applied sciences

MDPI

Article

DL-ReasonSuite: A Benchmark for Evaluating Description Logic Reasoning in Large Language Models

Müge Oluçoğlu * and Okan Bursa

Computer Engineering Department, Faculty of Engineering and Architecture, Izmir Bakircay University ; okan.bursa@bakircay.edu.tr

* Correspondence: muge.olucoglu@bakircay.edu.tr

Supplementary Materials: DL-ReasonSuite

S1. Overview and Scope

This supplementary document provides full methodological details for DL-ReasonSuite, including: (i) task schemas and prompt formats; (ii) metric definitions and aggregation; (iii) scoring rules and normalization; (iv) reproducibility and artifact checklist; and (v) recommended extensions for robustness and calibration.

S2. Benchmark Structure

S2.1 Tracks and Task Types

DL-ReasonSuite consists of three tracks and seven task types:

- **DL-Core Track (Formal Logic):**
  1. **Consistency Checking (COH):** Determine if an ontology/KB is satisfiable (consistent).
  2. **Subsumption (SUB):** Determine whether $C \sqsubseteq D$ is entailed by the ontology.
  3. **Instance Checking (INS):** Determine whether $a : C$ is entailed by the ontology.
- **DL-Query Track (SPARQL):**
  4. **SPARQL SELECT Answering (QRY):** Return bindings for variables under entailment closure.
- **DL-Bridge Track (NL↔OWL):**
  5. **NL→OWL Parse Validity (N2O-P):** Produce a syntactically valid OWL axiom/expression.
  6. **NL→OWL Semantic Equivalence (N2O-S):** Produce a logically equivalent formalization.
  7. **Entailment QA grounded in OWL (EQA):** Answer NL questions using ontology entailment.

S2.2 Task Distribution (Reference Configuration)

Reference configuration used in the report: **seed=13, 4,740 tasks**.

- DL-Core: 1,600 tasks (COH, SUB, INS)
- DL-Query: 1,200 tasks (QRY)
- DL-Bridge: 1,940 tasks (N2O-P, N2O-S, EQA)

Check for updates

Academic Editor: Christos Bouras

Received: 17 January 2026
Revised: 2 February 2026
Accepted: 5 February 2026
Published: 11 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

S3. Data Model and Formal Semantics

S3.1 Knowledge Base Representation

Each task instance is derived from a small OWL-like knowledge base (KB):

$$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$$

where $\mathcal{T}$ is the TBox (schema axioms) and $\mathcal{A}$ is the ABox (assertions about individuals). Class expressions and axioms are generated from a constrained grammar to maintain tractability and coverage.

S3.2 Reasoning Regime

Gold labels are determined under an entailment regime consistent with OWL RL-style closure reasoning (or a reasoner capable of OWL RL / OWL 2 profiles). The benchmark is compatible with:

- materialization-based reasoners (rule closure),
- reasoner-backed query answering (entailment regimes), and
- equivalence checks via mutual entailment of axioms.

S3.3 Formal Targets per Task Type

- COH: decide satisfiable($\mathcal{K}$).
- SUB: decide $\mathcal{K} \models C \sqsubseteq D$.
- INS: decide $\mathcal{K} \models C(a)$.
- QRY: compute $\text{Ans}(\mathcal{K}, q)$ for SPARQL SELECT query q (set of bindings).
- N2O-P/N2O-S: given NL utterance u, output OWL axiom α.
- EQA: given NL question x, output correct entailment-derived answer y.

S4. Prompt Formats and Task Schemas

S4.1 Canonical Input Blocks

Every task prompt contains:

1. **Task header**: task type identifier and required response format.
2. **Ontology block**: axioms in a target serialization (e.g., Manchester, Functional Syntax, or simplified triples).
3. **Query block**: a yes/no question, class relation, instance relation, SPARQL query, or NL instruction.
4. **Output constraint**: "Answer with {YES, NO} only" or "Return JSON bindings", etc.

S4.2 COH (Consistency Checking) Schema

Input: ontology axioms $\mathcal{T}$ and $\mathcal{A}$ (small KB).
Output: `CONSISTENT` or `INCONSISTENT` (or `YES/NO` per config).
Example output rule: accept only a single token (case-insensitive).

S4.3 SUB (Subsumption) Schema

Input: ontology axioms plus pair of class expressions (C, D).
Output: `YES` iff $\mathcal{K} \models C \sqsubseteq D$, else `NO`.

S4.4 INS (Instance Checking) Schema

Input: ontology axioms plus individual and class expression (a, C).
Output: `YES` iff $\mathcal{K} \models C(a)$, else `NO`.

https://doi.org/10.3390/app1010000

S4.5 QRY (SPARQL SELECT) Schema

Input: RDF-like triples (explicit) plus ontology axioms (for closure), and a SPARQL SELECT query.
Output: set of bindings for query variables.
Canonical output format:

```
[{"?x":"IRI_or_label","?y":"IRI_or_label"}, ...]
```

Bindings are evaluated as a set (order-insensitive) after normalization.

S4.6 N2O (NL→OWL) Schemas

N2O tasks come in two layers:

- **N2O-P (Parse)**: output must parse under the target OWL grammar.
- **N2O-S (Semantic)**: output must be logically equivalent to reference formalization.

Input: NL sentence(s) describing a constraint or definition, optionally with vocabulary hints.
Output: one OWL axiom in a specified syntax (e.g., Manchester).

S4.7 EQA (Entailment QA) Schema

Input: ontology axioms plus NL question about entailment outcomes (e.g., membership, subclass, constraint).
Output: short answer (boolean, label, or small set).

S5. Metric Definitions

S5.1 Accuracy (Boolean / Single-label Tasks)

For COH/SUB/INS/EQA where gold is a single label:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left[\hat{y}_i = y_i\right]$$

All answers are normalized (trim, case-fold, alias mapping) before comparison.

S5.2 Syntactic Validity Rate (N2O-P)

Let $\text{parse}(\hat{\alpha}_i) \in \{0, 1\}$ indicate whether the model output parses under the specified grammar.

$$\text{Validity} = \frac{1}{N}\sum_{i=1}^{N}\text{parse}(\hat{\alpha}_i)$$

Parsing uses a deterministic grammar toolchain (e.g., OWLAPI parser for the target serialization).

S5.3 Semantic Equivalence Accuracy (N2O-S)

Semantic equivalence is tested by mutual entailment:

$$\hat{\alpha} \equiv \alpha \iff (\mathcal{K} \cup \{\hat{\alpha}\} \models \alpha) \wedge (\mathcal{K} \cup \{\alpha\} \models \hat{\alpha})$$

Equivalence is computed by a reasoner or by checking entailments under the same inference regime used for gold generation.

$$\text{EquivAcc} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left[\hat{\alpha}_i \equiv \alpha_i\right]$$

S5.4 Micro-F1 for Binding Retrieval (QRY)

For each query instance i, let gold bindings be G_i and predicted bindings be P_i (sets of variable assignments). Define:

$$\mathrm{TP} = \sum_i |P_i \cap G_i|, \quad \mathrm{FP} = \sum_i |P_i \setminus G_i|, \quad \mathrm{FN} = \sum_i |G_i \setminus P_i|$$

Then:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \mathrm{MicroF1} = \frac{2\,\mathrm{Precision}\,\mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Normalization for bindings: each binding is canonicalized by:

- stripping whitespace/quotes,
- resolving prefixed names to full IRIs (if prefix map provided),
- label canonicalization when labels are permitted (optional mapping table),
- sorting variable keys within a binding record.

S6. Aggregation: From Task Metrics to Track Scores to DL Score

S6.1 Task-Type Score

Each task type t produces a score Score_t:

- COH/SUB/INS/EQA: accuracy in %.
- QRY: micro-F1 in %.
- N2O-P: validity in %.
- N2O-S: equivalence accuracy in %.

S6.2 Track Score

Each track score is the arithmetic mean of its task-type scores:

$$\mathrm{TrackScore}(\text{DL-Core}) = \frac{\mathrm{COH} + \mathrm{SUB} + \mathrm{INS}}{3}$$

$$\mathrm{TrackScore}(\text{DL-Bridge}) = \frac{\text{N2O-P} + \text{N2O-S} + \mathrm{EQA}}{3}$$

$$\mathrm{TrackScore}(\text{DL-Query}) = \mathrm{QRY}$$

S6.3 DL Score

The DL score is an average of the three track scores, optionally weighted by task count or by design preference. In the reference report we use an unweighted mean of track scores:

$$\mathrm{DLScore} = \frac{\text{DL-Core} + \text{DL-Query} + \text{DL-Bridge}}{3}$$

Alternative (task-count weighted):

$$\mathrm{DLScore}_{\mathrm{wt}} = \frac{1600 \cdot \text{DL-Core} + 1200 \cdot \text{DL-Query} + 1940 \cdot \text{DL-Bridge}}{4740}$$

S7. Scoring Rules and Edge-Case Handling

S7.1 Boolean Output Normalization

Accepted tokens for YES/NO tasks:

- YES: `YES`, `Y`, `TRUE`, `ENTAILS`
- NO: `NO`, `N`, `FALSE`, `NOT_ENTAILED`

Any other output is marked incorrect (strict mode) or optionally treated as abstain.

S7.2 Consistency Labels

Accepted tokens:

- CONSISTENT: `CONSISTENT`, `SAT`, `SATISFIABLE`
- INCONSISTENT: `INCONSISTENT`, `UNSAT`, `UNSATISFIABLE`

S7.3 OWL Parsing and Grammar Targets

A model output is considered parsable if:

1. it matches the expected serialization type (Manchester/Functional/Turtle),
2. it passes a deterministic parser without errors,
3. all referenced symbols exist in the provided vocabulary (optional strictness).

If parsing fails, N2O-P=0 and N2O-S=0 for that instance.

S7.4 Semantic Equivalence Checks

We check mutual entailment of axioms, with safeguards:

- **Timeout**: impose a maximum reasoning time per instance.
- **Inconsistency guard**: if $\mathcal{K}$ is inconsistent, equivalence checks are performed under a consistent repair or the instance is excluded (configurable). Default: ensure all N2O instances use consistent $\mathcal{K}$.
- **Normalization**: rewrite trivially equivalent constructs (e.g., commutativity of conjunction) before reasoning (optional).

S7.5 SPARQL Binding Canonicalization

Bindings are canonicalized before set comparison:

- Unify IRI forms (angle brackets, prefix forms).
- Convert literals to canonical lexical forms if typed literals are used.
- Enforce a stable ordering of variables inside each binding record.

After canonicalization, equality is set-based.

S8. Implementation Notes (Python Harness)

S8.1 Directory Layout (Recommended)

```
dl-reason-suite/
  tasks/
    dl_core_consistency.jsonl
    dl_core_subsumption.jsonl
    dl_core_instance.jsonl
    dl_query_sparql.jsonl
    dl_bridge_nl2owl_parse.jsonl
    dl_bridge_nl2owl_semantic.jsonl
    dl_bridge_entailment_qa.jsonl
  schemas/
  tools/
  reports/
src/
  generate_suite.py
  score_predictions.py
  normalize.py
  reasoner_backend.py
  sparql_eval.py
output/
```

```
  predictions/
  scores/
  comparison_report.md
```

S8.2 JSONL Task Schema (Canonical)

Each task instance is a JSON object:

```
{
  "id": "DL-COH-000123",
  "task_type": "COH",
  "track": "DL-Core",
  "seed": 13,
  "ontology": { "tbox": [...], "abox": [...] },
  "prompt": "...",
  "gold": { "label": "CONSISTENT" },
  "meta": { "difficulty": "medium", "constructs": ["disjointness","exists"] }
}
```

For QRY:

```
"gold": { "bindings": [{"?x":"...","?y":"..."}, ...] }
```

For N2O:

```
"gold": { "owl_axiom": "SubClassOf(...)" , "serialization": "manchester" }
```

S8.3 Scoring Pseudocode

Boolean tasks (COH/SUB/INS/EQA).

```
for task in tasks:
  pred = normalize_label(model_output[task.id])
  gold = normalize_label(task.gold.label)
  correct = int(pred == gold)
  accumulate(correct)
accuracy = sum(correct)/N
```

N2O parse + semantic.

```
for task in n2o_tasks:
  out = model_output[task.id]
  if not owl_parse(out, syntax=task.gold.serialization):
    parse_ok = 0; sem_ok = 0
  else:
    parse_ok = 1
    sem_ok = int(mutual_entailment(task.ontology, out, task.gold.owl_axiom))
  accumulate(parse_ok, sem_ok)
```

SPARQL micro-F1.

```
TP=FP=FN=0
for task in sparql_tasks:
  P = canonicalize_bindings(model_output[task.id])
  G = canonicalize_bindings(task.gold.bindings)
  TP += len(P & G)
  FP += len(P - G)
  FN += len(G - P)
```

```
precision = TP/(TP+FP) if TP+FP else 1.0
recall    = TP/(TP+FN) if TP+FN else 1.0
micro_f1  = 2*precision*recall/(precision+recall) if precision+recall else 0
```

S9. Reproducibility Checklist

S9.1 Determinism

- Fix suite generation seed (e.g., seed=13).
- Fix any paraphrase sampling seed for NL templates.
- Version-lock reasoner backend and SPARQL engine.
- Log exact prompt templates and normalization rules.

S9.2 Required Artifacts

Minimum artifacts for full replication:

- Task JSONL files for all 7 task types.
- Gold labels (including gold bindings for QRY and gold axioms for N2O).
- Scoring scripts and reasoner interface (Python).
- Model outputs (JSONL predictions) with model ID and timestamp.
- Aggregated CSV/JSON of scores and a generated report.

S9.3 Reporting Requirements

For each evaluated model, report:

- task-type scores, track scores, and DLScore,
- strictness mode (label-only vs. allow-abstain),
- any timeouts and failure handling,
- SPARQL canonicalization rules,
- equivalence-check method and reasoner.

S10. Recommended Extensions

S10.1 Harder DL Fragments

Extend generation to include:

- qualified cardinalities, nominals, and property chains,
- richer datatypes and constraints,
- larger TBox scaling to test compositional reasoning limits.

S10.2 Robustness Battery

Add perturbations:

- vocabulary renaming (alpha-renaming) to remove lexical cues,
- paraphrase of NL instructions,
- distractor axioms (non-interacting) to test focus and coherence,
- minimal edits that flip entailment (contrast pairs).

S10.3 Reliability and Selective Prediction

Enable:

- confidence extraction (if available) and calibration metrics (ECE/Brier),
- abstention strategies and risk-coverage curves.

S11. DL-ReasonSuite: Metric Glossary

- **COH**: ontology satisfiability/consistency correctness.
- **SUB**: correctness of subclass entailment decisions.

- **INS**: correctness of instance entailment decisions.
- **QRY**: micro-F1 on SPARQL SELECT binding sets.
- **N2O-P**: syntactic validity of generated OWL formalization.
- **N2O-S**: semantic equivalence of generated OWL formalization.
- **EQA**: entailment-grounded natural-language question answering.

https://doi.org/10.3390/app1010000