Journal Title
XX(X):2–26
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

CORE-LLM-Bench: A Controlled Neurosymbolic Benchmark for Ontology-Grounded Reasoning in Large Language Models

Julie Loesch[1] and Sara Falahatkar[1] and Nicole Kilk[2] and Rishabh Jakhar[1] and Raghava Mutharaju[3] and Michel Dumontier[1] and Remzi Celebi[1]

Abstract

Large language models (LLMs) are increasingly applied to tasks requiring symbolic and knowledge-intensive reasoning, yet reasoning over realistic ontological structures remains less systematically studied. We present CORE-LLM-Bench, a controlled neurosymbolic benchmark built from description-logic reasoner entailments and associated symbolic explanations. The finalized benchmark contains 9,048 ontology-grounded questions across Family, Pizza, and OWL2Bench datasets. It includes binary and open-ended questions. Each question is provided in three representations: natural-language, formal OWL/SPARQL, and abstracted natural-language representations. Each question is characterized by ontology-context depth and minimum explanation complexity. We evaluate GPT-5 mini, Gemini 2.5 Flash-Lite, and Qwen3-30B-A3B-Instruct on the finalized benchmark using Answer Exact Match, Answer F1, confidence–correctness alignment, and hallucination rate. Results show that binary questions consistently yield higher answer performance than open-ended questions. Broader ontology contexts generally coincide with lower performance. Formal-symbolic inputs show the strongest performance overall, particularly for open-ended questions, while the effect of semantic abstraction varies by ontology, model, and task. These results demonstrate the utility of CORE-LLM-Bench for diagnosing how neural language models interact with symbolic knowledge under controlled reasoning conditions.

Prepared using sagej.cls

Keywords
Neurosymbolic Artificial Intelligence, Benchmark, Ontology Reasoning, Large Language Model

Introduction

Large language models (LLMs) (Brown et al., 2020) have demonstrated strong performance across a wide range of natural-language tasks, including question answering, text generation, and knowledge-intensive inference (Kau et al., 2024). Despite these advances, their reliability on tasks requiring explicit symbolic reasoning remains uncertain. High question-answering accuracy alone does not establish whether a model has successfully used the underlying logical structure. It may instead rely on lexical associations, familiar terminology, or other representational cues. LLMs are also known to produce hallucinations, i.e., outputs that are factually incorrect or logically inconsistent (Chen et al., 2025), while their reasoning processes are often difficult to interpret (Pan et al., 2024). Evaluating symbolic reasoning in LLMs therefore requires benchmarks that provide verifiable ground truth while systematically controlling the properties and representation of the reasoning problems.

This challenge is particularly relevant in the context of ontology-based reasoning. Ontologies provide explicit, structured representations of domain knowledge and support logical inference through symbolic reasoners. They are widely used in knowledge-intensive applications where correctness, consistency, and explainability are essential. Standard ontology reasoners can provide sound and complete inferences within well-defined logical frameworks, but they are less suited to handling noisy, incomplete, or natural-language inputs (Ji et al., 2011). In contrast, LLMs offer flexible natural-language interaction and broad pretrained knowledge, but do not provide the same guarantees of logical consistency. This contrast motivates the neurosymbolic perspective (d'Avila Garcez et al., 2019; Besold et al., 2017), which seeks to combine the flexibility of neural models with the rigor and interpretability of symbolic reasoning.

Existing reasoning benchmarks have made important progress in evaluating LLMs on formal and semi-formal reasoning tasks. Benchmarks such as ProofWriter (Tafjord et al., 2020), ProntoQA (Saparov and He, 2023), and LogicBench (Parmar et al., 2024) test models on rule-based, proof-oriented, and logical inference settings. More recently, DL-ReasonSuite (Oluçoğlu and Bursa,

[1]Department of Advanced Computing Sciences, Maastricht University, Netherlands and [2]University College Maastricht, Maastricht University, Netherlands and [3]Mehta Family School of Data Science and AI, IIT Palakkad, India

Corresponding author:
Julie Loesch
Email: julie.loesch@maastrichtuniversity.nl

2026) has focused specifically on reasoning in description logic (DL), and covers tasks such as consistency checking, subsumption, instance checking, entailment-aware query answering, and natural-language-to-OWL translation. However, existing benchmarks have several limitations. Some rely on synthetic or templated environments, while others focus on canonical DL reasoning and translation tasks. They provide limited support for analyzing how LLMs behave over verifiable ontology-derived entailments of varying explanation complexity. As a result, they do not fully capture the challenges posed by ontology-grounded reasoning under varying levels of explanation-based reasoning complexity, abstraction, and answer format, where models must handle hierarchical relations, multi-hop dependencies, logical entailment, reduced domain cues, and open-ended answer generation.

To address this gap, we introduce CORE-LLM-Bench, a controlled neurosymbolic benchmark for evaluating LLMs on verifiable ontology-grounded reasoning tasks. CORE-LLM-Bench derives benchmark instances from entailments produced by a symbolic reasoner, providing independently verifiable answers together with reasoner-generated explanation structures. The benchmark evaluates two complementary tasks, binary verification and open-ended answer generation, and systematically varies ontology-context depth, knowledge representation, semantic abstraction, and explanation-based reasoning complexity. It contains natural-language, formal OWL/SPARQL, and abstracted natural-language representations. This enables systematic analysis of whether LLM performance depends on explicit symbolic structure, meaningful domain terminology, or broader contextual information. In this way, the benchmark connects neural and symbolic considerations directly: neural language models constitute the evaluated reasoning systems, while symbolic ontologies, reasoner-derived entailments, and explanation structures define the knowledge, ground truth, and diagnostic dimensions of the evaluation.

To guide this study, we formulate the following research questions:

1. How does knowledge representation affect LLM performance on ontology-grounded reasoning tasks?
2. How do ontology-context depth and explanation-based reasoning complexity affect LLM reasoning performance?
3. To what extent does semantic abstraction affect LLM performance when domain-specific lexical cues are removed?
4. How does task formulation, specifically binary verification versus open-ended answer generation, affect answer correctness, confidence–correctness alignment, and hallucination behavior?

The main contributions of this paper are:

1. **CORE-LLM-Bench**, a controlled neurosymbolic benchmark for evaluating LLMs on verifiable ontology-grounded reasoning tasks, covering binary verification and open-ended answer generation across multiple ontology domains.

2. A benchmark-construction methodology that derives questions and ground-truth answers from symbolic-reasoner entailments and associates them with a unified explanation schema. The schema represents alternative proofs for individual answers and complete evidence for multi-answer queries, enabling analysis by reasoning type and minimum complete explanation complexity.
3. A set of complementary evaluation conditions that systematically vary knowledge representation, ontology-context depth, and semantic abstraction through natural-language, formal OWL/SPARQL, and abstracted natural-language representations.
4. A multi-dimensional evaluation protocol combining Answer Exact Match, Answer F1, confidence–correctness alignment, and hallucination rate to characterize both answer quality and grounding behavior.
5. An empirical evaluation of three LLM families based on 81,432 model responses, characterizing differences across task formulation, representation, abstraction, context depth, and explanations.

Related Work

Tafjord et al. (2020) introduced ProofWriter, a framework for generating natural-language theories, implications, and supporting proofs. Given facts, rules, and a query, models are evaluated on tasks such as proof generation, implication enumeration, and abduction. ProofWriter and its associated datasets, including variants based on closed-world and open-world assumptions, provide a controlled setting for evaluating deductive reasoning in natural language. However, the benchmark relies on synthetic rules and templated language, with reasoning depths that remain relatively constrained. As a result, it does not fully capture the complexity of reasoning over richer ontology structures, where entailments may involve heterogeneous relations, class hierarchies, and combinations of inference patterns.

Saparov and He (2023) proposed ProntoQA, a synthetic benchmark designed to evaluate whether LLMs can follow formal proofs rather than rely only on surface-level answer patterns. Each example is generated from a controlled ontology and paired with a formal proof that is translated into a natural-language chain-of-thought explanation. ProntoQA is particularly valuable because it explicitly connects questions to proof structures and supports evaluation in abstract fictional domains. Nevertheless, its ontologies are small and synthetically generated, and its language remains tightly controlled. In contrast, our work evaluates LLMs on questions derived from symbolic-reasoner entailments over ontology-based knowledge structures, while also comparing original and abstracted variants to better isolate formal reasoning from domain familiarity.

Parmar et al. (2024) introduced LogicBench, a natural-language question-answering benchmark covering 25 inference rules. The benchmark enables fine-grained analysis of rule-specific reasoning and shows that even strong LLMs struggle with certain logical patterns, especially those involving negation or

more complex inference. However, LogicBench primarily focuses on individual inference rules rather than extended reasoning chains that combine multiple ontology-based inference mechanisms. Its primary focus is therefore different from evaluating entailments grounded in ontology fragments whose structure, representation, abstraction, and reasoner-derived explanation complexity can be varied systematically.

More recently, Oluçoğlu and Bursa (2026) introduced DL-ReasonSuite, a benchmark specifically designed to evaluate LLMs on DL reasoning. Unlike more general logical-reasoning benchmarks, DL-ReasonSuite directly targets ontology-related reasoning tasks and is organized around three complementary tracks: DLCore, which evaluates fundamental DL reasoning tasks such as consistency checking, subsumption, and instance checking; DLQuery, which assesses entailment-aware SPARQL query answering; and DLBridge, which examines the interaction between natural language and formal ontology representations through NL-to-OWL translation and tool-augmented entailment resolution. This makes DL-ReasonSuite an important recent contribution to the evaluation of LLMs in ontology-centered reasoning settings.

CORE-LLM-Bench is complementary to DL-ReasonSuite and has a different diagnostic focus. DL-ReasonSuite provides broad coverage of canonical DL reasoning, entailment-aware query answering, natural-language-to-OWL translation, and tool-assisted reasoning. CORE-LLM-Bench instead centers its evaluation on reasoner-derived ontology entailments that are linked to symbolic explanation structures and evaluated under systematically varied benchmark conditions. Each benchmark instance is associated with independently derived ground truth and explanation metadata that can be used to characterize reasoning type and structural complexity. In addition, the natural-language, formal-symbolic, and abstracted representations allow the same ontology-grounded reasoning setting to be examined under different levels of semantic and representational support, while binary and open-ended tasks distinguish candidate verification from complete answer-set generation. The contribution is therefore not broader coverage of DL tasks, but controlled diagnosis of how neural models behave as task formulation, representation, semantic cues, contextual scope, and symbolic explanation complexity vary.

Overall, existing benchmarks provide complementary perspectives on logical reasoning in LLMs. ProofWriter evaluates deductive inference over synthetic natural-language theories, ProntoQA emphasizes proof following in controlled and abstract domains, LogicBench enables fine-grained evaluation across individual inference rules, and DL-ReasonSuite covers a broad range of DL reasoning, query-answering, translation, and tool-assisted tasks. CORE-LLM-Bench contributes a complementary benchmark design centered on verifiable ontology-derived entailments and controlled diagnostic variation. Its combination of reasoner-derived explanations, binary and open-ended tasks, natural-language and formal-symbolic representations, semantic abstraction, and varying contextual and explanation complexity enables model behavior to be analyzed across several

factors that may otherwise be conflated in aggregate reasoning performance. In particular, the benchmark is designed to expose whether successful ontology reasoning persists when candidate answers are removed, semantic cues are reduced, symbolic structure is presented directly, or structurally more complex entailments must be recovered.

Methodology

This section describes the construction and design of CORE-LLM-Bench. Given an ontology, the framework extracts individual-centric subgraphs, applies a symbolic reasoner to derive verifiable entailments and associated explanations, generates binary and open-ended reasoning tasks from these entailments, and provides each task in natural-language, formal-symbolic, and abstracted representations. Benchmark instances are additionally characterized according to ontology-context depth, ontology size, and explanation-based reasoning complexity. Stratified sampling is used to retain diversity across these characteristics. The overall benchmark-construction pipeline is summarized in Table 1. Table 9 in the Appendix provides a multi-step example involving an uncle relation, illustrating how ontology fragments, inferred statements, and corresponding SPARQL and natural-language questions cohere.

Benchmark Design Dimensions

CORE-LLM-Bench is designed around four evaluation dimensions that characterize different aspects of ontology-grounded reasoning.

- **Task formulation:** Binary question answering (BQA) evaluates whether a specified statement is entailed by the supplied ontology context, whereas open-ended question answering (OEQA) requires the model to recover the complete set of entailed answers for a query.
- **Knowledge representation:** Benchmark instances are transformed into natural-language and formal-symbolic representations, enabling comparison of reasoning over linguistic and explicitly symbolic encodings of ontology knowledge.
- **Semantic abstraction:** To reduce the influence of domain-specific knowledge, we systematically replace domain-specific terms (i.e., class names, property names, and individual names) with domain-neutral identifiers. This abstraction reduces lexical and domain-specific cues while preserving the underlying logical structure.
- **Reasoning complexity:** Task complexity is characterized by two factors: ontology-context depth and minimum complete explanation complexity. The contextual depth of the ontology determines the depth to which relations around a particular individual are represented in the ontology fragment supplied to the model as context, whereas explanation complexity

Table 1. Overview of the CORE-LLM-Bench data creation pipeline using a running example centered around `John`. Individual-centric ontology subgraphs are processed by a symbolic reasoner to obtain ground-truth inferences. These inferences are transformed into positive and negative `ASK` queries for binary question answering and `SELECT` queries for open-ended question answering. Both the ontology context and generated SPARQL queries are verbalized before stratified sampling and LLM prompting.

Pipeline step	Running example
1. Ontology generation	An individual-centric subgraph is generated around `John`. **ABox:** `John rdf:type Person`; `John hasParent Mary`; `John hasSibling Lisa`. **TBox:** `Woman` ⊑ `Person`; `hasParent inverseOf hasChild`; `hasSibling symmetric`.
2. Symbolic reasoning	Pellet is applied to derive ground-truth inferences, which serve as benchmark answers. **Inferences:** `Mary hasChild John`; `Lisa hasSibling John`.
3. SPARQL query generation	Task-specific SPARQL queries are generated from each inferred triple. **BQA:** Positive `ASK` queries test entailed triples, e.g., `ASK WHERE { Mary hasChild John . }`. Matched negative `ASK` queries preserve the subject and predicate but replace the object with a candidate term for which the queried triple is not entailed by the ontology fragment, e.g., `ASK WHERE { Mary hasChild Lisa . }`. Here, `FALSE` denotes non-entailment by the supplied fragment; it does not denote that the explicit logical negation of the statement is entailed. Under the open-world assumption, such negative instances should therefore be interpreted as unknown (not entailed) rather than as logically false. **OEQA:** `SELECT` queries retrieve the inferred entity, class, or property value, e.g., `SELECT ?x WHERE { Mary hasChild ?x . }`.
4a. Query verbalization	Generated SPARQL queries are converted into natural-language questions. **Positive BQA:** Does Mary have a child named John? **Negative BQA:** Does Mary have a child named Lisa? **OEQA:** Who is Mary's child?
4b. Ontology verbalization	The ontology context is converted into natural language. **Context:** John is a person. John's parent is Mary. John has a sibling named Lisa. Every woman is a person. If someone is a parent of another person, then that other person is their child, and vice versa. If someone is a sibling of another person, then the other person is also their sibling.
5. Stratified sampling	Each generated query is assigned sampling metadata based on ontology size and reasoning complexity. **Reasoning complexity:** Reasoner-derived explanations receive tags for their supporting axioms. Complexity is the minimum complete primitive-tag count: the minimum support for BQA or the minimum union of supports covering every ground-truth answer for OEQA. Task-specific complexity bins are combined with ontology-size strata, and binary positive–negative pairs are sampled together. **Size:** Ontology size is measured as the total number of triples in the corresponding 1-hop or 2-hop individual-centric graph.
6. LLM prompting	Each final prompt contains the ontology context and one question.

characterizes the structural complexity of the minimum set of axioms required in the explanation to support an entailment.

These dimensions are intentionally separated because they capture different characteristics of the reasoning problem. Together, the dimensions allow CORE-LLM-Bench to test whether apparent reasoning performance remains stable when the task, representation, available semantic cues, and structural difficulty are varied systematically.

Individual-Centric Subgraph Generation

Let $G = (V, E)$ denote the complete ontology graph and let $r \in V$ be a selected root individual. A 1-hop individual-centric subgraph contains the triples directly incident to r, that is, triples in which r occurs as either subject or object. A 2-hop individual-centric subgraph additionally contains triples incident to entities reached from r within at most two graph edges. Relevant TBox axioms are included in both settings so that the semantic constraints required to derive inferences are preserved.

The terms 1-hop and 2-hop therefore describe the extraction depth of the ontology context, rather than the number of logical operations required to derive an answer. A question generated from a 2-hop subgraph does not necessarily require exactly two inference steps: its answer may follow from a direct assertion, a single logical operation, or a longer combination of axioms exposed by the broader context.

Question and Answer Generation from Symbolic Inferences

We use the Pellet reasoner (Sirin et al., 2007) to derive ground-truth inferences from the generated OWL subgraphs, focusing on class membership and object property assertions. Pellet is applied to both 1-hop and 2-hop subgraphs, producing the corresponding reasoner-derived inference sets.

Each inferred triple is treated as a verifiable benchmark answer. We then generate corresponding SPARQL queries as benchmark questions. The benchmark includes two task types: binary question answering (BQA) and open-ended question answering (OEQA).

For BQA, one representative positive `ASK` query is generated for each inferred statement and paired with a negative query that preserves the same subject and predicate while substituting a plausible alternative object. Candidate negatives are selected deterministically, prioritizing objects used with the same predicate elsewhere in the fragment and otherwise drawing from the same broad semantic kind (classes for membership, individuals for property assertions). Entailed candidates, duplicate negatives, and candidates already present among positive inferences are excluded. The final binary sample therefore contains equal numbers of `TRUE` and `FALSE` questions. A `FALSE` label means that the candidate statement is not entailed by the supplied fragment, not that its explicit logical negation is proven.

For OEQA, one `SELECT` query is generated for each subject–predicate group by replacing the object position with a variable. The expected answer is the complete

set of objects inferred for the corresponding subject and predicate. In contrast to BQA, where the model verifies a single candidate statement, OEQA requires the model to generate all inferred entities or classes associated with the query.

Natural-Language Conversion

To obtain natural-language questions, SPARQL queries are verbalized using `gpt-4.1-mini`. The same model is used to verbalize ontology contexts into natural language. For the abstract setting, classes, properties, and individuals are mapped consistently to neutral identifiers before verbalization so that the resulting prompts remain grammatical without exposing original domain-specific names.

This LLM-based verbalization strategy offers greater flexibility than fully rule-based approaches such as NaturalOWL (Androutsopoulos et al., 2014), which provide stricter syntactic control but require more manual engineering for diverse query patterns. Since verbalization serves as a transformation step for generating prompts rather than as the primary object of evaluation, using a lightweight LLM provides a practical balance between scalability, cost, and fluency.

To assess the quality of the generated verbalizations, we manually inspected a sample of ontology contexts and questions from the generated benchmark. The inspection focused on whether the verbalized text preserved the entities, relations, and intended entailment of the corresponding formal representation, and whether the resulting questions were understandable and unambiguous. Obvious semantic or formatting errors identified during this inspection were corrected during pipeline development. However, the complete generated dataset was not manually reviewed. Consequently, occasional verbalization errors or linguistic ambiguities may remain and are considered a potential source of noise in the natural-language and abstract settings.

Reasoner-Derived Explanations and Complexity

CORE-LLM-Bench stores symbolic explanations associated with reasoner-derived entailments in a unified format for BQA and OEQA. For each entailed answer, `answer_explanations` preserves alternative valid sets of supporting OWL axioms. Axioms are rendered in Manchester OWL syntax and annotated with reasoning tags. The stored tag sequence follows stored axiom order, not an inferred temporal order in which a reasoner executed logical rules.

For a positive BQA instance, explanations support the queried entailment. A negative instance instead references the explanations of its paired positive statement as provenance; these explanations are *not* proofs of the negative statement or its logical negation. This distinction is necessary under the open-world assumption, where non-entailment is different from logical falsity.

OEQA requires a complete set of entailed answers. An individual ground-truth answer may have multiple alternative explanations, any one of which is sufficient to support that answer. To explain the *complete* answer set, however, at least

one alternative must be selected for every ground-truth answer. Thus, alternative proofs *within* an answer are disjunctive, whereas the evidence requirements *across* answers are conjunctive. The supporting axioms from the selected alternatives are combined as a semantic union, with shared axioms counted only once. The `complete_explanation` metadata describes these unions and preserves all distinct tied minimum complete explanation sets rather than selecting an arbitrary representative. An illustrative formalization is provided in the Appendix.

The original taxonomy describes a broad set of ontology constructs. The primitive tags exercised by the finalized sampled explanations are direct assertions (`D`), hierarchy (`H`), inverse properties (`I`), property chains (`N`), domain/range restrictions (`R`), symmetry (`S`), transitivity (`T`), functional properties (`F`), disjointness (`J`), and reflexivity (`V`). The `M` marker is retained as derived multi-construct status and is excluded from primitive-tag counts.

We operationalize explanation-based complexity as the *minimum complete primitive-tag count*. For BQA, this is the minimum count among valid explanations supporting the positive entailment, inherited by its paired negative question for structural comparison. For OEQA, it is the minimum count across complete evidence unions that cover every ground-truth answer, counting shared axioms once. This annotation describes the recorded symbolic evidence; it does not measure the LLM's internal reasoning steps or guarantee that no unrecorded alternative proof exists. The stored tied minima additionally support reasoning-type analysis, distinguishing tags required by every recorded minimum complete explanation from tags occurring in only some minimum explanations.

Stratified Sampling

Eligible questions vary in context size and reasoner-derived explanation structure. Stratified sampling combines the number of triples in the relevant individual-centric context with task-specific bins derived from minimum complete explanation complexity. For BQA, low, medium, and high complexity correspond respectively to tag counts of 1, 2, and 3 or more. For OEQA, the corresponding thresholds are 1–3, 4–5, and 6 or more. These thresholds differ because a complete multi-answer explanation aggregates evidence for several answers. Positive and negative BQA questions are sampled as complete pairs, retaining balanced labels. The resulting sample captures variation in ontology size and observed explanation complexity without equating ontology-context depth with the number of inference steps.

Experiments

Our experiments evaluate the reasoning capabilities of three contemporary, cost-efficient LLMs from distinct model families using datasets generated with the CORE-LLM-Bench framework. We assess model performance across different ontologies, representation settings, context sizes, question formats, and evaluation metrics.

Datasets

CORE-LLM-Bench is instantiated using four ontology datasets: Family, Pizza 100, Pizza 250, and OWL2Bench. These ontologies were selected to provide variation in ontology size, domain, relational structure, and OWL construct coverage, allowing the benchmark to assess whether observed reasoning behavior generalizes across different symbolic knowledge structures rather than reflecting a single ontology or application domain.

The Family ontology contains 2,516 axioms describing individuals and familial relationships. Pizza 100 and Pizza 250 are two synthetic variants of the Pizza ontology containing 100 and 250 generated individuals, respectively, and comprise 2,151 and 2,628 axioms. The two Pizza variants allow the same underlying ontology schema to be evaluated under different ABox sizes. OWL2Bench is based on the OWL2Bench1-DL TBox together with a reduced, task-specific ABox and contains 1,841 axioms. Further details on the construction and characteristics of the underlying ontologies are provided in Loesch et al. (2026).

For each ontology, CORE-LLM-Bench generates individual-centric contexts at both 1-hop and 2-hop extraction depths and derives benchmark questions from symbolically verified class-membership and object-property entailments. Each generated instance is associated with its ontology context, query, ground-truth answer, task type, extraction depth, ontology-size information, and reasoner-derived explanation metadata. The benchmark provides binary question answering (BQA) and open-ended question answering (OEQA) instances in natural-language, formal-symbolic, and abstracted natural-language representations.

The resulting datasets therefore provide complementary forms of variation. Family emphasizes relational reasoning over familiar familial structures; the Pizza variants provide controlled changes in ABox scale while retaining a common ontology schema; and OWL2Bench exposes models to a broader range of OWL constructs. Together, these datasets allow CORE-LLM-Bench to evaluate the robustness of model behavior across ontology domains, scales, representation settings, task formulations, and levels of reasoning complexity.

The finalized sample contains 9,048 question–hop instances: 6,032 balanced binary questions and 3,016 open-ended questions. Each task is presented in all three representation conditions. Final sampled counts by ontology, hop, and task appear in Appendix Table 8.

Large Language Models

Given the financial and computational cost of large-scale LLM evaluation, we selected three relatively inexpensive models from distinct model families. This selection balances model diversity, evaluation feasibility, and reproducibility.

- **GPT-5 mini** (`gpt-5-mini-2025-08-07`), representing OpenAI's GPT model family.

- **Gemini 2.5 Flash-Lite** (`google/gemini-2.5-flash-lite`), representing Google's Gemini model family.
- **Qwen3-30B-A3B-Instruct** (`qwen/qwen3-30b-a3b-instruct-2507`), representing a cost-efficient open-weight model family.

Evaluation Metrics

CORE-LLM-Bench evaluates model behavior using Answer Exact Match, Answer F1, confidence–correctness alignment, and hallucination rate. Together, these metrics distinguish whether a model produces the complete expected answer, partially recovers the symbolically supported answer set, expresses confidence consistent with its observed correctness, and avoids generating answers unsupported by the ontology. All metrics are computed from the structured output schema produced by each model, which contains a final answer and a self-reported confidence score.

Answer Exact Match and Answer F1 We evaluate answer correctness using Answer Exact Match (A-EM) and Answer F1 (A-F1). Predicted and ground-truth answers are normalized and represented as sets of answer items. Answer ordering is ignored and duplicate items are removed.

Answer Exact Match is defined as

$$\text{A-EM}(A, E) = \begin{cases} 1, & \text{if } A = E, \\ 0, & \text{otherwise,} \end{cases}$$

where A denotes the normalized predicted answer set and E denotes the normalized ground-truth answer set.

For non-empty parsed answer sets, partial answer-set correctness is measured using precision and recall:

$$P = \frac{|A \cap E|}{|A|}, \qquad R = \frac{|A \cap E|}{|E|},$$

and Answer F1 is computed as

$$\text{A-F1} = \frac{2PR}{P + R}.$$

Confidence–Correctness Alignment We evaluate whether a model's self-reported confidence is consistent with the correctness of its answer at the individual-instance level. Each model reports a confidence value $C_i \in [0, 1]$, which is compared with the corresponding per-question Answer F1 score $F_{1,i}$:

$$\text{Alignment}_i = 1 - |C_i - F_{1,i}|.$$

Higher values indicate closer agreement between self-reported confidence and observed answer correctness.

Hallucination Rate For OEQA, we measure the proportion of generated answer items that are not supported by the complete reasoner-derived answer set for the query. Let A_i and E_i denote the normalized predicted and representation-specific ground-truth answer sets for question i. The aggregate hallucination rate is micro-averaged across all OEQA observations with generated answer items:

$$H = \frac{\sum_{i \in \mathrm{OEQA}} |A_i \setminus E_i|}{\sum_{i \in \mathrm{OEQA}} |A_i|}.$$

Experimental Setup

We evaluate each model under three controlled representations designed to examine how ontology-grounded reasoning changes when the same underlying symbolic knowledge is expressed through natural-language, formal-symbolic, or semantically abstracted inputs.

- **Natural-language representation (NL):** The model receives a verbalized ontology context together with a natural-language question. This condition evaluates ontology-grounded reasoning when symbolic knowledge is expressed through semantically meaningful natural language, allowing models to exploit both relational structure and lexical information. Contexts exceeding 10,000 characters are reduced by retaining their first 10,000 characters.
- **Formal-symbolic representation (FS):** The model receives the ontology context in formal OWL syntax, serialized as Turtle, together with the corresponding SPARQL query. This condition evaluates whether models can directly interpret explicit symbolic knowledge and query structure without requiring natural-language verbalization. Formal-symbolic contexts exceeding a 12,000-character prompt budget are reduced using a deterministic query-aware procedure. URI fragments and prefixed names are extracted from the SPARQL query, excluding generic RDF, RDFS, OWL, and XSD terms. The serialized ontology is divided into coherent blocks, and blocks containing at least one query term are prioritized. When no complete block matches, local character windows surrounding occurrences of the query terms are retained. Any remaining budget is filled with a prefix of the original ontology context, after which the resulting context is limited to 12,000 characters.
- **Abstracted natural-language representation (AR):** The model receives an abstracted natural-language representation in which domain-specific lexical cues are systematically removed while the underlying relational structure is preserved. Classes, properties, and individuals are replaced consistently with neutral identifiers such as `Class1`, `Property1`, and `Individual1` across the ontology context, questions, and answers before verbalization. This condition is designed to test whether reasoning performance persists when models cannot rely on familiar entity names,

domain terminology, or pretrained semantic associations. As in the NL setting, contexts exceeding 10,000 characters are reduced by retaining their first 10,000 characters.

All experiments use zero-shot instruction-only prompting with a fixed output schema. An example prompt template is provided in Figure 4 in the Appendix.

All three models were accessed through OpenRouter with pinned upstream providers and automatic fallback disabled. Qwen was accessed through the Alibaba provider at a temperature of 0.0, a `top_p` value of 0.9, a presence penalty of 0.0, a frequency penalty of 0.1, an output limit of 1,024, a seed of 0, and reasoning output disabled. Gemini used the pinned Google AI Studio provider with thinking disabled. GPT-5 mini used the pinned OpenAI provider, low reasoning effort, low verbosity, and a completion limit of 1,024; no explicit temperature was supplied. Exact model-specific configuration files and input hashes were retained for reproducibility. The final evaluation matrix contains one accepted observation for every model–question–representation cell (27,144 observations per model, 81,432 in total); repeated-run variance was not measured. A small number of malformed but technically accepted responses were retained and scored according to the same rules as other outputs (GPT-5 mini: 22; Gemini 2.5 Flash-Lite: 6; Qwen3-30B-A3B-Instruct: 58).

Results and Discussion

In this section, we present the performance of three LLMs, *GPT-5 mini*, *Gemini 2.5 Flash-Lite*, and *Qwen3-30B-A3B-Instruct*, on benchmark version 1.1.1, comprising 9,048 questions.

We evaluate model performance across different context depths (1-hop and 2-hop), input representations (natural language, formal-symbolic, and abstracted), and question types (binary and open-ended questions). The results are presented separately for each ontology to identify common patterns as well as dataset-specific differences.

Family Results

Table 2 presents the results for the Family ontology across natural-language (NL), formal-symbolic (FS), and abstracted (AR) representations. Overall, FS achieves the highest Answer Exact Match (A-EM) scores for all three models, across both question types and context depths. The performance gain is particularly pronounced for open-ended questions. On 2-hop open-ended questions, GPT-5 mini achieves 66.7% A-EM with FS, compared with 14.2% for NL and 3.2% for AR. Similar improvements are observed for Gemini 2.5 Flash-Lite and Qwen3-30B-A3B-Instruct. These results suggest that explicitly representing ontology axioms and queries in a formal language can lead to more accurate reasoning.

Despite these improvements, open-ended questions are more challenging than binary questions. In the FS setting with 2-hop context, A-EM scores for BQA

range from 87.4% to 92.3% across the three models, whereas open-ended A-EM ranges from 41.6% to 66.7%. This difference highlights the additional difficulty of generating a complete and correct answer set rather than binary verification of an inference. The higher answer F1 (A-F1) scores relative to A-EM further indicate that models sometimes recover part of the correct answer set without producing an exact match.

Context depth also affects performance. Across all models and input representations, A-EM decreases when the ontology-context extraction depth increases from 1-hop to 2-hop. The decline is more pronounced for open-ended questions in the NL and AR settings. Under AR, for open-ended questions with 2-hop context, GPT-5 mini and Qwen3-30B-A3B-Instruct achieve A-EM scores of 3.2% while Gemini 2.5 Flash-Lite achieves 4.0%. Hallucination rates are also high in this setting, reaching 57.6%, 95.4%, and 98.3%, respectively. These findings indicate that abstraction combined with increased context poses substantial challenges for language models.

The alignment results provide an additional perspective. Although FS generally achieves the highest answer accuracy, it does not consistently produce the highest confidence–correctness alignment results. For example, Gemini 2.5 Flash-Lite obtains a higher alignment on 2-hop open-ended questions with NL (63.7%) than with FS (47.7%), despite substantially lower A-EM (8.3% versus 41.6%). This illustrates the importance of considering answer accuracy and confidence alignment separately.

Overall, the results from the Family dataset suggest that formal-symbolic representations particularly improve LLM performance on ontology-based questions. In contrast, models achieve lower performance with abstracted representation, suggesting a potential benefit of familiar domain-specific labels. However, the extent to which familiar labels influence reasoning remains an open question.

Table 2. Performance on Family. A-EM and A-F1 denote Answer Exact Match and Answer F1; Align. denotes confidence–correctness alignment; Hall. is the micro-averaged OEQA hallucination rate over generated answers. For BQA, A-F1 equals A-EM. Bold values identify the maximum for answer and alignment metrics and the minimum for Hall. within each model and extraction depth. All metric values are percentages.

Model	Setting	Binary Questions				Open-Ended Questions							
		A-EM		Align.		A-EM		A-F1		Align.		Hall.	
		1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop
GPT-5 mini	NL	85.6	67.5	83.1	70.1	27.0	14.2	45.0	25.0	64.9	67.9	47.4	53.6
	FS	**95.7**	**92.3**	**95.7**	**92.5**	**75.1**	**66.7**	**78.1**	**69.4**	**78.1**	**70.3**	**12.7**	**16.4**
	AR	58.1	53.2	59.1	54.6	12.3	3.2	15.8	4.4	43.9	35.3	21.3	57.6
Gemini 2.5 Flash-Lite	NL	70.8	57.5	72.6	61.5	26.5	8.3	42.2	14.5	**57.0**	**63.7**	64.3	67.9
	FS	**94.5**	**90.5**	**94.5**	**90.5**	**47.5**	**41.6**	**50.4**	**47.7**	50.4	47.7	**40.5**	**49.7**
	AR	59.9	57.0	60.4	57.0	19.0	4.0	32.2	8.0	41.8	23.2	84.9	95.4
Qwen3-30B-A3B-Instruct	NL	80.4	67.2	79.8	66.8	19.8	4.8	34.0	14.7	40.0	18.9	80.9	92.9
	FS	**97.8**	**87.4**	**97.8**	**87.4**	**71.0**	**54.2**	**76.2**	**63.3**	**76.2**	**63.6**	**17.9**	**42.4**
	AR	67.6	60.0	67.3	59.4	18.3	3.2	31.2	9.1	42.5	21.1	68.6	98.3

Pizza Results

Tables 3 and 4 present the results for Pizza 100 and Pizza 250, respectively. As observed for Family, open-ended questions are generally more challenging than binary questions, and A-EM decreases from 1-hop to 2-hop questions across all models and input settings. However, the effect of input representation is less consistent than for Family. In particular, FS does not always outperform the other representations, and AR achieves comparable or higher accuracy in several cases.

For GPT-5 mini, FS consistently outperforms NL and AR for open-ended questions on both Pizza ontologies. On Pizza 100, FS achieves A-EM scores of 63.6% and 43.6% for questions with 1-hop and 2-hop contexts, respectively. Similar results are observed on Pizza 250, with scores of 64.6% and 43.7% A-EM. These scores exceed the corresponding NL and AR results. This indicates that formal-symbolic inputs remain effective for this model, particularly when generating complete answer sets.

However, AR is more competitive for Gemini 2.5 Flash-Lite and Qwen3-30B-A3B-Instruct. For 2-hop open-ended questions on Pizza 100, both models achieve higher A-EM with AR than with the FS representation. Gemini 2.5 Flash-Lite reaches 48.5% with AR compared with 42.4% with FS, while Qwen3-30B-A3B-Instruct achieves 46.7% compared with 41.8%. This pattern does not hold consistently on Pizza 250, where FS achieves higher A-EM scores for both models on 2-hop open-ended questions.

The NL setting is particularly challenging for some model–question combinations. Both Gemini 2.5 Flash-Lite and Qwen3-30B-A3B-Instruct achieve 0.0% A-EM on 2-hop open-ended questions for both Pizza ontologies. For Qwen3-30B-A3B-Instruct, this is accompanied by hallucination rates of 92.7% on Pizza 100 and 98.8% on Pizza 250. In contrast, FS and AR substantially improve open-ended answer accuracy for these models, although neither representation consistently performs best across all metrics.

Overall, the Pizza results highlight the challenges posed by increased context depth and open-ended question settings. However, unlike Family, they do not show a clear advantage for FS or a systematic deterioration with abstraction. Instead, performance variability suggests that outcomes depend on the interplay between ontology, model, and input representation. This variability underscores the importance of evaluating ontology-grounded reasoning across multiple datasets, models, and input representations.

OWL2Bench Results

Table 5 presents the results for OWL2Bench across NL, FS, and AR representations. Overall, FS achieves the highest Answer Exact Match (A-EM) scores for all three models on both 1-hop and 2-hop binary questions. GPT-5 mini, Gemini 2.5 Flash-Lite, and Qwen3-30B-A3B-Instruct achieve 89.9%, 88.3%, and 88.3% A-EM, respectively, on 2-hop questions. In comparison, the corresponding

Table 3. Performance on Pizza 100. A-EM and A-F1 denote Answer Exact Match and Answer F1; Align. denotes confidence–correctness alignment; Hall. is the micro-averaged OEQA hallucination rate over generated answers. For BQA, A-F1 equals A-EM. Bold values identify the maximum for answer and alignment metrics and the minimum for Hall. within each model and extraction depth. All metric values are percentages.

Model	Setting	Binary Questions				Open-Ended Questions							
		A-EM		Align.		A-EM		A-F1		Align.		Hall.	
		1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop
GPT-5 mini	NL	85.2	58.5	80.5	57.6	35.8	24.8	39.5	38.1	46.3	63.7	42.4	36.2
	FS	**85.5**	**62.1**	**86.3**	**65.6**	**63.6**	**43.6**	**75.1**	**57.6**	**74.5**	**64.9**	9.3	12.9
	AR	64.8	61.5	63.8	61.5	28.5	23.0	42.1	33.1	42.1	43.4	**0.3**	**8.1**
Gemini 2.5 Flash-Lite	NL	64.8	54.8	65.3	55.6	33.9	0.0	35.5	3.9	54.9	29.5	32.2	93.5
	FS	70.9	**68.2**	70.9	**68.2**	**60.6**	42.4	**71.9**	61.9	71.9	61.9	11.8	**11.7**
	AR	**73.6**	67.3	**73.8**	67.3	51.5	**48.5**	65.1	**64.3**	**77.2**	**65.6**	**0.0**	13.8
Qwen3-30B-A3B-Instruct	NL	84.8	66.1	84.8	66.0	1.2	0.0	1.6	0.7	2.8	11.9	96.9	92.7
	FS	**95.8**	**79.1**	**95.8**	**79.1**	63.0	41.8	63.9	55.2	63.9	55.3	18.4	**19.4**
	AR	71.8	58.8	71.9	58.5	**63.6**	**46.7**	**75.2**	**59.2**	**75.9**	**61.4**	**12.3**	39.6

Table 4. Performance on Pizza 250. A-EM and A-F1 denote Answer Exact Match and Answer F1; Align. denotes confidence–correctness alignment; Hall. is the micro-averaged OEQA hallucination rate over generated answers. For BQA, A-F1 equals A-EM. Bold values identify the maximum for answer and alignment metrics and the minimum for Hall. within each model and extraction depth. All metric values are percentages.

Model	Setting	Binary Questions				Open-Ended Questions							
		A-EM		Align.		A-EM		A-F1		Align.		Hall.	
		1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop
GPT-5 mini	NL	**85.0**	56.6	79.7	56.8	37.4	12.6	43.0	35.1	47.5	64.4	42.2	45.2
	FS	84.0	58.3	**85.3**	**62.9**	**64.6**	**43.7**	**75.8**	**58.5**	**75.3**	**66.6**	8.4	11.7
	AR	65.3	**59.5**	64.7	61.6	32.5	16.5	45.9	26.5	45.7	48.6	**0.0**	**11.7**
Gemini 2.5 Flash-Lite	NL	65.5	54.1	66.1	54.6	28.6	0.0	30.6	6.1	50.1	40.3	48.7	91.3
	FS	69.7	**66.3**	69.7	**66.3**	**60.7**	**43.2**	**71.1**	**63.8**	71.1	**63.8**	13.0	**10.8**
	AR	**71.1**	64.8	**71.1**	64.8	55.3	36.9	68.7	56.0	**78.4**	58.0	**0.0**	23.0
Qwen3-30B-A3B-Instruct	NL	83.7	60.9	83.6	60.9	0.5	0.0	1.5	1.2	3.0	19.5	97.4	98.8
	FS	**97.1**	**82.3**	**97.1**	**82.3**	64.6	**43.2**	65.1	**58.3**	65.1	58.6	16.7	**18.1**
	AR	72.6	61.7	72.6	61.3	**65.0**	37.9	**77.1**	53.4	**77.5**	**59.9**	**7.8**	23.2

NL results are 66.1%, 56.4%, and 68.5%. These findings are consistent with our earlier observation that explicitly representing ontology axioms and queries in a formal-symbolic format can support more accurate inference.

Open-ended questions remain substantially more challenging, with lower A-EM scores and substantial declines from 1-hop to 2-hop contexts. Although FS achieves the highest A-EM scores for all three models in 2-hop open-ended questions, AR and NL outperform FS in some 1-hop cases. In the FS setting, GPT-5 mini declines from 68.7% to 51.4%, Gemini 2.5 Flash-Lite from 41.3% to 29.2%, and Qwen3-30B-A3B-Instruct from 66.7% to 47.1%. However, this pattern does not extend to all binary results. Under FS, Gemini 2.5 Flash-Lite improves from 80.1% to 88.3%, while Qwen3-30B-A3B-Instruct improves from 86.6% to 88.3%. Thus, the effect of context depth is more consistent for open-ended questions than for binary questions.

The alignment results also vary across representations. Although FS generally provides the highest answer accuracy, it does not always achieve the strongest confidence–correctness alignment. For instance, on 1-hop open-ended questions, Gemini 2.5 Flash-Lite achieves 76.8% alignment with AR, compared with 49.7% with FS. This reinforces the need to assess answer accuracy and alignment separately.

Overall, the OWL2Bench results show a clear advantage for formal-symbolic representations, particularly for 2-hop questions. At the same time, open-ended answer generation remains challenging, and abstraction substantially reduces 2-hop accuracy across all three models. Compared with Family, where FS consistently achieves the highest A-EM, OWL2Bench also shows cases in which AR or NL performs better, particularly on 1-hop open-ended questions. These findings further demonstrate that the effect of input representation depends on the ontology, model, and question type.

Table 5. Performance on OWL2Bench. A-EM and A-F1 denote Answer Exact Match and Answer F1; Align. denotes confidence–correctness alignment; Hall. is the micro-averaged OEQA hallucination rate over generated answers. For BQA, A-F1 equals A-EM. Bold values identify the maximum for answer and alignment metrics and the minimum for Hall. within each model and extraction depth. All metric values are percentages.

Model	Setting	Binary Questions				Open-Ended Questions							
		A-EM		Align.		A-EM		A-F1		Align.		Hall.	
		1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop	1-hop	2-hop
GPT-5 mini	NL	90.7	66.1	89.2	66.0	62.8	14.7	73.1	26.5	75.1	65.6	10.3	37.2
	FS	**99.5**	**89.9**	**99.3**	**90.2**	**68.7**	**51.4**	**78.6**	**67.7**	**78.5**	**69.2**	12.4	13.7
	AR	77.8	58.5	77.5	59.1	56.0	13.4	67.8	22.7	69.1	47.5	**4.1**	**8.7**
Gemini 2.5 Flash-Lite	NL	71.8	56.4	71.9	52.9	53.8	7.0	67.1	16.4	75.7	33.3	**3.9**	74.0
	FS	**80.1**	**88.3**	**80.1**	**88.3**	41.3	**29.2**	49.7	**49.9**	49.7	**49.9**	32.1	**32.3**
	AR	77.3	58.2	77.4	58.4	**56.0**	16.9	**68.8**	30.9	**76.8**	35.3	10.5	70.9
Qwen3-30B-A3B-Instruct	NL	81.0	68.5	81.0	68.5	39.5	7.2	55.6	17.5	56.9	19.6	44.7	93.3
	FS	**86.6**	**88.3**	**86.6**	**88.3**	**66.7**	**47.1**	**77.4**	**71.3**	**77.4**	**71.5**	**10.8**	**17.5**
	AR	69.8	53.1	69.8	53.2	50.7	10.0	63.7	24.9	64.3	26.7	50.9	81.8

Performance by Explanation Tags

In Figure 1, we observe a clear trend in which performance gradually decreases as the number of tag types in the minimum explanation for abstracted representations (AR) increases. An exception occurs for open-ended questions with larger tag counts (6–7), where performance increases slightly. This is mainly because these tags are heavily fact-based (D) and are therefore naturally easier to answer than questions involving other axioms. For FS and NL-based representations, this trend is less clearly observable, as logical reasoning cannot be easily separated from pattern matching based on domain cues.

A similar trend can be observed when performance is plotted against tag combinations (Figure 2). In general, combinations involving more tags correspond to lower performance. For BQA, performance decreases as the number of tags increases: single-tag questions (D) perform better than two-tag questions

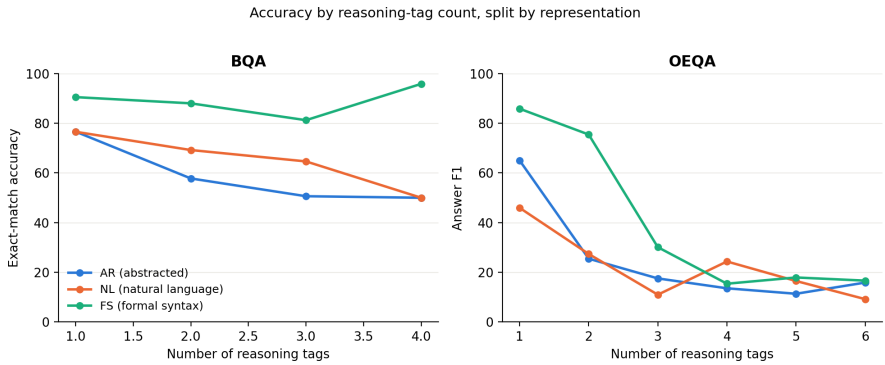


Figure 1. Accuracy by reasoning-tag count, split by representation (AR, NL, FS), averaged across the three models. Left: BQA (Answer Exact Match). Right: OEQA (Answer F1).

(DS, DH, and DR), which, in turn, perform better than three-tag questions (DIR). However, there is an exception: DN is considerably more challenging than DHR and DIR. Here, N represents a property chain, which introduces greater expressivity and is equivalent to an expression combining transitivity and subproperty axioms.

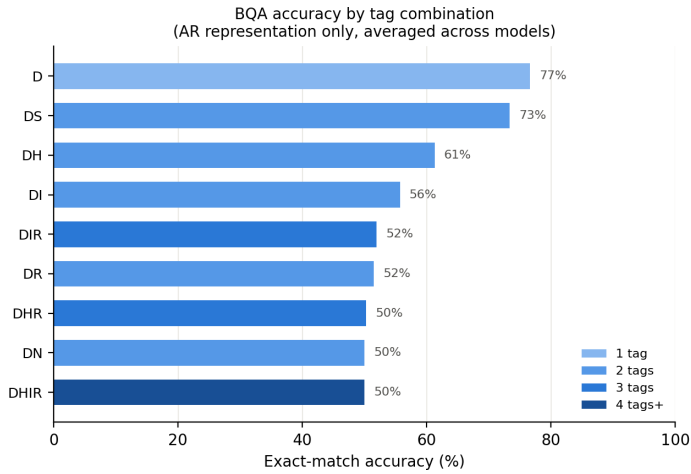


Figure 2. BQA accuracy by tag combination, AR representation only, average across models. Bar color indicates tag count (light to dark = 1 to 4+ tags).

This trend is also partially observable for open-ended questions. However, some tag combinations (DN, DHI, and DHR) are consistently more challenging across all representations. In general, certain question types appear in only one dataset,

while the tag combinations shared across all datasets are D, DH, and DI. This suggests that current datasets in the benchmark capture only certain levels of task complexity, largely determined by the complexity of the explanation tags. Consequently, the benchmark may not fully capture the broader range of reasoning complexities that can arise from different combinations of axioms.

To investigate this limitation more directly, Figure 3 examines the source-dataset composition of each tag combination for open-ended questions, restricting the analysis to the AR representation to isolate dataset effects from those associated with representation choice. The vast majority of tag combinations originate from a single source ontology. Among the eighteen combinations with sufficient support ($n \geq 20$), only four (D, DI, DH, and DHR) appear in more than one dataset, and only three (D, DI, and DH) are represented in all four datasets (Family, OWL2Bench, Pizza 100, and Pizza 250). The remaining combinations are restricted to specific datasets. For example, DN, DHI, and DFHRV occur only in OWL2Bench, whereas DHIR, DHIRS, DHRS, and DHNS are exclusive to Family. DFHIJR, in contrast, appears in both Pizza variants. These findings indicate that the coverage of tag combinations is uneven across source datasets, suggesting that the benchmark's representation of reasoning complexity may be influenced by the characteristics of individual ontologies rather than reflecting a systematic range of reasoning tasks.

Limitations

The benchmark has several limitations. First, symbolic explanation complexity reflects the reasoning steps recorded by the ontology reasoner, which may differ from how LLMs arrive at their answers. Moreover, the recorded explanations may not cover all possible reasoning paths. The uneven distribution and frequent overlap of reasoning types also make it difficult to assess the difficulty of individual OWL constructs.

Finally, the evaluation is limited to three language models, with one response per question and representation. Future work could extend model and ontology coverage, improve representation consistency and semantic validation, and investigate how symbolic explanations can support LLM reasoning.

Conclusion

In this study, we present *CORE-LLM-Bench*, a neurosymbolic benchmark generation framework for producing ontology-based question-answer pairs with explicit explanation sets. The generation pipeline instantiates questions over ontology-derived local graph slices and records explanations consisting of factual assertions and schema-level axioms sufficient to justify the answer. In this work, we use the Family, Pizza (i.e., Pizza 100 and Pizza 250), and OWL2Bench datasets in 1-hop and 2-hop settings.

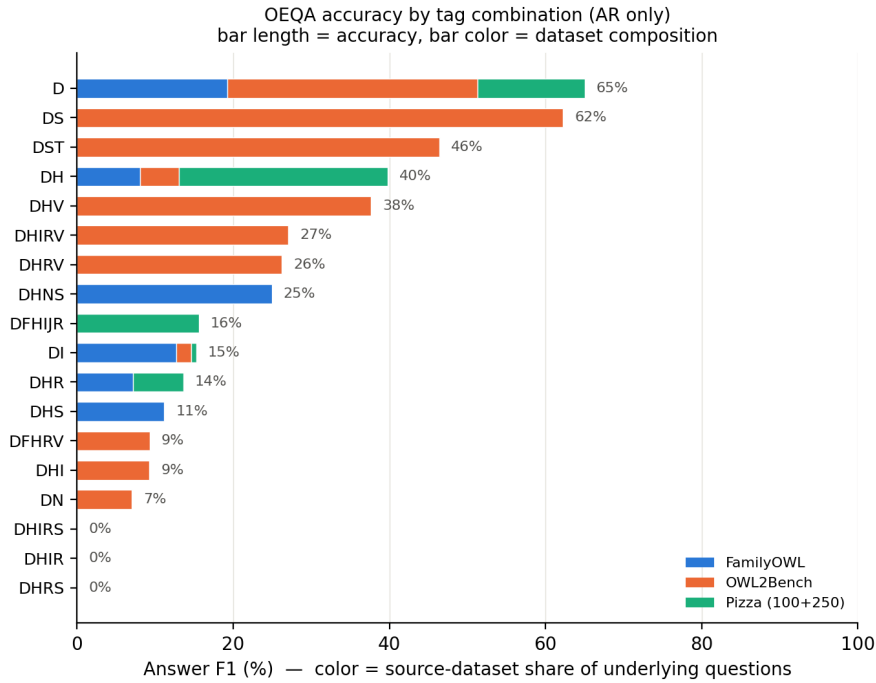


Figure 3. OEQA Answer F1 by tag combination (AR representation only). Bar length is Answer F1; bar color is split by the share of each combination's underlying unique questions drawn from each source dataset.

CORE-LLM-Bench provides a controlled framework for evaluating LLMs on verifiable multi-step ontology-based reasoning tasks. Our results show that formal-symbolic input representation generally improves performance, suggesting that explicit OWL/SPARQL structure helps models align with ontology-based inference. However, performance declines on 2-hop and open-ended questions, where models must generate complete inferred answer sets rather than verify individual claims. Abstraction further reduces accuracy and increases hallucination, indicating that models still rely in part on familiar semantic cues.

Overall, current LLMs can benefit from structured representations, but remain unreliable under abstraction, increased extraction depth, and open-ended question formulation settings. These findings highlight the value of *CORE-LLM-Bench* for identifying both apparent reasoning success and failures in robust, grounded multi-step inference.

However, the current framework may allow *shortcut reasoning*, where models produce correct answers without genuine multi-step inference, potentially inflating performance estimates. In addition, the benchmark has not yet systematically evaluated robustness under noisy, ambiguous, or conflicting information. Future

work will address these issues by removing cues that enable trivial conclusions and by introducing controlled noise, such as misleading or partially incorrect information. These extensions should provide a more rigorous assessment of multi-step reasoning and robustness under uncertainty.

Data and Code Availability

CORE-LLM-Bench v1.1.1, used in the experiments reported in this article, was published on October 4, 2026, and is archived on Zenodo under DOI 10.5281/zenodo.23138373. The reproducibility and source-code release is available at https://github.com/jloe2911/CORE-LLM-Bench/releases/tag/v1.1.1. The canonical versioned tabular dataset is publicly accessible on Hugging Face at https://huggingface.co/datasets/jloe2911/CORE-LLM-Bench/tree/v1.1.1, while the complete frozen archival package is provided through Zenodo.

The released benchmark contains 9,048 question–hop instances, comprising 6,032 binary and 3,016 open-ended questions. The archival package includes the finalized benchmark data, symbolic explanation and reasoning metadata, corrected observation-level evaluation scores, aggregated results, reasoning-tag analysis outputs, provenance manifests, validation reports, and checksums. Evaluation and analysis scripts are available in the versioned GitHub repository.

Licensing is component-specific. Repository software is distributed under the MIT License; separable original CORE-LLM-Bench content under CC BY 4.0; Pizza-derived material under CC BY 3.0; OWL2Bench-derived material under Apache-2.0; and Family/FHKB-derived material under CC BY-SA 3.0. Detailed attribution, provenance, and modification information is provided in the repository's `NOTICE.md`.

References

Androutsopoulos I, Lampouras G and Galanis D (2014) Generating natural language descriptions from OWL ontologies: the naturalowl system. *CoRR* abs/1405.6164. URL http://arxiv.org/abs/1405.6164.

Besold TR, d'Avila Garcez AS, Bader S, Bowman H, Domingos P, Hitzler P and Kühnberger KU (2017) Neural-symbolic learning and reasoning: A survey and interpretation. *Artificial Intelligence* 237: 214–251.

Brown T, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al. (2020) Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33: 1877–1901.

Chen B, Zhang Z, Langrené N and Zhu S (2025) Unleashing the potential of prompt engineering for large language models. *Patterns* 6(6): 101260. DOI: 10.1016/j.patter.2025.101260. URL http://dx.doi.org/10.1016/j.patter.2025.101260.

d'Avila Garcez AS, Lamb LC and Gabbay DM (2019) Neurosymbolic ai: The 3rd wave. *Artificial Intelligence Review* 52(4): 2025–2036.

Ji Q, Gao Z and Huang Z (2011) Reasoning with noisy semantic data. In: *The Semantic Web: Research and Applications*, *Lecture Notes in Computer Science*, volume 6644. Springer, pp. 497–502. DOI:10.1007/978-3-642-21064-8_42.

Kau A, He X, Nambissan A, Astudillo A, Yin H and Aryani A (2024) Combining knowledge graphs and large language models. URL https://arxiv.org/abs/2407.06564.

Loesch J, Singh G, Mutharaju R and Celebi R (2026) Nsorn: Designing a benchmark dataset for neurosymbolic ontology reasoning with noise. *Neurosymbolic Artificial Intelligence* 2: 29498732261469335. DOI:10.1177/29498732261469335. URL https://doi.org/10.1177/29498732261469335.

Oluçoğlu M and Bursa O (2026) Dl-reasonsuite: A benchmark for evaluating description logic reasoning in large language models. *Applied Sciences* 16(4). DOI:10.3390/app16041821. URL https://www.mdpi.com/2076-3417/16/4/1821.

Pan S, Luo L, Wang Y, Chen C, Wang J and Wu X (2024) Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering* 36(7): 3580–3599. DOI:10.1109/tkde.2024.3352100. URL http://dx.doi.org/10.1109/TKDE.2024.3352100.

Parmar M, Patel N, Varshney N, Nakamura M, Luo M, Mashetty S, Mitra A and Baral C (2024) Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. URL https://arxiv.org/abs/2404.15522.

Saparov A and He H (2023) Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. URL https://arxiv.org/abs/2210.01240.

Sirin E, Parsia B, Grau BC, Kalyanpur A and Katz Y (2007) Pellet: A practical owl-dl reasoner. *Journal of Web Semantics* 5(2): 51–53. DOI:https://doi.org/10.1016/j.websem.2007.03.004. URL https://www.sciencedirect.com/science/article/pii/S1570826807000169. Software Engineering and the Semantic Web.

Tafjord O, Mishra BD and Clark P (2020) Proofwriter: Generating implications, proofs, and abductive statements over natural language. *CoRR* abs/2012.13048. URL https://arxiv.org/abs/2012.13048.

Supplementary Material

This appendix provides supplementary documentation for CORE-LLM-Bench, including an illustration of multi-step ontology reasoning, the reasoner-derived explanation-tag taxonomy, detailed benchmark dataset statistics, the model output schema, and a complete example benchmark instance. Together, these materials document the benchmark's reasoning annotations, dataset composition, evaluation interface, and the relationship between symbolic ontology fragments, reasoner-derived entailments, and the corresponding reasoning tasks.

Multi-Step Reasoning

Multi-step reasoning over ontologies refers to the derivation of implicit knowledge through a sequence of logical inference steps. Such reasoning is central to ontology-based systems because many conclusions cannot be derived from a single asserted triple or axiom, but instead require combining multiple semantic relations.

For example, consider the following assertion and axioms:

- Assertion: `X worksAt Y`
- Axioms:
  - $\texttt{range}(\texttt{worksAt}) = \texttt{Organization}$
  - $\texttt{Organization} \sqsubseteq \texttt{LegalEntity}$

From the assertion `X worksAt Y`, the range axiom entails that `Y` is an instance of `Organization`. Applying the class hierarchy then entails that `Y` is also a `LegalEntity`. This conclusion is not explicitly stated in the ontology and can only be derived by combining multiple inference steps.

Explanation Tags and Complete Supports

The explanation-tag taxonomy describes OWL constructs represented in supporting axioms. The original taxonomy contains 20 entries, of which `M` is treated in v1.1.1 as *derived* multi-TBox metadata rather than a primitive explanation tag. Primitive tag sequences preserve stored axiom order and do not encode the temporal firing order of reasoner rules.

Table 7 reports *primary analysis* tag frequencies: a construct is counted for a question only when it appears in every tied minimum complete explanation. Counts overlap because several primitive tags may co-occur in the same supporting axioms. These minimum-explanation counts are deliberately more restrictive than counts over all stored proof alternatives. The derived `M` status is not included in primitive-tag attribution or complexity.

The EVERY definition identifies hierarchy, inverse-property, and domain/range tags with sufficient support in both tasks for the focal adjusted analysis. Property chains and symmetry are sparse in BQA, transitivity is absent from BQA and sparse in OEQA, and functional-property, disjointness, and reflexivity tags occur only in OEQA. Direct assertions are ubiquitous. These frequencies describe

Table 6. Explanation-tag taxonomy. `M` denotes derived metadata and is excluded from primitive-tag complexity.

Tag	Construct	Tag	Construct
D	Direct assertions	**I**	Inverse properties
H	Hierarchy	**F**	Functional properties
T	Transitivity	**V**	Reflexive properties
S	Symmetry	**Y**	Irreflexive properties
A	Asymmetric properties	**Q**	Equivalence
J	Disjointness	**R**	Domain/range restrictions
N	Property chains	**C**	Cardinality restrictions
E	Existential restrictions	**L**	Universal restrictions
∩	Intersection	**U**	Union
¬	Complement	**M**	Multi-TBox status (derived)

Table 7. Question-level coverage of tags present in every tied minimum complete explanation (EVERY definition). Counts for BQA include both members of each positive–negative pair; FALSE labels use paired-positive structural provenance. Tags are not mutually exclusive.

Primitive tag	**BQA** ($N = 6{,}032$)	**OEQA** ($N = 3{,}016$)
D (direct assertion)	6,032	3,016
H (hierarchy)	1,586	1,267
I (inverse property)	1,520	1,051
N (property chain)	78	42
R (domain/range)	664	640
S (symmetry)	40	124
T (transitivity)	0	10
F (functional property)	0	167
J (disjointness)	0	107
V (reflexive property)	0	256

recorded *minimum complete explanations* in the sampled questions; they do not measure the prevalence of the corresponding OWL constructs in the full source ontologies. A sensitivity analysis using the ANY definition includes additional questions whose tied minimum explanations contain a tag in at least one, but not every, minimum support.

For clarity, let $E = \{a_1, \ldots, a_m\}$ denote the complete ground-truth answer set for an OEQA query, and let $\mathcal{P}(a_i)$ be the recorded alternatives supporting answer a_i. A complete explanation is a union $P_1 \cup \cdots \cup P_m$ where $P_i \in \mathcal{P}(a_i)$. The complete-explanation annotation considers these alternative selections and retains every distinct tied minimum union. Identical supporting axioms appearing in two individual-answer proofs are counted once. The associated primitive tags

are counted over the selected union of axiom supports, not over concatenated answer-level tag strings.

Dataset Statistics

The finalized v1.1.1 sample is built from individual-centric contexts at 1-hop and 2-hop extraction depths. The total number of eligible entailments can differ by extraction depth and ontology. The reported figures here describe the *final sampled questions*. Each sampled OEQA question is associated with one balanced positive–negative BQA pair, subject to the task construction and stratified sampling described above.

Table 8. Final sampled v1.1.1 dataset composition. BQA counts are balanced, with positive/negative counts shown in parentheses. Each underlying question is supplied in NL, FS, and AR.

Ontology	Hop	BQA (T/F)	OEQA	Total
Family	1-hop	1254 (627/627)	627	1881
	2-hop	1254 (627/627)	627	1881
Pizza 100	1-hop	330 (165/165)	165	495
	2-hop	330 (165/165)	165	495
Pizza 250	1-hop	412 (206/206)	206	618
	2-hop	412 (206/206)	206	618
OWL2Bench	1-hop	978 (489/489)	489	1467
	2-hop	1062 (531/531)	531	1593
Total		**6032 (3016/3016)**	**3016**	**9048**

Across the four source ontologies, the final sample contains 3,016 positive BQA queries, 3,016 paired negative BQA queries, and 3,016 OEQA queries. Presenting each question in three representations yields 27,144 model inputs; the three-model evaluation has 81,432 responses.

Prompt Output Schema

Illustrative Benchmark Instance

Table 9. Illustrative ontology-reasoning scenario centered on `henry_jacobs_1849`; this pedagogical example is not asserted to be a member of the sampled v1.1.1 dataset. It shows multi-step reasoning involving a property-chain inclusion (`isBrotherOf` ∘ `isParentOf` ⊑ `isUncleOf`), a subproperty axiom (`isMotherOf` ⊑ `isParentOf`), and a domain restriction (`isBrotherOf` domain `Man`).

Ontology	**Inferences**	**Illustrative Question–Answer Pairs for Inference 4**

ABox:
`henry_jacobs_1849 isBrotherOf sarah_jacobs_1834`
`sarah_jacobs_1834 isMotherOf ada_rachel_heath_1868`
 	
TBox:
Classes:
`Man` ⊑ `Person`
Properties:
`isBrotherOf (domain: Man)`
`isMotherOf` ⊑ `isParentOf`
`isBrotherOf` ∘ `isParentOf` ⊑ `isUncleOf`
	1. `henry_jacobs_1849 rdf:type Man`
	2. `henry_jacobs_1849 rdf:type Person`
	3. `sarah_jacobs_1834 isParentOf ada_rachel_heath_1868`
	4. `henry_jacobs_1849 isUncleOf ada_rachel_heath_1868`
		SPARQL:
		
		1. `ASK WHERE { henry_jacobs_1849 isUncleOf ada_rachel_heath_1868 . }` → `TRUE`
		2. `ASK WHERE { henry_jacobs_1849 isUncleOf sarah_jacobs_1834 . }` → `FALSE`
		3. `SELECT ?x WHERE { henry_jacobs_1849 isUncleOf ?x . }` → `ada_rachel_heath_1868`

Natural Language: Henry is a brother of Sarah. Sarah is the mother of Ada. Every man is a person. The domain of `isBrotherOf` is `Man`: anyone occurring as the subject of an `isBrotherOf` relation is inferred to be a man. `isMotherOf` is a subproperty of `isParentOf`: every mother is also a parent. If someone is the brother of a person, and that person is a parent of another, then the first person is the uncle of the other.		**Natural Language:** 1. Is Henry the uncle of Ada? 2. Is Henry the uncle of Sarah? 3. Who is Henry the uncle of?

```
Binary Question Answering (BQA)
ANSWER: [TRUE or FALSE]
CONFIDENCE: [score from 0.0 to 1.0]

Instructions:
- In the ANSWER field, write only TRUE or FALSE.
- In the CONFIDENCE field, use 1.0 for complete certainty
  and 0.0 for a pure guess.

Open-Ended Question Answering (OEQA)
ANSWER: [local names only, separated by semicolons]
CONFIDENCE: [score from 0.0 to 1.0]

Instructions:
- In the ANSWER field, use local names only
  (e.g., Person; U0C4; caroline_lavinia_tubb_1840).
- Provide all possible answers.
- In the CONFIDENCE field, use 1.0 for complete certainty
  and 0.0 for a pure guess.
```

Figure 4. Output schema used for binary and open-ended question answering prompts.