFETY-CRITICAL SCENARIO GENERATION VIA PROMPT ENGINEERING

Prompt Engineering methods formalize scenario generation as text completion or code generation. Guided by strategies such as chain-of-thought reasoning and few-shot learning, these methods can generate logically consistent safety-critical scenarios without updating model parameters.

A. Method Characteristics and Key Technologies

The core technical characteristics of Prompt Engineering-based methods are primarily manifested in the following three dimensions:

1) **Structured Prompting**: Designing descriptors or markup languages to guide the model in transforming natural language into structured data interpretable by simulators.
2) **In-Context Learning (ICL)**: Utilizing instruction-to-code demonstrations to guide the model in analyzing mapping patterns, thereby converting novel requirements into compliant script files through analogical reasoning.
3) **CoT Reasoning**: Decomposing end-to-end tasks into step-by-step progressive stages—such as environment parsing, intention inference, and trajectory calculation—to ensure logical coherence.

B. Representative Works

This section compares typical prompt engineering-based works listed in Table I, which explore LLM potential in scenario construction via strategies like chain-of-thought, hierarchical prompting, and code generation.

As a representative work, LLMScenario proposes a closed-loop "Describe-Generate-Reflect" framework (Fig. 3). This method utilizes chain-of-thought to refine vague requirements into structured intermediate representations, maps semantics to parameters via JSON templates, and employs a self-reflection mechanism to autonomously correct logical flaws based on realism scores.

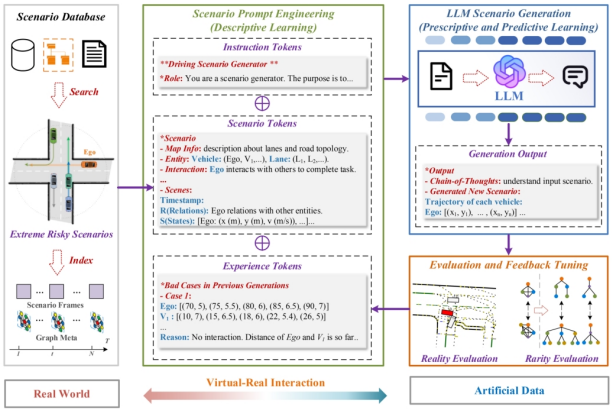

Fig. 3. LLMScenario's closed-loop "Describe-Generate-Reflect" framework [33]

LCTGen learns joint representations of text and HD maps via a Transformer to generate topology-compliant traffic flows, overcoming the limitation of TTSG which relies solely on road ID matching. ProSim further incorporates sketch-based interaction, enriching multimodal methods for spatial definition.

Txt2Sce innovatively employs LLMs to construct semantic scene trees while delegating numerical generation to rule-based operators, circumventing model randomness. Similarly, Sovar and LEGEND replace LLM calculations with mathematical solvers and multi-objective optimization, respectively, effectively addressing the lack of physical consistency in pure generative methods.

Gao et al. guide background vehicles to execute aggressive behaviors through multi-round games, proactively probing safety boundaries. Unlike CRITICAL, which focuses on log-based repair, this game-theoretic mechanism dynamically adjusts strategies to expand the verification dimension of system boundary capabilities.

C. Analysis

Prompt engineering-based methods enhance generation efficiency by leveraging semantic generalization, yet they face

TABLE I
COMPREHENSIVE ANALYSIS OF PROMPT ENGINEERING METHODS AND GENERATION CAPABILITIES

Paper	Input	Technique	Generation Capabilities			Key Limitation
			Output	Constr.	Closed	
LLMScenario [33]	Natural Language	CoT, Self-Reflection	Simulation Script	○	✓	Lacks spatial constraints.
LCTGen [34]	Latent Encoding	Semantic Injection	Trajectory Dist.	●	×	Open-loop; cannot be fine-tuned.
TTSG [35]	Natural Language	Key Entity Extraction	Map ID	●	×	Relies on pre-built maps.
SeGPT [36]	Natural Language	Curriculum Learning	Discrete Trajectory	○	×	Rigid and unrealistic motion.
Txt2Sce [37]	Accident Reports	Evolutionary Mutation	Scene Parameters	●	✓	Huge computational cost.
LeGEND [38]	Natural Language	Hierarchical Refinement	Scene Parameters	●	✓	Relies on traditional optimization.
SoVAR [39]	Accident Reports	Formal Translation + SMT	Trajectory Param.	●	×	Difficult interaction modeling.
Aiersilan et al. [40]	Natural Language	ICL (In-Context Learning)	Python Script	◐	×	Prone to API call errors.
Gao et al. [41]	Natural Language	Multi-view Adversarial	Attack Trajectory	○	✓	Simple physical environment.
CRITICAL [42]	Driving/Text	Iterative Optimization	Scene Parameters	●	✓	High inference cost.
ProSim [43]	Multimodal Prompts	Multimodal Fusion	Trajectory Flow	●	✓	Huge resource consumption.

Note: Constr. (Constraints): ●= Strong, ◐= Medium, ○= Weak. Closed (Closed-loop): ✓= Yes, × = No.

significant limitations. Physical inconsistency arises as the probabilistic nature of LLMs often yields parameters violating vehicle dynamics. Precision errors result from relying on general models without domain-specific fine-tuning, while high-quality data for SFT remains scarce. Furthermore, context constraints cause memory decay over long sequences due to limited window sizes, compromising the logical consistency required for complex scenarios.

IV. SAFETY-CRITICAL SCENARIO GENERATION VIA RAG

RAG-based methods introduce dynamic external knowledge bases, such as High-Definition maps and standard syntax. By utilizing retrieved reliable domain examples to constrain LLM reasoning, these methods significantly improve the precision and physical plausibility of scenario generation.

A. Methodological Characteristics and Key Technologies

The essence of this approach is to endow the model with external knowledge via real-time data querying.

1) Domain Knowledge Completion: Retrieving data from HD maps, reports, and regulations constrains generation within physical laws and ensures executability.

2) Retrieval-Augmented In-Context Learning: Leveraging in-context learning, the system retrieves matching code templates or cases to construct few-shot prompts. This guides the model to generate standardized scenarios adhering to specific syntax without fine-tuning.

B. Representative Works

As a representative work utilizing retrieval-augmented generation for safety-critical scenarios, *ChatScene* proposes a "hierarchical retrieval-modular assembly" generation framework, as shown in Fig. 4. This method constructs a specialized knowledge base and transforms the complex generation task into a multi-stage structured mapping process. Its core mechanism revolves around retrieval and generation: first, the system establishes road topology and traffic flow layout through retrieval; subsequently, it retrieves formally verified code snippets for specific behavioral logic. Finally, the LLM populates the retrieved standard templates with corresponding parameters. By confining generation to valid code, it circumvents compilation errors common in text-to-code conversion.

Other notable approaches also leverage retrieval mechanisms. *Chat2Scenario* directly retrieves trajectories from datasets such as HighD to replace abstract parameters, thereby aligning vehicle acceleration, deceleration, and lane-changing behaviors more closely with natural driving distributions. *OmniTester* addresses the spatial mismatch between pure text generation and road geometry by employing a spatial alignment strategy that projects generated agent behaviors onto retrieved HD map lanes. Furthermore, *Seeking to Collide* establishes a closed-loop adversarial strategy by inscribing attack patterns newly discovered during simulation into a memory bank.

C. Analysis

RAG mitigates the scarcity of domain knowledge in general LLMs by integrating external standards like OpenSCENARIO

TABLE II
COMPARISON OF RAG METHODS AND CAPABILITIES

Method	Source	Technique	Generation Capabilities			Limitation
			Output	Realism	Loop	
Chat2Scenario [44]	Naturalistic Driving Datasets	Param. Correction	OpenSCENARIO File	●	×	Relies on specific data formats.
ChatScene [22]	Scenario-Code Pair DB	Slot Filling	DSL / Python	◐	×	Generalization limited by DB scale.
OmniTester [45]	External Tools / Knowledge	Tool Integration	SUMO Script	●	✓	Relies on simulator interfaces.
RealGen [46]	Example Scenario Lib.	Latent Guidance	Trajectory Matrix	◐	×	Retrieval increases inference latency.
AutoScenario [47]	Multimodal Env. Features	Parametric Conv.	Python / XML	●	×	Info loss during modality conversion.
SAFE [48]	Structured Crash Reports	Causal Verification	Structured JSON	●	✓	High latency per generation.
Mei et al. [49]	Intent-Planner Memory	Feedback Loop	Adversarial Trajectory	●	✓	High online computational overhead.

Note: Realism: ●= High, ◐= Medium. Loop: ✓= Yes, × = No.

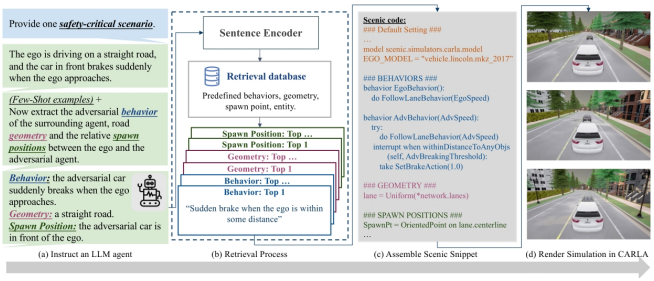

Fig. 4. The hierarchical retrieval-modular assembly framework proposed by ChatScene [22]

and HD maps. By referencing verified templates, this approach improves syntactic correctness and physical consistency without fine-tuning, while also offering interpretability for system verification.

However, it faces objective constraints. First, its performance relies heavily on the retrieval repository's coverage, struggling with out-of-distribution scenarios. Second, the prevalent open-loop structure often lacks feedback mechanisms to iteratively refine outputs based on simulation results.

V. SAFETY-CRITICAL SCENARIO GENERATION VIA AI AGENT

AI Agent methods advance scenario generation from static text output to interactive systems. By leveraging external tools and dynamic feedback, these agents optimize logical consistency and testing validity through closed-loop iteration.

A. Method Characteristics

AI Agents transform generation into a closed-loop optimization process via environmental perception and decision-making. Key characteristics include:

1) Multi-Agent Collaborative Planning: To manage complex spatiotemporal dependencies, tasks are decomposed into vertical subtasks. This division mitigates attention divergence in long-text reasoning and enhances scenario self-consistency through mutual constraints among models.

2) Environment Interaction and Closed-Loop Correction: Agents invoke HD maps or physics engines via APIs to monitor simulation results in real-time. Upon detecting physical violations or insufficient criticality, the system autonomously performs error diagnosis and parameter adjustment to re-execute the task.

B. Representative Works

Mimicking the accident reproduction logic of human experts, CrashAgent constructs a dual-agent collaborative architecture. As illustrated in Fig. 5, this framework first employs a reasoning agent to process accident reports containing unstructured text and simplified diagrams, extracting key environmental elements and deducing the causal chain of the accident. Subsequently, a construction agent transcribes this semantic information into the standard OpenSCENARIO format. Through strict parameter alignment, it ensures that the generated scenarios maintain high consistency with the original reports in terms of collision types and kinematic characteristics.

ChatSim integrates the McNeRF toolchain to edit geometric structures and texture details within scenarios via language instructions, achieving parameter alignment ranging from semantics to lighting and weather conditions. Generating OOD Scenarios incorporates the Tree of Thoughts (ToT) mechanism, utilizing a branching structure combined with rule-based pruning to systematically traverse rare, Out-Of-Distribution long-tail scenarios. Finally, LLM-Attacker establishes a closed-loop system equipped with dynamic game-theoretic capabilities. By filtering high-risk background vehicles in real time and

TABLE III
DETAILED ANALYSIS OF AGENT INTERACTION AND GENERATION CAPABILITIES

Method	Interaction Mech.	Core Task	Generation Capabilities			Limitation
			Output	Realism Focus	Loop	
ChatSim [50]	Hierarchical Mgmt.	Visual Editing	Blender Assets	Visual Appearance	×	Slow rendering inference.
CrashAgent [51]	Role-Based Coop.	Accident Recreation	Structured Scenario	Event Logic	×	Hard to complete missing info.
LLM-Attacker [52]	Adversarial Game	Attack Strategy Opt.	Adversarial Traj.	Behavioral Risk	✓	Search space explosion.
ChatSUMO [53]	Tool Integration	Macro Road Gen.	Python Scripts	Macro Traffic	×	Lacks microscopic game control.
Aasi et al. [54]	Tree-Search (ToT)	Long-tail Mining	Sim. Config	Data Diversity	×	Surge in computational cost.
Miao et al. [55]	VLM-Guided	Video-to-Sim Recon.	SCENIC Scripts	Sensor Fidelity	✓	Semantic detail loss.

Note: Loop: ✓= Yes (Closed-loop), × = No (Open-loop).

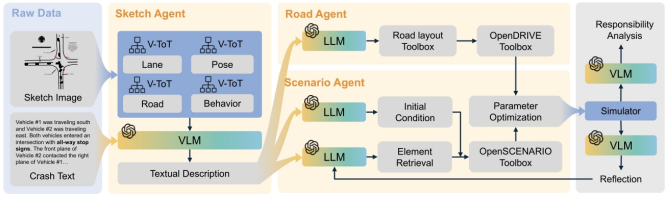

Fig. 5. The dual-agent collaborative architecture of CrashAgent [51]

adjusting attack strategies, it achieves the automated evolution of adversarial scenarios.

C. Analysis

AI Agent methods excel in automated construction and targeted adversarial testing, significantly reducing production costs while achieving alignment from semantic instructions to 3D geometry. By employing closed-loop mechanisms, these methods enhance adversarial attack success rates while maintaining human-like kinematics. Furthermore, the reduced collision rates of driving policies trained on such scenarios verify their practical utility.

However, practical applications remain subject to physical and computational constraints. Since model outputs lack intrinsic physical constraints, generated descriptions often violate kinematics, necessitating complex rule-based post-processing. Additionally, multi-agent collaboration introduces significant computational overhead and inference latency. Finally, generation accuracy relies on input quality; vague reports can cause reasoning bias to accumulate over long execution chains, compromising scenario reproduction precision.

VI. COMPARATIVE ANALYSIS AND PROSPECTS

A. Comparative Analysis

To intuitively demonstrate the performance discrepancies among different technical routes in constructing safety-critical scenarios, this section compares them across four key metrics: realism, adversarial effectiveness, scenario diversity, and response latency. The comparative results are presented in Table IV. The distinction between these technical routes lies in their information sources and constraint mechanisms. Prompt Engineering-based methods rely primarily on the internal parametric knowledge of LLMs. While they achieve high generation efficiency, the lack of external rule constraints makes them prone to physical hallucinations. RAG-based methods enhance scenario realism by reusing historical real-world cases; however, constrained by the scope of the retrieval repository, they struggle to generate OOD scenarios. AI Agent-based methods ensure the physical validity of generated scenarios through real-time closed-loop interactions with simulation tools, but this comes at the cost of significantly increased system complexity and inference latency.

TABLE IV
TECHNICAL CHARACTERISTICS COMPARISON

Metric	Prompt Eng.	RAG	AI Agent
Source	Parametric Memory	External Database	Environment Feedback
Constraint	Soft Constraints	Hard Constraints	Verifiable Constraints
Inference	Open-loop / One-pass	Retrieval + Generation	Closed-loop / Iterative
Search Space	Semantic Space	Database Manifold	Kinematic State Space
Latency	Token Seq. Length	Retrieval Scale	Interaction Turns

regarding physical rationality, Prompt Engineering relies entirely on statistical patterns memorized by the large model from training texts. However, since large models are insensitive to numbers, they frequently violate physical common sense. RAG methods ensure physical realism within fragments by directly searching for real data. AI Agent methods introduce rule constraints by invoking external tools such as mathematical solvers or simulation engines, thereby ensuring

that generated scenarios are strictly feasible at the physical level.

Regarding the effectiveness of safety-critical scenario generation, Prompt Engineering focuses on broad coverage; the dangerous scenarios generated mostly stem from randomness and are difficult to specifically trigger particular defects in the system under test. RAG can retrieve historical scenarios and excels at reproducing known accident forms with high fidelity, but it has limitations in exploring unknown risks. AI Agent methods probe the perception and decision-making boundaries of the vehicle under test through multi-round interaction, significantly improving the pertinence and attack success rate of generated scenarios, albeit with higher computational overhead and inference latency.

B. Prospects

The comparative analysis in Table IV highlights that large models still face numerous challenges in generating safety-critical scenarios, presenting many opportunities for future exploration. A promising direction is "Generative World Models with Embedded Physical Commonsense," shifting focus from low-level parameter output to an intermediate representation with physical constraints. This approach aims to achieve physical fidelity at the semantic planning level, as illustrated in Fig. 6.

In this proposed framework, the LLM is positioned as a high-level "scenario planner," responsible for outputting a "scenario concept blueprint" composed of symbols and relationships. Subsequently, a pre-trained "physics simulation compilation module," which integrates vehicle dynamics and interaction commonsense, is dedicated to transforming this abstract blueprint into physically executable simulation instances. A closed-loop iteration is established between the two components through a "learnable semantic-physics alignment layer": the compilation module verifies the physical feasibility of the blueprint in real-time and provides feedback constraints to the planner. The planner then dynamically revises the blueprint based on this feedback, thereby integrating physical constraints during the semantic planning stage and offering a novel technical pathway for generating high-fidelity safety-critical scenarios.

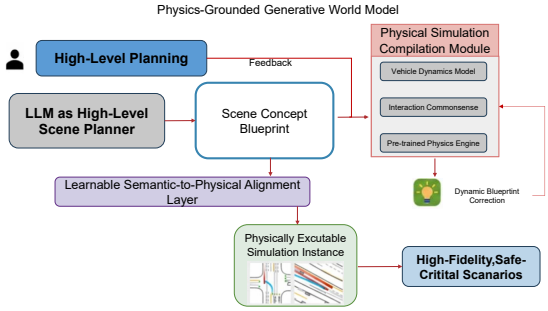

Fig. 6. The proposed technical route of Generative World Models with Embedded Physical Commonsense. The framework consists of a scenario planner, a physics simulation compilation module, and a semantic-physics alignment layer.

VII. Conclusion

This paper surveyed LLM-based autonomous driving safety-critical scenario generation. We clarified core challenges and categorized existing research into Prompt Engineering, RAG, and AI Agent-based methods. Finally, we analyzed existing constraints and proposed future research directions to promote further innovation.

References

[1] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[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] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation research part A: policy and practice*, vol. 94, pp. 182–193, 2016.

[4] 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.

[5] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.

[6] W. Ding, M. Xu, and D. Zhao, "Cmts: A conditional multiple trajectory synthesizer for generating safety-critical driving scenarios," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4314–4321.

[7] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty, and H. Kretzschmar, "Surfelgan: Synthesizing realistic sensor data for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 118–11 127.

[8] W. G. Najm, J. D. Smith, M. Yanagisawa *et al.*, "Pre-crash scenario typology for crash avoidance research," United States. Department of Transportation. National Highway Traffic Safety ..., Tech. Rep., 2007.

[9] D. Bogdoll, J. Breitenstein, F. Heidecker, M. Bieshaar, B. Sick, T. Fingscheidt, and M. Zöllner, "Description of corner cases in automated driving: Goals and challenges," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1023–1028.

[10] H. Sun, S. Feng, X. Yan, and H. X. Liu, "Corner case generation and analysis for safety assessment of autonomous vehicles," *Transportation research record*, vol. 2675, no. 11, pp. 587–600, 2021.

[11] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE intelligent vehicles symposium (IV)*. IEEE, 2018, pp. 1821–1827.

[12] M. Scholtes, L. Westhofen, L. R. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. H. Bollmann, F. Körtke *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.

[13] 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.

[14] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident analysis & prevention*, vol. 35, no. 3, pp. 427–433, 2003.

[15] F. Cunto and F. F. Saccomanno, "Calibration and validation of simulated vehicle safety performance at signalized intersections," *Accident analysis & prevention*, vol. 40, no. 3, pp. 1171–1179, 2008.

[16] M. M. Minderhoud and P. H. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89–97, 2001.

[17] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model

of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[18] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "Trafficgen: Learning to generate diverse and realistic traffic scenarios," *arXiv preprint arXiv:2210.06609*, 2022.

[19] W. Ding, W. Wang, and D. Zhao, "A multi-vehicle trajectories generator to simulate vehicle-to-vehicle encountering scenarios," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4255–4261.

[20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[21] D. Rempe, J. Philion, L. J. Guibas, S. Fidler, and O. Litany, "Generating useful accident-prone driving scenarios via a learned traffic prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 305–17 315.

[22] 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.

[23] 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.

[24] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "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)*. IEEE, 2018, pp. 2118–2125.

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

[26] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *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.

[27] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, J. Beißwenger, P. Luo, A. Geiger, and H. Li, "Drivelm: Driving with graph visual question answering," in *European conference on computer vision*. Springer, 2024, pp. 256–274.

[28] T. Deruyttere, S. Vandenhende, D. Grujicic, L. Van Gool, and M. F. Moens, "Talk2car: Taking control of your self-driving car," in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 2088–2098.

[29] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[30] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *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] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM computing surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[32] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou *et al.*, "The rise and potential of large language model based agents: A survey," *Science China Information Sciences*, vol. 68, no. 2, p. 121101, 2025.

[33] C. Chang, S. Wang, J. Zhang, J. Ge, and L. Li, "Llmscenario: Large language model driven scenario generation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 11, pp. 6581–6594, 2024.

[34] S. Tan, B. Ivanovic, X. Weng, M. Pavone, and P. Kraehenbuehl, "Language conditioned traffic generation," *arXiv preprint arXiv:2307.07947*, 2023.

[35] B.-K. Ruan, H.-T. Tsui, Y.-H. Li, and H.-H. Shuai, "Traffic scene generation from natural language description for autonomous vehicles with large language model," *arXiv preprint arXiv:2409.09575*, 2024.

[36] X. Li, E. Liu, T. Shen, J. Huang, and F.-Y. Wang, "Chatgpt-based scenario engineer: A new framework on scenario generation for trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 3, pp. 4422–4431, 2024.

[37] P. Ji, Y. Feng, Z. Li, X. Zhou, J. Liu, J. Sun, and Z. Zhao, "Txt2sce: Scenario generation for autonomous driving system testing based on textual reports," *arXiv preprint arXiv:2509.02150*, 2025.

[38] S. Tang, Z. Zhang, J. Zhou, L. Lei, Y. Zhou, and Y. Xue, "Legend: A top-down approach to scenario generation of autonomous driving systems assisted by large language models," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 1497–1508.

[39] A. Guo, Y. Zhou, H. Tian, C. Fang, Y. Sun, W. Sun, X. Gao, A. T. Luu, Y. Liu, and Z. Chen, "Sovar: Build generalizable scenarios from accident reports for autonomous driving testing," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 268–280.

[40] A. Aiersilan, "Generating traffic scenarios via in-context learning to learn better motion planner," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 14, 2025, pp. 14 539–14 547.

[41] Y. Gao, M. Piccinini, K. Moller, A. Alanwar, and J. Betz, "From words to collisions: Llm-guided evaluation and adversarial generation of safety-critical driving scenarios," *arXiv preprint arXiv:2502.02145*, 2025.

[42] H. Tian, K. Reddy, Y. Feng, M. Quddus, Y. Demiris, and P. Angeloudis, "Enhancing autonomous vehicle training with language model integration and critical scenario generation," *arXiv preprint arXiv:2404.08570*, 2024.

[43] S. Tan, B. Ivanovic, Y. Chen, B. Li, X. Weng, Y. Cao, P. Krähenbühl, and M. Pavone, "Promptable closed-loop traffic simulation," *arXiv preprint arXiv:2409.05863*, 2024.

[44] Y. Zhao, W. Xiao, T. Mihalj, J. Hu, and A. Eichberger, "Chat2scenario: Scenario extraction from dataset through utilization of large language model," in *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2024, pp. 559–566.

[45] Q. Lu, X. Wang, Y. Jiang, G. Zhao, M. Ma, and S. Feng, "Omnitester: Multimodal large language model driven scenario testing for autonomous vehicles," *Automotive Innovation*, vol. 8, no. 4, pp. 838–852, 2025.

[46] W. Ding, Y. Cao, D. Zhao, C. Xiao, and M. Pavone, "Realgen: Retrieval augmented generation for controllable traffic scenarios," in *European Conference on Computer Vision*. Springer, 2024, pp. 93–110.

[47] Q. Lu, M. Ma, X. Dai, X. Wang, and S. Feng, "Realistic corner case generation for autonomous vehicles with multimodal large language model," *arXiv preprint arXiv:2412.00243*, 2024.

[48] S. Luo, Y. Zhang, Y. Deng, L. Liang, and X. Zheng, "Safe: Harnessing llm for scenario-driven ads testing from multimodal crash data," 2025. [Online]. Available: https://arxiv.org/abs/2502.02025

[49] Y. Mei, T. Nie, J. Sun, and Y. Tian, "Seeking to collide: Online safety-critical scenario generation for autonomous driving with retrieval augmented large language models," *arXiv preprint arXiv:2505.00972*, 2025.

[50] Y. Wei, Z. Wang, Y. Lu, C. Xu, C. Liu, H. Zhao, S. Chen, and Y. Wang, "Editable scene simulation for autonomous driving via collaborative llm-agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 077–15 087.

[51] M. Li, W. Ding, H. Lin, Y. Lyu, Y. Yao, Y. Zhang, and D. Zhao, "Crashagent: Crash scenario generation via multi-modal reasoning," *arXiv preprint arXiv:2505.18341*, 2025.

[52] Y. Mei, T. Nie, J. Sun, and Y. Tian, "Llm-attacker: Enhancing closed-loop adversarial scenario generation for autonomous driving with large language models," *arXiv preprint arXiv:2501.15850*, 2025.

[53] S. Li, T. Azfar, and R. Ke, "Chatsumo: Large language model for automating traffic scenario generation in simulation of urban mobility," *IEEE Transactions on Intelligent Vehicles*, 2024.

[54] E. Aasi, P. Nguyen, S. Sreeram, G. Rosman, S. Karaman, and D. Rus, "Generating out-of-distribution scenarios using language models," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 616–10 623.

[55] Y. Miao, G. Fainekos, B. Hoxha, H. Okamoto, D. Prokhorov, and S. Mitra, "From dashcam videos to driving simulations: Stress testing automated vehicles against rare events," *arXiv preprint arXiv:2411.16027*, 2024.