Enhancing Transparency and Compliance in Automated Decision-Making: A Multi-Agent System Approach Using Language Models[⋆]

Ya Wang[1,2], Raja H. Seggoju[1] and Adrian Paschke[1,2]

[1]Fraunhofer FOKUS, Berlin
[2]Freie Universität Berlin

Abstract

The emergence of large language models has significantly advanced the feasibility of automated problem-solving using agents. However, despite promising results, these systems often function as "black boxes", raising concerns about their ability to comply with requirements due to opaque decision-making processes. To mitigate these issues, we introduce a multi-agent system powered by language models. This system segments the decision-making process into three agent-driven stages: proposing queries, identifying norms, and retrieving facts, while delegating final judgment to a logical reasoner. We evaluated our system in simulated driving scenarios governed by a limited set of traffic regulations. Results indicate that our approach markedly enhances compliance with decision-making accuracy and offers a more interpretable and traceable method compared to methods that rely solely on language models.

Keywords

Multi-Agent Systems, Large Language Model, Ontological Reasoning, Rule Compliance

1. Introduction

The rapid advancement and widespread adoption of Artificial Intelligence (AI) are revolutionizing various industries and reshaping human society [1, 2]. The increasing deployment of robotaxis, such as Waymo [3] in the United States and Baidu [4] in China, has garnered significant attention. However, the extensive integration of AI agents into societal frameworks demands rigorous compliance with established human societal norms [5]. Consider the specific challenge within autonomous driving [6], illustrated in Figure 1 (left). Here, a vehicle encounters a lane blockage with a solid line to the left, presenting a decision-making dilemma. The vehicle must assess whether and when it is permissible to cross the solid line to overtake the obstacle, considering the uncertain duration of the blockage. This scenario requires the AI agent to not only understand the traffic scene and its rules but also to apply these rules in making legal and reasoned decisions. Recent advancements in language models, particularly through Reinforcement Learning from Human Feedback [7, 8] (RLHF), have significantly improved AI alignment with human preferences [9]. However, these models still face logical inconsistencies and hallucinations during complex reasoning [10, 11]. Existing enhancements, including tool usage and extended contextual interactions with environment [12, 13], do not guarantee consistent and rule-compliant outcomes. Another category of approach, safety assurance, involves verifying AI systems against predefined specifications after training or deployment [14, 15, 16]. These methods are typically rule-based, providing a deterministic and transparent process, yet they face limitations in flexibility and scalability for real world runtime applications. To address this, our proposed system, illustrated in Figure 1 (right), employs multiple language model powered agents working collaboratively to derive and verify decisions. These agents actively search for formalized rules and extract relevant facts from a domain-specific ontology. An integrated logical reasoner within the workflow continuously

SGKi 2025: Scaling Knowledge Graphs for Industry Workshop, co-located with SEMANTiCS'25: International Conference on Semantic Systems, September 3–5, 2025, Vienna, Austria

ya.wang@fokus.fraunhofer.de (Y. Wang); raja.havish.seggoju@fokus.fraunhofer.de (R. H. Seggoju); adrian.paschke@fokus.fraunhofer.de (A. Paschke)

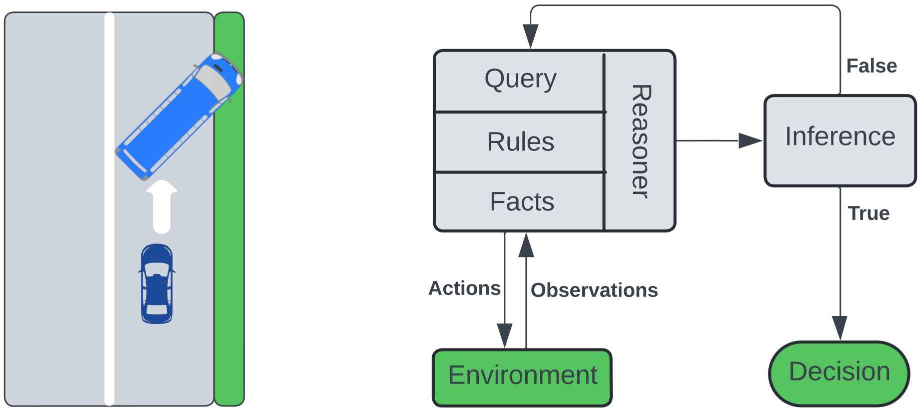


Figure 1: Decision-making process in autonomous vehicles when encountering obstructions. Left panel: An autonomous vehicle assesses the legality of crossing a solid line to bypass a lane blockage. Right panel: A breakdown of the vehicle's reasoning workflow into three agent-driven stages—action querying, rule searching, and fact retrieval—integrated with a symbolic logical reasoner for final decision-making.

assesses these inputs, ensuring efficient and effective compliance with established rules. Our principal contributions include:

- **A rule-compliant decision-making system:** Integrates multiple LLM agents, each specialized in different aspects of the decision-making process from query generation to fact retrieval, thereby streamlining the logical reasoning workload.
- **Evaluation in simulated driving scenarios:** Our system outperformed those relying solely on language models, achieving not only higher decision accuracy but also greater interpretability and transparency.

2. Related Works

Aligning the behavior of automated agents with established norms is crucial for safe deployment in real-world applications [17, 18]. Formal verification, which rigorously checks that systems conform to predefined specifications, has been extensively researched and implemented across various domains, offering significant advantages in interpretability, traceability, and determinism [19, 20]. Previous approaches [21, 22] have utilized ontology-based frameworks to verify system behavior against predefined rules and queries. More recently, Hanif et al. [23] introduced an innovative automatic regulatory framework employing a defeasible deontic logic solver, enabling vehicles to comply with rules through reasoning over driving conditions and legal contexts. Despite their deterministic and transparent reasoning processes, these methods are not yet suitable for runtime applications in complex scenarios that demand a comprehensive understanding of the environment and the ability to efficiently manage and assess a vast array of rules and facts. Recent advancements in LLMs have significantly enhanced capabilities for understanding and reasoning over unstructured data [24, 25]. Techniques such as ReAct [13] and RAG [26, 27] have been developed to mitigate the problem of hallucination by enhancing contextual interactions with language models. However, these methods still fall short in terms of interpretability and robustness, which are crucial for effective rule evaluation. To address this, Pan et al. [11] integrated LLMs with symbolic solvers, achieving improved accuracy in some specific domains. Trinh et al.'s AlphaGeometry [28] utilized LLMs to propose innovative constructs guiding symbolic solvers in solving Olympiad-level geometry problems. Our work shares a similar idea, leveraging language models to interpret scenarios, propose actions, and generate context-aware search queries. By employing ontologies to establish and derive facts and a logical reasoner to verify proposals, our method creates a more reliable and interpretable framework. This integration facilitates effective navigation of complex traffic situations, balancing flexibility with precision to ensure compliance with rules.

3. Preliminaries

3.1. Problem Statement

We define a scenario by a set $\mathcal{T}$ that includes n distinct objects $\{o_1, o_2, \ldots, o_n\}$. Each object o_i is associated with a set of properties $\mathcal{P}_i = \{p_{i1}, p_{i2}, \ldots, p_{im}\}$, where m represents the number of properties each object possesses. An automated agent operating within this scenario can perform actions from a predefined set $\mathcal{A} = \{a_1, a_2, .., a_k\}$. These actions are either explicitly or implicitly regulated by a set of legal norms $\mathcal{R}$. Each rule in $\mathcal{R}$ takes the form $\Phi \to \Psi$, stipulating that the occurrence of condition Φ mandates or prohibits the outcome of actions Ψ. The task is to determine a subset of actions $A \subseteq \mathcal{A}$ that, when executed by the agent, complies with all the regulations in $\mathcal{R}$. Specifically, the problem can be formally expressed as:

$$\forall(\Phi \to \Psi) \in \mathcal{R}, (\mathcal{T} \models \Phi) \Rightarrow (A \models \Psi),$$

where $\models$ signifies the satisfaction relation, indicating that if the scenario $\mathcal{T}$ satisfies the condition Φ, then the chosen actions A must ensure the outcome Ψ, thus adhering to the stipulated legal norms. The primary challenge in this task is the indirect evaluability of rule conditions based on the available facts. For instance, a rule for overtaking requires that the oncoming lane be clear, which involves assessing the number of vehicles visible to the ego vehicle. These essential facts are not directly available from the properties of objects; they must be inferred using domain-specific knowledge and mathematical and physical principles. Leveraging the extensive knowledge encoded in large language models can aid in making rule-compliant decisions. However, this approach risks generating hallucinations, which are unacceptable in safety-critical tasks. In contrast, traditional rule-based methods are more robust and deterministic but face significant computational challenges due to the processing of numerous rules, predicates, predicate arguments, and relevant facts. Therefore, a multi-agent decision-making system that combines the strengths of both methodologies is needed to enhance the effectiveness, robustness, and safety of the decision-making process.

3.2. Language Model-Based Multi-Agent Collaboration

The concept of "Intelligent Agents" [29, 30], developed in the late 20th century, defines autonomous entities capable of observing and acting upon an environment to achieve goals. This concept spans various domains, as in robotics, where intelligent agents perceive their environment through sensors and act through actuators [31, 32], and in reinforcement learning, where they aim to maximize cumulative rewards by taking actions in dynamic environments [33, 34]. With the advent of LLMs, LLM agents have evolved into systems capable of complex reasoning, planning, tool usage, and memory, thereby solving problems autonomously [35, 36]. An LLM functions as the system coordinator, activated via a prompt template that outlines the agent's operations and available tools. This setup enables the LLM to control the workflow and complete tasks efficiently. Each agent can be assigned a specific persona within the prompt, including information about the agent's role, personality, social characteristics, and other demographic data [37]. Complex tasks often require multiple agents working collaboratively. Chen et al. proposed ChatDev [38], which segments the workflow $\mathcal{F}$ into sequential phases $\mathcal{P}$, each comprising multiple subtasks $\mathcal{T}$ (see Equation 1). In each subtask, a dual-agent system [39] collaborates to derive solutions: one agent acts as the instructor $\mathcal{I}$ providing specific requirements, while the other acts as the assistant $\mathcal{A}$, completing the task by actively asking for additional details over multiple rounds of t.

$$\mathcal{F} = \langle \mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^{|C|} \rangle \qquad \mathcal{P}^i = \langle \mathcal{T}^1, \mathcal{T}^2, \ldots, \mathcal{T}^{|\mathcal{P}^i|} \rangle \qquad \mathcal{T}^j = \tau(\langle \mathcal{I}, \mathcal{A} \rangle^t) \tag{1}$$

The limited context length of LLMs often restricts maintaining a complete communication history among all agents and phases. To address this, agents' context memories are segmented into short-term and long-term memory [40]. Short-term memory sustains dialogue continuity within a single phase, while long-term memory preserves contextual awareness across different phases.

3.3. Normative and World Knowledge Representation

3.3.1. Legal Norms Formalization

Formalizing legal norms is essential for enabling rule-based logical reasoning. However, the inherent vagueness, abstract expressions, exceptions, and potential conflicts within established norms pose significant challenges. Westhofen et al. [41] characterize these challenges as a congruence problem between legal interpretation and system implementation. Building on legal theory elements, Chitashvili et al. [42, 43] introduce an intuitive normal form structure to represent norms, aimed at facilitating collaboration between computer scientists and legal experts. They propose a four-dimensional framework—space R_ϕ, time T_ϕ, subject S_ϕ, and action O_ϕ—to structure legal norms ϕ. R_ϕ defines where the rule applies. T_ϕ specifies the duration or activation moments. S_ϕ indicates who is bound by the rule. O_ϕ describes what is obligated, prohibited, or permitted. To adapt this framework to our use case, we introduce a fifth dimension, exceptions E_ϕ, which represents prioritized exceptional rules that override standard rules in specific cases, such as crossing a solid line. The validity of this dimension is dynamically computed based on environmental conditions. We focus mainly on formalizing obligations and prohibitions, incorporating permissions only when they provide actionable guidance under exceptional circumstances. Starting with the legal texts, we analyzed and encoded the norms into the structured formula $R_\phi \wedge T_\phi \wedge S_\phi \wedge E_\phi \rightarrow O_\phi$. This formalization into a normal form structure serves as a pivotal intermediate step. Each dimension incorporates detailed textual descriptions, which not only simplify the translation of legal norms into specific logical sentences but also facilitate more efficient searching due to the clarity of this structure.

Ontological Representation Ontology is a fundamental method for modeling domain-specific knowledge, providing a formal and explicit specification of shared conceptualizations that facilitate consistent and unambiguous knowledge exchange and management [44, 45, 46]. By incorporating ontologies, LLMs gain access to domain-specific knowledge, significantly enhancing their reliability and logical reasoning capabilities. Ontology for traffic scene modelling has attracted considerable interest due to its ability to accurately represent complex real-world situations, support automated reasoning, and maintain interpretability by humans [47, 48, 49, 50]. Typically, an ontology is seen as a knowledge base $\mathcal{KB} = (\mathcal{TB}, \mathcal{AB})$, where the TBox $\mathcal{TB}$, or Terminological Box, outlines the hierarchical structure of classes through object and data properties, axioms, and logical constructs [51]. The ABox $\mathcal{AB}$, or Assertional Box, contains the specific instances and facts derived from situational knowledge, usually represented as a graph in RDF [52]. The development of an effective ontology typically begins with a review of existing ontologies [53]. We utilize the OpenX ontology developed by ASAM [54], which offers a robust foundation with widely recognized definitions, properties, and relationships pertinent to road traffic. We further enrich this base by integrating additional concepts, relationships, and rules tailored to our specific use cases, thereby extending its applicability and effectiveness in real-world traffic scenarios.

4. Multi-Agent Rule-Compliant Decision-Making System

4.1. Rule-Compliant Decision Making Workflow

We structure the rule-compliant decision workflow into three agent-driven phases, $F = \langle P_q(T), P_r(T), P_f(T) \rangle$, where each phase handles a single task T, collaboratively aiming to create a streamlined logic program that consists of queries, facts, and rules for efficient processing by a logical reasoner. The output from the reasoner provides iterative feedback, ensuring that decisions conform strictly to established norms. As illustrated in Figure 2 , the initial phase involves two agents that interpret the traffic scene and suggest an action, such as an overtaking maneuver denoted by the query

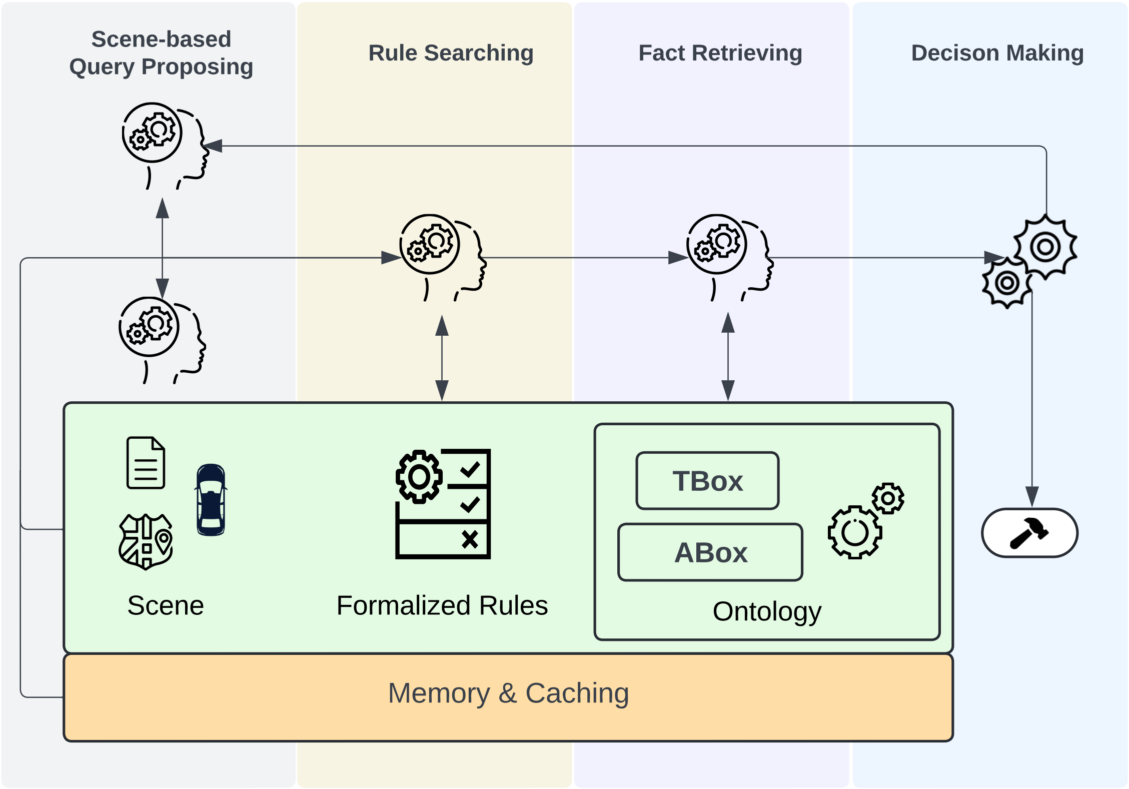


Figure 2: Visual representation of the interaction and collaboration patterns among multiple agents within a rule-compliant decision-making framework.

$Overtake(X)$, through mutual communication. The extracted scene features and the proposed query are then processed in the second phase by a semantic search agent, which identifies applicable rules for the case from the perspective of a legal expert. These rules are evaluated based on the available facts; any missing or unknown facts are forwarded to the third phase. In this phase, agents consider the evaluation context and the predicates of facts to generate SPARQL queries. These queries are executed on the ontology to retrieve necessary information through reasoning. The logical reasoner then processes the queries, rules, and facts to produce a decision, which either reenters the loop for further refinement or stands as the final decision. For more details about our algorithm, we refer to Appendix A.1.

4.2. Data Mapping and Storage

To connect different components within a system, we have integrated three distinct types of data streams. The first type encompasses environmental data sourced from vehicles for environment perception and communication, such as traffic conditions and road infrastructure. This data is assumed to be accessible via the Controller Area Network (CAN) from Electronic Control Units (ECUs) within the vehicle. The second type involves semantic data, which is understandable and executable by ontologies and logical reasoners. The third type comprises natural language, generated by LLMs to provide instructions or answers. The environmental data, characterizing a traffic scene as key-value pairs, lacks the semantic meaning required for direct evaluation of traffic rule predicates, such as $hasOncomingTraffic(X, Y)$. To enable the ontology to infer new facts, such as spatial relationships and object counts, we map the environmental data to semantic data using the OpenX ontology [54]. To better suit our requirements, we reduce its scope and then extend it to maintain its utility while optimizing query performance. The refined version includes a total of 396 axioms, along with 113 classes, 35 object properties, 6 data properties, and 2 SWRL rules within the TBox. Specifically, we use the Owlready2 [55] library to map environmental data into the ABox, making it accessible for a variety of SPARQL queries. In the first phase, only some predefined basic facts are available in the ABox. During the rule search and evaluation in the subsequent phases, more facts and rules are added, providing the rationale for the logical reasoner. The agents in the first phase are endowed with long-term memory, which allows them to adjust their strategies for proposing actions. Other agents possess short-term memory, prompting them to specialize in their own tasks. Generated SPARQL queries and facts are cached, making them accessible to each agent via its interface to boost system performance.

4.3. Module Implementation

Scene-based Query Generation In the first phase, basic facts are available from the ontology and represented as a list of predicates that characterize the traffic scene. We employ a dual-agent system to complete the scene interpretation and action proposal. This system follows an instruction-following cooperation model, which has been shown to advance the progression of productive communications and achieve meaningful solutions [39]. The instructor agent initiates instructions, guiding the discourse toward the completion of the task, while the assistant agent adheres to these instructions and responds with appropriate solutions.

$$C(I, A) = \langle I \to A, A \leftarrow I \rangle_{\text{loop}} \tag{2}$$

We prompt the instructor agent to describe the scene based on the available predicates and the feedback from the solver if available. The assistant agent is then instructed to interpret this scene and propose a driving action. To reduce communicative hallucination [38], we encourage the assistant to actively seek more facts from the instructor before delivering a final response. They engage in a multi-turn dialogue C, working cooperatively until they achieve consensus, ultimately leading to the completion of the task.

Rule Formalization and Search Working closely with legal experts, we gather German traffic rules from written legislation, legal precedents, and court decisions. We then analyze these rules and convert them into a normal form structure [42] with detailed descriptions across five dimensions. To translate the rules into executable programs, we represent them in predicate logic, limiting them to a maximum of two arguments and avoiding explicit quantifiers to maintain simplicity and coherence (see examples in Appendix A.2). This formalization enables efficient querying within the description logic-based ontology and ensures reasoning through a Prolog-based solver [56]. In total, we formalized 25 rules of prohibitions, obligations, and exceptions for our traffic scenarios. Benefiting from the clear and more implementable representation of the normal form structure, the translation into predicate logic is semi-automated by prompting a language model with logical syntax and legal terms from the ontology, followed by a thorough review. The search for related rules is carried out by a semantic search agent, which maps rules into an embedding space using text embedding models. The agent then queries the rules based on proposed actions and key features extracted from the traffic scene.

Fact Retrieval SPARQL Query generation connects common-sense knowledge from LLMs with domain-specific ontology expertise. While other studies and applications [57, 58, 59] have used LLMs to generate SQL queries by providing syntax, schema, and examples, our approach follows a similar principle, guiding LLMs step-by-step through SPARQL syntax and structure. In our application, we explored using rule context and ontology segments for query generation. We propose three methods for query generation in predicate evaluation for a rule, each providing a different level of flexibility and contextual information.

1. **Zero-Informed**: This method focuses on unary predicates, specifically designed for class hierarchical reasoning, characterized by a invariable and consistent query structure. It generates queries aimed at searching for instances that belong to the class required by the rule evaluation.
2. **Rule-Informed**: This method generates queries based on the context of the evaluated rule, which can be answered by the ontology reasoning. For example, given the context of the rule that states a solid lane marking on the left that connects to the ego lane and requires generating a query

about the predicate *LeftConnectedTo(X, Y)*, this method would incorporate the information about X as the ego lane type and Y as the solid line type into the query construction. This method limits the range of possible answers derived from the ontology.

3. **Ontology-Informed**: This method targets queries that can not directly inferred from the ontology. It incorporates additional ontological information, including comments about the predicate and available predicates, to construct the query.

These three query generation methods offer increased context and flexibility but reduced semantic correctness. In our work, most queries use the first two methods, while only two queries use the third (see examples in Appendix B.1).

Logical Reasoning We use Prolog [56] solver for legal reasoning to verify the proposed action. Prolog is a declarative language derived from a subset of first-order logic, operating under the Closed World Assumption (CWA) to maintain decidability. In our pipeline, any facts or rules not retrieved from prior agents are considered false. The rule compliance of the proposed action is validated using available information through the Prolog query with backward chaining. Once the action is deemed consistent, it is added to the knowledge base. Subsequently, other driving actions, which are regulated by related prohibition and obligation rules, are iteratively queried and derived. As Prolog doesn't inherently support deontic logic reasoning for different modalities of rules, we devise a mechanism to manage exceptional rules as a priority when assessing the current scene for possible rule exceptions. We then assign truth values to the "exception"dimension of corresponding rules. An action is deemed rule-compliant when it aligns with both prohibition and obligation rules.

5. Experiment

Providing explicit rules as extended context in prompts is a common method for managing the behavior of language models. In contrast, our method restricts its output by employing ontology-based fact retrieval to evaluate rules via query generation. While agent-based language models [60, 61] have demonstrated higher accuracy in decision-making, our experiment aims to assess whether our approach delivers more accurate, reliable, and traceable rule-compliant decisions compared to rule-prompting methods.

5.1. Dataset

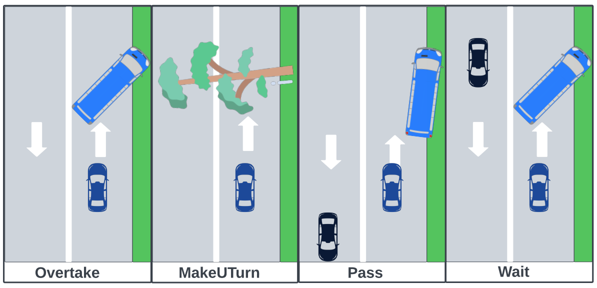


Figure 3: Driving scenarios classified by primary decision types.

Despite the availability of diverse datasets in autonomous driving, there are few designed specifically for rule compliance assessment. In our experiment, we create a compact synthetic dataset consisting of 60 randomly generated two-lane road scenarios. Each scenario contains detailed information about traffic participants, road infrastructure, environmental factors, and applicable traffic rules, all stored in key-value pairs in JSON format. We label each scenario with rule-compliant driving decisions involving primary and secondary actions. The primary action space A_p (see Figure 3), intended for escaping dilemma situations, comprises maneuvers targeting an inoperative vehicle X: A_p={Overtake(X), MakeUTurn(X), Pass(X), Wait(X)}, governed by rules of prohibition, obligation and exception. "Pass" is distinct from "Overtake" as it occurs within the same lane. The secondary action space A_s comprises A_s={SpeedLimit(X), KeepSafeLateralDistance(X), LaneChangeTo(X), Cross(X)}, where X indicates specific values or road elements. These actions adhere to obligation rules and depend on primary actions, such as maintaining lateral distance during overtaking. These scenarios are equally distributed across four classes based on primary actions, where each class encompasses various scenarios. These include differences in the number, size, type, location, and speed of vehicles, as well as road markings, traffic signs, weather conditions, and congestion levels, all of which may influence driving actions. Our data generation follows principles of flexibility, extensibility, and scalability through random sampling for variables considering physical and rule constraints, supplemented by thorough manual review. While our dataset, programmatically configured, may not fully reflect real-world driving complexities, it aims to test our hypotheses and serve as an example for collaboration between legal experts and computer scientists in dataset creation.

5.2. Baselines and Metrics

Language models primarily trained on general natural language corpora [62], have limited understanding of less common key-value data structures. While targeted instructions can help, the results may not always be reliable. To maximize the reasoning potential of our baseline models, we implement a rule-based approach that programmatically generates narratives for each scenario from key-value pairs, enabling language models to derive rule-compliant actions from these textual descriptions. We use Few-shot-CoT [63, 64] as our baseline method, enabling complex reasoning by prompting detailed intermediate reasoning steps. To guide decision-making, we provide four representative examples that follow different reasoning paths involving traffic rules. For a fair comparison of reasoning abilities, we exclude the rule search part by specifying applicable traffic rules as natural language text in the baseline models and as logical forms in our method. We use GPT-3.5-Turbo and GPT-4o as language models for both methods. To evaluate the accuracy of derived actions, we use precision P, recall R, and the $F1$ score as our metrics in each scenario. A True Positive is counted when both the predicate and its argument are correctly predicted. Subsequently, we calculate the average and standard deviation for each metric across all classes and methods.

5.3. Results

As presented in Table 1, our method, NeSy-LAD (Neuro-Symbolic Legal Guidely Automated Decision-making System), outperforms the Few-shot-CoT baseline models across GPT-3.5 and GPT-4o, with significant gains in precision, recall, and F1 Score. The NeSy-LAD with GPT-4o achieved the best performance, exhibiting a 13.75% increase in precision, a 7.25% increase in recall, and a 10.54% increase in F1 score compared to the Few-shot-CoT with GPT-4o. Remarkably, even when utilizing GPT-3.5, NeSy-LAD still outperforms the Few-shot-CoT with GPT-4o by 0.75%, which indicates that integrating ontology-based fact retrieval with a symbolic solver offers a significant advantage over the approach directly prompting rules for the decision-making. GPT-4o generally outperforms GPT-3.5 across both methods. Notably, in the Few-shot-CoT approach, GPT-4o demonstrates a recall that is 12.5% higher than the GPT-3.5 implementation.

Table 1
Comparison of performance between Few-shot-CoT and NeSy-LAD across GPT models.

Metric	Few-shot-CoT GPT-3.5	Few-shot-CoT GPT-4o	NeSy-LAD GPT-3.5	NeSy-LAD GPT-4o
Precision	79.64±3.29%	82.08±5.45%	85.50±3.01%	**95.83±1.85%**
Recall	73.83±2.85%	86.33±2.47%	84.31±3.81%	**93.58±4.39%**
F1 Score	76.13±3.30%	84.15±7.91%	84.90±6.82%	**94.69±6.24%**

To explore the performance of these two methods across various scenario categories, we plotted the average F1 scores for both in Figure 4 (left). Our method demonstrates higher F1 scores in the “Wait”, “Pass”, and “MakeUTurn” classes, with the exception of the “Overtake” class. The Few-shot-CoT approach tends to favor the “Overtake” action to escape these challenging scenarios, whereas our method relies more on the rule evaluation. Particularly, in the “Wait” and “Pass” classes, which involve a higher number of rules and predicates (Figure 4, left), our method achieved very high F1 scores.

The errors in our approach for the class of “Overtake” stem mainly from incorrect query generation, including both semantic and syntactic inaccuracies. Semantic errors are more common, largely due to the misuse of the rule context and available predicates from the ontology for query generation. The errors in the Few-shot-CoT approach arise from several sources: omitted traffic rules, inconsistencies between the reasoning process and conclusions, and an inability to accurately capture the semantic meaning of the rules. For example, the action ”Wait” is mostly regulated implicitly by the prohibition of ”Overtake” or ”MakeUTurn” in traffic rules, which Few-shot-CoT may not capture. With respect to interpretable and traceable reasoning, our approach provides detailed insights into evaluated rules and generated queries for unary and binary predicates (Figure 4, right), offering a more reliable and trustworthy process than Few-shot CoT.

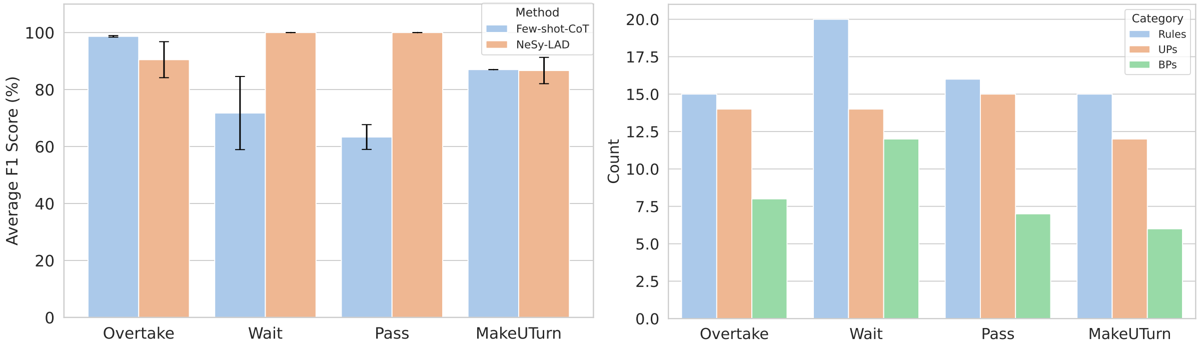


Figure 4: Comparison of average F1 scores (left), and distribution of the average number of evaluated rules and generated queries for unary (UPs) and binary (BPs) predicates across all classes (right).

5.4. Discussion

As demonstrated in our experiments, compared to providing rules as the context to language models, our method, which segments the decision process into three agent-driven phases and delegates the final reasoning to a symbolic logical reasoner, exhibits significant advantages in rule-compliant decison making accuracy, transparency, and interpretability. Despite these achievements, we acknowledge certain limitations in our experiment and approach. First, we tested only a limited set of rules and predicates, which may not fully represent the language model’s query generation capabilities. To scale the approach, future work should explore more efficient mechanisms that utilize various contexts or consider fine-tuning the model for SPARQL query generation. Secondly, our system is heavily dependent on formalized rules in an executable format. However, we argue that legislation regarding automated agents should address both implementability for systems and interpretability for humans, which calls for collaboration between computer scientists and legal experts. This lays the groundwork for the safe and lawful deployment of agents in real-world applications. Our LAD system integrates traffic rules explicitly into a symbolic solver, providing an interpretable and traceable decision-making process for humans. This setup allows for flexible and rapid rule updates as regulations evolve. Additionally, our system’s modular design allows different modules to be replaced with varying techniques. For instance, the ontology query part can be replaced with knowledge graph embeddings, and the symbolic solver can be substituted with a neural-based reasoner. Though our system was originally designed for decision-making in autonomous driving, it can be adapted to other domains requiring scene recognition and rule compliance. In conclusion, our approach provides a scalable and adaptable framework that can serve as a foundational solution for a wide range of applications, enabling reliable and interpretable rule-compliant decision making in diverse contexts.

6. Conclusion

In this paper, we present a framework that combines language model agents with a symbolic logical reasoner for rule-compliant decision-making in autonomous driving. Our approach regulates automated agents with formalized rules, providing an adaptable solution for safer and more interpretable automated decision-making. Despite its effectiveness in traffic scenarios, our system has limitations when dealing with complex rules and probabilistic reasoning. Future work should explore more implementable rule formats and develop more scalable query methods.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4o in order to: Grammar and spelling check. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. J. Bryson, The future of ai's impact on society, MIT Technology Review (2024). URL: https://www.technologyreview.com/2019/12/18/102365/the-future-of-ais-impact-on-society/, accessed: 2024-07-20.

[2] J. Karsten, D. M. West, How artificial intelligence is transforming the world, Brookings (2024). URL: https://www.brookings.edu/articles/how-artificial-intelligence-is-transforming-the-world/, accessed: 2024-07-20.

[3] BBC Newsround, Robotaxis: Driverless cars arriving in us cities, BBC Newsround (2024). URL: https://www.bbc.co.uk/newsround/68777656, accessed: 20-July-2024.

[4] CNN, China's baidu apollo go robotaxi, CNN (2024). URL: https://edition.cnn.com/2024/07/18/cars/china-baidu-apollo-go-robotaxi-anxiety-intl-hnk/index.html, accessed: 20-July-2024.

[5] V. Conitzer, W. Sinnott-Armstrong, J. S. Borg, Y. Deng, M. Kramer, Moral decision making frameworks for artificial intelligence, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 31, 2017.

[6] Y. Wang, M. Grabowski, A. Paschke, An ontology-based model for handling rule exceptions in traffic scenes, in: Proceedings of the International Workshop on AI Compliance Mechanism (WAICOM 2022), 2022, pp. 87–100.

[7] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al., Training a helpful and harmless assistant with reinforcement learning from human feedback, arXiv preprint arXiv:2204.05862 (2022).

[8] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, D. Amodei, Deep reinforcement learning from human preferences, Advances in neural information processing systems 30 (2017).

[9] OpenAI, Gpt-4v(ision) system card, https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023. Accessed: 2023-07-21.

[10] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, P. Stone, Llm+ p: Empowering large language models with optimal planning proficiency, arXiv preprint arXiv:2304.11477 (2023).

[11] L. Pan, A. Albalak, X. Wang, W. Y. Wang, LOGIC-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning, Findings of the Association for Computational Linguistics: EMNLP 2023 (2023) 3806–3824. doi:10.18653/v1/2023.findings-emnlp.248. arXiv:2305.12295.

[12] B. Paranjape, S. Lundberg, S. Singh, H. Hajishirzi, L. Zettlemoyer, M. T. Ribeiro, Art: Automatic multi-step reasoning and tool-use for large language models, arXiv preprint arXiv:2303.09014 (2023).

[13] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, arXiv preprint arXiv:2210.03629 (2022).

[14] H. Bhuiyan, G. Governatori, A. Bond, A. Rakotonirainy, Traffic rules compliance checking of automated vehicle maneuvers, Artificial Intelligence and Law 32 (2024) 1–56.

[15] L. Dennis, M. Fisher, M. Slavkovik, M. Webster, Formal verification of ethical choices in autonomous systems, Robotics and Autonomous Systems 77 (2016) 1–14.

[16] B. Mermet, G. Simon, Formal verication of ethical properties in multiagent systems, in: 1st Workshop on Ethics in the Design of Intelligent Agents, 2016.

[17] M. L. Kubica, Autonomous vehicles and liability law, The American Journal of Comparative Law 70 (2022) i39–i69.

[18] H. Prakken, On the problem of making autonomous vehicles conform to traffic law, Artificial Intelligence and Law 25 (2017) 341–363.

[19] P. Pauwels, S. Zhang, Semantic rule-checking for regulation compliance checking: an overview of strategies and approaches, in: 32rd international CIB W78 conference, 2015.

[20] B. Zhong, C. Gan, H. Luo, X. Xing, Ontology-based framework for building environmental monitoring and compliance checking under bim environment, Building and Environment 141 (2018) 127–142.

[21] P. Morignot, F. Nashashibi, An ontology-based approach to relax traffic regulation for autonomous vehicle assistance, IASTED Multiconferences - Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, AIA 2013 (2013) 122–129. doi:10.2316/P.2013.793-024. arXiv:1212.0768.

[22] D. Guo, E. Onstein, A. D. La Rosa, A semantic approach for automated rule compliance checking in construction industry, IEEE Access 9 (2021) 129648–129660.

[23] H. Bhuiyan, G. Governatori, A. Bond, A. Rakotonirainy, Validation of autonomous vehicle overtaking under queensland road rules, Proceedings of the 6th International Joint Conference on Rules and Reasoning (2022).

[24] J. Huang, K. C.-C. Chang, Towards reasoning in large language models: A survey, arXiv preprint arXiv:2212.10403 (2022).

[25] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, arXiv preprint arXiv:2303.18223 (2023).

[26] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[27] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, Retrieval-augmented generation for large language models: A survey, arXiv preprint arXiv:2312.10997 (2023).

[28] T. H. Trinh, Y. Wu, Q. V. Le, H. He, T. Luong, Solving olympiad geometry without human demonstrations, Nature 625 (2024) 476–482. doi:10.1038/s41586-023-06747-5.

[29] K. M. Tolle, Intelligent Agents, Springer US, Boston, MA, 1997, pp. 275–290. URL: https://doi.org/10.1007/978-1-4615-5915-3_23. doi:10.1007/978-1-4615-5915-3_23.

[30] M. Wooldridge, N. R. Jennings, Intelligent agents: Theory and practice, The knowledge engineering review 10 (1995) 115–152.

[31] L.-J. Lin, Reinforcement learning for robots using neural networks, Carnegie Mellon University, 1992.

[32] S. J. Russell, P. Norvig, E. Davis, Artificial intelligence: a modern approach, Prentice Hall series in artificial intelligence, 3rd ed ed., Prentice Hall, Upper Saddle River, 2010.

[33] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement learning: A survey, Journal of artificial intelligence research 4 (1996) 237–285.

[34] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[35] J. Ruan, Y. Chen, B. Zhang, Z. Xu, T. Bao, G. Du, S. Shi, H. Mao, Z. Li, X. Zeng, R. Zhao, TPTU: Large Language Model-based AI Agents for Task Planning and Tool Usage, 2023. doi:10.48550/ARXIV.2308.03427, version Number: 3.

[36] Significant Gravitas, AutoGPT, ???? URL: https://github.com/Significant-Gravitas/AutoGPT.

[37] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, et al., A survey on large language model based autonomous agents, Frontiers of Computer Science 18

(2024) 186345.

[38] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong, J. Xu, D. Li, Z. Liu, M. Sun, ChatDev: Communicative Agents for Software Development, 2024. URL: http://arxiv.org/abs/2307.07924, arXiv:2307.07924 [cs].

[39] G. Li, H. Hammoud, H. Itani, D. Khizbullin, B. Ghanem, Camel: Communicative agents for" mind" exploration of large language model society, Advances in Neural Information Processing Systems 36 (2023) 51991–52008.

[40] X. Zhu, Y. Chen, H. Tian, C. Tao, W. Su, C. Yang, G. Huang, B. Li, L. Lu, X. Wang, et al., Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory, arXiv preprint arXiv:2305.17144 (2023).

[41] L. Westhofen, I. Stierand, J. S. Becker, E. Möhlmann, W. Hagemann, Towards a congruent interpretation of traffic rules for automated driving-experiences and challenges, 2022, pp. 8–21.

[42] M. Chitashvili, M. Hermann, D. Sasdelli, C. Wüst, A Normal Form for Representing Legal Norms and its Visualisation Through Normative Diagrams, Proceedings of the 19th International Conference on Artificial Inteligence and Law (2023).

[43] D. Sasdelli, A. T. G. Trivisonno, Normative diagrams as a tool for representing legal systems, The Review of Socionetwork Strategies (2023) 217–231.

[44] A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220. doi:https://doi.org/10.1006/knac.1993.1008.

[45] N. Guarino, D. Oberle, S. Staab, What is an ontology?, Handbook on ontologies (2009) 1–17.

[46] S. Staab, R. Studer, Handbook on ontologies, Springer Science & Business Media, 2013.

[47] M. Hülsen, J. M. Zöllner, C. Weiss, Traffic intersection situation description ontology for advanced driver assistance, in: 2011 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2011, pp. 993–999.

[48] M. A. Mohammad, I. Kaloskampis, Y. Hicks, R. Setchi, Ontology-based framework for risk assessment in road scenes using videos, Procedia Computer Science 60 (2015) 1532–1541.

[49] M. Scholtes, L. Westhofen, L. R. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. H. Bollmann, F. Körtke, et al., 6-layer model for a structured description and categorization of urban traffic and environment, IEEE Access 9 (2021) 59131–59147.

[50] L. Zhao, R. Ichise, Y. Sasaki, Z. Liu, T. Yoshikawa, Fast decision making using ontology-based knowledge base, in: 2016 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2016, pp. 173–178.

[51] G. De Giacomo, M. Lenzerini, et al., Tbox and abox reasoning in expressive description logics., KR 96 (1996) 10.

[52] W3C RDF & SPARQL Working Group, Rdf 1.2 concepts and abstract syntax (resource description framework), https://www.w3.org/TR/rdf12-concepts/, 2025. Working Draft (last updated July 2025), RDF 1.2 suite.

[53] N. Noy, Ontology development 101: A guide to creating your first ontology, 2001. URL: https://api.semanticscholar.org/CorpusID:500106.

[54] J. Tao, A. L. GMBH, F. F. Informatik, D. AI, E. C. Research, F. Software, T. S. GmbH, et al., ASAM OpenX Ontology User Guide, 2021. URL: https://www.asam.net/standards/asam-openxontology/.

[55] J.-B. Lamy, Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies, Artificial intelligence in medicine 80 (2017) 11–28.

[56] J. Wielemaker, Swi-prolog: A comprehensive prolog implementation, https://github.com/SWI-Prolog/swipl-devel, 2024. Accessed: April 21, 2024.

[57] Z. Gu, J. Fan, N. Tang, L. Cao, B. Jia, S. Madden, X. Du, Few-shot text-to-sql translation using structure and content prompt learning, Proceedings of the ACM on Management of Data 1 (2023) 1–28.

[58] R. Sun, S. O. Arik, H. Nakhost, H. Dai, R. Sinha, P. Yin, T. Pfister, Sql-palm: Improved large language modeladaptation for text-to-sql, arXiv preprint arXiv:2306.00739 (2023).

[59] Q. Zhang, J. Dong, H. Chen, W. Li, F. Huang, X. Huang, Structure guided large language model for sql generation, arXiv preprint arXiv:2402.13284 (2024).

[60] S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou,

et al., Metagpt: Meta programming for multi-agent collaborative framework, arXiv preprint arXiv:2308.00352 (2023).

[61] Z. Ma, Y. Mei, Z. Su, Understanding the benefits and challenges of using large language model-based conversational agents for mental well-being support, in: AMIA Annual Symposium Proceedings, volume 2023, American Medical Informatics Association, 2023, p. 1105.

[62] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[63] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, L. Zettlemoyer, Rethinking the role of demonstrations: What makes in-context learning work?, 2022. arXiv:2202.12837.

[64] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.

A. Appendix A

A.1. Rule-Compliant Decision-Making Algorithm

Algorithm 1 Rule-Compliant Decision Making through LLM Agents

```
Input: TBox $TB$, scenario $T$, rules $R$, possible actions $A$
Output: Primary and secondary actions $\{A_p, A_s\}$, compliant with the rules $\{I_p, I_s\}$
Hyperparameter: Number of trails $N$, language model $LM$, symbolic solver $SL$
$A_p, A_s, I_p, I_s \leftarrow \emptyset$
$AB \leftarrow f(\ TB,\ T)$                                        ▷ Mapping data into ABox
$KB = \{\ AB,\ R\}$                            ▷ Build knowledge base with rule sets ready for query
$facts \leftarrow query(\ KB)$                                     ▷ Retrieve basic available facts
for $n = 1, \dots, N$ do                          ▷ Iterate N trials for LLM primary action proposal
    $\hat{a_p}, \hat{ft} \leftarrow\ LM_1(\ AB,\ T, facts, prompts)$      ▷ Output candidate action and scene features
    $\hat{R} \leftarrow$ SEARCH( $R, \hat{a_p}, \hat{ft},\ LM_2$)                   ▷ Search for all related rules
    for $\hat{r_i}$ in $\hat{R}, i = 1, ...d$ do
        $AP \leftarrow$ EVALUATE$(\hat{r_i}, facts)$                ▷ Identify rule predicates absent from facts
        for $p_j$ in $AP, j = 1, ...k$ do
            $q \leftarrow\ LM_3(p_j, \hat{r}_i, prompts)$                ▷ Generate queries for absent predicates
            $new\_facts \leftarrow query(\ KB, q)$          ▷ Retrieve new facts with ontology reasoning
            $facts \leftarrow facts \cup new\_facts$
        end for
    end for
    $c, a_p, i_p \leftarrow SL(facts, \hat{a}_p,\ \hat{R})$          ▷ Verify action consistency via backwards chaining
    if c is True then
        $A_p \leftarrow A_p \cup a_p$
        $I_p \leftarrow I_p \cup i_p$
        break                                              ▷ Stop loop when primary action found
    else
        UPDATE( $LMs_1, a_p$)                                  ▷ Update prompts for next iteration
    end if
end for
$A_s, I_s \leftarrow SL(facts, A_p,\ \hat{R})$                                ▷ Derive secondary actions
return $A_p, A_s, I_p, I_s$
```

We propose the Rule-Compliant Decision-Making Algorithm, which combines multiple language models with a symbolic solver to derive rule-compliant actions. As presented in Algorithm 1, it takes as input a TBox TB, a scene T represented by key-value pairs, a set of formalized rules together with their corresponding logical representations R, and possible actions A. It outputs primary A_p and secondary A_s actions that comply with the corresponding rules I_p and I_s. The process begins by mapping data to an ABox AB. Together with the formalized rules, this forms a knowledge base KB, which is then ready for querying. Basic facts about the current scene are then extracted from this knowledge base. In the first loop of N trials, the language model agents LMs collaboratively proposes a primary action $\hat{a_p}$ and identifies relevant scene features $\hat{ft}$ based on the blockage of the front vehicle. It then searches for all rules related to the proposed action and the identified features. In the second loop, for each relevant rule $\hat{r_i}$, the algorithm evaluates which rule predicates are not currently supported by the available facts. For each absent predicate, the language model generates queries, which are used to retrieve new facts from the knowledge base through ontology reasoning. Upon identifying a candidate primary action $\hat{a_p}$, the symbolic solver SL verifies the action's consistency with the rules through backward chaining. If the action is found to be compliant, the loop terminates, marking the action as the primary compliant action. If not, the process iterates, updating the prompts for the language model agents to refine the action proposal. After determining the primary action, the algorithm employs the symbolic solver to derive secondary actions that are compliant with the rules. Our algorithm reduces the workload for the symbolic solver by suggesting candidate actions, pinpointing the most relevant rules, and extracting the necessary facts though context-based query generation. This approach significantly narrows the search space for rule evaluation, while offering more interpretable and traceable results compared to purely language-based models.

A.2. Traffic Rule Formalization

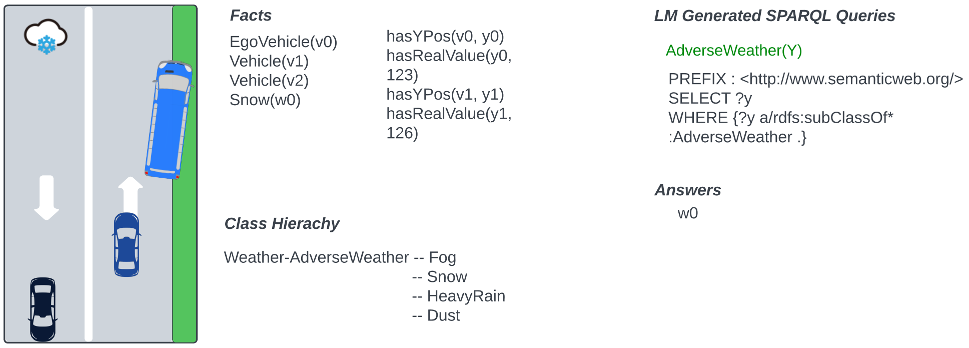
We begin by collecting traffic regulations from various sources, then analyze these rules and convert them into a normal form structure with detailed descriptions across five dimensions. At last, we employ large language models as tools to help formalize the rules in predicate logic. In total, we've formalized 25 rules, with examples shown as follows.

```
{
  "id": 5,
  "R": "Motorcycle(X), Overtake(X)",
  "T": "None",
  "S": "Driver(J)",
  "E": "None",
  "Q": "KeepSafeLateralDistance(1.50)",
  "condition": "driver overtaking a motorcycle",
  "consequence": "maintain a minimum lateral distance of 1.5 meters"
},
{
  "id": 11,
  "R": "TrafficLight(X), Red(Y), hasColor(X, Y), Vehicle(V), inFrontOf(V, J)",
  "T": "None",
  "S": "Driver(J)",
  "E": "None",
  "Q": "Overtake(V), !Pass(V), !MakeUTurn(V)",
  "condition": "driver approaching a red traffic Light",
  "consequence": "must not overtake, pass or make a U-turn"
},
{
  "id": 18,
  "R": "EgoLane(G), SolidWhiteLine(Y), LeftConnectedTo(G, Y), Vehicle(X), Inoperative(X), OnComingLane(Z),
       InFrontof(X, J), !hasOncomingVehicle(J, W), Vehicle(W), block(X, G)",
  "T": "None",
  "S": "Driver(J)",
  "E": "None",
  "Q": "Overtake(X), Cross(Y), LaneChangeTo(Z)",
  "condition": "obstacle or stationary vehicle on road partially blocking the ego lane with solid white line,
       not predictable when cleared",
  "consequence": "permitted to cross the solid white line and use oncoming lane for overtaking, if safe and no
       oncoming traffic"
}
```

B. Appendix B

B.1. Examples of Context-Based Query Generation

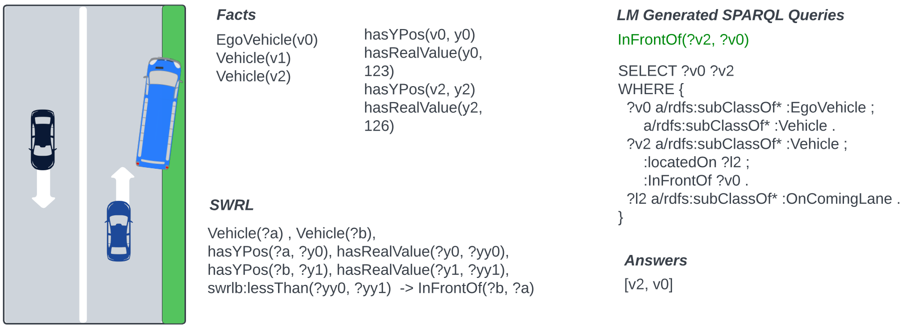
Zero-informed method. This method rarely uses any context for query generation. It primarily focuses on the evaluation of unary predicates, involving class hierarchical reasoning, which generated less diverse query structures. For example (see Figure 5), when evaluating the predicate *AdverseWeather(Y)* within a rule, this method generates a SPARQL query to the ontology, which retrieves the answer *w0* because the facts indicate that the weather is snow, which is defined as adverse weather in the ontology.



```
PREFIX : <http://www.semanticweb.org/>
SELECT ?y
WHERE {?y a/rdfs:subClassOf*
:AdverseWeather .}
```

Figure 5: Query generated by the zero-informed method.

Rule-informed method. This method generates queries based on the context of the rules being evaluated, which can be answered by ontology reasoning, as illustrated in Figure 6. Given a rule under evaluation, the language model uses predicates such as *Vehicle(X)*, *Vehicle(Y)*, *EgoVehicle(X)*, *OnComingLane(Z)*, *locatedOn(Y, Z)*, and *InFrontOf(Y, X)* within the rule to construct a detailed query for the *InFrontOf* predicate. It reduces the number of instances retrieved from the ontology.



```
Vehicle(?a) , Vehicle(?b),
hasYPos(?a, ?y0), hasRealValue(?y0, ?yy0),
hasYPos(?b, ?y1), hasRealValue(?y1, ?yy1),
swrlb:lessThan(?yy0, ?yy1)  -> InFrontOf(?b, ?a)
```

```
SELECT ?v0 ?v2
WHERE {
  ?v0 a/rdfs:subClassOf* :EgoVehicle ;
      a/rdfs:subClassOf* :Vehicle .
  ?v2 a/rdfs:subClassOf* :Vehicle ;
      :locatedOn ?l2 ;
      :InFrontOf ?v0 .
  ?l2 a/rdfs:subClassOf* :OnComingLane .
}
```

Figure 6: Query generated by the rule-informed method.