The Case for Negative Data: From Crash Reports to Counterfactuals for Reasonable Driving

Jay Patrikar[1,*], **Apoorva Sharma**[2], **Sushant Veer**[2], **Boyi Li**[2], **Sebastian Scherer**[3], **Marco Pavone**[2,4]
[1]Field AI [2]NVIDIA [3]Carnegie Mellon University [4]Stanford University

Abstract: Learning-based autonomous driving systems are trained mostly on incident-free data, offering little guidance near safety–performance boundaries. Real crash reports contain precisely the contrastive evidence needed, but they are hard to use: narratives are unstructured, third-person, and poorly grounded to sensor views. We address these challenges by normalizing crash narratives to ego-centric language and converting both logs and crashes into a unified scene–action representation suitable for retrieval. At decision time, our system adjudicates proposed actions by retrieving relevant precedents from this unified index; an agentic counterfactual extension proposes plausible alternatives, retrieves for each, and reasons across outcomes before deciding. On a nuScenes benchmark, precedent retrieval substantially improves calibration, with recall on contextually preferred actions rising from 24% to 53%. The counterfactual variant preserves these gains while sharpening decisions near risk.

Keywords: Autonomous Driving, Retrieval-Augmented Reasoning, Safety

1 Introduction

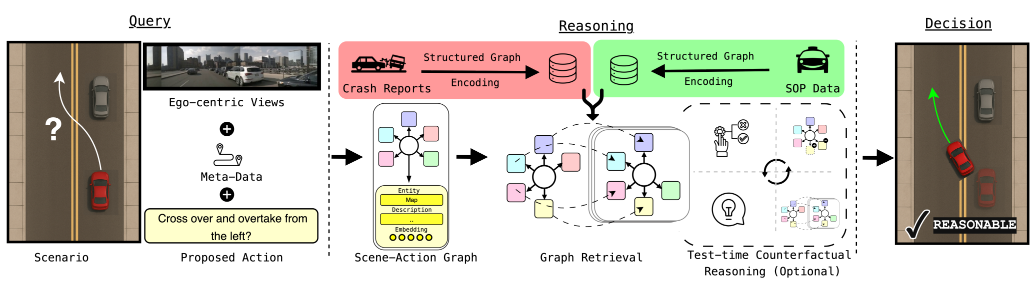

Figure 1: Overview of our pipeline. **Input**: a driving scene with front-facing video and a candidate ego action. **Processing**: convert the scene to a structured scene–action graph and retrieve precedents from a unified index of positive logs and crash narratives; optionally run an agentic counterfactual loop that proposes and evaluates alternative actions. **Output**: a justified label for the proposed action (`UNSAFE | SAFE | REASONABLE`) grounded in retrieved evidence.

End-to-end learning-based autonomous vehicle (AV) systems are trained primarily through imitation learning on positive, incident-free driving data [1, 2]. This data is typically collected by expert human drivers driving sensor-instrumented vehicles in a variety of driving scenarios, resulting in a dataset pairing the sensor observations that the AV will encounter with the action that the human driver chose in that moment. While this data helps define "good" driving that an AV should imitate, it does not provide direct supervision of what behaviors are to be avoided. Some have aimed to address this gap through auxiliary reward functions defining a rules-based definition of risky driving

[*]This work was done in-part as an intern at NVIDIA Research and a PhD student at the CMU.

[3], but such rules can be challenging to specify: Risk is difficult to quantify due to uncertainty over other road user's behaviors. Moreover, competent driving requires appropriately *managing* the risk that is taken on to make progress; remaining stopped is the safest policy, but not competent driving behavior.

Instead, in this work, we consider an alternative data-driven approach to provide negative supervision for AV decision making. Specifically, we explore the use of crash reports as a complementary source of driving knowledge. Agencies such as the National Highway Traffic Safety Administration (NHTSA) collect structured narrative accounts of real-world accidents, including the actions taken and the conditions under which failures occurred. While these reports lack the rich multimodal data of first-person human-driven AV logs, they contain valuable causal and contextual information that can support counterfactual reasoning. While these reports can't directly be used in policy training, recent advances in vision-language models (VLMs) capable of reasoning across sensor and text domains offer a compelling avenue for bringing such valuable sources of negative data into AV decision making.

In this paper, we study how negative data influences VLM reasoning capabilities in AV decision making tasks by developing a retrieval-augmented-generation (RAG) pipeline for AV safety adjudication. Specifically our contributions are as follows: (i) a GraphRAG [4] style retrieval pipeline for both positive and negative driving precedent, using a unified structured language representation for both sensor-domain positive data and language-domain negative data; (ii) an agentic extension which uses additional test-time compute to reason about counterfactuals prior to making a safety judgment; (iii) evaluation of both approaches in terms of alignment with human judgement on the safety of possible actions in driving scenarios, showing the impact of negative crash report data on VLM decision making capability.

2 Related Works

2.1 Safety Assessment of Motion Plans

The increasingly black-box nature of autonomous driving systems raises significant challenges towards safety assessment of the motion plans they produce. Classically, methods drawn from reachability theory [5] have been used to provide an understanding of "criticality" of a particular interaction. Vanilla reachability uses the worst-case assumption on the behavior of other agents which makes it overly conservative limiting its use in real-world situations. While methods that assume a more reasonable behavior from the other agent have been proposed [6, 7], the effectiveness of these methods in distribution shifts and high-dimensional data are not well studied. Uncertainty quantification (UQ) [3, 8] is another paradigm that can be used to identify unsafe scenarios given a dataset of safe scenarios. While UQ methods are effective in identifying scenarios that are far from the safe distribution, these methods fail to offer counterfactuals which are often essential to reason about safety [9, 10]. Rules [11] and catalog-based [12] methods that rely on humans to craft hand-engineered rules have also been proposed, but these methods are painstaking to develop and can struggle with far edge-case scenarios that were overlooked during development. Yet another category of approaches, that fall within the sub-field of explainable AI, furnish human-understandable "reasoning" that the model plausibly underwent for generating a particular outcome [13]. Some popular methods include the use of visual saliency maps [14] and vision and language captions for providing decision explanations [15, 16, 17, 18] among others. However, these approaches predominantly focus on positive behaviors, limiting their ability to distinguish good from bad driving behaviors, unlike this paper.

2.2 Leveraging Crash Report Data

A significant bottleneck to using negative or crash data is its lack of availability. Most regular driving logs—e.g., nuScenes [1], nuPlan [19], and Waymo Open Motion Dataset [20]—which include vectorized scene information only provide safe driving data. To address this limitation, NHTSA crash reports [21], which are in the form of textual descriptions of real-world crashes, have been leveraged for synthesizing crash simulations [22, 23, 24]. However, it remains unclear how realistic these simulations are to be able to stake a safety claim using them. Directly using the crash reports without converting them to simulations first can alleviate the need for validating the simulations. However, most prior works that directly use the crash reports look at aggregate bechmarking of the safety of autonomous driving rather than a case-by-case safety analysis [25]. In this paper, we develop an approach that uses RAG to retrieve the nearest scenario to the current one from a database that consists of the NHTSA crash reports along with regular driving data from nuScenes and supply it to a VLM-based safety monitor. Importantly, we show that the use of the crash reports help reduce the over-conservatism inherent to VLM-based monitors.

3 Problem Statement

To investigate the impact of negative data on VLM decision-making, we create a benchmark based on action adjudication. Specifically, we set up a retrieval-augmented pipeline for a multi-class classification problem. Given a driving scene $x \in \mathcal{X}$, and a candidate action $a \in \mathcal{A}$, the VLM is asked to classify whether action a is either `UNSAFE` (would likely result in a collision, rule violation, or hazardous interaction), `SAFE` (legal and physically safe, but not an action that a reasonable driver would choose), and `REASONABLE` (safe, and likely to be chosen by a reasonable driver). We distinguish `SAFE` and `REASONABLE` to better test the VLM's ability to adjudicate safety in alignment with human drivers – often, the `REASONABLE` action is the one that takes on the appropriate level of risk to maintain reasonable progress; marking `REASONABLE` actions as `UNSAFE` is an indication of an overconservative safety assessment.

We represent the scene x in a VLM compatible manner using (i) a forward-facing camera image and (ii) a natural-language description of the scene. We choose a discrete action space containing the following high-level actions:

$$\mathcal{A} = \left\{ \begin{array}{c} \texttt{MERGE LEFT, TURN LEFT, NUDGE LEFT,} \\ \texttt{STRAIGHT, STOP, ACCELERATE, DECELERATE,} \\ \texttt{NUDGE RIGHT, TURN RIGHT, MERGE RIGHT} \end{array} \right\}$$

For a given scene-action pair (x, a) we query the VLM to predict the label $y \in \mathcal{Y} = \{\texttt{UNSAFE}, \texttt{SAFE}, \texttt{REASONABLE}\}$, and use the classifier performance as a signal for VLM safety adjudication capabilities.

4 A Unified Data Representation for Structured Retrieval of Positive and Negative Data

A key challenge in enabling a decision maker to reason about both positive and negative driving examples is unifying the starkly different types of data – high-quality, annotated camera data in the case of positive driving data, and relatively terse, technical crash reports in the case of negative data. We construct a unified retrieval corpus by converting both crash reports and AV driving logs into a shared structured representation suitable for retrieval.

In particular, we embed each into structured *scene-action graphs*. Each graph G consists of a set of nodes, each with a type $t \in \mathcal{T}$ where the set of types is $\mathcal{T} = \{\mathsf{Ego}, \mathsf{Obstacles}, \mathsf{Map}\}$. Each node has an associated natural language description that summarize the state or behavior of each element in the scene. For positive data, we construct this graph using a pipeline involving video summarization using a VLM, followed by LLM processing of the result into the structured representation. For negative crash report data, we use an LLM-only pipeline to obtain the graphs. For details on these pipelines, and examples of inputs the output structured representation, see Appendix B and Figure 2.

These graphs serve as the corpus for GraphRAG-style retrieval: the text of each node is converted to a vector via NV-EmbedQA-Mistral-7B-v2, a state-of-the-art text embedding model optimized for retrieval tasks ϕ. For a query graph G_q and a corpus graph G, we compute similarity as a sum over

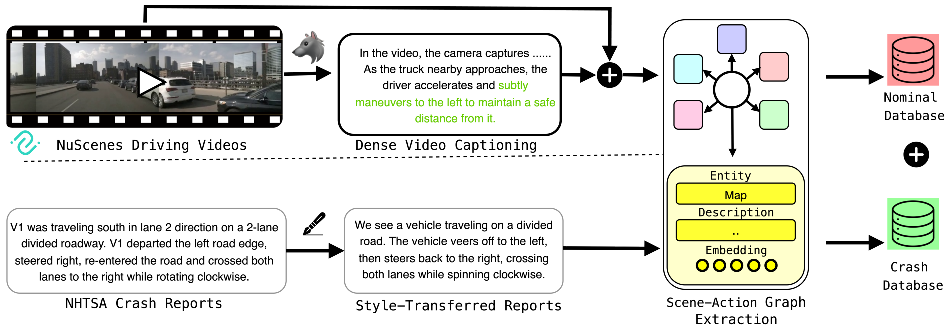

Figure 2: Unified representation. Positive driving logs and negative crash narratives are converted to structured scene–action graphs with canonical node types. Nodes carry natural language summaries and embeddings for GraphRAG retrieval.

node types, where the graph similarity for a node type is highest pairwise similarity between nodes of the same type:

$$S(G_q, G) = \sum_{t \in \mathcal{T}} w_t \max_{v \in \mathcal{V}_t(G_q)} \max_{u \in \mathcal{V}_t(G)} \mathrm{Sim}_{\cos}\big(\phi(d(v)), \phi(d(u))\big)$$
$$\text{where } \mathcal{V}_t(G) = \{u \in \mathcal{V}(G) : \tau(u) = t\},\ w_t \geq 0,\ \sum_t w_t = 1.$$

where $\mathcal{V}(G)$ is the set of nodes of graph G, and $\tau(u)$ returns the type of node u, and $\mathrm{Sim}_{\mathrm{cosine}}$ is the cosine similarity of two vectors. We use top-k retrieval under S to form the precedent set for decoding. By explicitly comparing the query scene against the corpus on each relevant axis, such graph-based retrieval enables more precise and controllable action-conditioned retrieval than directly using holistic embeddings of the full scene descriptions.

4.1 Structuring Negative Data from Crash Reports

We process NHTSA crash reports through a three-step pipeline: (1) sourcing free-text crash narratives, (2) style normalization to convert third-person templated language into ego-centric descriptions, and (3) scene-action graph construction using prompt-based extraction to identify canonical entities (*Map, Ego, Ego Action, Obstacles, Obstacles Action, Collision/No Collision*). Each node is embedded to enable fine-grained graph-level similarity for retrieval. See Appendix A for detailed implementation.

4.2 Structuring Positive Data from Driving Logs

We extract positive driving scenarios from the nuScenes dataset and process them through a three-step pipeline: (1) sourcing sensor-rich logs, (2) dense video captioning to summarize driving segments, and (3) scene-action graph construction using prompt-based extraction to identify canonical entities (*Ego, Ego Action, Obstacles, Map*). Each node is embedded to support fine-grained GraphRAG retrieval. See Appendix A for detailed implementation.

5 VLM Reasoning Engines for Adjudicating Driving Actions

In order to study the impact of negative data on adjudicating the safety of driving actions, we compare three reasoning paradigms that incorporate precedent data to varying degrees: (1) a base vision-language model (VLM) operating without external context f_{base}, (2) a VLM augmented with precedent retrieval via GraphRAG and chain-of-thought (CoT) decoding f_{rag}, and (3) an agentic extension that iteratively explores counterfactuals to improve robustness f_{agentic}, each illustrated in Figure 3.

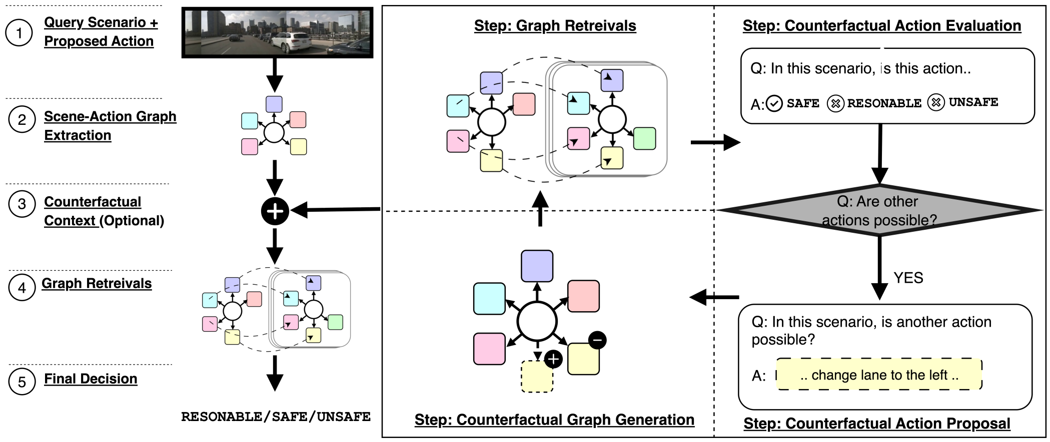

Figure 3: Reasoning engines. We compare a VLM-only baseline (only steps 1 and 5, no additional context), a one-step CoT with precedent retrieval via GraphRAG (1,2,4,5), and an agentic counterfactual loop that proposes alternatives and retrieves precedents for each before adjudication (1,2,3,4,5).

For all reasoning engines, we use GPT-4o as the primary reasoning engine, which provides strong multimodal reasoning capabilities for both visual scene understanding and textual precedent analysis. We leave the impact of the choice of different VLMs and spatially-aware VLMs, including systems that use specialized models for spatial reasoning like GroundingDINO [26], for future conference-level iterations of this work.

Next, to study the impact of providing driving precedent, we provide the VLM with context retrieved using the GraphRAG approach detailed above, and augment the prompt to encourage chain-of-thought (CoT) decoding to factor in the retrievals into the judgment, f_{RAG}. Finally, we explore the impact of additional test-time compute by constructing an agentic extension that iteratively explores counterfactuals, performing multiple rounds of retrievals from the precedent corpus, f_{agentic}. We detail each of these approaches below.

5.1 VLM-Only Baseline

In the simplest configuration, the VLM receives a natural language description of the driving scene and a proposed alternate action for the ego vehicle. It is prompted to reason about the action's legality, safety, and efficiency, and assign one of three outcome labels: `UNSAFE`, `SAFE`, or `REASONABLE`. This method reflects a purely generative model without access to precedent scenarios, which we use as a baseline for comparison.

5.2 One Step RAG + Chain-of-Thought Decoding

To enrich the VLM's judgment with external behavioral context, we augment it with precedent retrieval via a GraphRAG-style method tailored to our structured graph format.

Given a query scene and proposed ego action, we construct the query graph G_q, setting the *Ego Action* node with the candidate, and retrieve the top-k most similar graphs from the precedent corpus according to the similarity function $S(G_q, G)$ detailed in Section 4.

We append the full textual description of the retrieved scenes, as well as their class (positive/negative) to the VLM context, together with a CoT prompt which encourages the VLM to draw analogies to the retrieved scenarios when adjudicating the action. See the Appendix for more details and the prompt template used.

5.3 Agentic Counterfactual GraphRAG

The agentic counterfactual loop builds on recent trends in test-time reasoning and planning, where models dynamically adjust their computational effort based on task complexity. This aligns closely with recent Agentic paradigms like ReAct [27], Reflexion [28], and test-time counterfactual planning in embodied agents [29], where additional reasoning steps, retrievals, or internal simulations are performed only when needed. Rather than producing a single output in one pass, these systems reframe decision-making as a multi-step process—evaluating, reflecting, and refining based on intermediate outputs.

In our setting, each retrieval–evaluation loop represents a unit of deliberation. Simple scenarios —e.g., clear STOP or obvious collision—may resolve in one step. But complex scenarios may require multiple iterations of proposal, evidence gathering, and adjudication before a confident classification is made. To enable this, we allow the VLM to propose alternative actions with which to query the current scene. Instead of issuing a single judgment, the model actively generates and evaluates plausible alternatives, each grounded in precedent cases. This mirrors human-like counterfactual reasoning: "What if I turned instead of braking? Have others made similar choices in this scenario—and what happened next?" The model's internal state evolves over time, accumulating both counterfactual actions and their justifications to make better-informed final decisions.

The process follows a generation–retrieval–evaluation loop, implemented as a LangGraph [30] pipeline:

- **(Step 1) Generation:** A new counterfactual action is proposed. The model internally reflects on prior retrievals and decisions to avoid repetition and focus on unexplored decision boundaries.
- **(Step 2) Retrieval:** The query graph is updated with the proposed action, and top-k similar scenes are retrieved from the precedent database using GraphRAG. These retrievals enhance behavioral diversity while preserving semantic alignment.
- **(Step 3) Evaluation:** Retrieved scene descriptions are used to assess the plausibility and safety of the counterfactual action. Evaluation is guided by precedent-aligned reasoning and incorporates justification feedback, which is used to refine future proposals.

This cycle repeats till the agent decides to stop or for a max number of iterations. At each step, introspective reasoning—akin to ReACT or Reflexion—allows the model to learn from previous steps and improve future iterations. The accumulated evidence—drawn from diverse plausible behaviors—forms a more robust context for evaluating the original action. The final classification for the candidate action is made after this deliberative process, incorporating precedent-informed reasoning comparing the candidate action against a richer behavioral space.

6 Results and Discussions

We investigate how structured precedent data—particularly crash reports—can inform the classification of counterfactual driving actions as UNSAFE, SAFE, or REASONABLE. Human annotators labeled a balanced dataset of 1,275 action-scene pairs, based on visual scenes, textual descriptions, and ten standard driving actions. We evaluate how vision-language models (VLMs) perform across four configurations: without retrieval (VLM-only), with precedent retrieval (VLM+RAG), with RAG using only positive examples (VLM+RAG-PosOnly), and with an agentic counterfactual refinement mechanism (Agentic RAG). This section summarizes our key findings and hypotheses, see the Appendix for more details on experimental set-up.

Retrieval Enables Nuanced Reasoning Beyond Binary Safety Judgments Retrieval-augmented models significantly improve the ability to distinguish between actions that are merely safe and those that are both safe and contextually preferred. Without access to precedents, the VLM-only model overpredicts danger—labeling 37% of `REASONABLE` actions as `UNSAFE`. By incorporating retrieved

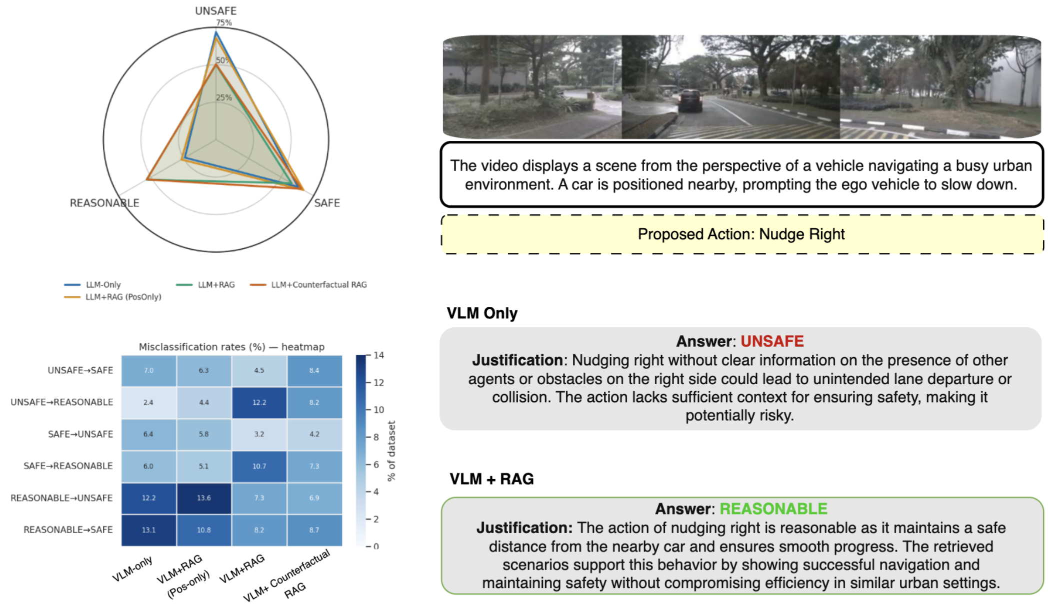

Figure 4: Overall results. Left: consolidated quantitative summary combining class-wise recall and misclassification breakdown across four engines (VLM-only, VLM+RAG (PosOnly), VLM+RAG, Agentic RAG). Right: qualitative example where the ground-truth outcome is REASONABLE. The VLM-only model labels the proposed action UNSAFE, while precedent-augmented VLM+RAG correctly judges it REASONABLE, citing retrieved precedents. See text for discussion of recall gains and critical error reductions.

scenarios, the VLM+RAG model increases REASONABLE recall from 24% to 53%, demonstrating a more refined understanding of assertive yet appropriate behavior. This improvement, however, comes with a tradeoff: UNSAFE recall drops from 72% to 50%, suggesting the base VLM tends towards conservative caution without precedent, but when offered the possibility of negative precedent but lacking a similar enough negative example, may falsely assume an action is SAFE. In practice, we observe that many UNSAFE $\rightarrow$ SAFE errors arise when retrieved reports are not semantically close to the scene at hand. Because our crash corpus is limited in size and coverage (we do not use the full NHTSA dataset due to budget constraints), the system can implicitly treat the absence of a matching negative precedent as weak evidence of safety. We expect this effect to diminish with a larger and more diverse set of crash reports. As an additional mitigation, when no sufficiently similar precedent is retrieved, the model could be prompted to explicitly reason about potential failure modes rather than defaulting to SAFE. These shifts suggest that precedent retrieval largely improves the model's ability to align with human preferences on the actions that human labellers deem efficient – those that appropriately manage risk while maintaining efficiency.

Negative Precedents Are Crucial for Contrastive Reasoning When crash reports are withheld from the retrieval pool, REASONABLE classification drops sharply—from 53% to 27%—highlighting the model's inability to recognize assertive but acceptable behavior without negative examples to identify the threshold at which actions become unacceptable. At the same time, UNSAFE recall improves from 50% to 68%, but with 41% of REASONABLE actions misclassified as UNSAFE—worse than the VLM—only baseline. This indicates that in the absence of negative examples, the VLM leans towards a brittle, overly cautious prior. Without real-world accounts of failure, the model defaults to treating any deviation from passive behavior as unsafe. Negative precedents restore balance by providing grounded evidence for when safety is truly violated, leading to more selective and meaningful UNSAFE predictions.

Agentic Counterfactual Adjudication Enhances Decision Boundary Calibration Compared to vanilla VLM+RAG, the agentic counterfactual approach maintains strong `REASONABLE` recall (53%) while improving both precision and calibration at the class boundaries. SAFE recall increases from 58% to 66%, indicating better recognition of legally sound but less aggressive decisions. `UNSAFE`-to-`REASONABLE` errors drop notably from 36% to 24%—a sign of improved caution on borderline dangerous actions. These improvements suggest that the generate–retrieve–evaluate cycle helps the model simulate plausible alternatives, weigh them with precedent, and return more robust, context-aware classifications than flat retrieval alone.

7 Limitations

While our results are encouraging, several limitations remain, and we view this work as a workshop-stage exploration rather than a production-ready system.

Spatial reasoning in VLMs. Current vision–language models struggle with fine-grained spatial reasoning (e.g., occlusions, relative lane geometry, and precise right-of-way logic). Ongoing efforts aim to improve grounding and geometric consistency via better visual grounding, structured scene representations, and spatially-aware prompting/fine-tuning [26, 31]. Our approach partially mitigates this by retrieving precedent narratives that encode spatial context in natural language, but residual failures in spatial inference persist. We expect improvements in spatial reasoning capabilities will mitigate these in the future.

Dataset and scope. Our evaluation dataset is limited in size and coverage. The crash corpus used for negative precedents comprises only 1.4k reports, and our benchmark subsamples to 1,275 action–scene pairs (425 per class). Larger and more diverse datasets—both in nominal driving and in negative incidents—are needed to stress-test generalization, rare interactions, and long-tail scenarios beyond the present scope.

Runtime and deployment. The full pipeline may be too slow for strict real-time operation with current VLMs. That said, inference times are trending down, and our method is immediately useful for offline scenario mining and policy auditing. For online use, the graph index can be made more efficient through compression and hierarchical organization (e.g., coarse-to-fine node pruning), approximate nearest-neighbor search over node embeddings, and lightweight re-ranking. These changes would preserve precedent alignment while improving retrieval latency.

8 Conclusion

We introduced precedent-guided adjudication for autonomous driving by converting logs and crashes into a unified scene–action representation and retrieving precedents with VLM+RAG. We further proposed an agentic counterfactual VLM+RAG engine that proposes alternatives, retrieves precedents for each, and adjudicates across outcomes.

On a balanced nuScenes benchmark, VLM+RAG substantially improves calibration over a VLM-only baseline (e.g., recall on contextually preferred actions 24% $\rightarrow$ 53%), and the agentic counterfactual variant maintains these gains while further sharpening decisions near risk. Beyond AVs, precedent-guided, counterfactual adjudication can support domains that balance safety and performance under uncertainty. Future work includes scaling multimodal inputs, learning retrieval weights end-to-end, and integrating causal structure from simulators and maps.

Acknowledgments

This work was completed as part of NVIDIA Research Internship. This work also used Bridges-2 at Pittsburgh Supercomputing Center through allocation cis220039p from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] 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*, pages 11621–11631, 2020.

[2] X. Weng, B. Ivanovic, Y. Wang, Y. Wang, and M. Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024.

[3] J. Patrikar, S. Veer, A. Sharma, M. Pavone, and S. Scherer. Rulefuser: Injecting rules in evidential networks for robust out-of-distribution trajectory prediction. *arXiv preprint arXiv:2405.11139*, 2024.

[4] T. Procko. Graph retrieval-augmented generation for large language models: A survey. *Available at SSRN*, 2024.

[5] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.

[6] K. Chakraborty, Z. Feng, S. Veer, A. Sharma, W. Ding, S. Topan, B. Ivanovic, M. Pavone, and S. Bansal. Safety evaluation of motion plans using trajectory predictors as forward reachable set estimators. *arXiv preprint arXiv:2507.22389*, 2025.

[7] K. Nakamura and S. Bansal. Online update of safety assurances using confidence-based predictions. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12765–12771. IEEE, 2023.

[8] M. Itkina and M. Kochenderfer. Interpretable self-aware neural networks for robust trajectory prediction. In *Conference on Robot Learning*, pages 606–617. PMLR, 2023.

[9] A. Zanardi, A. Censi, M. Atzei, L. Di Lillo, and E. Frazzoli. A counterfactual safety margin perspective on the scoring of autonomous vehicles' riskiness. *arXiv preprint arXiv:2308.01050*, 2023.

[10] S. Roy-Singh, S. Kolekar, D. P. Bonny, and K. Foss. Fractional collisions: A framework for risk estimation of counterfactual conflicts using autonomous driving behavior simulations. *arXiv preprint arXiv:2506.07540*, 2025.

[11] B. Stoler, I. Navarro, M. Jana, S. Hwang, J. Francis, and J. Oh. Safeshift: Safety-informed distribution shifts for robust trajectory prediction in autonomous driving. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 1179–1186. IEEE, 2024.

[12] H. Weber, J. Bock, J. Klimke, C. Roesener, J. Hiller, R. Krajewski, A. Zlocki, and L. Eckstein. A framework for definition of logical scenarios for safety assurance of automated driving. *Traffic injury prevention*, 20(sup1):S65–S70, 2019.

[13] F. Sado, C. K. Loo, W. S. Liew, M. Kerzel, and S. Wermter. Explainable goal-driven agents and robots-a comprehensive review. *ACM Computing Surveys*, 55(10):1–41, 2023.

[14] K. Renz, K. Chitta, O.-B. Mercea, A. Koepke, Z. Akata, and A. Geiger. Plant: Explainable planning transformers via object-level representations. *arXiv preprint arXiv:2210.14222*, 2022.

[15] B. Jin, X. Liu, Y. Zheng, P. Li, H. Zhao, T. Zhang, Y. Zheng, G. Zhou, and J. Liu. Adapt: Action-aware driving caption transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7554–7561. IEEE, 2023.

[16] H. Ben-Younes, É. Zablocki, P. Pérez, and M. Cord. Driving behavior explanation with multi-level fusion. *Pattern Recognition*, 123:108421, 2022.

[17] J. Yuan, S. Sun, D. Omeiza, B. Zhao, P. Newman, L. Kunze, and M. Gadd. Rag-driver: Generalisable driving explanations with retrieval-augmented in-context learning in multi-modal large language model. *arXiv preprint arXiv:2402.10828*, 2024.

[18] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024.

[19] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari. nuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition ADP3 workshop*, 2021.

[20] 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*, pages 9710–9719, 2021.

[21] N. H. T. S. Administration. Crash viewer. https://crashviewer.nhtsa.dot.gov/, 2025. Accessed: 2025-08-19.

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

[23] 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.

[24] Z. Wang, S. Lan, X. Sun, N. Chang, Z. Li, Z. Yu, and J. M. Alvarez. Enhancing autonomous driving safety with collision scenario integration. *arXiv preprint arXiv:2503.03957*, 2025.

[25] J. M. Scanlon, K. D. Kusano, L. A. Fraade-Blanar, T. L. McMurry, Y.-H. Chen, and T. Victor. Benchmarks for retrospective automated driving system crash rate analysis using police-reported crash data. *Traffic Injury Prevention*, 25(sup1):S51–S65, 2024.

[26] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[27] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[28] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 8634–8652, 2023.

[29] S. Sun, X. Li, D. Di, M. Wei, L. Nie, W.-N. Zhang, D. Zhan, Y. Song, and L. Fan. Llapa: A vision-language model framework for counterfactual-aware procedural planning. *arXiv preprint arXiv:2507.08496*, 2025.

[30] LangChain. LangChain Documentation. https://python.langchain.com/, 2025. Accessed: August 28, 2025.

[31] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[32] B. Li, L. Zhu, R. Tian, S. Tan, Y. Chen, Y. Lu, Y. Cui, S. Veer, M. Ehrlich, J. Philion, et al. Wolf: Dense video captioning with a world summarization framework. *arXiv preprint arXiv:2407.18908*, 2024.

A Data Structuring Details

A.1 Structuring Negative Data from Crash Reports

Step 1: Sourcing
We use free-text crash narratives from the National Highway Traffic Safety Administration (NHTSA) crash database. These narratives describe the sequence of events leading up to and following vehicle collisions, often written in a templated but unstructured format.

Step 2: Style Normalization via Prompt-Based Conversion
NHTSA crash reports contain detailed descriptions of real-world collisions, but are written in templated, third-person language with identifiers like ”V1” and ”V2” and cardinal directions (e.g., ”eastbound”). To make them compatible with AV data and downstream retrieval, we apply a prompt-based style transfer procedure using a language model.

The model is prompted to reframe each scenario from the ego vehicle’s perspective, replacing entity tags with relational terms (e.g., ”ego vehicle,” ”car in front”) and converting absolute references into relative spatial language. It also removes irrelevant technical detail while preserving causal structure and intent. This conversion is guided by a system prompt and a set of in-context examples (provided in the appendix). This step preserves causal structure while eliminating technical jargon and irrelevant formatting.

Step 3: Scene-Action Graph Construction
Each normalized crash description is parsed into a structured *scene-action graph*. We use a constrained prompt-based extraction method to identify entities from a fixed schema of node types:

$$Nodes = \left\{ \begin{array}{c} \texttt{Map, Ego, Ego Action,} \\ \texttt{Obstacles, Obstacles Action,} \\ \texttt{Collision or No Collision} \end{array} \right\}$$

For each node, we extract a canonicalized name, type, and natural language description. If multiple nodes of the same type exist (e.g., several nearby vehicles), their descriptions are merged into a unified summary under the same node type. This ensures compact yet complete semantic representations.

In addition to storing the textual descriptions, we compute a vector embedding for each individual node using NV-EmbedQA-Mistral-7B-v2, a pre-trained embedding model optimized for retrieval tasks. This allows us to retrieve scenes based on fine-grained graph-level similarity by computing similarity over sets of node embeddings, rather than relying on holistic sentence-level encodings alone.

A.2 Structuring Positive Data from Driving Logs

Step 1: Sourcing
To complement crash reports, we extract positive driving scenarios from the nuScenes dataset, which provides sensor-rich logs of urban driving. Each scene includes ego vehicle pose, high-resolution camera footage, annotated tracks of surrounding agents, and detailed map context. However, this data is not naturally in a textual or structured narrative format and thus cannot be directly compared to processed crash reports.

Step 2: Dense Video Captioning
We begin by summarizing short temporal segments of driving using WOLF [32], a vision-to-language model designed to produce natural language descriptions from multi-camera input and associated metadata. These captions describe what the ego vehicle did, its context (e.g., intersections, obstacles), and agent interactions. We additionally append structured metadata including agent types, their positions relative to the ego vehicle, and coarse map information (e.g., whether the ego is approaching a merger or intersection).

Step 3: Scene-Action Graph Construction
Each datapoint is converted into a structured *scene-action graph* using a prompt-based extraction process, consistent with the format used for crash data. The model identifies canonical entities (e.g.,

Ego, Ego Action, Obstacles, Map) along with their attributes and relationships. To ensure spatial grounding, the forward-facing ego camera image is included as part of the prompt context, anchoring the language model's interpretation to the actual visual layout of the scene. If multiple entities of a given type are present (e.g., several vehicles or pedestrians), their descriptions are merged into a single node. Each node is associated with both a natural language summary and a learned embedding, computed independently using NV-EmbedQA-Mistral-7B-v2, to support fine-grained GraphRAG retrieval.

B Prompt Templates and Examples

This section demonstrates the conversation structure and prompt templates used in our proposed system. The following shows a complete multi-turn conversation illustrating how the system analyzes driving scenarios and provides safety assessments.

Goal You are the decision-making brain of an autonomous vehicle engaged in a multi-step reasoning process. Your task involves analyzing scenarios, actions, and their alternatives through the following structured approach:

1. Initial Action Evaluation:

- Analyze the provided action in the current scenario
- Consider the retrieved similar scenarios to inform your judgment
- Assess safety (collision avoidance, traffic rule compliance) and performance (efficiency, comfort, goal achievement)

2. Counterfactual Generation & Analysis:

- Generate physically possible alternative actions that could have been taken
- For each alternative:
  - Use retrieved scenarios to evaluate its viability
  - Consider both conventional and creative solutions
  - Assess safety and performance implications
- Repeat this process 2–3 times to explore different possibilities

3. Comparative Analysis:

- Compare all considered actions (original and alternatives)
- Use retrieved scenarios to support comparisons
- Consider trade-offs between safety and performance

4. Final Action Assessment:

- Evaluate any new proposed action against all previous insights
- Use the accumulated knowledge from retrievals and counterfactuals
- Make a final determination based on comprehensive analysis

A preferred action must:

- **Ensure safety** (collision avoidance, adherence to traffic laws)
- **Enhance performance** (efficiency, comfort, goal achievement)

Remember:

- Use retrieved scenarios actively in your reasoning
- Consider both conventional and creative alternatives
- Build upon insights from each iteration

Examples: `{in_context_example_1}`

Now begin your multi-step analysis of the scenario.

Real Data This video provides a first-person perspective of a vehicle navigating through an urban environment. Several cars are seen close to the ego vehicle throughout the drive. The ego vehicle adjusts its speed and maintains a safe distance from nearby cars, ensuring a smooth and safe progression through traffic.

Following is the meta-data: The types of agents in proximity are: ['VEHICLE' 'VEHICLE' 'VEHICLE' 'VEHICLE' 'VEHICLE'] The relative positions of these agents with respect to ego agent are: ['REAR_RIGHT' 'FRONT_RIGHT' 'LEFT' 'REAR_LEFT' 'FRONT_LEFT'] The ego vehicle is on a road with no mergers or intersections. Retrieved scenario:

Introspection A feasible alternate action is to engage in a temporary stop to assess the surroundings and re-evaluate the situation. [Detailed reasoning omitted...]

Alternate Action temporary stop to assess surroundings

Introspection The action of performing a temporary stop to assess surroundings is classified as SAFE. [Detailed analysis omitted...]

Label SAFE

Introspection A feasible alternate action is to employ a brief deceleration followed by a gentle arc to the left within the current lane. [Detailed reasoning omitted...]

Alternate Action brief deceleration followed by a gentle arc to the left within the lane

Introspection The action of brief deceleration followed by a gentle arc to the left within the lane is classified as REASONABLE. [Detailed analysis omitted...]

Label REASONABLE

Proposed Action NUDGE LEFT

Introspection ... The retrieved scenarios indicate that slight lane adjustments, such as veering to the left, are effective for maintaining safe distances from other vehicles. This action ensures safety by avoiding close proximity to nearby cars and enhances performance by maintaining a smooth flow through traffic. [Detailed analysis omitted...]

Label REASONABLE

End of conversation.

C Human Labelling Interface

We developed a custom labelling interface to collect human annotations for our driving action classification task. The interface presents human labellers with a driving scene image and a text summary of the scenario, then asks them to evaluate each of the 10 standard driving actions from our problem statement as either ”SAFE”, ”UNSAFE”, or ”REASONABLE”.

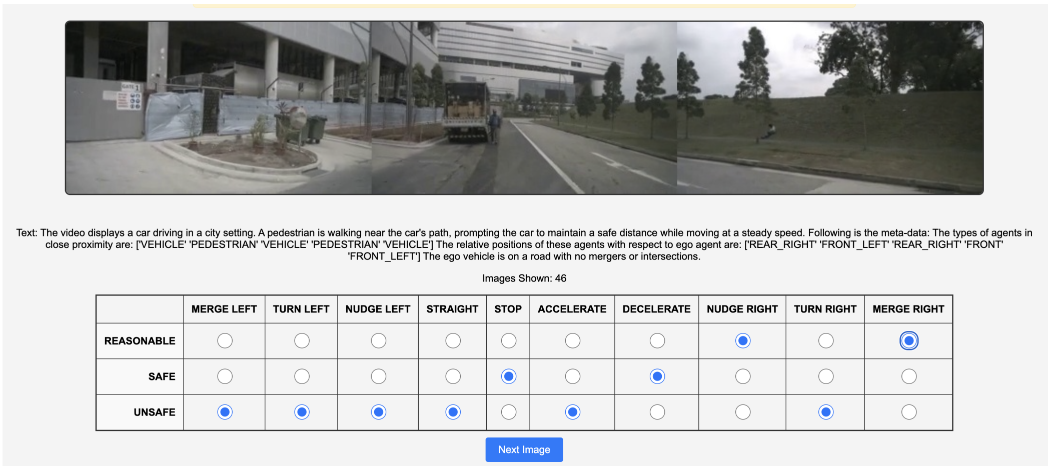

Figure 5: The custom human labelling interface. Labellers are shown a driving scene image and text description, then asked to classify each of the 10 possible ego vehicle actions as SAFE, UNSAFE, or REASONABLE based on the current driving context.

The labelling process ensures that each action is evaluated in the context of the specific driving scenario, allowing labellers to consider factors such as traffic conditions, road geometry, presence of other vehicles, and overall driving context when making their safety assessments. This human-annotated dataset serves as the ground truth for evaluating our systems.