

# AutoMine Solution for AV2 2026 Scenario Mining Challenge

Songliang Cao<sup>1,2</sup> \* Jiele Zhao<sup>1</sup> \* Yuru Wang<sup>1</sup> Hao Li<sup>1</sup> Daqi Liu<sup>1</sup> Zehan Zhang<sup>1</sup> †  
Fangzhen Li<sup>1</sup> † Yu Wang Yue Zhang Bing Wang<sup>1</sup> Guang Chen<sup>1</sup> Hao Lu<sup>2</sup> Hangjun Ye<sup>1</sup>

<sup>1</sup>Xiaomi EV <sup>2</sup>Huazhong University of Science and Technology

## Abstract

*With the development of autonomous driving systems, mining high-value, safety-critical, and planning-relevant scenarios from large-scale driving logs has become essential for data-driven evaluation. In this paper, we propose **AutoMine**, a robust self-refining scenario mining method based on LLMs and VLMs. AutoMine uses semantics-preserving prompt augmentation to reduce LLM prompt sensitivity, combines robust trajectory atomic functions with VLM-based functions to handle perception noise and open-world visual cues, and refines generated code through execution feedback from real logs. In the Argoverse 2 Scenario Mining Competition at CVPR 2026, AutoMine achieves a HOTA-Temporal score of **36.38** and a Timestamp BA score of **77.21**.*

## 1. Introduction

Autonomous driving datasets contain massive sensor logs, while rare and safety-critical events remain sparse. Scenario mining enables targeted evaluation by retrieving logs, timestamps, and 3D actors that match a natural language description.

This task is challenging because query wording is precise, predicted tracks are noisy, and some scenarios require visual evidence beyond 3D trajectories. For example, *passing* and *overtaking* may imply different conditions, while tracks may contain missing detections, heading noise, fragmentation, and ID switches.

The rapid development of LLMs and VLMs brings new possibilities for scenario mining. RefProg [1] shows that LLMs can translate natural language descriptions into composable atomic function calls for interpretable trajectory-based mining. However, manually specified atomic functions are hard to scale to open-world visual concepts. In addition, one-shot generated code can fail when the LLM misunderstands the prompt, selects the wrong category,

reverses a relation, or imposes incorrect constraints; this is consistent with prior findings on LLM sensitivity to meaning-preserving prompt design choices [3].

We propose **AutoMine**, a multimodal scenario mining framework that strengthens LLM-generated programs with semantic-preserving prompt augmentation, robust atomic functions, VLM-based visual functions, perception post-processing, and execution-driven self-refinement. AutoMine executes generated code on real logs, summarizes the observed outputs, and uses the feedback to repair systematic errors, improving robustness to language ambiguity and perception noise.

## 2. Method

### 2.1. Overview

Given a natural language description, driving logs, and sensor data, AutoMine outputs referred actors and valid timestamps. As shown in Fig.1, AutoMine first refines perception tracks, then combines semantic-preserving prompt augmentation, LLM-generated atomic functions, multimodal execution over tracks, maps, and images, and execution-feedback-based code refinement. We describe each module in the following sections.

### 2.2. Trajectory Refinement

AutoMine uses the detection and tracking results from Le3DE2E [5] as initial trajectory inputs. Although these results provide strong 3D tracks, we observe ID switches, fragmented trajectories, missed detections, false positives, and duplicated boxes, which directly affect temporal localization and referred-actor selection.

Inspired by Immortal Tracker [4], we keep short unmatched tracklets alive instead of terminating them immediately, and reconnect compatible fragments using spatial, temporal, category, and size consistency. We also apply backward tracking to recover early segments that are easier to associate from later frames. In addition, we use the grounding capability of Qwen3.5-27B [2] to verify projected boxes in camera views and remove additional false positives. This trajectory refinement improves track con-

\*This work was done during internship at Xiaomi.

†Project leader.

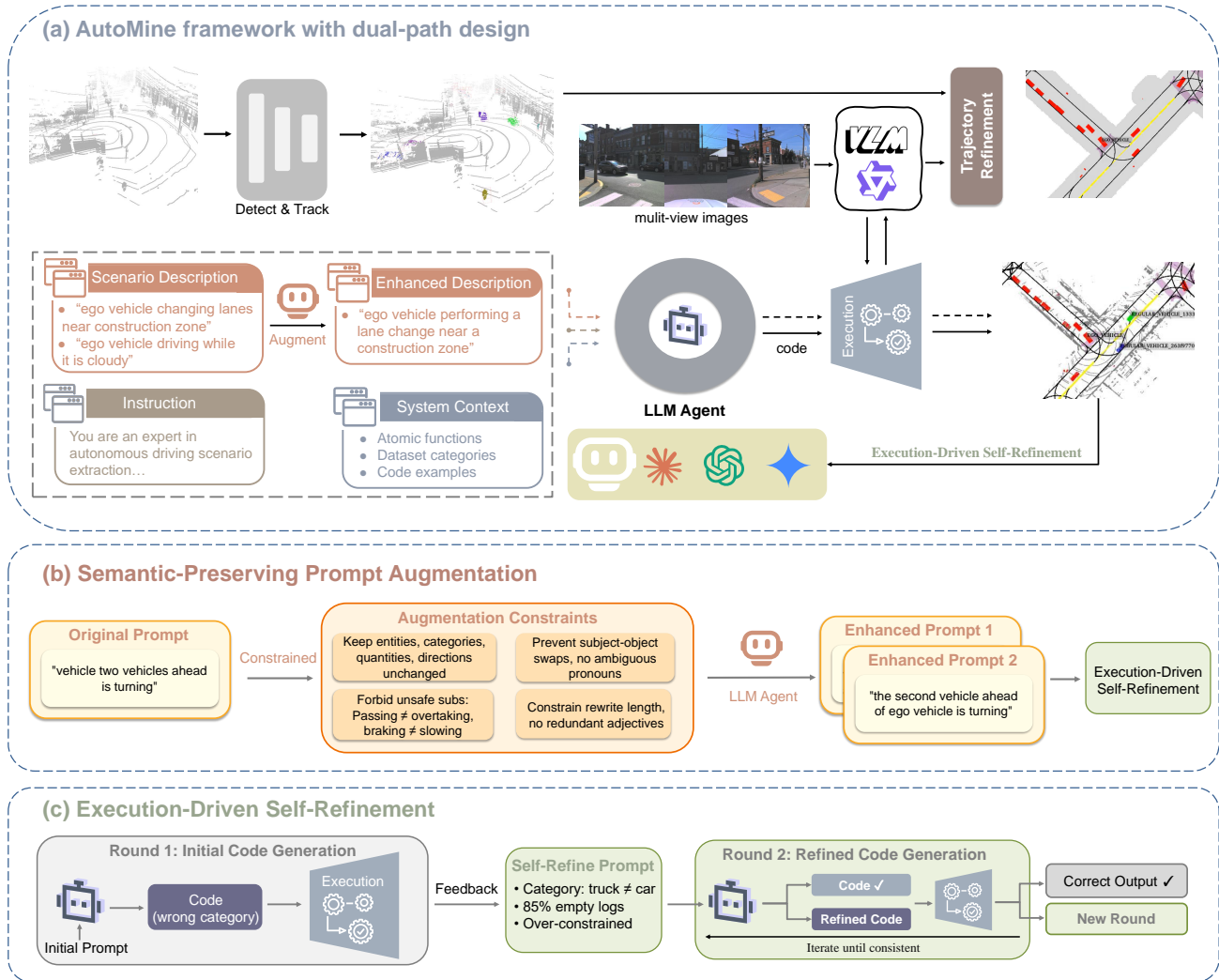


Figure 1. (a): Overview of the AutoMine framework with dual-path design (perception + language). (b): Semantic-preserving prompt augmentation. (c): Execution-driven self-refinement loop.

tinuity and provides more reliable inputs for downstream atomic functions.

### 2.3. Semantic-Preserving Prompt Augmentation

As mentioned above, prior work shows that LLMs are sensitive to prompt wording and formatting [3], and we observe the same issue when generating scenario mining code. To reduce this instability, AutoMine augments the natural language scenario descriptions before code generation.

The augmentation is constrained to preserve the original semantics rather than freely paraphrase the query. The rewrite prompt explicitly keeps all entities, categories, quantities, directions, road context, spatial-temporal relations, numerical values, and the referred target unchanged. It also forbids unsafe substitutions such as *passing*  $\neq$  *overtaking*, *braking*  $\neq$  *slowing*, *changing lanes*  $\neq$  *merging*,

*stopped*  $\neq$  *parked*, and *near*  $\neq$  *next to*. To avoid misleading the LLM’s judgment of referred objects, related objects, and relation directions, we further enforce constraints that prevent subject-object swaps, demoting the referred actor into a modifier, changing category granularity, or introducing ambiguous pronouns. We also constrain the rewrite length to prevent over-introducing redundant adjectives or extra background.

### 2.4. Robust Trajectory-Based Atomic Functions

AutoMine represents scenario logic as compositions of atomic functions over tracks, ego poses, maps, and time windows. Since predicted tracks are noisy, we refactor motion and relation functions to use temporally aggregated evidence instead of brittle single-frame measurements. Directional and spatial relations are evaluated over multiple valid

frames, with relaxed continuity checks and ego-relative geometry.

We also extend the library for planning-centric behaviors not covered by the baseline functions, including U-turns, three-point turns, side parking, special stopping behavior, object interactions, and map-aware road constraints. We use LLM clustering over all descriptions to identify missing function categories, then manually revise and implement the final function set.

## 2.5. VLM-Enhanced Atomic Functions

Some scenarios require visual evidence beyond 3D tracks and maps. AutoMine therefore provides VLM-enhanced atomic functions for fine-grained object type, visual attributes, environment, road surface, zones, pedestrian actions, traffic lights, occlusion, and attached objects. We use Qwen3.5-72B [2] as the underlying vision-language model for all visual reasoning calls.

For visual reasoning about candidate actors, we collect all camera views where the candidate appears and draw the projected 3D box on the original image instead of cropping it, preserving context under projection noise. For environment-level conditions, we use a representative front-view image because each log is short and the environment is usually stable. For road and zone conditions, we stitch candidate-visible frames into a panel annotated with camera names and timestamp indices, and the VLM returns the valid timestamps. Ego-related visual functions use separate prompts because the ego vehicle is not represented by a normal projected object box.

The VLM-enhanced function library includes the functions in Table 1. These functions allow AutoMine to preserve the advantages of symbolic program execution while extending coverage to open-world visual concepts.

## 2.6. Execution-Driven Self-Refinement

LLM-generated mining code often fails in systematic ways, such as selecting the wrong referred category, reversing relation-function arguments, missing `reverse_relationship`, using overly strict thresholds, or misjudging front/back and left/right geometry. Since these errors are hard to identify from code alone, AutoMine refines code with feedback from real execution.

In each round, a code generator produces scenario-mining code, and an executor runs it on up to `max_logs` logs with trajectory and VLM atomic functions. AutoMine then summarizes what the code retrieves, including per-track category, temporal coverage, size, ego-object geometry, and cross-log statistics such as referred-category distribution, empty-log ratio, and related-object distribution. These diagnostics expose category confusion, noisy short tracks, over-constrained logic, and relation-direction errors.

The next refinement prompt includes the original de-

Table 1. VLM-enhanced atomic functions.

Function	Description
<code>is_specific_object_type()</code>	Identifies fine-grained object types.
<code>in_environment()</code>	Checks weather or environment conditions.
<code>has_attribute()</code>	Verifies visual attributes or states.
<code>on_road_type()</code>	Determines the road type.
<code>on_road_surface()</code>	Checks road surface conditions.
<code>in_zone()</code>	Determines whether an actor is in a visual zone.
<code>pedestrian_action()</code>	Recognizes pedestrian actions.
<code>object_color()</code>	Identifies object color.
<code>at_traffic_lights()</code>	Checks traffic-light-related conditions.
<code>is_occluded_by()</code>	Detects visual occlusion relations.
<code>has_attached_object()</code>	Checks attached or carried objects.

scription, function library, category definitions, previous code, structured feedback, and the referred category from `REFERRED_DICT` as a hard constraint. The LLM keeps the code unchanged if the feedback is consistent; otherwise, it repairs function choices, argument order, reverse relations, thresholds, category filters, or spatial reasoning. This forms an execution-grounded self-refinement loop.

## 3. Experiments

### 3.1. Dataset and Evaluation Metrics

We conduct experiments on the official benchmark of the CVPR 2026 Argoverse 2 Scenario Mining Challenge. The benchmark is built upon the Argoverse 2 Sensor Dataset [6], which contains 1,000 driving logs (700 training, 150 validation, 150 test) with a total of approximately 4.2 hours of driving data. Each log lasts about 15 seconds. The sensor suite includes two 32-beam LiDARs (10 Hz), nine global shutter cameras (20 fps), HD maps, and 6-DOF ego-vehicle poses. The dataset provides 10,000 planning-centric natural language queries. The evaluation metrics are as follows:

**HOTA-Temporal** ( $\uparrow$ ). The primary ranking metric, computed only on the time window where the scenario description holds. For each prompt, predictions are filtered by both class and a per-prompt confidence threshold (selected from 10 recall-based candidates), and the standard HOTA score is computed using center-distance similarity (zero-distance = 2m). HOTA jointly measures detection and association

Table 2. Ablation results on the Argoverse 2 validation set.

Configuration	HOTA-Temporal(↑)	HOTA-Track(↑)	Timestamp BA(↑)	Log BA(↑)
<b>Different LLM backends (single-query baseline)</b>				
gpt-5.3-codex-9	15.72	24.93	61.19	63.76
Gemini-2.5-Pro	21.32	29.54	64.49	64.95
Claude-Sonnet-4.6	22.14	31.05	67.17	67.97
<b>Ablations on Claude-Sonnet-4.6</b>				
+ Trajectory Refinement	24.04	35.96	67.51	71.59
+ Atomic Function Optimization (Robust + VLM)	31.46	41.86	74.61	76.98
+ Execution-Driven Self-Refinement	33.96	44.95	75.98	<b>80.04</b>
+ Semantic-Preserving Prompt Augmentation	<b>34.99</b>	<b>45.91</b>	<b>77.87</b>	79.83

Table 3. Official leaderboard results of the AV2 2026 Scenario Mining Challenge on the HOTA-Temporal track.

Rank	Team	HOTA-Temporal(↑)	HOTA-Track(↑)	Timestamp BA(↑)	Log BA(↑)
1	HYU_OASIS	<b>38.50</b>	52.63	74.32	77.12
2	MTL (Argonaut)	37.04	<b>55.11</b>	75.50	<b>80.75</b>
3	<b>MISI (AutoMine) (Ours)</b>	36.38	49.32	<b>77.21</b>	76.26
4	Zelos_Agent_Driving	33.45	45.23	72.45	73.66
5	Lilly	32.51	42.06	71.93	73.11
6	DataClaw	31.40	44.40	74.05	79.10

quality:

$$\begin{cases} \text{HOTA}_\alpha = \sqrt{\text{DetA}_\alpha \cdot \text{AssA}_\alpha}, \\ \text{HOTA} = \frac{1}{|A|} \sum_{\alpha \in A} \text{HOTA}_\alpha, \end{cases} \quad (1)$$

where  $A = \{0.05, 0.10, \dots, 0.95\}$  is the set of localization thresholds, and DetA, AssA denote detection and association accuracy, respectively. The final score per prompt is the maximum HOTA over the 10 candidate thresholds.

**HOTA-Track** (↑). Same as HOTA-Temporal, except that any track ever marked as REFERRED\_OBJECT in a sequence is treated as referred across all of its frames. This evaluates the full lifetime of referred tracks rather than only the description-active interval.

**Timestamp BA** (↑). A frame-level retrieval metric that asks whether any referred object exists in each timestamp, independent of localization. Using the optimal thresholds from HOTA-Temporal, each frame is classified as positive iff it contains at least one REFERRED\_OBJECT. Per prompt, we aggregate frame-level TP/FP/TN/FN and compute balanced accuracy:

$$\text{Timestamp BA} = \frac{1}{2} \left( \frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}} \right). \quad (2)$$

**Log BA** (↑). The log-level counterpart of Timestamp BA. A (log, prompt) sequence is positive iff *any* frame contains a referred object. BA is then computed across all sequences using the same formulation as above.

### 3.2. Ablation Study

To validate the contribution of each design in AutoMine, we conduct ablation studies on the validation set, using the publicly available Le3DE2E [5] trajectories as initial input. We first compare three state-of-the-art LLMs under the single-query baseline and adopt Claude-Sonnet-4.6 as the default code generator, since it is the most stable on relation-argument ordering and fine-grained category selection. For all VLM-enhanced atomic functions, we use Qwen3.5-27B [2] as the underlying vision-language model. We then incrementally add each component on top of this baseline. The results are shown in Table 2.

We observe that each component improves the system through a distinctly different mechanism rather than simply boosting all metrics uniformly. Trajectory refinement contributes a much larger HOTA-Track gain than HOTA-Temporal gain, indicating that re-linking fragmented tracklets mainly extends the lifetime of already-correct referred objects rather than enlarging the temporal window of new scenarios. Atomic function optimization brings the largest single jump, because temporally aggregated relation predicates absorb per-frame heading noise on directional queries, while VLM-enhanced functions cover attributes that are unobservable from 3D boxes alone (e.g., traffic-light state, road surface, attached cargo); the two are largely complementary and fail on disjoint subsets of queries.

Execution-driven self-refinement further improves

Table 4. Official leaderboard results of the AV2 2026 Scenario Mining Challenge on the Timestamp BA track.

Rank	Team	Timestamp BA(↑)	Log BA(↑)
1	<b>AutoMine (Ours)</b>	<b>77.21</b>	76.26
2	MTL (Argonaut)	75.50	<b>80.75</b>
3	HYU_OASIS	74.69	77.84
4	DataClaw	74.18	78.67
5	EABOT_AMD	72.86	71.09
6	Lilly	72.70	75.77

all metrics by repairing three recurring error patterns we observe in the round-0 code: missing `reverse_relationship` calls, over-strict thresholds that yield zero candidates, and wrong referred categories caught against `REFERRED_DICT`. Semantic-preserving prompt augmentation yields only marginal gains on the raw baseline but becomes effective once stacked on top of the optimized pipeline, since it reduces prompt-induced variance across semantically equivalent rewrites rather than introducing new capability—an effect that is naturally amplified when the downstream pipeline is already strong.

### 3.3. Leaderboard Results

Our final submission, AutoMine, achieves excellent results on the test set of the CVPR 2026 Argoverse 2 Scenario Mining Challenge. Table 3 shows the leaderboard sorted by the primary metric HOTA-Temporal: we rank **3rd** with **36.38** and HOTA-Track of 49.32. Table 4 shows the leaderboard sorted by Timestamp BA: we rank **1st** with **77.21**, demonstrating the advantage of AutoMine in temporal localization accuracy.

## 4. Conclusion

In this report, we presented **AutoMine**, a robust multimodal scenario mining framework for the AV2 2026 Scenario Mining Challenge. AutoMine addresses several key challenges in natural-language-driven scenario mining, including LLM prompt sensitivity, noisy and fragmented perception tracks, ambiguous spatial-temporal relations, and open-world visual concepts that cannot be captured by 3D trajectories alone. To this end, AutoMine refines raw trajectories, applies semantics-preserving prompt augmentation, builds robust trajectory-based and VLM-enhanced atomic functions, and further improves generated mining programs through execution-driven self-refinement using feedback from real logs. Experiments and ablation studies on the Argoverse 2 benchmark show that these components provide complementary gains, with atomic function optimization and self-refinement substantially improving actor retrieval and temporal localization. On the official leaderboard, AutoMine achieves the best Timestamp BA score

of 77.21, demonstrating strong temporal localization ability, and ranks 3rd in HOTA-Temporal with a score of 36.38. These results show the promise of integrating symbolic programs, perception refinement, and visual reasoning for scenario mining.

## References

- [1] Cainan Davidson, Deva Ramanan, and Neehar Peri. Refav: Towards planning-centric scenario mining. *arXiv preprint arXiv:2505.20981*, 2025. 1
- [2] Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. 1, 3, 4
- [3] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, 2024. 1, 2
- [4] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, and Zhaoxiang Zhang. Immortal tracker: Tracklet never dies. *arXiv preprint arXiv:2111.13672*, 2021. 1
- [5] Zhepeng Wang, Feng Chen, Kanokphan Lertniphonphan, Siwei Chen, Jinyao Bao, Pengfei Zheng, Jinbao Zhang, Kaer Huang, and Tao Zhang. Technical report for argoverse challenges on unified sensor-based detection, tracking, and forecasting. *arXiv preprint arXiv:2311.15615*, 2023. 1, 4
- [6] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *ArXiv*, abs/2301.00493, 2023. 3