Enhancing Large Language Models through Neuro-Symbolic Integration and Ontological Reasoning for Automotive Maintenance

Ruslan Idelfonso Magaña Vsevolodovna[a] **and Marco Monti**[b,c]

[a]IBM Client Innovation Center Italy, Segrate (MI), Italy
[b]Free University of Bozen-Bolzano (UNIBZ), Bozen-Bolzano (BZ), Italy
[c]IBM, Segrate (MI), Italy

Abstract. Large Language Models (LLMs) in safety-critical automotive applications risk non-compliance and hazardous failures due to hallucinations. We present a lightweight, auditable, and plug-and-play neuro-symbolic guardrail that enhances LLM reliability for industrial use. The framework validates outputs against a domain ontology, detects inconsistencies with a Description Logic reasoner, and guides any black-box LLM to self-correct through iterative, ontology-informed feedback—without retraining or noticeable latency. In a controlled sandbox evaluation on 82 induced hallucination cases, our system corrected over two-thirds of genuinely hallucinated answers (70.4%), showing that even a minimal ontology can substantially improve robustness. While still at prototype scale, this work provides a practical blueprint for developing safer and regulation-ready AI assistants in the mobility sector and other high-stakes domains.

Keywords: LLM, ontology, neuro-symbolic reasoning, consistency checking, knowledge representation, hallucination mitigation, automotive maintenance, trustworthy AI, validation

1 Introduction

Artificial Intelligence (AI) is becoming increasingly critical in the automotive industry, particularly within *predictive maintenance* applications. Advanced AI systems enable manufacturers and fleet operators to proactively manage vehicle health, minimize downtime, and optimize operational efficiency. However, the adoption of powerful generative AI models, such as Large Language Models (LLMs), in these high-stakes industrial contexts introduces substantial risks.

LLMs excel at natural language understanding and generation but are prone to producing *hallucinations*—outputs that are fluent and seemingly coherent yet factually incorrect or logically inconsistent [10]. In regulated domains such as automotive diagnostics, these hallucinations may cause hazardous misdiagnoses, regulatory violations, and increased maintenance costs . For instance, suggesting that a fault in a combustion engine is due to an electric stator not only misleads the technician but may also breach safety standards (e.g., ISO 26262) and compliance requirements outlined in the emerging EU AI Act. Beyond hallucinations, risks include the propagation of training-set biases, lack of explainability, and misalignment between user queries and model assumptions.

Existing techniques for hallucination mitigation include *Retrieval-Augmented Generation* (RAG) [5], *prompt engineering* [7], and fine-tuning on domain-specific datasets [1]. While effective in specific scenarios, these methods create significant engineering bottlenecks in enterprise settings: RAG pipelines depend on brittle retrieval infrastructures, prompt engineering provides no formal guarantees, and fine-tuning requires costly retraining and dataset maintenance. Such limitations make them just partially suitable for scalable deployment in safety-critical environments, especially when working with proprietary black-box LLMs.

Neuro-symbolic AI, which integrates statistical learning with symbolic reasoning [2, 4], offers a promising alternative. By embedding logical constraints into the generation loop, it enables LLMs to preserve fluency while ensuring factual and logical consistency.

Contribution. We propose a lightweight, plug-and-play neuro-symbolic framework designed as an external guardrail for LLMs in automotive maintenance. Our system cross-checks LLM outputs against a domain-specific OWL 2 DL ontology, detects inconsistencies using a Description Logic (DL) reasoner, and iteratively guides the LLM to self-correct through ontology-informed re-prompting. This process operates without access to model internals or retraining, ensuring compatibility with commercial APIs.

Our contributions are threefold: (1) a closed-loop neuro-symbolic pipeline that enforces consistency between LLM outputs and ontological knowledge, (2) a *Hallucination Induction Protocol* that systematically simulates domain-specific hallucinations to evaluate robustness, and (3) empirical evidence on 82 diagnostic cases showing that our prototype corrected over 70% of genuinely hallucinated answers.

Metaphorically, the framework resembles a *service triad*: the LLM plays the role of an improvising mechanic, the ontology acts as the official service manual specifying what is possible or impossible in engine diagnostics, and the reasoner functions as an expert inspector verifying claims and requesting revisions when necessary. This illustrates how symbolic reasoning can provide a lightweight but decisive safeguard in industrial practice. The remainder of this paper is organized as follows. Section 2 reviews the industrial context and surveys related work. Section 3 details our proposed neuro-symbolic framework. Section 4 presents our automotive case study and introduces the Hallucination Induction Protocol. Section 5 provides a thorough experimental evaluation. Finally, Section 6 discusses our findings and a roadmap for future development, before Section 7 concludes the paper.

2 Industrial Context and Related Work

Predictive maintenance has evolved significantly in industrial operations, shifting from reactive or scheduled approaches to data-driven strategies aimed at reducing downtime and enhancing reliability. Early methods often relied on rule-based systems or supervised machine learning models trained on sensor data, which proved effective for well-defined fault categories within narrow operational envelopes. However, these traditional approaches often struggle to generalize across diverse vehicle configurations or adapt to dynamic and evolving fault taxonomies.

The advent of Large Language Models (LLMs) has opened new possibilities in automotive maintenance, particularly in enabling conversational interfaces for diagnostics, intelligent ticketing systems, and dynamic explanation generation. Their ability to process and generate human-like text makes them attractive for automating tasks such as interpreting technician reports or assisting users with troubleshooting. However, the inherent statistical nature of LLMs, lacking embedded formal domain knowledge, makes them susceptible to producing *hallucinations*—outputs that are linguistically fluent but factually incorrect or logically inconsistent [10]. In regulated and safety-critical domains like automotive maintenance, such hallucinations pose significant risks.

2.1 Existing Approaches for Hallucination Mitigation

Several methodological families have been proposed to reduce the occurrence of hallucinations in large language models, each reflecting a different balance between practicality and robustness. Retrieval-Augmented Generation (RAG) represents one of the most prominent strategies, grounding responses in external sources to enhance factual reliability [5]. This approach has shown considerable promise in domains that are text-rich, yet it is dependent on the continuous curation of reliable document repositories and the design of resilient retrieval pipelines. In settings where structured ontologies rather than document corpora serve as the natural knowledge source, these pipelines may exhibit brittleness. A second strategy, often referred to as prompt engineering, attempts to guide the generative process by carefully crafting input instructions [7]. Although attractive for its low implementation overhead, this technique offers limited guarantees of correctness and remains vulnerable to producing fluent but factually inconsistent outputs. A third family of approaches focuses on fine-tuning models on domain-specific data [1]. This method can meaningfully reduce hallucinations when sufficient and representative training data are available, but it demands access to the internal model parameters and substantial computational resources, conditions that are rarely met in enterprise contexts where models are often accessed via proprietary APIs.

2.2 Neuro-Symbolic Approaches

An emerging line of research on hallucination mitigation emphasizes the integration of statistical learning with symbolic reasoning, often referred to as neuro-symbolic (NeSy) methods [2, 4]. This hybrid paradigm seeks to combine the generative flexibility of neural models with the formal rigor of symbolic systems. One direction of this research involves the use of ontological or graph-based structures to detect and repair inconsistencies in model outputs. For instance, OntoMit [6] and knowledge-graph–based repair approaches [3] demonstrate how structured knowledge can be injected into the validation loop to constrain generations within semantically permissible boundaries. Another line of inquiry, exemplified by DeepOnto [11], explores how LLMs themselves can assist ontology learning and alignment, thereby reinforcing the robustness and interpretability of semantic tasks. Complementing these individual efforts, surveys of the field [12] emphasize the growing recognition of ontology-driven consistency checking as a central pillar for building trustworthy AI.

Although these studies collectively confirm the feasibility of combining LLMs with structured knowledge resources, they also reveal practical limitations. Many of the existing systems presuppose access to internal model parameters or remain confined to small-scale research settings, limiting their direct applicability in industrial contexts. In contrast, the contribution advanced in this work is designed as a plug-and-play repair loop that operates entirely externally to the LLM. By aligning outputs with a domain ontology through iterative reasoning and re-prompting, our framework offers a pathway toward industrial deployment without sacrificing the constraints of black-box access that characterize most commercial environments.

2.3 Comparative Overview

Table 1 contrasts representative approaches with our proposed framework along key engineering dimensions.

Table 1: Comparison of hallucination mitigation approaches.

Approach	**Strengths**	**Limitations**	**Industrial Suitability**
RAG [5]	Leverages external documents; scalable in text-rich domains	Requires curated corpora; retrieval errors propagate	Medium (document-heavy domains)
Prompt Engineering [7]	Lightweight, no extra infra	Fragile, no factual guarantees	Low
Fine-tuning [1]	Tailors model to domain	Costly retraining; API restrictions	Low (when black-box APIs)
OntoMit [6]	Ontology-based validation	Limited experiments, academic scale	Medium
KG-based repair	Flexible KG integration	Complex KG construction; efficiency issues	Medium
DeepOnto	Robust ontology alignment	Focus on ontology tasks, not diagnostics	Low/Medium
Our Framework	Plug-and-play; ontology reasoning; real-time repair	Current ontology minimal; parser brittle	**High (black-box APIs, regulated domains)**

This comparative analysis highlights that, unlike many existing methods, our framework is specifically designed for enterprise integration, addressing regulatory requirements (ISO 26262, EU AI Act), latency constraints (< 1 s), and plug-and-play compatibility with commercial LLMs.

3 Methodology: Ontology-Guided Neuro-Symbolic Repair Loop

To enhance the factual reliability of Large Language Models (LLMs) in industrial automotive settings, we propose a neuro-symbolic val-

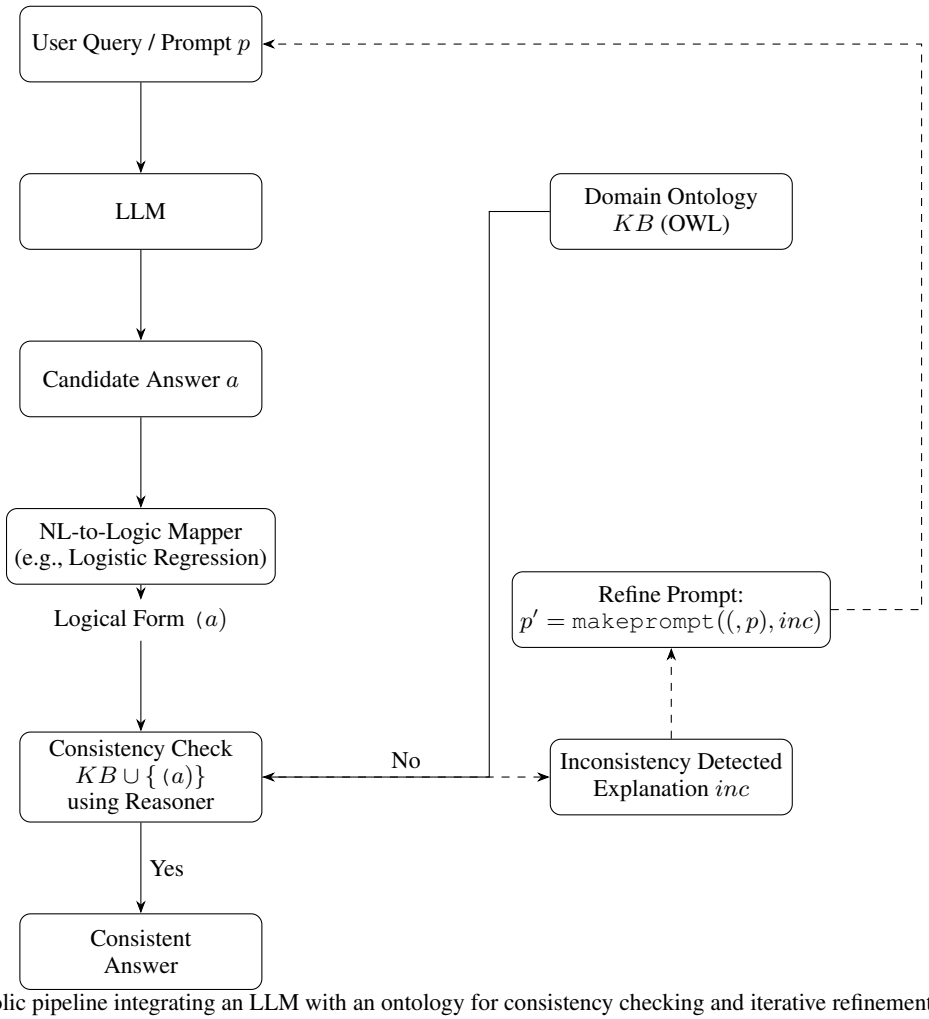

Figure 1: Neuro-symbolic pipeline integrating an LLM with an ontology for consistency checking and iterative refinement. The process involves receiving a query, generating an initial answer via the LLM, mapping this answer to a logical form, checking its consistency against the domain ontology (KB), and either accepting the answer or generating a refined prompt based on the inconsistency explanation to guide the LLM in a feedback loop.

idation framework that integrates natural language generation with formal ontological reasoning. This section outlines the system architecture, the design of the underlying ontology, and its applicability in real-world industrial deployments.

3.1 Architecture Overview

Our framework integrates statistical and symbolic reasoning into a looped architecture that validates and corrects the outputs of LLMs without requiring access to their internals. The pipeline, illustrated in Figure 1, unfolds as a sequential process. It begins with a user submitting a maintenance-related query in natural language. This input is processed by a commercial LLM (e.g., Llama 3.2 Instruct via the Watsonx.ai API), which generates a free-form textual response without direct awareness of the ontology.

The response is subsequently parsed by a lightweight natural-language-to-logic mapper trained as a supervised logistic regression classifier. The mapper extracts subject–predicate–object triples that capture the core semantic relations and represent them as structured logical atoms. Training data for this parser were generated automatically by enumerating valid and invalid logical statements from the ontology and paraphrasing them into natural language, thereby broadening coverage and reducing manual bias. The modularity of this component allows future substitution with more sophisticated extractors such as transformer-based models [**?**].

Once extracted, the triples are projected onto a domain-specific OWL 2 DL ontology, aligning the semantic content of the response with the ontology's vocabulary. The enriched ontology, comprising both terminological (TBox) and assertional (ABox) components, is then checked for logical consistency using a Description Logic reasoner such as HermiT [8]. Inconsistencies, including class disjointness violations or erroneous property assertions, are detected at this stage.

If contradictions arise, a repair prompt is generated. In the current prototype, this prompt is produced by a fixed natural-language template that incorporates all relevant axioms. For example:

> According to the domain knowledge:
> (1) A stator is an ElectricEngineComponent.
> (2) A piston is an OilEngineComponent.
> (3) Electric and oil engine components are disjoint.
>
> Please revise your answer to the question:
> *"What could cause irregular engine idle?"*

The revised prompt is resubmitted to the LLM, initiating an iterative repair loop that continues until a consistent answer emerges or

a timeout condition is reached. On average, each cycle completes in less than one second. The final stage of the pipeline delivers a validated, ontologically coherent response to the user, accompanied by status information and traceability logs. Conceptually, the architecture functions as a *compliance firewall*: the LLM generates freely, but its outputs are systematically verified and corrected against formal domain knowledge.

3.2 Ontology Design

At the core of the reasoning pipeline lies a domain-specific OWL 2 DL ontology, initially developed for experimental purposes and made available as open-source software.[1] The ontology encodes essential knowledge about automotive systems. It specifies classes representing both component and system types, such as `Engine`, `Piston`, and `Battery`. It further defines object properties that formalize relationships, for example the `CausesFailure` relation linking a component to an engine. Logical axioms enforce domain constraints, including disjointness conditions distinguishing oil-engine from electric-engine components and causal rules such as `Piston CausesFailure some OilEngine`.

These axioms are crucial for detecting inconsistencies whenever extracted triples conflict with established domain truths. Although intentionally minimalistic in its present form to facilitate pipeline testing, the ontology provides a scalable foundation for expansion. Analogously, the LLM acts as an improvising mechanic generating hypotheses, the ontology serves as the official service manual delineating permissible configurations, and the reasoner is the inspector ensuring compliance with these rules.

3.3 Independence and Industrial Integration

A defining feature of the framework is its plug-and-play compatibility, which allows it to operate with any commercial LLM without retraining. This characteristic directly addresses enterprise constraints where model internals remain inaccessible. The architecture wraps around existing LLM APIs, ensuring modularity and eliminating the need for modification of underlying models. Its performance profile supports real-time applications, with validation loops consistently completing in less than one second, thereby making it suitable for synchronous tools such as chatbots or interactive diagnostic assistants. Hardware demands remain minimal, permitting deployment on commodity infrastructure.

Beyond technical feasibility, the system offers auditability by logging axioms, inconsistencies, and repairs, a feature that aligns with compliance requirements specified by standards such as ISO 26262 and the EU AI Act. Collectively, these properties ensure that the framework can serve as an external guardrail within industrial diagnostic pipelines, providing not only factual reliability but also explainability and regulatory alignment.

3.4 Systems Engineering Considerations

The architectural design of our framework was guided by key engineering principles aimed at ensuring feasibility and scalability in industrial environments:

- Choice of Parser: The selection of a lightweight logistic regression classifier over a larger transformer-based model was a deliberate engineering trade-off. We prioritized minimal latency and a low computational footprint—critical for synchronous applications—accepting a manageable loss in parsing robustness. This ensures that the guardrail can operate on commodity hardware without introducing significant delays.
- Ontology Design: The OWL 2 DL ontology was intentionally designed to be minimalistic yet expressive enough to capture critical domain constraints (e.g., class disjointness). This strategy ensures fast and decidable reasoning, essential for real-time validation, while providing a scalable foundation. Focusing on a compact, logic-first knowledge base avoids the maintenance overhead of large, unstructured document corpora used in RAG pipelines.
- Reasoning Scalability: The use of OWL 2 DL allows expressive yet decidable reasoning. For larger datasets, modularization or lightweight OWL profiles (EL, QL) can be adopted to guarantee polynomial-time reasoning, ensuring scalability without sacrificing correctness.

[1] https://github.com/ruslanmv/Neuro-symbolic-interaction

4 Case Study: Automotive Predictive Maintenance

To evaluate the practical feasibility of our neuro-symbolic validation framework, we conducted a controlled simulation in the domain of automotive predictive maintenance—a field where factual accuracy of AI-generated suggestions is crucial due to safety, economic, and regulatory implications. This setup was designed to test the framework's ability to detect inconsistencies in LLM-generated outputs and guide the model toward ontology-aligned corrections via symbolic reasoning.

4.1 Industrial Scenario and Diagnostic Challenge

Consider the query: *"What could cause irregular engine idle?"*, submitted to a commercial LLM via API. Conditioned by misleading associations injected through our protocol, the LLM may respond: *"An irregular idle may be caused by a malfunctioning stator or piston."*

Although linguistically fluent, this output introduces a factual inconsistency: `stator` is an `ElectricEngineComponent`, while `piston` is an `OilEngineComponent`. Since these two classes are declared disjoint in the ontology, the generated answer violates domain constraints. The inconsistency is flagged by the reasoner, triggering a repair prompt that injects the relevant axioms in natural language. The revised LLM response becomes: *"An irregular idle in a combustion engine may be caused by a faulty piston or oil pump."*, which is factually consistent.

This example illustrates the type of domain-specific hallucinations our framework targets: plausible-sounding but ontologically invalid associations between components and failure modes.

4.2 Hallucination Induction Protocol

A core difficulty in evaluating hallucination mitigation lies in the absence of standardized benchmarks. To address this limitation, we developed a Hallucination Induction Protocol designed to systematically generate test cases in which large language models are prone to errors. The protocol operates along three dimensions. The first involves targeted component confusions, where prompts are deliberately constructed to conflate electric and combustion engine components. For instance, an LLM may be encouraged to associate stators with irregular idle problems in combustion engines or to suggest that pistons could be responsible for battery failures. The second dimension concerns cross-system mismatches. In this setting, queries are framed

to link unrelated subsystems, such as asking whether a faulty alternator could cause brake fluid leakage. Finally, the third dimension emphasizes redundancy or over-specified symptoms. Here, prompts are designed to elicit verbose and contradictory responses, often mixing components that are compatible with others that are not, within the same diagnostic explanation. Each of these scenarios is anchored in violations of specific ontological axioms, including class disjointness or domain-range constraints, thereby ensuring that the contradictions produced are formally detectable by the reasoning system.

4.3 Knowledge Assumptions

The design of the experimental framework rests on a clear separation between two categories of knowledge. On one side lies the ontology knowledge, which encodes formalized truths concerning component classes, their disjointness, and causal relationships. An example is the axiom specifying that a piston can cause failures in oil engines, expressed as `Piston CausesFailure some OilEngine`. On the other side lies the knowledge embedded within the LLM itself, which is derived from extensive pre-training on technical manuals, diagnostic reports, and broader textual corpora. This statistical knowledge allows the model to capture plausible associations, such as recognizing the role of pistons in engine idling, but it lacks explicit constraints to prevent it from also proposing irrelevant components like stators in the same context. The recognition of this separation clarifies the necessity of inducing hallucinations in a controlled manner: while the LLM may contain broad associations, only a structured induction protocol ensures that contradictions against the ontological truths consistently emerge and can therefore be tested and addressed systematically.

4.4 Dataset Description

The evaluation dataset consists of 82 prompt–answer pairs simulating typical diagnostic queries. All hallucinations were explicitly induced following the above protocol to guarantee controlled and reproducible failure cases. This setting ensures that our framework is rigorously tested on scenarios where LLMs generate outputs that are linguistically plausible yet formally inconsistent with the ontology.

5 Experimental Evaluation

We conducted a structured experimental evaluation to assess the ability of our neuro-symbolic framework to correct hallucinated outputs from a large language model in the context of automotive maintenance. The experimental design simulated an industrial diagnostic environment deliberately infused with misleading information, thereby allowing us to demonstrate the operation of the full iterative pipeline with a live LLM.

5.1 Experimental Setup

The evaluation was based on a curated dataset of 82 automotive diagnostic queries, as described in Section 4.2. Each prompt was submitted to a commercial LLM under two distinct conditions. In the first, referred to as the baseline condition with ontology disabled, the LLM was allowed to respond freely without any validation. This condition established the initial rate of hallucinations when the model was exposed to misleading associations introduced through the Hallucination Induction Protocol. In the second, with ontology-enabled repair loop active, the complete neuro-symbolic pipeline was applied. Here, the initial outputs of the LLM were validated against the ontology, and the repair mechanism iteratively guided the model until a consistent answer was achieved. Out of the 82 pairs, 13 resulted in systemic failures due to parser errors, timeouts, or empty outputs, which we denote with error code E=9. These were excluded from the main analysis, leaving 69 evaluable responses, distributed as 36 in the baseline-only condition and 33 in the repair-enabled condition. The asymmetry stems from the fact that some baseline responses corresponded to systemic errors and therefore had no repaired counterpart.

5.2 Evaluation Criteria

Each evaluable response produced by the LLM was manually coded using a structured error taxonomy. Responses were considered fully correct and ontologically aligned when labeled E=1, partially correct when labeled E=0.5 (for example, when answers were incomplete, verbose, or slightly inaccurate), and incorrect when labeled E=0. Responses that failed due to systemic issues were assigned E=9 and excluded. To better capture performance trends, we introduced a weighted correctness score (WCS) defined as:

$$\text{WCS} = \frac{1.0 \times \#(E = 1) + 0.5 \times \#(E = 0.5)}{\#\text{Total Responses}}$$

This measure distinguishes between fully and partially correct answers, thereby offering a more nuanced assessment of the repair loop's effectiveness.

5.3 Paired Error-Flip Analysis

To evaluate improvements directly attributable to the repair loop, we conducted a paired error-flip analysis. This focused on the 27 cases in which baseline outputs were initially incorrect (E=0) but remained evaluable after repair. Out of these, 14 cases were corrected to fully correct (E=1), 5 improved to partially correct (E=0.5), and 8 remained incorrect. Thus, the direct correction rate was 70.4% (19 out of 27 cases), counting both full and partial improvements. The weighted correctness score for these repaired cases reached 0.61, compared to 0.00 for their baseline counterparts. This clearly confirms that the majority of hallucinated responses were successfully improved, even though not all reached full correctness.

Table 2: Summary of evaluation outcomes comparing baseline and ontology-enabled conditions.

	Baseline	Ontology-Enabled	Improvement
Evaluable Responses	36	33	–
Correct (E=1)	–	14	+14
Partial (E=0.5)	–	5	+5
Incorrect (E=0)	27	8	-19
WCS	0.00	0.61	+0.61

5.4 Residual Failures

Analysis of the eight unrepaired cases provided insights into the limitations of the current system. Four cases were attributable to parser gaps, where triple extraction failed and consequently prevented accurate ontology projection. Two cases involved over-pruned correction, in which the repair prompt inadvertently removed correct information along with the incorrect content. The remaining two cases were linked to ontology coverage limitations, where specific components or fault modes were absent from the ontology and therefore could not be

checked for consistency. These observations highlight the importance of improving parser robustness, refining repair prompt strategies, and extending ontology coverage.

5.5 Automatic Metrics

Beyond manual coding, automatic evaluation metrics were used to provide complementary perspectives. Across the 69 evaluable responses, BLEU increased from 0.124 to 0.186 after repair, indicating improved semantic overlap with reference answers. At the same time, ROUGE-1 F1 decreased from 0.612 to 0.455, and ROUGE-L F1 decreased from 0.558 to 0.438. Coherence also decreased from 0.379 to 0.286. These declines in surface-level metrics reflect the fact that repaired answers were often more concise, thereby reducing lexical overlap with references, even as factual accuracy improved. This finding underscores the limitations of standard text-similarity metrics for regulated domains where factual consistency, rather than lexical similarity, is paramount.

5.6 Variability Considerations

For comparability, the evaluation relied on a single run per prompt. It is well known, however, that LLM outputs may vary across runs. A more comprehensive analysis would require multiple runs per prompt and reporting of averaged performance statistics. This remains an important direction for future work, aiming to strengthen the statistical robustness of the evaluation and to capture variability inherent in generative systems.

6 Discussion and Roadmap

Our results demonstrate the feasibility and effectiveness of a neuro-symbolic guardrail for improving LLM reliability in industrial domains such as automotive maintenance. Even under deliberately induced misleading associations, the framework recovered factual correctness and ontological alignment in over 70% of hallucinated cases. This section discusses the industrial significance of these findings, outlines our methodological contributions, identifies current limitations, and presents a roadmap for advancing the framework toward industrial deployment.

6.1 Industrial Impact and Significance

The direct correction rate of 70.4% is not just a statistical improvement; it represents a meaningful reduction in operational risk. In an industrial context, this translates into preventing the majority of potentially hazardous maintenance actions, reducing unnecessary costs from misdiagnoses, and strengthening technician trust in AI-powered diagnostic tools. The framework's core properties directly address the needs of regulated industries:

- Trustworthiness:Contradictions are systematically detected and corrected against formal, auditable knowledge, moving beyond probabilistic guarantees.
- Explainability: Each correction is linked to explicit ontological axioms and reasoner outputs, providing a transparent rationale for why an answer was considered inconsistent. This traceability is foundational for compliance with standards such as ISO 26262 and the EU AI Act.
- Modularity and Real-Time Operation: The plug-and-play nature and sub-second validation cycles make the framework attractive for deployment as a compliance firewall, filtering LLM outputs before they reach technicians or customers without disrupting workflows.

Our analysis also showed that surface-level NLP metrics like ROUGE may decline even when factual correctness improves. This highlights the insufficiency of standard benchmarks in safety-critical domains and points to the need for domain-specific, logic-aware evaluation protocols—a key insight for the broader NeSy community.

6.2 Methodological Contributions

Beyond its applied value, this work makes several contributions to the neuro-symbolic AI community. It shows how ontology-based inconsistency detection can be transformed from a mere evaluation tool into a dynamic feedback signal that actively guides Large Language Models during live repair. Building on this idea, we introduce a Hallucination Induction Protocol, a controlled procedure for systematically provoking factual errors so that correction mechanisms can be assessed under reproducible conditions. To evaluate effectiveness, we combine manual error coding with automatic metrics, thus capturing both the semantic correctness of answers and their surface-level similarity to references. Finally, the framework demonstrates a lightweight and pragmatic architecture that balances symbolic rigor with industrial feasibility, offering a blueprint for plug-and-play integration of neuro-symbolic validation into enterprise pipelines.

6.3 Identified Limitations

Despite encouraging results, several limitations constrain immediate large-scale adoption. The ontology employed so far models only a small fragment of automotive knowledge, and scaling to industrial scenarios will require modular, fine-grained representations that incorporate temporal and probabilistic aspects. At the parsing level, the lightweight logistic regression model has proved efficient but remains brittle when faced with complex or ambiguous sentences, which explains part of the residual failures observed in our evaluation. The repair strategy, while effective, is also fragile: injecting all relevant axioms ensures coverage but often results in verbose prompts and, in some cases, unintentionally prunes correct information alongside the erroneous parts. Evaluation is further complicated by the reliance on surface-form metrics such as ROUGE and Coherence, which tend to penalize concise yet factually correct outputs; this highlights the need for domain-specific factuality measures that better reflect semantic accuracy. Finally, the current analysis is based on single-shot runs, and stronger statistical confidence will require averaging over multiple generations to capture the variability inherent in large language models. Taken together, these limitations not only mark the boundaries of the current prototype but also define a clear roadmap for scaling the framework toward industrial-grade reliability.

6.4 Conceptual Considerations

Why not query only the ontology? The ontology provides the skeleton of truths and logical constraints, but it lacks the fluency and breadth of natural language generation. The LLM is indispensable for handling the long tail of non-formalized knowledge and for producing explanations in human language. The two components are therefore complementary rather than interchangeable.

Relation to RAG approaches. Our method is not an alternative to Retrieval-Augmented Generation (RAG) but a complement. While

RAG grounds answers in external factual sources such as manuals or technical documents, our neuro-symbolic guardrail ensures that these grounded outputs remain logically consistent with domain axioms. Future hybrid systems could integrate both paradigms.

Scalability under Industrial Workloads. A natural concern is whether the framework can handle high-volume industrial scenarios, such as thousands of diagnostic queries per hour in large fleet operations. The current prototype achieves sub-second validation on commodity hardware, making horizontal scaling straightforward: multiple guardrail instances can be deployed in parallel behind API gateways. Ontology modularization further enables selective loading of subsystems (e.g., engine, brakes, electronics), reducing reasoning overhead. These properties suggest compatibility with enterprise workloads, though stress-testing at scale remains future work.

6.5 Roadmap for Industrial Scaling

Our roadmap is designed to systematically advance the framework from its current prototype stage toward industrial readiness. The first priority is to enhance the parsing component: replacing the current lightweight classifier with fine-tuned transformer-based parsers (for example, compact BERT variants) should significantly reduce failures on complex technician queries. In parallel, the ontology must be scaled to cover a broader and more realistic fragment of the domain, starting with the top fifty Diagnostic Trouble Codes (DTCs) for a major vehicle model and enriched with real-world maintenance logs to ensure both completeness and relevance.

A third avenue involves human-in-the-loop integration. By developing a technician-facing interface that allows corrections to be validated or rejected, the system can support ontology refinement and gradually improve its prompting strategies, potentially through Reinforcement Learning from Human Feedback (RLHF). At the same time, smarter prompting mechanisms are needed: rather than injecting all relevant axioms into each repair prompt, future iterations will explore selective minimization strategies [9] and reinforcement learning-based optimization to balance conciseness with coverage.

Once these components mature, we plan to pursue pilot deployment in a live diagnostic ticketing system for a small fleet. Such a pilot will provide concrete measurements of impact on key performance indicators, including First-Time-Right repair rates and average diagnosis time. Beyond the automotive setting, the same guardrail principles could be generalized to other regulated sectors such as aerospace or medical devices, where factual reliability is equally critical. In this way, the roadmap moves step by step from targeted technical refinements to full industrial scaling, offering a clear trajectory toward robust and domain-compliant neuro-symbolic guardrails.

6.6 Summary

In summary, this research advances the vision of trustworthy generative AI by combining the fluency of LLMs with the rigor of ontological reasoning. While the current prototype demonstrates effectiveness on induced hallucinations, the outlined roadmap provides a concrete pathway toward robust, scalable, and regulation-ready deployment in industrial settings.

7 Conclusion

This paper introduced a neuro-symbolic framework designed to enhance the factual reliability of Large Language Models (LLMs) in automotive predictive maintenance. By integrating an OWL 2 DL ontology and a Description Logic reasoner into an external repair loop, our approach systematically detects and corrects hallucinated outputs without requiring access to model internals or retraining.

Our experimental evaluation on 82 induced hallucination cases demonstrated that the framework successfully corrected 70.4% of erroneous responses, improving factual reliability while maintaining real-time feasibility (validation loop < 1 s). This represents not only a measurable statistical gain but also a reduction in operational risk: fewer hazardous maintenance suggestions, lower diagnostic costs, and greater technician trust in AI-powered support tools.

Beyond empirical validation, the proposed framework contributes a practical integration strategy: lightweight, auditable, and regulation-ready. Corrections are linked to explicit axioms and reasoning outputs, offering transparent justification and aligning with compliance requirements such as ISO 26262 and the EU AI Act. At the same time, we acknowledge current limitations in ontology scope, parser robustness, and prompting strategies. Addressing these gaps requires systematic scaling, as outlined in our roadmap: ontology expansion to cover Diagnostic Trouble Codes, transformer-based semantic parsing, smarter prompt generation, and technician-in-the-loop validation. Importantly, the architecture is designed for horizontal scaling across enterprise workloads, with ontology modularization enabling selective subsystem loading to ensure scalability.

In conclusion, this work advances the vision of trustworthy generative AI by combining the fluency of LLMs with the rigor of ontological reasoning. The framework provides a concrete step toward reliable, auditable, and regulation-compliant AI assistants in the automotive sector, and offers a blueprint for establishing neuro-symbolic guardrails as standard components of AI pipelines across other high-stakes industrial domains.

Acknowledgements

We thank IBM Corporation and the collaborating institutions, including the Free University of Bozen-Bolzano (UNIBZ), for supporting this research, and the anonymous reviewers for their valuable feedback which helped improve this manuscript. We are also deeply grateful to Professors Oliver Kutz and Nicolas Troquard, the second author's Ph.D. supervisors, for their insightful guidance, critical support, and inspiring mentorship throughout this work.

Acknowledgment of Generative AI Use

Generative AI tools (e.g., ChatGPT, Grammarly) were employed for language editing. In addition, the main author used them as assistive technology to compensate for visual impairment, in compliance with the UNIBZ AI Guidelines. The authors have reviewed and edited all content and take full responsibility for the final version.

References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, et al. On the opportunities and risks of foundation models. *arXiv preprint*, 2021.

[2] A. S. d'Avila Garcez and L. C. Lamb. Neurosymbolic ai: the 3rd wave. *Artificial Intelligence Review*, 56:12557–12591, 2023. doi: 10.1007/s10462-023-10503-w.

[3] T. Hartl, F. L. T. Santos, F. Lecue, and H. Stuckenschmidt. Repairing llm responses with knowledge graph-based ontology checks. *arXiv preprint*, 2024.

[4] L. C. Lamb, A. S. d'Avila Garcez, M. Gori, M. O. R. Prates, P. H. C. Avelar, and M. Y. Vardi. Graph neural networks meet neural-symbolic computing: A survey and perspective. *ACM Transactions on Computational Logic*, 23(4):1–36, 2022. doi: 10.1145/3476832.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474, 2020.

[6] G. Martino, O. Indika, and M. Theiler. Ontomit: An ontology-based approach for mitigating hallucinations in llms. In *The Semantic Web: ESWC 2023 Satellite Events*, volume 13998 of *Lecture Notes in Computer Science*, pages 127–132. Springer, 2023. doi: 10.1007/978-3-031-35714-4_11.

[7] L. Reynolds and K. McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. *arXiv preprint arXiv:2102.07350*, 2021. doi: 10.48550/arXiv.2102.07350.

[8] R. Shearer, B. Motik, and I. Horrocks. Hermit: A highly-efficient owl reasoner. In *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2008)*, 2008.

[9] N. Troquard, R. Confalonieri, P. Galliani, R. Peñaloza, D. Porello, and O. Kutz. Repairing ontologies via axiom weakening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 1981–1988, 2018.

[10] H. Zhang, L. Li, B. Yan, L. Chen, J. Han, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint*, 2024.

[11] Y. Zhang, Y. Luo, and W. Hu. Deeponto: A toolkit for ontology learning, matching, and evaluation. *arXiv preprint arXiv:2306.03128*, 2023.

[12] Y. Zhang, Y. Cai, X. Zuo, X. Luan, K. Wang, et al. The fusion of large language models and formal methods for trustworthy ai agents: A roadmap. *arXiv preprint*, 2024.