Robust Scenario Mining Assisted by Multimodal Semantics

Anonymous ICCV submission

Paper ID *****

Abstract

Scenario mining from large-scale autonomous driving datasets, such as Argoverse 2, is crucial for the development and validation of autonomous driving systems. The RefAV framework represents a promising approach by employing Large Language Models (LLMs) to translate natural-language queries into executable code for identifying relevant scenarios. However, the performance of this method is constrained by its reliance on the quality of upstream 3D multi-object tracking data, the absence of a direct linkage between natural-language descriptions and RGB images, runtime errors stemming from LLM-generated code, and inaccuracies in interpreting parameters for functions that describe complex multi-object spatial relationships. To address these issues, we introduce a method that utilizes a CLIP encoder for multimodal semantic similarity filtering, first performing a coarse-grained selection by comparing raw images against the natural-language description, followed by fine-grained mining using an LLM-generated script composed of atomic functions. Additionally, a fault-tolerant iterative code generation mechanism is introduced, which refines code by reprompting the LLM with error feedback, along with specialized prompt engineering to enhance the LLM's comprehension and correct application of spatial-relationship functions. Experiments on Argoverse 2 with various LLMs show that our method achieves consistent improvements across multiple metrics. These results underscore the efficacy of the proposed techniques for reliable, high-precision scenario mining.

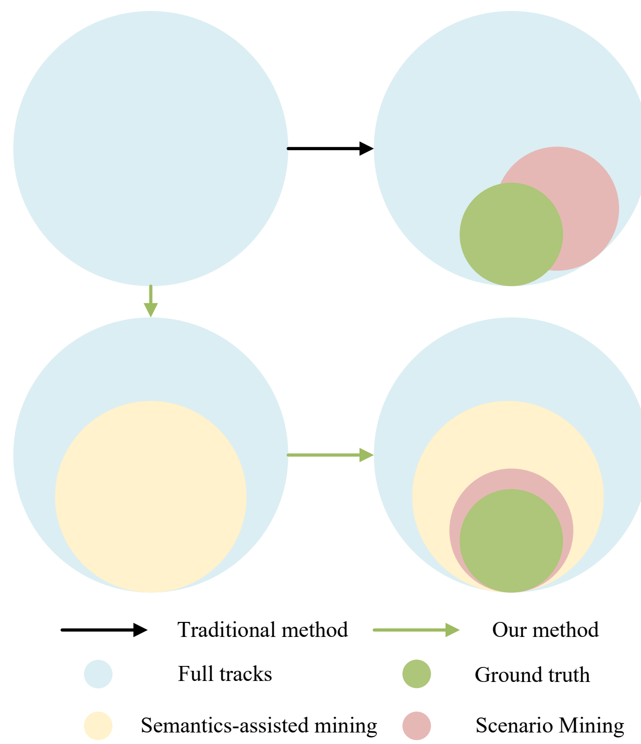

Figure 1. In contrast to traditional scenario-mining pipelines that interrogate the entire collection of 3D tracks—risking substantial drift from the intended query when trajectories share confounding similarities in colour, object class, or event labels—our multimodal, semantics-assisted method first subjects the raw RGB imagery to a semantic filter, isolates a candidate subset of 3D tracks, and only then executes the natural-language query within this reduced search space, which markedly enhances retrieval precision and curtails computational overhead.

1. Introduction

The deployment of Autonomous Vehicles (AVs) necessitates rigorous testing and validation, for which the identification of interesting, rare, or safety-critical scenarios from vast operational data is paramount. This process is vital not only for evaluating ego-behavior and safety testing but also for enabling active learning at scale [12]. Traditional methods relying on manual inspection or predefined heuristics are often prohibitively time-consuming and prone to errors when faced with the terabytes of multimodal data collected by AV fleets [29]. Previous methods that used database queries for scenario mining lacked flexibility compared to methods based on LLMs [10, 13, 16]. The sheer volume and complexity of this data pose a major challenge, making efficient and accurate scenario mining a major ongoing challenge.

The RefAV [6, 30] framework is a method for retrieving specific scenarios from sensor data via natural language queries, leveraging the powerful zero-shot capabilities of Large Language Models. RefAV translates natural-language descriptions of scenarios into composable function calls, which are then executed to identify relevant events within driving logs. This approach offers flexibility and expressiveness beyond structured query languages.

Despite the promise of LLM-based scenario mining, practical implementations like RefAV encounter specific limitations. First, the method hand-crafts 28 atomic functions that detect trajectory states, articulate relations between a focal object and its surrounding entities, and implement basic Boolean logic; an LLM then composes these atoms into scripts that operate directly on the 3D tracks. As a result, mining accuracy is tied to the quality of the upstream data, since the preceding 3D object-detection and tracking modules determine the object labels—and therefore whether the trajectories themselves are computed correctly. Consequently, poor performance in the upstream 3D multi-object tracking directly degrades the performance of scenario mining. Furthermore, this approach contravenes the conventional intuition of video retrieval by neglecting the association between the raw image and the natural-language description. Secondly, code generated directly by LLMs can frequently contain syntactic or logical errors, leading to runtime failures. These failures disrupt the mining pipeline and result in incomplete scenario discovery. LLMs may also struggle with the nuanced semantics of functions describing relative spatial relationships between multiple objects. For instance, functions such as *has objects in relative direction()* or *facing toward()* require precise parameter assignment to reflect the intended meaning (e.g., distinguishing "a car in front of a pedestrian" from "a pedestrian in front of a car"). Misinterpretation of these parameters leads to semantic inaccuracies in the retrieved scenarios, even if the code executes without error. This issue is a manifestation of a known failure mode in LLMs, often termed 'factual hallucination' or a breakdown in understanding relational knowledge [17, 23]. These represent fundamental hurdles in reliably converting complex human language into precise and correct machine-executable instructions.

In light of the aforementioned limitations, we propose a robust, multimodally-aware scenario mining methodology that enhances the RefAV framework. Our approach introduces a dual-branch architecture comprising an image-semantic branch and a text-semantic branch. The image-semantic branch employs the YOLOv8 model for object detection on raw RGB frames. Subsequently, a pre-trained CLIP image encoder [22] is utilized to extract offline feature embeddings for each detected object. Concurrently, the text-semantic branch processes the input natural-language query using the spaCy [14] NLP toolkit to perform keyword extraction. This process isolates critical terms, including colors, nouns, and spatial prepositions, which are then encoded into offline feature embeddings using a CLIP text encoder. During inference, a coarse-grained filtering stage is executed by computing the cosine similarity between the keyword features and the object features across all frames within a complete log. Based on a Top-K selection, the tracklets corresponding to frames with the highest similarity scores are shortlisted. Simultaneously, leveraging the Fault-Tolerant Iterative Code Generation mechanism and spatially-aware prompting, the LLM generates an executable script from the natural-language description. This script then performs a fine-grained search directly on the shortlisted tracklets—subsets of the overall 3D tracks—to precisely identify the target scenario.

In summary, our main contributions are the following:

- We propose a multimodal semantics enhancement to the RefAV methodology. This method addresses a critical deficiency in the original pipeline by establishing a direct association between RGB images and natural language descriptions.
- We introduce the Fault-Tolerant Iterative Code Generation (FT-ICG) mechanism, specifically designed for the paradigm of using Large Language Models to compose atomic functions. This contribution significantly enhances the robustness of the method.
- We propose the integration of enhanced prompting for spatial relationship functions. This technique mitigates the propensity of the LLM to misinterpret parameters for atomic functions that describe complex spatial relationships.

2. Related Works

2.1. Scenario mining

The safety and reliability of autonomous driving (AD) are of paramount importance, necessitating rigorous testing and validation protocols before deployment. While real-world road testing is indispensable, it is prohibitively expensive, time-consuming, and fails to provide sufficient coverage of rare but critical "edge cases." Consequently, simulation-based testing has emerged as an essential component of the verification and validation pipeline. A core challenge in this paradigm is the generation of a comprehensive and challenging suite of test scenarios. This has given rise to the field of Scenario Mining, which focuses on systematically creating diverse, critical, and realistic driving scenarios to test AD in simulation efficiently. [9, 29] The primary goal is to pinpoint trajectory snippets, within the set of annotated scenes, that satisfy the given natural-language description.

Early efforts in scenario mining leveraged explicit human knowledge. These methods encode traffic laws, domain

ICCV #*****

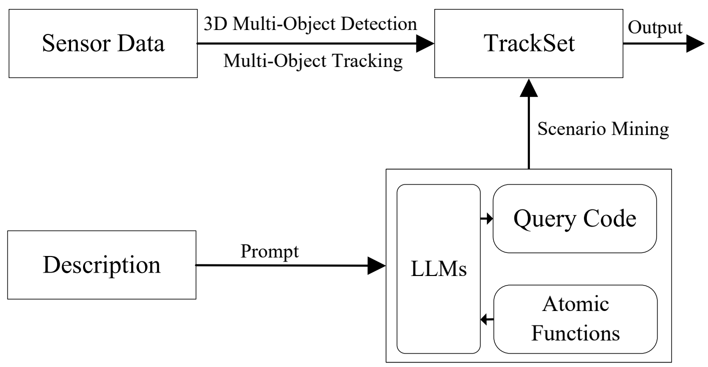

Figure 2. Overview of the RefAV framework: RefAV harnesses large language models to transmute natural-language descriptions into executable, structured code and then runs these scripts automatically to mine the dataset. The trajectory labels consumed by the queries originate from upstream 3D multi-object detection and tracking modules, while handcrafted prompts and a library of twenty-eight atomic functions—each expressing a particular object state—steer the LLM to generate the final retrieval program.

expertise, and parameters from accident databases into formal languages and ontologies. By defining a logical scenario space with parameters (e.g., road curvature, number of vehicles, weather) and their valid ranges, scenarios can be generated through techniques like combinatorial testing to cover a wide array of predefined conditions. A significant initiative in this area is the PEGASUS project [20], which established a systematic, knowledge-driven workflow for defining scenarios. Formal languages such as ASAM OpenSCENARIO [1] have become industry standards for describing the dynamic content of driving scenarios. The primary advantage of knowledge-based methods is the high degree of control and interpretability, making them ideal for testing system compliance with known rules. However, their main limitation is that they are bound by existing knowledge and manual effort, often failing to uncover novel modes and lacking the behavioral complexity of real-world traffic.

With the advent of large-scale, real-world driving datasets, data-driven methods have become prominent. These approaches mine vast logs of sensor data to extract realistic scenarios or learn generative models of traffic behavior. The typical pipeline involves data acquisition from datasets like the Waymo Open Motion Dataset [24], nuScenes [4] or Argoverse2 [30], using scenario identification by mining the data for events that exceed a certain criticality threshold. By articulating the task in a native database query language, a bespoke domain-specific language (DSL), or a general-purpose programming language, the problem is recast as one of label retrieval. Erwin de Gelder et al. [10] present a label-based scenario-mining system for autonomous driving that operates on datasets pre-annotated either automatically or by hand. Although the labels are applied in a semi-manual fashion, they remain coarse-grained; as a result, the framework is rigid and scales poorly—supporting richer, more nuanced scenes would require an unwieldy proliferation of tags. Motional's scenario mining pipeline [19] adopts a continual-learning paradigm: the system cyclically discovers scenes, annotates them (both manually and automatically), retrains its models with the expanded data, and then performs automatic evaluation. Acting as the data-sourcing engine, it maintains a tag vocabulary whose compositions encode basic spatio-temporal relations between the ego vehicle and surrounding traffic participants. Given 3D trajectories, the ego path, and the HD map, it automatically labels both the ego and other actors, stores the tags in a relational database, and exploits SQL for efficient retrieval. The strength of data-driven methods lies in their ability to produce highly realistic scenarios grounded in real-world behavior. The coverage of the source data is inherently limiting their primary drawback; discovering truly novel edge cases remains a "needle-in-a-haystack" problem, and the generated scenarios are often descriptive rather than actively challenging.

To overcome the limitations of passive methods, the most recent trend in scenario mining involves the application of large foundation models [28]. As outlined in the comprehensive survey by Gao et al. [8], this new paradigm leverages the power of Large Language Models (LLMs), Vision-Language Models (VLMs), and Diffusion Models to generate scenarios from high-level, often semantic, inputs. For instance, a user can provide a natural language prompt like, "Create a challenging scenario where a truck illegally overtakes a bicycle on a rainy night," and the model generates the corresponding scene parameters for the simulator.

Works such as ChatScene [34] have demonstrated the ability of LLMs to understand complex spatial and behavioral relationships to produce diverse and contextually rich scenarios. This approach holds immense promise for bridging the gap between abstract human knowledge and concrete simulation data. While still an emerging area, the key challenges include ensuring the physical plausibility and controllability of generated scenarios and managing the significant computational resources required by these large models. RefAV [6] is a large-scale scenario-mining framework that houses 10,000 distinct natural-language queries describing the complex multi-agent interactions present in the 1,000 driving logs of the Argoverse 2 sensor suite. It exposes 28 handcrafted atomic functions capable of recognizing trajectory states, expressing relational predicates between a target agent and its surrounding entities, and supporting basic Boolean logic. At its core, the framework feeds the natural-language query, the atomic function inventory, and carefully engineered prompts into an LLM, which synthesizes an executable script com-

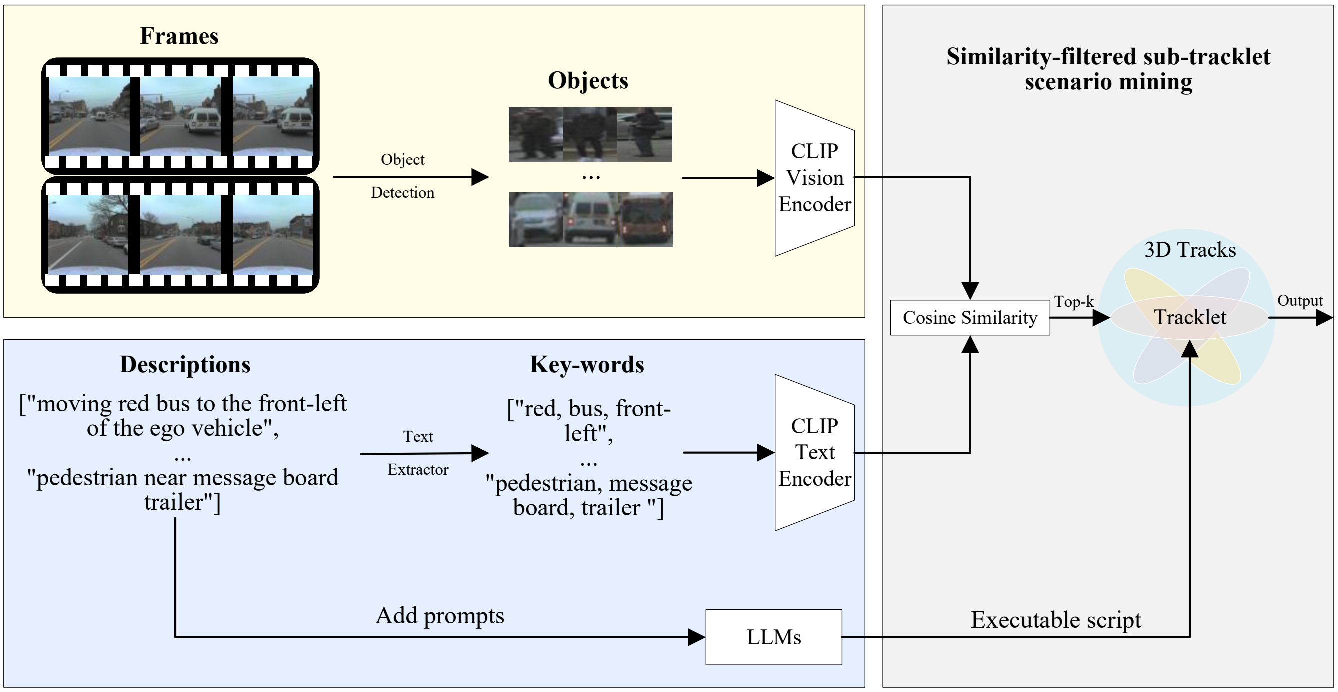

Figure 3. Framework of the multimodal semantics assisted scenario mining.

posed of these atoms; running the script over the dataset then retrieves the trajectories that satisfy the query, as shown in Figure 2. Our work builds on RefAV to deliver robust scenario mining assisted by multimodal semantics.

In conclusion, the field of scenario mining is evolving from static, knowledge-driven methods towards more dynamic and intelligent approaches. Current research increasingly focuses on hybrid methods, such as using data-driven models to create a realistic basis for subsequent scenario mining. Key future directions include richer multimodal annotations, more faithful modeling of complex multi-agent interactions, and greater explainability of the critical scenarios that are retrieved.

2.2. Video retrieval

Text–video retrieval is a task closely aligned with, and analogous to, scenario mining: both seek to locate semantically coherent segments within large-scale, long-sequence data streams based on natural-language descriptions. Multimodal fusion–based approaches constitute the most prevalent and emblematic paradigm within the text–video retrieval literature. We may glean valuable insights for scenario mining by studying advances in video retrieval. Since 2020, research on text–video retrieval has advanced through successive innovations in cross-modal alignment and temporal modeling: in 2020, Gabeur et al. proposed a multi-modal Transformer that jointly encodes visual modalities and explicitly models temporal dependencies via cross-modal attention, optimizing language embeddings together with video features [7]; in 2021, Wang et al. released T2VLAD, which introduces shared semantic centers to perform computationally efficient global-local alignment for fine-grained comparison [27]; Gorti et al. presented X-Pool, enabling text to attend selectively to semantically relevant frames and thereby filtering visual noise for improved accuracy [11]; in 2023, Wu et al. developed Cap4Video, leveraging zero-shot video-generated captions for data augmentation, cross-modal interaction, and an auxiliary inference branch, pushing performance on multiple benchmarks [31]; in 2024, Wang et al. introduced T-MASS, a stochastic text-embedding strategy that treats queries as deformable semantic masses, employing a similarity-aware radius and support-text regularization to boost expressiveness and set new records on five datasets [26]; entering 2025, Zhang et al. proposed TokenBinder, a two-stage framework adopting a one-to-many coarse-to-fine alignment paradigm inspired by comparative judgment and equipped with a Focused-view Fusion Network for cross-attention, achieving state-of-the-art results across six benchmarks [33], while Bian et al. introduced the SMA framework, which performs selective multi-grained alignment at both video-sentence and object-phrase levels with token aggregation and similarity-aware keyframe selection, attaining strong performance on MSR-VTT, ActivityNet, and beyond [3].

The motivation driving these video-retrieval tasks is akin to that of scenario mining: both seek to locate a contiguous scene that matches a natural-language description. Scenario mining, however, often requires finer-grained retrieval targeting the trajectory of a specific agent or a set of interacting agents. In almost all video-retrieval work, retrieval is achieved by directly aligning or contrasting visual and textual semantics, which constitutes a latent opportunity for scenario mining: cross-modal semantic matching can be leveraged for coarse retrieval, after which detailed scenario mining can be confined to the trajectory subset thus obtained.

3. Methods

This section first details the multimodal semantics assisted scenario mining pipeline, and then presents the work's two further robustness-oriented contributions: a Fault-Tolerant Iterative Code Generation (FT-ICG) mechanism and enhanced prompting for spatial-relation functions.

3.1. CLIP-based Multimodal Semantic Filter

Within the original RefAV pipeline, scenario mining is performed exclusively via scripts assembled from atomic functions, a design that overlooks the direct correspondence between natural-language queries and raw image frames. In RefAV, the pipeline begins with 3D object detection to extract each target's class, heading, velocity, and related attributes, assigns corresponding labels, and then waits for an LLM-generated script to query them. The correctness of those attributes rests wholly on the upstream detector's performance, and the elongated processing chain renders the connection between natural-language queries and raw sensor data highly indirect—raising the likelihood that crucial information will be missed or inaccurately captured. Inspired by advances in text–video retrieval, we enhance RefAV with a coarse-to-fine filtering stage: natural-language descriptions first delimit tracklets that are likely to contain the target situation, and LLM-generated code then probes only those candidates. This hierarchical procedure sharply reduces false positives and scene ambiguities, yielding results that hew more closely to ground truth.

Specifically, on the visual side, we enumerate every object visible in the nine synchronized camera views and encode each one offline with a pretrained CLIP image encoder, so that every embedding captures a localized slice of the frame at its timestamp. On the language side we employ spaCy—an industrial-grade NLP library—to extract colour, entity, and spatial-relation words from each query; these discrete keywords succinctly convey the sentence semantics and, when embedded by the CLIP text encoder, have been shown by Xie et al [32]. to be more discriminative than full-sentence encodings. We rank frames by cosine similarity to the keyword embeddings, retain the top-k matches, and record their timestamps to assemble candidate tracklets—subsets of the full 3D trajectories. This design enables raw RGB imagery to be compared directly against the textual description, thereby discarding tracklets whose semantics deviate markedly from the query; confining the LLM-generated scripts to mine only within these semantically aligned candidates greatly diminishes the risk of false positives. Executing the original RefAV scripts on this pruned search space markedly improves the HOTA-T metric while reducing inference consumption.

3.2. Fault-Tolerant Iterative Code Generation

A significant challenge in the practical application of LLMs for code generation is the propensity for the generated code to contain errors. These errors can range from simple syntax mistakes to more complex logical flaws or incorrect usage of the provided atomic functions, all of which lead to runtime exceptions. Such failures can terminate the scenario mining process prematurely, resulting in missed scenarios and reduced overall system reliability. The pseudocode for the fault-tolerant iterative code generation mechanism is shown in Algorithm 1. Algorithm 1 proceeds as follows: first, the natural-language scenario query is concatenated with a description of the available atomic-function library to form an initial prompt, giving the LLM full context about which functions are permissible and how they behave so that its

Algorithm 1: Fault-Tolerant Iterative Code Generation

Input: Natural-language query $NLQuery$; set of atomic functions $\mathcal{A}$; maximum iterations K
Output: Executable Python code $ValidCode$

$Prompt \leftarrow$ $\textsc{Compose}(NLQuery,\ \textsc{Describe}(\mathcal{A}))$;
for $i \leftarrow 1$ **to** K **do**
  try
    $Code \leftarrow \textsc{LLMGenerate}(Prompt)$;
    $\textsc{PythonExec}(Code)$;
    Break;
  catch (**RuntimeError** ε)
    $ErrorMsg \leftarrow \textsc{Message}(\varepsilon)$;
    $IterationPrompt \leftarrow$ `"This is the code generated last time:` $\{Code\}$`, with the error message:` $\{ErrorMsg\}$`. Please avoid code runtime errors.";`
    $Prompt \leftarrow$ $\textsc{Compose}(NLQuery,\ IterationPrompt)$;

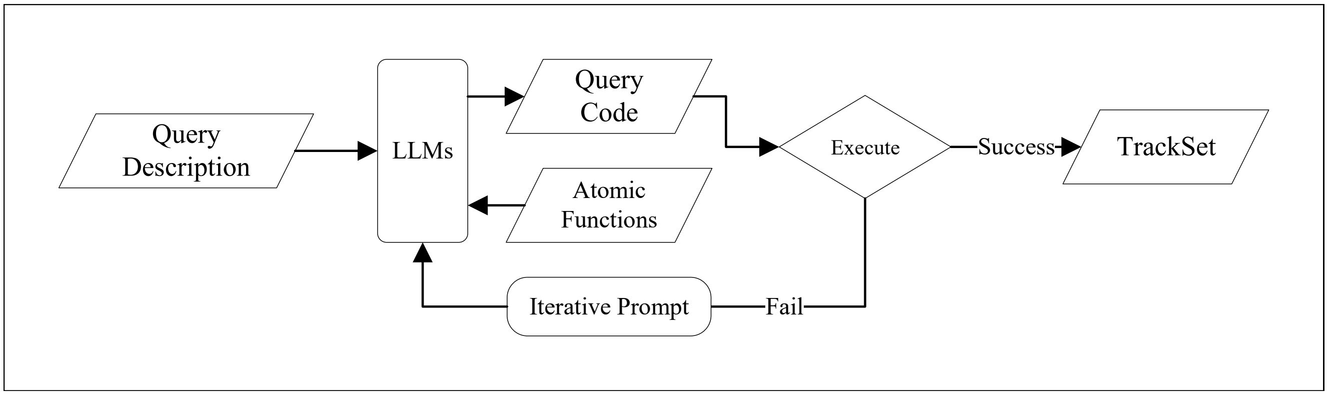

Figure 4. Framework of the Fault-Tolerant Iterative Code-Generation mechanism: whenever the executable script raises a runtime exception, the error trace is fed back to the LLM, which regenerates a revised script; this loop repeats for a preset number of iterations, thereby substantially reducing mining failures caused by execution errors.

first attempt is well informed. The model then generates a Python snippet that chains the atomic functions to implement the query logic and retrieve the desired scenarios. This code is executed immediately in a sandboxed environment; if it runs without error, the resulting track set is accepted and the pipeline terminates successfully. If, however, execution raises an exception—such as a 'NameError' for an undefined variable, a 'TypeError' due to incorrect argument counts, or any other syntactic or logical fault—the error is caught and its message recorded. The system then feeds both the faulty code and the accompanying error message back to the LLM in a new prompt that explicitly instructs the model to correct the identified problem. Armed with this feedback, the LLM produces a revised snippet, which is executed and validated again. This refinement loop continues, with the model iteratively "learning" from each failure, until the code executes cleanly or a maximum of K iterations is reached. In our implementation $K = 5$; if a runnable solution still has not been produced after five attempts, the query is flagged for manual review. This upper bound prevents pathological infinite-loop behavior while still allowing most errors to be resolved through a handful of feedback cycles. This iterative approach treats the LLM not as a single-shot code generator but as an entity capable of learning from explicit feedback on its errors. By providing the context of the previous failure, the LLM is guided towards a correct solution. This significantly increases the success rate of code generation, thereby enhancing the robustness and coverage of the scenario mining pipeline, allowing it to handle a broader spectrum of queries and code complexities without manual intervention. This process mirrors a human programmer's debugging cycle, iteratively refining code based on observed errors.

3.3. Enhanced Prompting for Spatial Relational Functions

Beyond syntactic correctness, the semantic accuracy of the generated code is paramount. LLMs often fail to correctly interpret and parameterize functions that describe the relative spatial relationships between multiple objects in a specific domain. For example, a query like "a cyclist to the left of a bus" requires the LLM to correctly assign the 'cyclist' and 'bus' tracks to the appropriate parameters of a function like has *objects in relative direction()*. An incorrect assignment could lead the system to search for "a bus to the left of a cyclist," fundamentally misinterpreting the query. A more comprehensive prompt-engineering strategy can effectively suppress ambiguities and hallucinations in large models. [5] To mitigate such semantic errors, Enhanced Prompting for Spatial Relational Functions is introduced. This involves augmenting the initial prompt provided to the LLM with specific instructions that clarify the argument semantics for these critical functions. Before the LLM attempts to generate code involving functions that define relative positions or orientations, it receives the following guiding information:

If you use has objects in relative direction(), being crossed by(), heading in relative direction to() *functions, direction parameter specifies the orientation of related candidates relative to track candidates. The* facing toward() *and* heading toward() *functions indicate that the track candidates parameter is oriented toward the related candidates parameter.*

This explicit instruction serves as a form of contextual disambiguation. It clearly defines the roles of *track candidates* (often the primary subject of the relation) and *related candidates* (the reference object) within the context of each specified function. For directional functions like *has objects in relative direction*, it clarifies which entity's perspective defines the direction. For orientational functions like *facing toward*, it specifies which entity is performing the action of

ICCV #*****

facing. By providing this upfront clarification, the LLM is better equipped to map the natural language description of spatial relationships to the correct functional representation and parameter assignment. This leads to a higher fidelity in translating complex spatial queries, ultimately improving the semantic accuracy and relevance of the mined scenarios. This addresses the challenge that code might run correctly but perform the wrong semantic operation if the LLM misunderstands these subtle but critical distinctions.

4. Experiments

4.1. Implementation Details

The experiments were conducted using the Argoverse 2 dataset. The dataset provides rich multi-modal information, including RGB camera frames, LiDAR point clouds, HD Maps, and 3D track annotations for 26 object categories.

The primary metric is HOTA-Temporal. It is a spatial tracking metric that considers only the scenario-relevant objects during the precise timeframe when the scenario is occurring. HOTA[18] was introduced to provide a unified evaluation of multi-object tracking by jointly accounting for detection, association, and localization—three facets that together reflect human intuition of tracking quality. Secondary metrics include HOTA, Timestamp F1, and Log F1. Timestamp F1 treats the video as a sequence of frames, labeling each timestamp as "scenario" or "non-scenario." Precision and recall are computed from the comparison of predicted and ground-truth frame labels. Log F1 simplifies the task to a single binary decision per log. After aggregating true positives, false positives, and false negatives across all logs, a conventional F1-score is produced.

In our setup, we adopt the pretrained ViT-B/32 size of CLIP—both its image and text encoders—as the backbone of the multimodal semantic filter. We employ YOLOv8-l [15] as the object detector. The Qwen2.5-VL-7B model [2] was deployed locally on a workstation outfitted with an NVIDIA RTX 4090 GPU, whereas the Gemini model [25] was accessed remotely via API calls. For 3D object detection and tracking, we utilized the track obtained directly from the LT3D method [21]. We set K in Algorithm 1 to 5. For the generated code, if the number of iterations of the fault tolerance mechanism exceeds the K value, we manually edit the generated code, manually modify the reported errors, and fill in the correct track candidates, related candidates, and direction parameters. Method evaluation is conducted on the validation set.

4.2. Comparative Experiments

As Table 1-3 demonstrate, our method outperforms the baseline under both upstream 3D tracking pipelines—Le3DE2E and TransFusion. We report results with three distinct LLMs acting as code generators, and the largest performance gain arises when Qwen2.5-VL-7B is used. This is likely because Qwen2.5-VL-7B, relative to Gemini Flash and Gemini Pro, exhibits a weaker innate understanding of spatial relations and atomic functions; our pipeline compensates for this shortcoming. The semantics-assisted filtering stage confines Qwen's search to a much smaller candidate subset, the FT-ICG loop produces more robust and executable code, and the spatially informed prompts help Qwen correctly interpret and invoke the atomic functions. Collectively, these elements drive the substantial improvement over the original RefAV baseline when Qwen serves as the LLM. When Gemini Flash and Gemini Pro are used as the LLM, our method likewise outperforms the RefAV baseline—an advantage attributable to its coarse-to-fine mining cascade, the greater robustness of the generated code, and a deeper, more accurate handling of the atomic-function semantics.

As reported in Table 4, we benchmark the end-to-end inference time for generating and executing a single query under RefAV and under our framework. Our pipeline delivers a notable speed-up, attributable to its coarse-to-fine mining strategy: CLIP features for all images and query tokens are

3D Track	Method	HOTA-T	HOTA	TS-F1	Log-F1
Le3DE2E	RefAV*	33.27	36.72	61.94	58.12
	Our method	44.54	44.71	70.37	71.47
TransFusion	RefAV*	30.06	31.27	59.96	59.31
	Our method	44.11	44.67	69.44	68.66

Table 1. With Qwen2.5-VL-7B as the LLM, comparison of our method and baseline across two distinct 3D tracking pipelines—Le3DE2E and TransFusion. * represents the baseline reproduced in our implementation.

3D Track	Method	HOTA-T	HOTA	TS-F1	Log-F1
Le3DE2E	RefAV*	40.17	40.33	66.70	62.71
	Our method	48.30	49.52	72.30	73.41
TransFusion	RefAV*	35.50	35.93	59.89	59.13
	Our method	47.07	47.19	69.79	70.93

Table 2. With Gemini 2.5 Flash as the LLM, comparison of our method and baseline across two distinct 3D tracking pipelines—Le3DE2E and TransFusion.

3D Track	Method	HOTA-T	HOTA	TS-F1	Log-F1
Le3DE2E	RefAV*	42.73	44.27	69.84	66.13
	Our method	52.10	51.07	74.21	70.45
TransFusion	RefAV*	38.76	39.22	60.36	60.31
	Our method	47.37	47.79	69.73	71.66

Table 3. With Gemini 2.5 Pro as the LLM, comparison of our method and baseline across two distinct 3D tracking pipelines—Le3DE2E and TransFusion.

pre-extracted offline, making their embeddings immediately available at inference time. The system therefore performs only a lightweight cosine-similarity check to complete the coarse retrieval stage, swiftly pruning trajectories that are semantically irrelevant; by sparing the subsequent fine-grained miner from comparing against obviously incorrect tracks, the overall runtime is substantially reduced.

LLMs	Method	Time(s)
Qwen2.5-VL-7B	RefAV*	47.3
	Our method	19.4
Gemini Flash	RefAV*	48.9
	Our method	17.7
Gemini Pro	RefAV*	42.7
	Our method	18.7

Table 4. A comparison of the inference time required by RefAV and our approach to generate and execute a single query.

MSA	FT-ICG	EP-SRF	HOTA-T	HOTA	TS-F1	Log-F1
✓			41.06	39.98	67.31	67.04
	✓		34.71	39.32	62.77	58.09
		✓	35.37	39.21	62.93	60.11
✓	✓		41.12	40.38	66.68	68.45
✓		✓	43.27	44.13	70.12	69.93
✓	✓	✓	44.54	44.71	70.37	71.47

Table 5. With Qwen2.5-VL-7B as the LLM, performance comparison under different configurations.

4.3. Ablation study

To demonstrate the broad performance gains delivered by our three contributions to scenario mining, we perform ablation studies under three distinct LLM configurations. In the table, the multimodal semantics–assisted filter is denoted **MSA**, Fault-Tolerant Iterative Code Generation appears as **FT-ICG**, and Enhanced Prompting for Spatial-Relational Functions is labeled **EP-SRF**; all results are reported using the Le3DE2E 3D tracker.

Across all three LLMs, the multimodal semantic filter (MSA) raises performance consistently—most conspicuously on the TS-F1 metric—demonstrating that the CLIP-based filter effectively selects image frames whose content aligns with the query keywords, thereby producing more accurate timestamps and tracklets. Restricting scenario mining to these subsets of the full 3D tracks improves multi-agent retrieval precision and minimizes false positives. The FT-ICG mechanism likewise yields uniform gains, particularly in HOTA-T, underscoring the practical benefit of resolving runtime code errors: each iterative refinement produces scripts with higher correctness and executability, which in

MSA	FT-ICG	EP-SRF	HOTA-T	HOTA	TS-F1	Log-F1
✓			44.34	45.98	70.97	66.45
	✓		44.13	45.07	70.44	60.66
		✓	41.77	40.93	69.73	60.95
✓	✓		44.20	46.37	72.01	67.17
✓		✓	45.97	46.63	71.60	69.31
✓	✓	✓	48.30	49.52	72.30	73.41

Table 6. With Gemini Flash as the LLM, performance comparison under different configurations.

MSA	FT-ICG	EP-SRF	HOTA-T	HOTA	TS-F1	Log-F1
✓			47.28	47.97	67.70	66.95
	✓		44.10	46.37	65.90	60.32
		✓	43.74	45.62	69.90	59.13
✓	✓		51.98	51.10	74.05	69.98
✓		✓	50.56	49.47	71.30	70.01
✓	✓	✓	52.10	51.07	74.21	70.45

Table 7. With Gemini Pro as the LLM, performance comparison under different configurations.

turn lifts the HOTA-T score. Subsequent incorporation of EP-SRF provides additional enhancements—most notably in HOTA-Temporal, Timestamp-F1, and Log-F1—highlighting the critical role of semantically precise parameterization of spatial-relation functions and revealing untapped LLM potential that can be unlocked through better prompt engineering. The consistency of these improvements across different LLM backbones indicates that our approach tackles fundamental challenges in LLM-driven code generation and interpretation rather than exploiting model-specific quirks.

5. Conclusion

In this paper we presented a robust, multimodal scenario-mining framework that augments the RefAV pipeline with CLIP-based semantic filtering, a fault-tolerant iterative code-generation loop, and relation-explicit prompt engineering; through a coarse-to-fine retrieval strategy that first constrains the search space via image–text similarity and then refines results with LLM-composed atomic-function scripts, the proposed method simultaneously mitigates error propagation from upstream tracking, suppresses LLM runtime failures, and corrects common mis-parameterizations of spatial-relation functions. Comprehensive experiments on the Argoverse 2 benchmark—covering two distinct 3D tracking backbones and three heterogeneous LLMs—demonstrate consistent gains across all evaluation metrics. These results confirm that tightly coupling vision–language alignment with error-aware code synthesis delivers substantial practical benefits for large-scale autonomous-driving data mining, and they suggest a clear path toward even richer multimodal integration and adaptive, self-refining prompting in future work.

References

[1] ASAM e.V. ASAM OpenSCENARIO XML, 2022. Online; accessed 13 Jul 2025. 3

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. 7

[3] Ziyi Bian, Cong Jiang, Fangzhi Zhu, and Zheng Zhang. Selective multi-grained alignment for text-video retrieval. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 873–877, 2025. 5

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020. 3

[5] Lucas Choi and Ross Greer. Beyond general prompts: Automated prompt refinement using contrastive class alignment scores for disambiguating objects in vision-language models, 2025. 6

[6] Cainan Davidson, Deva Ramanan, and Neehar Peri. Refav: Towards planning-centric scenario mining. *arXiv preprint arXiv:2505.20981*, 2025. 2, 3

[7] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, pages 214–229. Springer, 2020. 4

[8] Yuan Gao, Mattia Piccinini, Yuchen Zhang, Dingrui Wang, Korbinian Moller, Roberto Brusnicki, Baha Zarrouki, Alessio Gambi, Jan Frederik Totz, Kai Storms, Steven Peters, Andrea Stocco, Bassam Alrifaee, Marco Pavone, and Johannes Betz. Foundation models in autonomous driving: A survey on scenario generation and scenario analysis, 2025. 3

[9] Erwin de Gelder, Jeroen Manders, Corrado Grappiolo, Jan-Pieter Paardekooper, Olaf Op den Camp, and Bart De Schutter. Real-world scenario mining for the assessment of automated vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8, 2020. 2

[10] Erwin de Gelder, Jeroen Manders, Corrado Grappiolo, Jan-Pieter Paardekooper, Olaf Op den Camp, and Bart De Schutter. Real-world scenario mining for the assessment of automated vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, page 1–8. IEEE, 2020. 1, 3

[11] Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. X-pool: Cross-modal language-video attention for text-video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5006–5015, 2022. 4

[12] Ross Greer, Bjørk Antoniussen, Andreas Møgelmose, and Mohan Trivedi. Language-driven active learning for diverse open-set 3d object detection. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 980–988, 2025. 1

[13] Elmar Haussmann, Michele Fenzi, Kashyap Chitta, Jan Ivanecky, Hanson Xu, Donna Roy, Akshita Mittel, Nicolas Koumchatzky, Clement Farabet, and Jose M Alvarez. Scalable active learning for object detection. In *2020 IEEE intelligent vehicles symposium (iv)*, pages 1430–1435. IEEE, 2020. 1

[14] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. 2

[15] Muhammad Hussain. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, 11(7):677, 2023. 7

[16] Aryan Keskar, Srinivasa Perisetla, and Ross Greer. Evaluating multimodal vision-language model prompting strategies for visual question answering in road scene understanding. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1027–1036, 2025. 1

[17] Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand, 2024. Association for Computational Linguistics. 2

[18] Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2):548–578, 2020. 7

[19] Motional. Technically speaking: Mining for scenarios to help better train our avs, 2022. Accessed 14 Jul 2025. 3

[20] PEGASUS Project. Scenario Description: Requirements & Conditions. In *Proc. PEGASUS Symposium*, Aachen, Germany, 2017. Online; accessed 13 Jul 2025. 3

[21] Neehar Peri, Achal Dave, Deva Ramanan, and Shu Kong. Towards long-tailed 3d detection, 2023. 7

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[23] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models, 2023. 2

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2020. 3

[25] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser,

Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, and Angeliki Lazaridou. Gemini: A family of highly capable multimodal models, 2025. 7

[26] Jiamian Wang, Guohao Sun, Pichao Wang, Dongfang Liu, Sohail Dianat, Majid Rabbani, Raghuveer Rao, and Zhiqiang Tao. Text is mass: Modeling as stochastic embedding for text-video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16551–16560, 2024. 4

[27] Xiaohan Wang, Linchao Zhu, and Yi Yang. T2vlad: global-local sequence alignment for text-video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5079–5088, 2021. 4

[28] Yuping Wang, Shuo Xing, Cui Can, Renjie Li, Hongyuan Hua, Kexin Tian, Zhaobin Mo, Xiangbo Gao, Keshu Wu, Sulong Zhou, Hengxu You, Juntong Peng, Junge Zhang, Zehao Wang, Rui Song, Mingxuan Yan, Walter Zimmer, Xingcheng Zhou, Peiran Li, Zhaohan Lu, Chia-Ju Chen, Yue Huang, Ryan A. Rossi, Lichao Sun, Hongkai Yu, Zhiwen Fan, Frank Hao Yang, Yuhao Kang, Ross Greer, Chenxi Liu, Eun Hak Lee, Xuan Di, Xinyue Ye, Liu Ren, Alois Knoll, Xiaopeng Li, Shuiwang Ji, Masayoshi Tomizuka, Marco Pavone, Tianbao Yang, Jing Du, Ming-Hsuan Yang, Hua Wei, Ziran Wang, Yang Zhou, Jiachen Li, and Zhengzhong Tu. Generative ai for autonomous driving: Frontiers and opportunities, 2025. 3

[29] Hiroki Watanabe, Lukas Tobisch, Julia Rost, Johannes Wallner, and Günther Prokop. Scenario mining for development of predictive safety functions. In *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–7, 2019. 1, 2

[30] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting, 2023. 2, 3

[31] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10704–10713, 2023. 4

[32] Dong Xie, Linhu Liu, Shengjun Zhang, and Jiang Tian. A unified multi-modal structure for retrieving tracked vehicles through natural language descriptions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5419–5427, 2023. 5

[33] Bingqing Zhang, Zhuo Cao, Heming Du, Xin Yu, Xue Li, Jiajun Liu, and Sen Wang. Tokenbinder: Text-video retrieval with one-to-many alignment paradigm. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4957–4967. IEEE, 2025. 4

[34] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles, 2024. 3