Multi-Sample Voting solution for the 2026 AV2 Scenario Mining Challenge

Huailong Peng, Jiatong Du, Yanjun Huang, Hong Chen
College of Automotive and Energy Engineering, Tongji University

Abstract

We present our solution to the Argoverse 2 Scenario Mining Challenge, which asks systems to retrieve objects and time intervals matching a natural-language scenario description from large-scale driving logs. Our approach follows the RefAV program-synthesis paradigm: an LLM generates executable Python scripts over a fixed library of atomic predicates, and these scripts are executed on the organizer-provided tracking outputs. We incorporate Global Context-Aware Generation and Multi-Agent Refinement from prior work, and further study Multi-Sample Voting (MSV), a prediction-level ensemble that votes over (timestamp, track) pairs across independently generated predictions. Experiments compare four LLMs, several global-context batch sizes, and different voting settings; the strongest single sample reaches 30.28 HOTA-Temporal on the Test split, and the final five-sample MSV ensemble reaches 30.79 HOTA-Temporal, 42.12 HOTA-Track, 71.44 Timestamp BA, and 73.73 Log BA. Detailed quantitative results are reported in the experiment section, and the source code of our solution is publicly released.

1. Introduction

The validation of autonomous driving systems requires identifying rare and safety-critical scenarios within large volumes of logged sensor data. Scenario mining formulates this need as a retrieval task: given a natural-language description, a system must return the objects and time intervals in which the described situation holds. The Argoverse 2 Scenario Mining Challenge provides a standardized benchmark for this task and evaluates submissions under Spatio-Temporal Localization and Temporal Localization tracks.

We build on the RefAV baseline[1], which casts scenario mining as program synthesis. In this paradigm, an LLM converts each natural-language description into an executable script that calls a predefined library of atomic functions describing geometric, motion, and semantic predicates. The script is then executed on the predictions of a fixed, organizer-provided tracker. Since the tracking input is shared, the main design space of our work is the generation and combination of scenario-mining programs.

This report therefore studies a planning-centric code-generation pipeline rather than a new detector or tracker. We use Global Context-Aware Generation and Multi-Agent Refinement from prior work[3] as pipeline components, and focus on two questions: how LLM choice and global-context batch size affect the generated programs, and whether independently generated predictions can be combined effectively. To address the second question, we introduce Multi-Sample Voting (MSV) at the prediction level and evaluate the conditions under which it improves performance.

Our contributions are: (1) a controlled empirical comparison of four LLMs and several global-context batch sizes under a fixed tracking input; (2) a prediction-level MSV scheme that merges independently generated outputs by voting over (timestamp, track) pairs; and (3) an analysis of how the relative strength of contributing samples affects voting behavior. The implementation and experimental results are described in the following sections.

2. Method

2.1. Overall framework

Our pipeline follows the RefAV baseline and contains three stages. First, in the tracking stage, we use the organizer-provided Le3DE2E tracking predictions[4], which are held fixed across all experiments. Second, in the code-generation stage, an LLM agent translates each scenario description into a Python script that composes calls to a fixed library of 32 atomic predicates. Each predicate is exposed through its function signature and docstring, and the model is constrained to write code against this library. Third, in the execution and filtering stage, the generated script is executed in a controlled namespace where the atomic functions and per-log data are available; the resulting set of (track, timestamp) pairs is written as the prediction. The same prediction file is submitted to both challenge tracks.

This paper was supported in part by the National Natural Science Foundation of China under Grant 525B2180, in part by the National Natural Science Foundation of China under Grant U23B2061, in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100

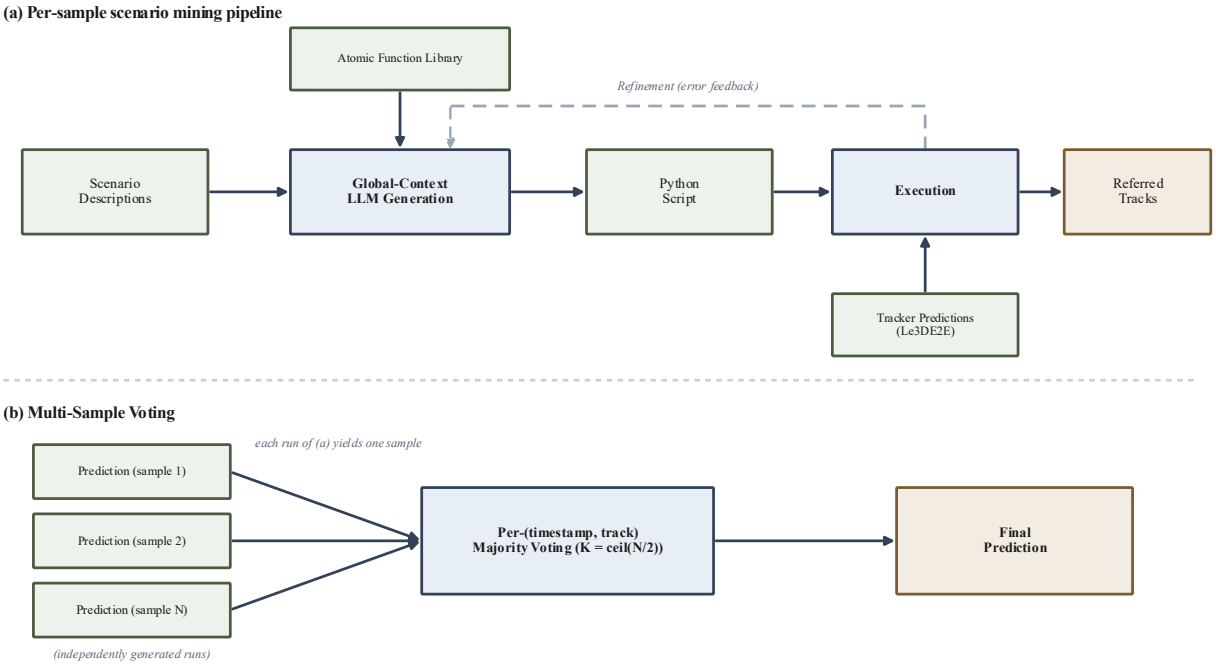

Figure 1. Overall architecture of the RefAV2026 scenario-mining pipeline.

2.2. Global context and refinement

We adopt two strategies from prior work[3]. Global Context-Aware Generation bundles a batch of B descriptions into a single prompt and asks the LLM to emit the corresponding code blocks in one response, encouraging consistent treatment of related descriptions. Multi-Agent Refinement re-prompts the LLM with the failed script and its error traceback whenever execution raises an exception, so that runtime failures can be repaired automatically. We treat both strategies as inherited components of the pipeline. Our implementation removes a fixed output-token limit on the OpenAI-compatible interface, which had truncated long batches, and treats the batch size B as a hyper-parameter evaluated in Section 3.3.

2.3. Multi-sample voting

Self-consistency[5] improves reasoning by sampling multiple outputs and voting over their answers. We adapt this idea to scenario-mining predictions. Given N independently generated prediction files for the same log and description, MSV retains each (timestamp, track) pair that appears in at least K of the N files. The threshold K controls the voting rule: K = ceil(N/2) gives majority voting, K = N gives unanimous intersection voting, and K = 1 gives union voting. Voting is performed at the (timestamp, track) level rather than at the whole-scenario level, so partial temporal disagreements do not automatically remove an entire scenario. Section 3.4 evaluates how this voting rule behaves under different sample-strength settings.

3. Experiments

3.1. Dataset and metrics

We evaluate on the Argoverse 2 scenario-mining benchmark[1], built on the Argoverse 2 Sensor dataset[2], which provides 750 training, 150 validation, and 150 test logs. Each scenario description is associated with both positive and negative instances. We report the four official metrics: HOTA-Temporal, the primary metric of the Spatio-Temporal track, which jointly assesses detection and association over the precise interval when the scenario occurs; HOTA-Track; Timestamp Balanced Accuracy (Timestamp BA), the primary metric of the Temporal track; and Log-level Balanced Accuracy (Log BA). All scores are produced by the official evaluator; Test scores are computed on the Test split.

3.2. Implementation details

Each LLM is accessed through an OpenAI-compatible interface. Single-sample runs use Global Context-Aware Generation and Multi-Agent Refinement, with at most two

LLM	Batch	HOTA-Temporal	HOTA-Track	Timestamp BA	Log BA
claude-opus-4.6	15	28.31	38.48	69.94	71.85
claude-opus-4.6	30	28.59	39.06	70.24	72.26
claude-opus-4.7	20	29.31	39.64	70.42	71.83
claude-opus-4.7	30	30.28	41.27	70.91	72.73
claude-opus-4.8	30	28.27	38.59	70.32	71.52
claude-opus-4.8	32	28.89	38.85	70.74	71.46
gpt-5.5	30	29.56	40.49	70.62	72.58
gpt-5.5	32	28.52	38.80	69.53	71.86

Table 1. Single-sample Test results for different LLMs and global-context batch sizes. All runs use Global Context-Aware Generation and Multi-Agent Refinement; higher is better for every metric.

Split	Scheme	Samples	HOTA-Temporal
Val	Best single	opus-4.6 / gpt-5.5	24.69
Val	MSV K=2 of 3 (majority)	2x opus-4.6 + gpt-5.5	25.04
Val	MSV K=3 of 3 (unanimous)	2x opus-4.6 + gpt-5.5	21.42
Test	Best single	opus-4.7 (batch 30)	30.28
Test	MSV K=2 (unequal)	opus-4.7 + 2x opus-4.6	29.45
Test	MSV K=2 (same-model)	3x opus-4.7 (batch 30/32/35)	30.22
Test	Final MSV (5 samples)	opus-4.7 b30/b32/b35 + 2x gpt-5.5 b30	30.79

Table 2. Multi-Sample Voting results. The final five-sample cross-model ensemble obtains the best Test result; higher HOTA-Temporal is better.

repair rounds per failing script and no additional review stage. Because generation uses temperature sampling, two runs of the same model differ; we measured a single-run variation of approximately 0.07 HOTA-Temporal and treat differences below this level as noise. For 3D tracking, we use the organizer-provided Le3DE2E predictions[4]. The same prediction file is submitted to both challenge tracks.

3.3. Comparison of LLMs and batch sizes

Table 1 compares single-sample submissions under varying LLMs and global-context batch sizes, with all other components held fixed. At a fixed batch size of 30, which isolates the effect of the model, claude-opus-4.7 attains the strongest single-sample result among the evaluated models (HOTA-Temporal 30.28), followed by gpt-5.5 (29.56), claude-opus-4.6 (28.59), and claude-opus-4.8 (28.27). Within this evaluated set, a higher version label in the same model family does not guarantee better performance on structured scenario-mining code generation.

Increasing the batch size from a smaller value to 30 improves both claude-opus-4.6 (28.31 to 28.59 from batch 15 to 30) and claude-opus-4.7 (29.31 to 30.28 from batch 20 to 30). Beyond 30, the effect is smaller and model-dependent in sign: from batch 30 to 32, gpt-5.5 decreases by 1.04, whereas claude-opus-4.8 increases by 0.62. These results suggest that the benefit of larger global-context batches saturates around 30 in our setting, and that no single batch size is uniformly optimal.

3.4. Multi-sample voting

Table 2 summarizes the MSV study. On the validation split, we vote over three samples whose single-run HOTA-Temporal scores lie between 24.58 and 24.69, a spread of 0.11. Majority voting (K = 2 of N = 3) raises HOTA-Temporal to 25.04, improving over the best contributing sample by 0.35 and exceeding the measured single-run noise level. In contrast, unanimous voting (K = 3 of 3) over the same samples reduces HOTA-Temporal to 21.42, indicating that strict intersection voting removes too much temporal coverage.

On the Test split, the relative strength of the contributing samples is decisive. Majority voting over claude-opus-4.7 at batch 30 together with two weaker claude-opus-4.6 samples has a spread of 1.97 and yields only 29.45, below the best single sample. A same-model ensemble of three claude-opus-4.7 samples reaches 30.22. The best MSV result comes from a five-sample cross-model ensemble

combining claude-opus-4.7 at batches 30, 32, and 35 with two gpt-5.5 batch-30 predictions; it reaches HOTA-Temporal 30.79, HOTA-Track 42.12, Timestamp BA 71.44, and Log BA 73.73. Because not every ensemble member was separately evaluated as a standalone submission, we report the spread only for contributors with known individual scores and interpret the ensemble comparison conservatively.

4. Conclusion

We described our solution to the 2026 Argoverse 2 Scenario Mining Challenge. The solution keeps the organizer-provided tracker fixed, follows the RefAV program-synthesis baseline, and incorporates Global Context-Aware Generation and Multi-Agent Refinement from prior work. Our experiments compare LLM choices and batch sizes, and evaluate MSV as a prediction-level ensemble. The results show that majority voting can improve over the best single sample when the contributing predictions are of comparable strength, whereas a large strength gap or unanimous intersection voting is harmful. Among our evaluated configurations, the strongest single sample reaches HOTA-Temporal 30.28 on the Test split, and the final five-sample MSV ensemble reaches HOTA-Temporal 30.79, HOTA-Track 42.12, Timestamp BA 71.44, and Log BA 73.73. The source code of our solution is publicly released.

References

[1] Cainan Davidson, Deva Ramanan, and Neehar Peri. RefAV: Towards Planning-Centric Scenario Mining. arXiv preprint arXiv:2505.20981, 2025.

[2] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint arXiv:2301.00493, 2023.

[3] Dubing Chen, Huan Zheng, Wencheng Han, Runzhou Tao, Zhongying Qiu, Jianfei Yang, and Jianbing Shen. SM-Agent solution for AV2 2025 scenario mining challenge. CVPR 2025 Workshop on Autonomous Driving technical report, 2025.

[4] Feng Chen, Kanokphan Lertniphonphan, Yaqing Meng, Ling Ding, Jun Xie, Kaer Huang, and Zhepeng Wang. Le3DE2E solution for AV2 2024 unified detection, tracking, and forecasting challenge. CVPR 2024 Workshop on Autonomous Driving technical report, 2024.

[5] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain-of-thought reasoning in language models. In International Conference on Learning Representations, 2023.