ANSWER SET CONSISTENCY OF LLMS FOR QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) sometimes contradict themselves when answering factual questions, especially when asked to enumerate all entities that satisfy the question. We formalize such self-contradiction as answer-set inconsistency: Given two enumeration questions whose answers satisfy a set-theoretic relation (equivalence, disjointness, containment, etc.), the LLM generates responses violating the relation. To diagnose this phenomenon, we create a benchmark dataset comprising tuples of enumeration questions over which a variety of set-theoretic relations hold, and propose related metrics to quantify answer-set inconsistency. Our evaluation of several state-of-the-art LLMs reveal pervasive inconsistency across models, even in cases where the LLM can identify the correct relation. This leads us to further analyze potential causes and propose mitigation strategies wherein the LLM is prompted to reason about such relations before answering, which lead to improved answer-set consistency. This work thus provides both a benchmark and a systematic approach for evaluating, explaining, and addressing answer-set inconsistency in LLM question answering, towards deriving practical insights to improve the reliability of LLMs.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated impressive capabilities not only in natural language understanding and generation, but also in question answering and other tasks involving complex reasoning (Tan et al., 2023; Ding et al., 2024; Li et al., 2024; Saxena et al., 2024; Sui et al., 2024). However, in the context of the latter tasks, they are prone to various types of contradiction (Ghosh et al., 2025; Calanzone et al., 2025) since they are not based on computational techniques that guarantee formal notions of consistency, soundness, etc.

One type of inconsistency that LLMs exhibit relates to factual question answering. Specifically, in the context of *enumeration questions*, which ask to list all entities that satisfy a question, the responses across different questions may exhibit inconsistency with respect to evident set-theoretic relations that hold between such questions. Take for example the four questions in Table 1. All such questions are enumeration questions: they expect a set of entities as answers. Let $[\![Q]\!]$ denote the expected set of answers for a question Q. Among these four questions, we can see that certain set-theoretic relations should be expected to hold, including *equality* ($[\![Q_1]\!] = [\![Q_2]\!] = [\![Q_3]\!] \cup [\![Q_4]\!]$), *containment* ($[\![Q_3]\!] \subseteq [\![Q_1]\!], [\![Q_4]\!] \subseteq [\![Q_1]\!], [\![Q_3]\!] \subseteq [\![Q_2]\!], [\![Q_4]\!] \subseteq [\![Q_2]\!]$, and such relations entailed by equality), *disjointness* ($[\![Q_3]\!] \cap [\![Q_4]\!] = \emptyset$), etc. However, the answers returned by a particular model may not satisfy these relations, even sometimes in cases where the model can recognize the correct expected relation. Specifically, let $[\![Q]\!]_M$ denote the set of answers enumerated by model M. Then, for example, given Q_1 and Q_3, when asked what relation holds between their answers, M may correctly recognize the containment of the latter in the former ($[\![Q_3]\!] \subseteq [\![Q_1]\!]$), but still enumerate $[\![Q_1]\!]_M$ and $[\![Q_3]\!]_M$ such that $[\![Q_3]\!]_M \not\subseteq [\![Q_1]\!]_M$, thus contradicting itself.

We formalize this issue as *answer-set consistency*, wherein *the answers for a tuple of factual enumeration questions generated by a particular model does satisfy the set-theoretic relations that are expected to hold for that tuple*. We further consider an *answer-set contradiction* whereby *the model enumerates answers that do not satisfy the set-theoretic relation it itself predicts*. Related topics have been well-studied in database theory literature wherein the notions of *query containment*, *equivalence*, etc., are textbook topics (Abiteboul et al., 1995), with decades of theoretical and practical re-

Table 1: Illustrative example with four enumeration questions

Q_1	What are the tributaries of the Madeira River?
Q_2	Which rivers and streams flow directly into the Madeira River?
Q_3	What are the right-bank tributaries of the Madeira River?
Q_4	What are the left-bank tributaries of the Madeira River?

sults covering a variety of query languages, data models, and reasoning formalisms. Unlike database systems, generative A.I. models are not designed to guarantee satisfaction of such formal relations in the answers they provide. Perhaps for this reason, answer-set (in)consistency has not been well-studied in the context of generative A.I. To the best of our knowledge, the closest work is that of Elazar et al. (2021), which evaluates consistency across paraphrased versions of cloze-style phrases, but these permit only a single answer. Hogan et al. (2025) briefly discuss this issue as "*coherence*", but do not investigate it further. Given that such models are increasingly used to answer users' enumeration questions, we believe the topic merits more analysis.

Research questions In this paper, we address the following research questions (RQs):

RQ1 To what extent do LLMs produce (in)consistent answer sets for enumeration questions?

RQ2 Can LLMs recognize the set-theoretic relations that exist between enumeration questions?

RQ3 Which set-theoretic relations cause the most difficulty for LLMs?

RQ4 Which key factors cause answer-set inconsistency in LLMs?

RQ5 Can we mitigate answer-set inconsistency with prompting strategies?

Contributions. This paper develops four main contributions: (1) We highlight and formalize the notion of answer-set consistency for enumeration questions. (2) To quantify answer-set consistency, we develop and release a novel handcrafted benchmark of 600 question quadruples, with 2,400 questions in total, where fixed set-theoretic relations are expected to hold between the questions of each tuple. We further propose measures to quantify answer-set consistency with respect to such a dataset. (3) We present an empirical analysis of 18 state-of-the-art LLMs on the benchmark. (4) We further present preliminary prompting strategies to mitigate answer-set inconsistency and evaluate their effectiveness.

Our methodology allows for a systematic analysis of how model size, architecture, and prompting strategy affect the logical coherence of LLM outputs, providing insights into both the strengths and limitations of current models for generating consistent answers to enumeration questions. Our findings reveal that LLMs exhibit significant answer-set inconsistency, the extent of which depends on the particular model and the particular relation (interestingly, newer or bigger models do not universally outperform older or smaller variants). Such inconsistency often occurs even when the model is able to correctly recognize the relation expected to hold. The prompting strategies we propose to mitigate this issue significantly improve consistency across all tested models.

The code, datasets and results of experiments are available on (anonymous) GitHub (authors, 2025).

2 Related Work

We present an overview of related works on datasets for question answering, consistency of LLMs, and query containment of databases.

Question answering datasets A wide range of datasets have been proposed for different flavors of question answering tasks (e.g., Tan et al. (2023); Wang et al. (2023); Lee & Kim (2024); Singhal et al. (2024); Wang et al. (2024); Zheng et al. (2024); Zhou & Duan (2024); Zhu et al. (2024); Allemang & Sequeda (2025); Ma et al. (2025)). These datasets and evaluation frameworks focus on accuracy with respect to a predefined ground truth. While such benchmarks are essential to verify the correctness and completeness of answers, our focus is rather on the internal consistency of the models, which we see as complementary to these existing datasets.

Consistency of LLMs Various works have addressed different notions of consistency for LLMs.

Ghosh et al. (2025) study the logical consistency of LLMs in a boolean fact-checking setting. They define the logical consistency of an LLM in terms of the boolean response $LLM(p)$ for some propositional statement p representing a fact. Specifically, they test *negation* ($LLM(\neg p) = \neg LLM(p)$, i.e., the response of the LLM to a negated fact should be the negation of the response to the original fact), *disjunction* ($LLM(p \vee q) = LLM(p) \vee LLM(q)$, i.e., the response to a conjunction of facts is the same as the conjunction of the responses to each fact), and *conjunction* ($LLM(p \wedge q) = LLM(p) \wedge LLM(q)$, as before). The authors further construct complex statements using the combinations of these three boolean operators $\{\neg, \vee, \wedge\}$. Their results showed that consistency improved with an increase in the number of model parameters.

Similarly, Calanzone et al. (2025) investigated whether a fine-tuning approach that integrates neuro-symbolic reasoning can enforce logical consistency in language models. Here, the authors checked also for negation consistency, but also included implication consistency, to check whether the LLM can follow implication rules. Empirical results demonstrate that the approach outperforms conventional fine-tuning and external solver-based methods in factual accuracy and logical consistency, while also generalizing effectively to unseen but related knowledge.

While these works address statements in propositional logic, Liu et al. (2024b) investigate logical consistency in LLMs as a property of the relationships between sets of items, rather than isolated predictions. Instead of evaluating responses in isolation, the model is queried across sets of comparisons to assess whether its preferences form a coherent structure. They define consistency through properties like transitivity, order-invariance, and semantic negation, applied across sets of judgments. Their experiments reveal that popular LLMs frequently violate these properties, even on simple domains, and that improving consistency correlates with better downstream task performance.

Jang et al. (2022) propose a behavioral definition of consistency, categorizing it into semantic, logical, and factual types. They introduce BECEL, a benchmark designed to evaluate consistency through controlled input changes across several NLP tasks. It tests logical consistency indirectly by checking whether a model's predicted labels remain logically coherent across natural language sentence pairs that are logically related.

Addressing factual questions and statements with a single response, Elazar et al. (2021) evaluate the consistency of (L)LMs across paraphrased versions of cloze-style phrases (i.e., facts with entities masked, such as "*___ is the largest tributary of the Madeira River*"). The authors found notable inconsistency in the models tested at that time.

Cohen et al. (2023) propose having one LLM cross-examine another LLM asking it diverse follow-up questions on the facts it states – for example, by rephrasing the fact and posing it as a question, asking about logical implications of the stated fact, etc. – checking if the response of the latter model remains consistent under cross-examination.

Although the consistency of LLMs with respect to boolean statements, facts, and cloze-style phrases has been studied, the consistency of their responses with respect to questions that permit sets of answers has, to the best of our knowledge, not been explored in depth.

Query containment Given a database D and a structured query Q, let $[\![Q]\!]_D$ denote the answers for the query on that database. Given two queries Q_1 and Q_2, query containment asks if, for any database D (whose schema is compatbile with Q), it holds that $[\![Q_1]\!]_D \subseteq [\![Q_2]\!]_D$, while *query equivalence* asks whether or not $[\![Q_1]\!]_D = [\![Q_2]\!]_D$ likewise holds for any database (Abiteboul et al., 1995). These problems have been studied for a variety of query languages and database models. While these problems are decidable for simple query formalisms like conjunctive queries under set semantics (akin to queries that only allow relational-style joins), they are undecidable for more expressive query languages (like the relational algebra, full SQL, etc.) (Abiteboul et al., 1995).

3 Answer-Set Consistency

In this section, we define the notion of answer-set consistency, describe the dataset we create to evaluate it, the metrics we use to quantify it, and the mitigation strategies we propose to address it.

Table 2: Relations for questions in our evaluation quadruples (implied relations in parentheses)

	Q_1	Q_2	Q_3	Q_4
Q_1	$=(\subseteq, \supseteq, \top)$	$=(\subseteq, \supseteq, \top)$	$\supseteq (\top)$	$\supseteq (\top)$
Q_2	$=(\subseteq, \supseteq, \top)$	$=(\subseteq, \supseteq, \top)$	$\supseteq (\top)$	$\supseteq (\top)$
Q_3	$\subseteq (\top)$	$\subseteq (\top)$	$=(\subseteq, \supseteq, \top)$	$\perp$
Q_4	$\subseteq (\top)$	$\subseteq (\top)$	$\perp$	$=(\subseteq, \supseteq, \top)$

3.1 Definition

We provide a formal definition for answer-set consistency and contradictions.

Let $\mathbf{E}$ denote the universe of entities from an arbitrary domain (in practice, strings). Given a natural language question Q, and its expected answer set $[\![Q]\!]$, we say that Q is an *enumeration question* if $[\![Q]\!] \subseteq \mathbf{E}$. Let M be a model (e.g., an LLM, potentially primed with a prompt), and again let $[\![Q]\!]_M \subseteq \mathbf{E}$ be the answer set generated by model M for question Q.

Given two sets A and B, let $A \perp B$ denote disjointness, shorthand for $A \cap B = \emptyset$, and let $A \top B$ denote overlap, shorthand for $A \cap B \neq \emptyset$. Let $\star \in \{=, \subseteq, \perp, \top\}$ denote a binary set relation.

Given two enumeration questions Q_1 and Q_2, we define a model M as *answer-set consistent* with respect to (Q_1,Q_2,$\star$) if and only if it holds that $[\![Q_1]\!] \star [\![Q_2]\!]$ and $[\![Q_1]\!]_M \star [\![Q_2]\!]_M$. Note that we do not need to know the ground-truth answer sets for $[\![Q_1]\!]$ or $[\![Q_2]\!]$ in order to diagnose answer-set consistency: rather we simply need to know that $[\![Q_1]\!] \star [\![Q_2]\!]$, which we can determine from static analysis of the questions (for example, a reader may be satisfied that $[\![Q_3]\!] \subseteq [\![Q_1]\!]$ holds from Table 1 without knowing any of the actual answers for either question). If this property does not hold, we call the model *answer-set inconsistent* with respect to (Q_1,Q_2,$\star$).

Given two enumeration questions Q_1 and Q_2, let $[\![Q_1, Q_2]\!]_M \subseteq \{=, \subseteq, \perp, \top\}$ denote the model's prediction (primed with an appropriate prompt giving the possible relations as alternatives) for what binary relations hold from Q_1 to Q_2. We define (Q_1,Q_2,$\star$) as an *answer-set contradiction* of M if and only if it holds that $[\![Q_1]\!]_M \star [\![Q_2]\!]_M$ and $\star \notin [\![Q_1, Q_2]\!]_M$. This is an internal contradiction. For reasons of space, we delegate discussion of this issue to Appendix F.

We also consider answer-set consistency and contradictions between sets constructed from multiple base sets via union, intersection, difference, etc. (e.g., for $Q_1 = Q_3 \cup Q_4$ in Table 1).

3.2 Dataset

In order to evaluate the *answer-set consistency* of various LLMs, and to address our research questions, we require a collection of pairs of questions of the form (Q_1, Q_2) such that $[\![Q_1]\!] \star [\![Q_2]\!]$ for $\star \in \{=, \subseteq, \perp, \top\}$. Given that, to the best of our knowledge, no such dataset exists, we design and construct such a dataset.

Given the manual cost of constructing datasets, instead of defining tuples $(Q_1, Q_2, \star)$, we rather define quadruples of the form (Q_1, Q_2, Q_3, Q_4) wherein different fixed relations hold between the different positions. Specifically, as illustrated in Table 1, we define this quadruple such that $Q_1 = Q_2 = Q_3 \cup Q_4$ and $Q_3 \perp Q_4$. This design is more efficient in terms of manual effort: per Table 2, each question appears in more than one relation. Conceptually, for one such quadruple, we can model 12 pairs of questions with a primary relation (and other implied relations).

We selected base questions (Q_1) that satisfied the following criteria: (i) the answers are drawn from a finite set of elements; (ii) the answers to the questions are non-empty and upper-bounded ($2 \leq [\![Q_1]\!] \leq 100$), where we require at least two answers to ensure both Q_3 and Q_4 are non-empty, and we avoid questions with an excessive number of answers that LLMs may struggle to generate; (iii) the questions and answers are objective. Example questions meeting these criteria are *Which countries are members of the European Union?*, *What are the tributaries of the Madeira River?*, etc.

Rather than constructing the question quadruples from the ground up, we identified that existing (open) knowledge graph question answering (KGQA) datasets provide us with a good starting point since they consist mostly of enumeration questions, and referring to the aforementioned criteria: (i) the answers are bounded to the domain of the knowledge graph; (ii) we can use the associated

structured query to ensure that the number of results are within a fixed bound; (iii) the questions and answers are mostly objective. Furthermore, some datasets provide paraphrases that cover Q_1 and Q_2 (equivalence). Unfortunately, only one KGQA dataset, namely QAWIKI (Moya Loustaunau & Hogan, 2025), provides containment relations, and only relatively few. No dataset provides the precise relations for the quadruples we need.

We constructed our dataset, which we call the Answer-Set Consistency Benchmark (ASCB), from three base KGQA datasets: LC-QUAD 2.0 (Dubey et al., 2019), QALD (Usbeck et al., 2024), and QAWIKI (Moya Loustaunau & Hogan, 2025). From each dataset, we evaluated the structured queries associated with questions to ensure that they are enumeration questions and to test that they satisfy the cardinality bounds. This created a candidate list of base questions (Q_1), which we manually reviewed and filtered down to a smaller set of selected questions that best meet our criteria, as well as additional desiderata prioritizing the "crispness"[1], diversity, fluency and objectiveness of questions. We enriched this candidate set with available paraphrases (for Q_2), and in the case of QAWIKI, with available containment relations (for Q_3). Given the base questions, we further used LLMs to generate candidate suggestions for Q_3 and Q_4, satisfying the relations we require. These candidate questions were manually revised, pruned, modified, etc., to ensure a high-quality dataset, resulting in 150 question quadruples from each dataset. We further added an additional source, which we call SYNTHETIC, of questions generated from scratch by LLMs. It is important to note that in many cases, the questions extracted merely served as "inspiration" for the final quadruple, even for the base query: the questions in many cases were heavily modified. Furthermore, suggestions by LLMs for Q_3 and Q_4, though useful, often did not satisfy the formal relations expected[2], and needed to be modified. As a final step, we used LLMs to revise the quadruples and suggest improvements for phrasing, corrections, etc., which were revised manually and applied if deemed suitable. The manual revision, curation and modification of questions was conducted by three of the authors.

The resulting ASCB dataset (available online (authors, 2025)) comprises 600 quadruples of hand-crafted questions in English – 2,400 questions in total – satisfying the relations in Table 2.

3.3 EVALUATION TASKS AND MITIGATION STRATEGIES

To evaluate the answer-set consistency of LLMs using our ASCB, we perform three distinct tasks, the latter two of which involve mitigation strategies to try to achieve better consistency. For all such tasks, the LLMs under test are configured to use the lowest possible temperature. The full prompts used for these tasks are provided in Appendix A for reference.

Task 1: Base evaluation Our first task evaluates the base answer-set consistency of the LLM under test, and involves two subtasks.

In Task 1.1, *Classification*, the models are presented pairs of questions (Q_i, Q_j) projected from each quadruple and asked to identify the first set-theoretic relation that holds, in the following order, from the answer set of Q_1 to that of Q_2: equivalence ($=$), contained by ($\subseteq$), contains ($\supseteq$), disjointness ($\perp$) and overlap ($\top$). Although any pair of questions must satisfy at least one such relation, we further add an *unknown* option in case the model cannot confidently determine the relation.

In Task 1.2, *Enumeration*, all $2{,}400$ questions are posed to the models independently, each within its own isolated context, requesting the LLM to enumerate all answers for the question. The prompt further instructs the model to return an exhaustive list separated by '|', to avoid additional text, to return '`idk`' in case it cannot answer, to return '`no answer`' if there is no answer, and to expand acronyms and use full names whenever possible. The corresponding answers are then collected.

Task 2: Classification-then-Enumeration (CtE) Following preliminary experiments on Task 1, we investigated a mitigation strategy that operates within a single conversational context. The model is first asked to identify the set-theoretic relationship between the question pair (or triplet, in the case of the difference relation), and is then prompted to provide answers to the individual questions. Our

[1]This relates to the domain being a crisp set: some questions, such as "*What are the jobs in finance?*" in LC-QuAD 2.0, do not clearly define a "crisp" base set for Q_1, and are excluded.

[2]Often the suggestions for Q_3 and Q_4 did not form a true dichotomy, for example: *What counties of North Dakota use the Mountain Time Zone?* and *What counties of North Dakota use a timezone other than the Mountain Time Zone?* is not a true dichotomy as some counties are in multiple time zones.

hypothesis is that this will improve answer-set consistency by allowing the model to first reason about the relation that the answer sets should satisfy before enumerating them.

Task 3: Oracle Here we first run Task 1.1, and if we detect an answer-set inconsistency with respect to $(Q_i, Q_j, \star)$, then we instruct the LLM to observe that the relation $\star$ holds from the answer set of Q_i to Q_j, and request the answers to be enumerated again. This task is an ideal version of Task 2, as it assumes an oracle that knows what relation holds between the questions. This will give us insights into what the model could achieve in Task 2 if it could always classify the relation correctly.

Question and relations considered Per Table 2, our quadruples provide a large number of potential binary relations to test. However, many such relations are redundant. Thus, we only consider each pair of questions once (skipping symmetric or inverse relations), and only test for the primary relation; specifically, we test for the following relations (R_1) $(Q_1, Q_2, =)$, (R_2) $(Q_3, Q_1, \subseteq)$, (R_3) $(Q_4, Q_1, \subseteq)$, (R_4) $(Q_3, Q_4, \perp)$. We also test a relation for a constructed set: (R_5) $(Q_4, Q_1 \setminus Q_3, =)$. Finally, to establish a referential result for non-determinism of the model over time, we test (R_*) $(Q_1, Q_1^*, =)$, where Q_1^* is the question Q_1 run at a different time. We will use this as a control later.

3.4 Evaluation Measures

We present various measures to evaluate the answer-set consistency of LLMs with respect to the previous tasks. The first two address the performance for enumeration, while the second addresses the performance for classification.

Classification accuracy We first quantify the ability of LLMs to correctly classify the set-theoretic relation that holds between the answer-sets of the questions. For each relation $R_1, \ldots, R_5$, the *classification accuracy* of a model M over n test instances is defined as $\propto R_i(M) = \frac{\text{\# correct classifications by } M}{n}$. We further denote by $\propto R(M)$ the average of $\{\!\{\propto R_1(M), \ldots, \propto R_5(M)\}\!\}$ (or equivalently, $\propto R(M) = \frac{\text{\# total correct classifications by } M}{5n}$).

Consistency rates For each pair of enumerated answer sets, we evaluate whether or not the six expected relations $R_1, \ldots, R_5, R_*$ are satisfied. Each such relation R_i is checked independently over n test instances ($n = 600$ for ASCB) of the form $T_i = \{(Q_{i1}, Q'_{i1}, \star_i), \ldots (Q_{in}, Q'_{in}, \star_i)\}$. For a given model M, and test instance $t \in T_i$ we define $\rho_M(t) = 0$ if M is answer-set inconsistent with respect to t, and $\rho_M(t) = 1$ otherwise. The *consistency rate* of M for R_i is then defined as $\%R_i(M) = \frac{1}{n}\sum_{t \in T_i} \rho_M(t)$. We further denote by $\%R(M)$ the average of $\{\!\{\%R_1(M), \ldots, \%R_5(M)\}\!\}$ (or equivalently, $\%R(M) = \frac{1}{5n}\sum_{t \in T_1 \cup \ldots \cup T_5} \rho_M(t)$). Note that we do not include R_* in the average as it is intended as a control.

Jaccard similarity The consistency rates consider each $(Q_i, Q'_i, \star_i)$ as a discrete result (1 for correct, or 0 for incorrect) – irrespective of how close the answer sets of the two questions are to satisfying the relation. To complement the first measure, we consider a second continuous measure that captures the degree to which an instance is satisfied. Specifically, we use the well-known *Jaccard similarity*, defined for sets S_1 and S_2 as $J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$. For a given model M, and test instance $t = (Q, Q', \star) \in T_i$, we define $\sigma_M(t) = J([\![Q]\!]_M, [\![Q']\!]_M)$. The Jaccard similarity pf a model M for a relation R_i is then defined as $\sim R_i(M) = \frac{1}{n}\sum_{t \in T_i} \sigma_M(t)$. We apply this measure for relations involving equivalence (R_1, R_5, R_*) and disjointness (R_4), where a score close to 1 is good in the former case, and poor in the latter case.

Hypotheses and significance testing We propose two hypotheses: (H_1) The **CtE** and **Oracle** strategies yield less answer-set inconsistency than **Base**. (H_2) LLMs with better general performance produce more consistent responses compared to older models.

To test statistical significance, we apply the one-sided McNemar test (Lachenbruch, 2014), which is suitable for paired nominal data. We adopt a significance level of $\alpha = 0.05$. The null hypothesis of no improvement is rejected if the one-sided p-value satisfies $p < 0.05$. In such cases, we conclude that the alternative strategy or model yields a statistically significant improvement in consistency.

Stochasticity control The causes of answer-set inconsistency in LLMs can be attributed to two main factors. (1) *Stochasticity* in the generation process, such as during token sampling, or latent variability, leading the model to produce different outputs for the same query across runs. (2) *Semantic misunderstanding*, where the model misinterprets or overlooks the logical or semantic relations among queries, resulting in violations of restrictions. We introduce R_* precisely to estimate and control for the effects of (1), which varies across models.

The impact specifically of the two factors on the answer-set inconsistency of model M for R_1 can be assessed by comparing $\%R_1(M)$ with $\%R_*(M)$ and $\sim R_1(M)$ with $\sim R_*(M)$: a large gap indicates (2) plays a smaller role, and a small gap indicates (1) plays a larger role. On the other hand, we can compare the difference of consistency for equivalence-based relations like R_*, R_1 with other relations like R_2, R_3, R_4 where (assuming stocasticity plays a similar role for all such relations), where a large gap indicates (2) plays a larger role, and a small gap indicates (2) plays a smaller role.

4 RESULTS

We now present the results of our experiments on 18 LLMs from the DeepSeek, Gemini, Grok, GPT, Llama and Mistral families. For all models, the temperature was set to the lowest possible value (zero, if possible). The results will be reported considering the overall dataset, that is, the dataset obtained by merging the four sources presented in Section 3.2. Results for individual datasets are available (anonymously) on GitHub (authors, 2025). For reasons of space, we provide additional visualizations that help to identify trends in these results in Appendix C.

4.1 CLASSIFICATION TASK

Appendix D reports the accuracy of different LLMs on the relation classification task across the relation restrictions R1–R5. The highest accuracy score in each column is indicated in bold. All evaluated LLMs are ranked in ascending order based on their Global Average scores reported by White et al. (2025) from A (worst) to R (best). Some open-source models are not included in this ranking; therefore, we loosely position them alongside models of similar parameter size.

The results reveal substantial variability among the evaluated models. Smaller-scale models such as Llama-3.1-8b exhibit poor performance across all relations, with accuracy often below 20%. Similarly, GPT-oss-20b, GPT-4.1-nano and Mistral-small:24b perform inconsistently, though notably the relations they struggle on sometimes differ. In contrast, larger models such as Gemini-2.5-pro, GPT-5-nano, GPT-5, GPT-o3, and Grok-3-mini achieve accuracies consistently above (or close to) 90% across all relations. Among the evaluated relations, R_1 and R_4 are the least challenging overall (though not for all models), while R_3 emerges as the most challenging, revealing the limitations of current models in reliably capturing the containment relation.[3] GPT-5 demonstrates the best performance over all relations, followed closely by Gemini-2.5-pro.

4.2 ANSWER-SET CONSISTENCY

Table 3 lists the results for answer-set consistency, considering the control relation, all five test relations, and the measures of consistency rate and Jaccard similarity. Please note that exceptionally for $\sim R_4$, a score lower to 0 for Jaccard similarity is better as the answer sets should be disjoint. We also show the percentage of questions that return `idk` or an empty answer.

In the base case, we see high rates of answer-set inconsistency for all relations, including even the control relation. The small gap between R_* and R_1 suggests that stochasticity plays an important role in answer-set inconsistency for equivalence across all models, even though temperature was lowered as much as possible in all such cases. However, the gap between R_*, R_1 and containment / ternary relations suggests that semantic misunderstanding is also a key cause of inconsistency for these latter relations in particular. The most inconsistent relation is the ternary relation of R_5, with the most consistent relation overall being the disjointness relation R_4. The best models again tend to be the large GPT models. There are notable improvements for the mitigation strategy Classify-then-

[3]Regarding why R_3 is more challenging than R_2, R_2 is based on Q_3, and R_3 is based on Q_4, where Q_4 questions tend to negate the restriction that Q_3 adds over Q_1, and this negation appears more challenging.

Enumerate (CtE), which (surprisingly) even outperforms the Oracle in many cases, perhaps due to forcing the LLM itself to reason about the questions when classifying the relation between them.

Table 3: Per-relation consistency (0-100 scale) for each model and strategy (Str).

ID	Model	Str	$\%R_*$	$\sim R_*$	$\%R_1$	$\%R_2$	$\%R_3$	$\%R_4$	$\%R_5$	$\%R$	$\sim R_1$	$\sim R_4$	$\sim R_5$	%idk
A	Llama-3.1-8b	Base	2.91	21.52	1.67	3.83	2.17	84.33	0.00	18.40	16.77	3.19	11.63	1.96
		CtE	–"–	–"–	16.33	**100.00**	33.83	68.17	5.00	44.67	56.03	9.74	50.18	25.08
		Ora.	–"–	–"–	12.67	15.50	12.33	61.83	1.83	20.83	45.49	11.26	33.16	0.21
B	GPT-oss-20b	Base	21.92	43.78	21.83	38.83	33.83	83.50	10.17	37.63	43.41	7.52	39.22	13.75
		CtE	–"–	–"–	67.17	**100.00**	73.17	84.67	38.00	72.60	79.39	10.73	76.77	37.04
		Ora.	–"–	–"–	58.33	80.67	92.33	95.33	62.00	77.73	76.50	3.97	84.43	13.33
C	GPT-4.1-nano	Base	58.43	68.06	28.00	47.33	37.83	63.33	3.83	36.06	50.83	17.90	39.07	12.71
		CtE	–"–	–"–	**97.33**	71.67	56.67	38.17	0.00	52.77	**97.57**	61.49	72.72	**80.87**
		Ora.	–"–	–"–	54.67	59.17	60.17	61.83	21.50	51.47	62.37	22.57	50.43	19.37
D	Mistral-small:24b	Base	43.00	63.98	44.33	50.67	45.00	51.00	1.50	38.50	64.42	34.41	51.91	29.42
		CtE	–"–	–"–	84.50	**100.00**	80.67	79.83	42.33	77.47	92.57	17.74	81.10	33.58
		Ora.	–"–	–"–	72.00	74.67	90.00	95.83	50.17	76.53	82.96	1.48	75.63	15.54
E	Llama-3.1-70b	Base	20.33	48.82	20.50	29.17	24.33	71.67	3.17	29.77	47.21	9.55	35.40	3.58
		CtE	–"–	–"–	80.33	**100.00**	86.33	93.50	56.17	83.27	90.71	2.95	86.21	26.91
		Ora.	–"–	–"–	61.00	72.67	82.17	93.50	50.67	72.00	75.29	3.37	79.57	5.04
F	Gemini-2.0-flash	Base	48.56	70.09	33.33	43.33	36.50	64.00	5.67	36.57	59.78	12.21	45.47	1.83
		CtE	–"–	–"–	88.17	**100.00**	94.50	64.67	54.00	80.27	93.84	27.65	92.57	40.66
		Ora.	–"–	–"–	79.00	84.17	92.00	96.67	73.83	85.13	82.33	1.37	90.66	2.96
G	GPT-4.1-mini	Base	28.00	53.78	33.83	48.33	38.67	56.33	5.17	36.47	62.75	15.74	49.84	4.96
		CtE	–"–	–"–	89.83	64.00	94.83	77.17	64.00	77.97	95.73	21.83	94.82	23.25
		Ora.	–"–	–"–	77.50	87.67	96.33	89.67	68.83	84.00	88.81	6.10	88.35	8.75
H	GPT-4o	Base	37.83	60.70	45.67	53.50	47.17	63.50	6.17	43.20	65.70	26.48	53.93	29.83
		CtE	–"–	–"–	97.33	99.33	**98.67**	38.00	32.50	73.17	58.67	61.27	97.91	66.46
		Ora.	–"–	–"–	85.31	91.82	94.82	86.48	67.95	85.28	73.38	12.11	73.90	33.93
I	GPT-4.1	Base	28.75	59.62	39.67	50.50	39.00	64.17	10.17	40.70	67.74	10.96	54.27	3.96
		CtE	–"–	–"–	91.83	70.33	97.50	87.00	74.83	84.30	97.54	11.74	**97.11**	13.75
		Ora.	–"–	–"–	82.50	91.83	94.50	95.33	74.83	87.80	92.07	3.09	92.52	6.62
J	Grok-3-mini	Base	37.67	63.09	34.33	52.50	44.33	87.83	23.17	48.43	63.30	5.50	57.31	8.21
		CtE	–"–	–"–	90.83	**100.00**	90.00	86.67	66.83	86.87	96.39	12.77	93.66	35.08
		Ora.	–"–	–"–	88.17	92.67	98.50	95.50	81.00	91.17	93.63	4.00	94.48	13.46
K	DeepSeek-V3.1	Base	38.92	61.44	32.50	45.50	39.83	55.83	7.17	36.17	56.96	16.65	46.37	11.33
		CtE	–"–	–"–	95.00	93.33	92.00	69.83	56.33	81.30	97.34	28.44	94.61	30.21
		Ora.	–"–	–"–	81.67	88.83	95.00	94.67	71.00	86.23	91.17	3.96	91.64	10.00
L	Gemini-2.5-flash	Base	38.00	64.54	33.67	50.83	43.17	86.17	23.17	47.40	59.72	2.88	53.20	7.37
		CtE	–"–	–"–	89.83	**100.00**	96.83	90.67	84.83	**92.43**	93.83	8.82	96.07	31.96
		Ora.	–"–	–"–	89.17	90.17	95.33	95.67	**85.50**	91.17	92.56	4.08	91.72	8.00
M	GPT-5-nano	Base	33.59	49.39	59.33	64.33	63.33	67.67	17.33	54.40	72.62	29.88	63.45	41.96
		CtE	–"–	–"–	77.83	**100.00**	76.83	19.50	0.50	54.93	78.16	**80.39**	76.08	31.00
		Ora.	–"–	–"–	84.17	83.17	97.17	77.83	53.33	79.13	89.65	21.40	85.36	41.25
N	DeepSeek-reasoner	Base	27.64	49.39	17.83	40.00	30.33	81.67	10.50	36.07	45.58	3.96	41.09	4.21
		CtE	–"–	–"–	71.50	93.33	87.33	68.33	53.17	74.73	76.87	30.82	88.53	29.00
		Ora.	–"–	–"–	68.33	75.67	86.33	93.00	58.33	76.33	82.21	5.37	83.57	6.25
O	Gemini-2.5-pro	Base	36.75	66.34	30.83	44.00	39.33	81.50	19.50	43.03	62.91	3.10	56.57	3.71
		CtE	–"–	–"–	85.33	**100.00**	88.00	93.00	71.33	87.53	89.78	6.22	86.16	33.00
		Ora.	–"–	–"–	77.33	80.17	92.17	**97.83**	74.83	84.47	81.97	2.06	85.79	10.62
P	GPT-5-mini	Base	**63.17**	76.14	65.33	63.67	68.17	61.50	14.50	54.63	77.02	35.93	67.13	47.08
		CtE	–"–	–"–	88.00	**100.00**	75.00	67.33	34.00	72.87	92.58	32.67	76.67	55.08
		Ora.	–"–	–"–	86.83	91.67	95.00	73.67	54.50	80.33	91.04	26.17	87.30	50.04
Q	GPT-o3	Base	31.25	58.88	34.33	50.83	38.50	86.50	19.33	45.90	62.68	5.47	54.42	8.83
		CtE	–"–	–"–	82.00	88.33	91.33	92.50	74.67	85.77	93.24	6.74	91.24	26.58
		Ora.	–"–	–"–	73.67	92.00	94.50	97.33	77.17	86.93	82.18	2.44	85.65	13.88
R	GPT-5	Base	58.50	**76.93**	60.67	63.83	65.00	73.83	21.50	56.97	78.33	22.05	67.92	32.04
		CtE	–"–	–"–	82.52	93.35	83.89	71.65	48.24	75.93	87.60	25.62	59.50	37.06
		Ora.	–"–	–"–	73.50	80.36	87.05	87.83	60.70	77.89	82.21	8.66	69.69	16.33

4.3 HYPOTHESIS TESTING

In Appendix E, we present an analysis of the statistical significance of our results. Regarding hypothesis H_1, that "The **CtE** and **Oracle** strategies yield less answer-set inconsistency than **Base**", this is confirmed by a p-value < 0.001 for almost all models for both strategies. Regarding hypoth-

esis H_2, the analysis reveals that prompting strategy significantly affects both absolute consistency and relative model rankings.

Table 4 shows significant positive correlations between $\%R_4$, $\%R_5$, and $\sim R_4$ and the average score of the models on the external benchmark (White et al., 2025). Models not in this benchmark are excluded. In particular, R_5 exhibits strong positive correlations across both measures. Since R_5 is the most challenging relation to identify, this finding supports H2, suggesting that consistency is more pronounced in the more complex answer-set tasks.

Table 4: Pearson correlations between external model scores and reasoning performance metrics. Asterisks indicate statistical significance: * $p < 0.05$, ** $p < 0.001$.

	$\%R_1$	$\%R_2$	$\%R_3$	$\%R_4$	$\%R_5$	$\%R$	$\sim R_1$	$\sim R_4$	$\sim R_5$
r	0.310	0.368	0.400	0.623*	0.800**	0.659*	0.418	-0.201	0.847**
p	0.281	0.195	0.156	0.017	0.001	0.010	0.137	0.492	<0.001

5 Discussion

We have highlighted and formalized the phenomenon of answer-set inconsistency in LLMs, proposed a dataset to evaluate it, defined various measures to quantify it, and presented the results for 18 contemporary LLMs.

Research questions We address the RQs presented in the introduction:

- **RQ1** LLMs exhibit high degrees of answer-set inconsistency for enumeration questions, with the particular degree depending on the model and relation (see Table 3).
- **RQ2** Contemporary large LLMs can recognize the set-theoretic relations that hold between enumeration questions with accuracy often about 90%, though smaller/older models struggle with many types of relations (see Appendix D).
- **RQ3** Binary equivalence relations appear to be the easiest relations for the LLMs to reason about, where they struggle most with containment relations and n-ary relations (see Appendix D and Table 3).
- **RQ4** Based on our control (R_*), much of answer-set inconsistency for equivalence relations is due to the stochastic nature of LLMs, whereas semantic misunderstanding plays a more dominant role for containment, disjointness and n-ary relations (Table 3).
- **RQ5** Answer-set inconsistency can be mitigated (i.e., improved by a wide margin, with statistical significance) by prompting strategies that ask the LLM to reason about the set-theoretic relations that holds between enumeration questions and their answer sets (Table 3).

The performance of LLMs for enumeration questions is remarkable considering that the technology was not designed for this sort of workload (unlike, say, databases). But their performance is *far* from perfect. Users should exercise caution when using contemporary LLMs to answer enumeration questions, and should not expect consistent responses (even for the same query at different times). Further research is required to understand and address this issue, potentially combining LLMs with other technologies that provide consistency guarantees.

Future Work The notion of answer-set (in)consistency could be extended further, for example, to include set cardinality (counts). More work is needed on how to improve the consistency of LLMs in this regard, which may include prompting strategies that instruct the LLM to reason about relations such as containment, disjointness, etc., inherent to such questions. Overall, one cannot expect an LLM by itself to provide consistency guarantees, so it is of interest to conduct further research on combining LLMs with technologies that provide such guarantees by design.

REFERENCES

Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. ISBN 0-201-53771-0. URL http://webdam.inria.fr/Alice/.

Dean Allemang and Juan Sequeda. Increasing the accuracy of LLM question-answering systems with ontologies. In Gianluca Demartini, Katja Hose, Maribel Acosta, Matteo Palmonari, Gong Cheng, Hala Skaf-Molli, Nicolas Ferranti, Daniel Hernández, and Aidan Hogan (eds.), *The Semantic Web – ISWC 2024*, pp. 324–339, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-77847-6. doi: 10.1007/978-3-031-77847-6_18.

Anonymous authors. Answer set consistency of LLMs for question answering, 2025. URL https://anonymous.4open.science/r/ASCS-7412/.

Diego Calanzone, Stefano Teso, and Antonio Vergari. Logically consistent language models via neuro-symbolic integration. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=7PGluppo4k.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. LM vs LM: Detecting factual errors via cross examination. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 12621–12640. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.778.

Junhua Ding, Huyen Nguyen, and Haihua Chen. Evaluation of question-answering based text summarization using LLM invited paper. In *2024 IEEE International Conference on Artificial Intelligence Testing (AITest)*, pp. 142–149, 2024. doi: 10.1109/AITest62860.2024.00025.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. LC-QuAD 2.0: A large dataset for complex question answering over Wikidata and DBpedia. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon (eds.), *The Semantic Web – ISWC 2019*, pp. 69–78, Cham, 2019. Springer International Publishing. ISBN 978-3-030-30796-7. doi: 10.1007/978-3-030-30796-7_5.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410. URL https://aclanthology.org/2021.tacl-1.60/.

Bishwamittra Ghosh, Sarah Hasan, Naheed Anjum Arafat, and Arijit Khan. Logical consistency of large language models in fact-checking. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SimlDuN0YT.

Aidan Hogan, Xin Luna Dong, Denny Vrandečić, and Gerhard Weikum. Large language models, knowledge graphs and search engines: A crossroads for answering users' questions, 2025. URL https://arxiv.org/abs/2501.06699.

Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. BECEL: Benchmark for consistency evaluation of language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3680–3696, 2022.

Peter A Lachenbruch. McNemar test. *Wiley StatsRef: Statistics Reference Online*, 2014. doi: 10.1002/9781118445112.stat04876.

Yanggyu Lee and Jihie Kim. Evaluating consistencies in LLM responses through a semantic clustering of question answering. *arXiv preprint arXiv:2410.15440*, 2024. doi: 10.48550/arXiv.2410.15440.

Nianqi Li, Jingping Liu, Sihang Jiang, Haiyun Jiang, Yanghua Xiao, Jiaqing Liang, Zujie Liang, Feng Wei, Jinglei Chen, Zhenghong Hao, and Bing Han. CR-LLM: A dataset and optimization for concept reasoning of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13737–13747,

Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.815. URL https://aclanthology.org/2024.findings-acl.815/.

Shicheng Liu, Sina J Semnani, Harold Triedman, Jialiang Xu, Isaac Dan Zhao, and Monica S Lam. Spinach: SPARQL-based information navigation for challenging real-world questions. *arXiv preprint arXiv:2407.11417*, 2024a. doi: 10.48550/arXiv.2407.11417.

Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vulić, and Nigel Collier. Aligning with logic: Measuring, evaluating and improving logical consistency in large language models. *arXiv preprint arXiv:2410.02205*, 2024b. doi: 10.48550/arXiv.2410.02205.

Jiatong Ma, Linmei Hu, Rang Li, and Wenbo Fu. Local: Logical and causal fact-checking with LLM-based multi-agents. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 1614–1625, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712746. doi: 10.1145/3696410.3714748.

Alberto Moya Loustaunau and Aidan Hogan. QAWiki: A knowledge graph question answering & SPARQL query generation dataset for wikidata. In *Wikidata Workshop, colocated with ISWC 2025*, 2025.

Yash Saxena, Sarthak Chopra, and Arunendra Mani Tripathi. Evaluating consistency and reasoning capabilities of large language models. In *2024 Second International Conference on Data Science and Information System (ICDSIS)*, pp. 1–5, 2024. doi: 10.1109/ICDSIS61070.2024.10594233.

Aryan Singhal, Thomas Law, Coby Kassner, Ayushman Gupta, Evan Duan, Aviral Damle, and Ryan Luo Li. Multilingual fact-checking using LLMs. In Daryna Dementieva, Oana Ignat, Zhijing Jin, Rada Mihalcea, Giorgio Piatti, Joel Tetreault, Steven Wilson, and Jieyu Zhao (eds.), *Proceedings of the Third Workshop on NLP for Positive Impact*, pp. 13–31, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.nlp4pi-1.2. URL https://aclanthology.org/2024.nlp4pi-1.2/.

Yuan Sui, Yufei He, Zifeng Ding, and Bryan Hooi. Can knowledge graphs make large language models more trustworthy? an empirical study over open-ended question answering. *arXiv preprint arXiv:2410.08085*, 2024.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. Can ChatGPT replace traditional KBQA models? an in-depth analysis of the question answering performance of the GPT LLM family. In Terry R. Payne, Valentina Presutti, Guilin Qi, María Poveda-Villalón, Giorgos Stoilos, Laura Hollink, Zoi Kaoudi, Gong Cheng, and Juanzi Li (eds.), *The Semantic Web – ISWC 2023*, pp. 348–367, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-47240-4. doi: 10.1007/978-3-031-47240-4_19.

Ricardo Usbeck, Xi Yan, Aleksandr Perevalov, Longquan Jiang, Julius Schulz, Angelie Kraft, Cedric Möller, Junbo Huang, Jan Reineke, Axel-Cyrille Ngonga Ngomo, Muhammad Saleem, and Andreas Both. QALD-10 – the 10th challenge on question answering over linked data: Shifting from DBpedia to Wikidata as a KG for KGQA. *Semantic Web*, 15(6):2193–2207, 2024.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*, 2023. doi: 10.48550/arXiv.2310.07521. URL https://arxiv.org/abs/2310.07521.

Yuxia Wang, Minghan Wang, Hasan Iqbal, Georgi N. Georgiev, Jiahui Geng, and Preslav Nakov. Openfactcheck: A unified framework for factuality evaluation of LLMs. *CoRR*, abs/2405.05583, 2024. doi: 10.48550/ARXIV.2405.05583. URL https://doi.org/10.48550/arXiv.2405.05583.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha V. Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger,

and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=sKYHBTAxVa.

Danna Zheng, Mirella Lapata, and Jeff Z Pan. How reliable are LLMs as knowledge bases? rethinking facutality and consistency. *arXiv preprint arXiv:2407.13578*, 2024. doi: 10.48550/arXiv.2407.13578.

Wenjie Zhou and Xiangyu Duan. Exploring and improving consistency in large language models for multiple-choice question assessment. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2024. doi: 10.1109/IJCNN60899.2024.10650668.

Qinrui Zhu, Derui Lyu, Xi Fan, Xiangyu Wang, Qiang Tu, Yibin Zhan, and Huanhuan Chen. Multi-model consistency for LLMs' evaluation. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2024. doi: 10.1109/IJCNN60899.2024.10651158.

A PROMPTS USED WITH LLMS

This section presents the three prompting strategies employed to evaluate the LLMs. The placeholder question denotes the text segment that is replaced with a question from the dataset.

A.1 *Zero-shot*

```
{question}
If you can't answer, return 'idk'.
If the question has no answer, return 'no answer'.
In the response, do not use abbreviations or acronyms, but spell out the full terms, i.e.  "United States of America" instead of "USA".
If the response contains numbers or digits, use Arabic numerals. For example, if the answer contains Star Wars V, indicate it with Star Wars 5. Do not use Roman numerals (such as V) or text (such as five).
Please, Return me an exhaustive list separated by the symbol '|' don't add any other text.
```

A.2 *Classification & Question*

```
You are given two questions, q1 and q2.
Your task is to determine the logical relationship between their respective sets of correct answers
Choose only one of the following relations:
- Equivalence: The answer sets of q1 and q2 are exactly the same.
- Contains: All answers to q2 are also answers to q1, but q1 includes additional answers.
- ContainedBy: All answers to q1 are also answers to q2, but q2 includes additional answers.
- Overlap: q1 and q2 share some, but not all, answers. Neither fully contains the other.
- Disjoint: q1 and q2 have no answers in common.
- Unknown: The relation between the answer sets cannot be confidently determined based on the given questions.
Here are the two questions:
q1: {q1}
q2: {q2}
Return **only** the name of the most appropriate relation from the list above.
Do **not** provide any explanation or commentary.
```

A.3 *Oracle*

Below is the oracle prompt used for the logical equivalence relation. For the other two other logical relations tested, the prompt is adapted by replacing the true_logical_relation placeholder and modifying certain parts of the sentence to ensure coherence.

```
Pay attention, the questions I asked you before are {true_logical_relation}, but you returned me different values.
In the response, do not use abbreviations or acronyms, but spell out the full terms, i.e. "United States of America" instead of "USA".
If the response contains numbers or digits, use Arabic numerals. For example, if the answer contains Star Wars V, indicate it with Star Wars 5. Do not use Roman numerals (such as V) or text (such as five).
Please, Return me an exhaustive list separated by the symbol '|' don't add any other text.
```

A.4 QUESTION PIPELINE

The multi-agent process consists of four steps:

- **Agent 1 (Validate Q1):** checks whether the input question Q_1 is well-formed and suitable; aborts if invalid.
- **Agent 2 (Generate Q2):** produces a companion question Q_2 consistent with the intent and constraints of Q_1.
- **Agent 3 (Generate Q3 & Q4):** creates Q_3 and Q_4 such that the set (Q_1, Q_2, Q_3, Q_4) satisfies the desired logical relations (e.g., equivalence, containment, or disjointness).
- **Agent 4 (Validate All):** verifies that Q_1–Q_4 jointly meet the required logical and formatting rules, corrects minor issues, and outputs the final validated set.

Overall, the workflow follows: validate $Q_1 \rightarrow$ generate $Q_2 \rightarrow$ generate $Q_3, Q_4 \rightarrow$ final validation. The process aborts if any step fails.

Agent1:

```
Evaluate whether the following question meets all of the following
criteria for acceptable answer types:

1. Returns a limited number of distinct answers (between 2 to 50).
2. Does **not** return a binary answer (e.g., "yes" or "no").
3. Does **not** return a single specific value (e.g., a date, name,
or number).
4. Does **not** require multiple answer dimensions (e.g., combining "
what" and "where" in the same question).

Question: "{question}"

Answer only with "Yes" or "No".
```

Agent2:

```
You are a rephrasing expert. Generate question Q2 which means exactly
 the same thing as the original question but uses different syntax
wording.

Original Question: "{question}"

Format your response as:
Q2: ...
```

Agent3:

```
Given the original question below, generate Q3 and Q4 to ensure:
- Q3 and Q4 are objective,
- answer set of Q1 equal to union of answer set of Q3 and Q4,
- answer set of Q3 disjoint from Q4,
- answer set of Q3 and Q4 subset of Q1 (more restrictive).

Original Question: "{question}"

Format your response as:
Q3: ...
Q4: ...
```

Agent4:

```
You are reviewing four related questions.

Q1: {q1}
Q2: {q2}
Q3: {q3}
Q4: {q4}

Tasks:
1. Check if Q2 is equivalent to Q1.
```

```
    2. Check if Q3 is a more restrictive version of Q1.
    3. Check if the union of Q3 and Q4 covers Q1 (Q1 = Q3 + Q4).
    4. Check all Q1~Q4 are objective questions.

    If everything is correct, answer only:
    "Yes"

    If not, return a corrected version in this format:
    Corrected Q1: ...
    Corrected Q2: ...
    Corrected Q3: ...
    Corrected Q4: ...
```

B DATASET CONSTRUCTION

To construct the question sets, we relied on four data sources: QALD (Usbeck et al., 2024), a handcrafted question-answering dataset for Wikidata and DBpedia; LC-QuAD 2.0, a large-scale question-answering dataset providing SPARQL queries with corresponding answers from both Wikidata and DBpedia[4] ; and a Synthetic dataset, generated entirely through LLMs.

QALD and LC-QuAD 2.0. To extract questions from QALD (Liu et al., 2024a; Usbeck et al., 2024) and LC-QuAD 2.0 (Dubey et al., 2019) datasets that conformed to our criteria, we employed a filtering step using an LLM-based pipeline. The model was instructed to assess each question for compliance with the inclusion conditions and, if satisfied, to generate the corresponding Q_2, Q_3, and Q_4 questions. The LLM model used for this task was GPT-4.1-2025-04-14 and the initial datasets contained 320 for QALD and $30,000$ for LC-QuAD 2.0. All generated sets were subsequently reviewed by three researchers to ensure adherence to the defined logical relationships and criteria. The final filtered QALD dataset contains 150 distinct question types per logical relation, for a total of 600 questions.

QAWiki. For the QAWiki dataset, many Q_1, Q_2, and Q_3 pairs were directly retrieved via SPARQL queries. However, only 54 Q_1–Q_3 pairs and 951 Q_1–Q_2 pairs were available. Therefore, for the 54 Q_1–Q_3 pairs, the corresponding Q_2 and Q_4 questions were manually constructed. Conversely, for the 951 Q_1–Q_2 pairs, Q_3 and Q_4 were manually derived. As with the other sources, this dataset was curated to include 150 distinct question types per logical relation, totaling 600 questions.

Synthetic. The Synthetic dataset was generated using *gpt-4.1-2025-04-14*, designed to produce 500 complete sets of Q_1, Q_2, Q_3, and Q_4 questions. Each set was then manually reviewed by three researchers to ensure correctness, adherence to the inclusion criteria, and to eliminate duplicates. Also, this dataset was curated to include 150 distinct question types per logical relation, totaling 600 questions.

The overall dataset contains 600 question types per logical relation (Q_1 to Q_4), yielding a total of $2,400$ questions.

C VISUALIZATIONS

In this section, we present visualizations of the main results for answer-set inconsistency across the models.

Figure 1 presents the Jaccard similarity $\sim R_1$, $\sim R_4$ and $\sim R_5$. Figure 1a illustrates that for $\sim R_1$ there is high consistency across models, particularly for CtE and Oracle, with Jaccard similarity values frequently exceeding 0.8. In contrast, Figure 1b demonstrates very low consistency, indicating that this is the most challenging relation for LLMs. Even when adjusting the prompting strategy, the improvement remains limited and not comparable to the performance observed for $\sim R_1$ and $\sim R_5$.

[4]DBpedia: https://www.dbpedia.org/

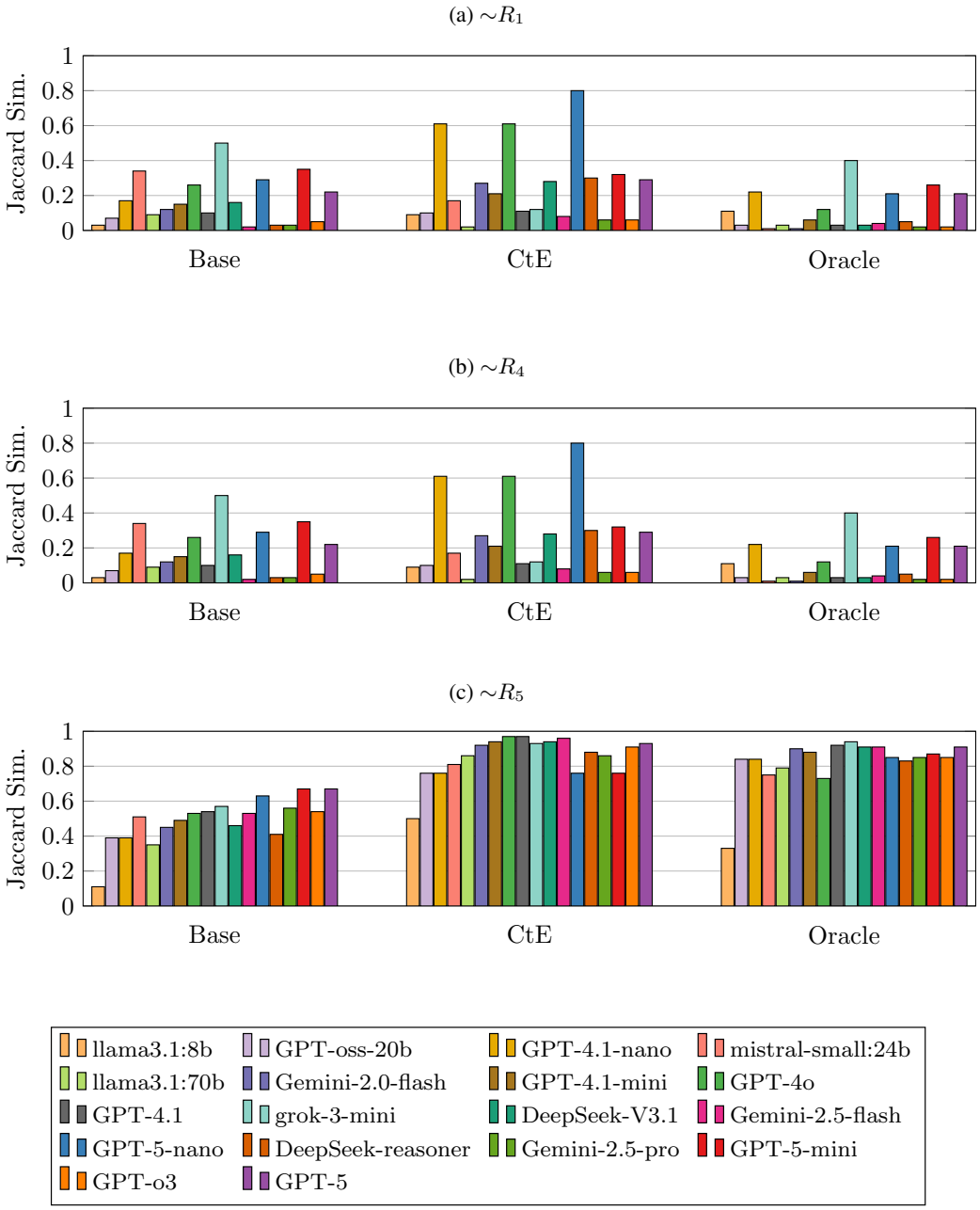

Figure 1: Visualization of answer-set (in)consistency across models

Finally, similar to $\sim R_1$, Figure 1c shows higher consistency for $\sim R_5$, with scores improving progressively from Base to Oracle. Moreover, for some models (such as GPT-4.1 and DeepSeek-V3.1), the CtE strategy proves more effective than Oracle.

Figure 2 shows an alternative view to the bar chart, using a line chart and adding the consistency rate. It is interesting to see how the **Base** strategy (blue dot) is almost always below **CtE** and **Ora**. Furthermore, the sorting of the models follows the benchmark White et al. (2025), but there is no linear increase in performance as the model performance scales (for all strategies), as we would expected. What happen instead is that models such as GPT-4.1-nano is better than DeepSeek-V3.1 for the $\%R4$.

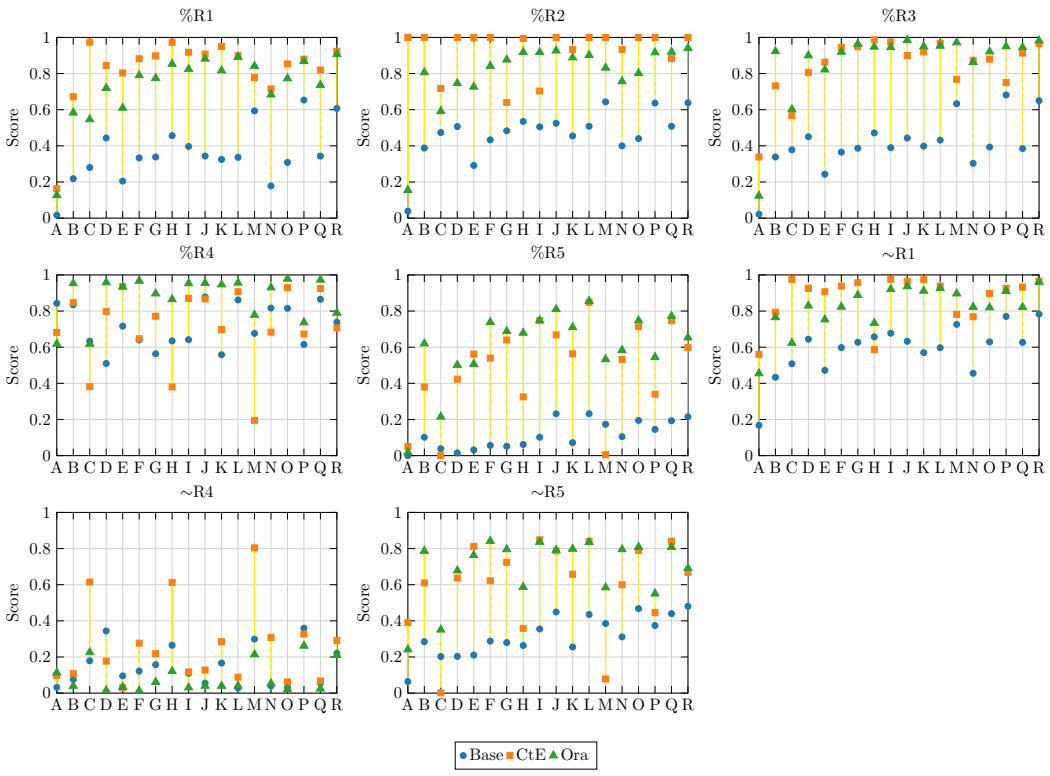

Figure 2: Per-relation consistency for each LLM and strategy. LLMs are represented by ID from Table 5.

Table 5: Accuracy (%) of different LLMs on the relation classification task across relations. In bold, the model that achieves the highest accuracy for the relation(s). $\propto R$ indicates mean accuracy.

ID	Model	$\propto R_1$	$\propto R_2$	$\propto R_3$	$\propto R_4$	$\propto R_5$	$\propto R$
A	Llama-3.1-8b	5.83	11.50	11.17	19.67	3.00	10.63
B	GPT-oss-20b	92.89	4.00	1.78	0.89	94.22	38.76
C	GPT-4.1-nano	68.67	41.11	42.67	66.00	0.00	43.29
D	Mistral-small:24b	91.33	0.22	1.10	**99.56**	13.33	41.91
E	Llama-3.1-70b	**97.78**	8.00	27.33	99.33	98.44	66.98
F	Gemini-2.0-flash	89.56	82.22	20.44	95.33	80.22	73.15
G	GPT-4.1-mini	91.33	34.89	19.56	99.33	96.44	68.31
H	GPT-4o	94.00	33.78	11.11	**99.56**	65.33	60.76
I	GPT-4.1	93.93	**98.83**	67.76	99.53	**99.30**	91.87
J	Grok-3-mini	90.44	95.11	96.67	97.78	97.56	95.11
K	DeepSeek-V3.1	97.33	36.44	20.67	**99.56**	91.33	69.87
L	Gemini-2.5-flash	85.56	96.00	94.44	97.56	95.56	93.82
M	GPT-5-nano	91.11	85.78	91.56	88.00	90.67	89.82
N	DeepSeek-reasoner	87.76	96.83	96.60	96.60	95.92	94.74
O	Gemini-2.5-pro	90.22	97.11	93.78	98.22	96.67	95.20
P	GPT-5-mini	89.33	97.78	98.00	94.89	96.00	95.20
Q	GPT-o3	91.33	97.33	97.33	94.67	93.56	94.84
R	GPT-5	89.33	98.00	**98.44**	96.44	94.44	**95.33**
Average		85.43	61.94	55.02	**85.72**	77.89	73.31

D CLASSIFICATION TASK RESULT

Table 5 reports in detail the accuracy of the different models in identifying the different relationship. GPT-5 proves to be the strongest model, consistently detecting different relationships. The most difficult relationship to recognize for all models war $R3$.

E STATISTICAL SIGNIFICANCE

In this section, we present the statistical significance of the results. Table 6 presents the McNemar p-value which shows the statistically significant improvements over the **Base** prompting with the **Classification-then-Enumerate** (CtE) strategy and **Oracle** (Orac.) on the overall dataset. Asterisks denote statistical significance: *** for $p < 0.001$, ** for $p < 0.01$, and * for $p < 0.05$. The empty cell indicates a p-value ≥ 0.05.

The table shows that the two proposed strategies are statistically effective in improving consistency on almost all tested models. The improvement does not occur in almost all cases for the $R4$ relationship for the **CtE** strategy, partuculary for models LLama-3.1-8b, GPT-oss-20b, Gemini-2.0-flash, GPT-4o, Grok-3-mini, GPT-5-nano, DeepSeek-reasoner and GPT-5.

The **Oracle** strategy, on the other hand, does not show a statistically significant improvement only for $R4$ and onl on LLama-3.1-8b and GPT-4.1-nano.

Table 6: McNemar p-value which shows the statistically significant improvements over the zero-shot prompting with the Classification-then-Enumerate (CtE) strategy and Oracle (Orac.) on the overall dataset. Asterisks denote statistical significance: *** for $p < 0.001$, ** for $p < 0.01$, and * for $p < 0.05$. The empty cell indicates a p-value ≥ 0.05

	CtE					Ora.				
LLM	**R1**	**R2**	**R3**	**R4**	**R5**	**R1**	**R2**	**R3**	**R4**	**R5**
Llama-3.1-8b	***	***	***		***	***	***	***		***
GPT-oss-20b	***	***	***		***	***	***	***	***	***
GPT-4.1-nano	***	***	***			***	***	***		***
Mistral-small:24b	***	***	***	***	***	***	***	***	***	***
Llama-3.1-70b	***	***	***	***	***	***	***	***	***	***
Gemini-2.0-flash	***	***	***		***	***	***	***	***	***
GPT-4.1-mini	***	***	***	***	***	***	***	***	***	***
GPT-4o	***	***	***		***	***	***	***	***	***
GPT-4.1	***	***	***	***	***	***	***	***	***	***
Grok-3-mini	***	***	***		***	***	***	***	***	***
DeepSeek-V3.1	***	***	***	***	***	***	***	***	***	***
Gemini-2.5-flash	***	***	***	***	***	***	***	***	***	***
GPT-5-nano	***	***	***			***	***	***	***	***
DeepSeek-reasoner	***	***	***		***	***	***	***	***	***
Gemini-2.5-pro	***	***	***	***	***	***	***	***	***	***
GPT-5-mini	***	***	***	***	***	***	***	***	***	***
GPT-o3	***	***	***	***	***	***	***	***	***	***
GPT-5	***	***	***		***	***	***	***	***	***

F ANWSER-SET CONTRADICTIONS

In the body of the paper, we have looked at answer-set inconsistencies with respect to gold-standard relations between questions. We further define *answer-set contradictions* as the case where – irrespective of the gold standard – the relation $\star$ predicted by a model M for questions Q_1 and Q_2 does not hold for the answer sets that M itself generates for Q_1 and Q_2. In these cases, the model contradicts itself. We describe here some measures and results that we extracted to analyze this issue.

F.1 MEASURES

We consider the following measures to quantify the self-contradictions of the LLMs in this setting.

Contradiction-free rates To complement classification accuracy and answer-set consistency, we measure whether a model is *internally consistent* with respect to the logical relation it *predicts*. Given two questions Q_i, Q_j, We say the case is *contradiction-free* iff the predicted relation is

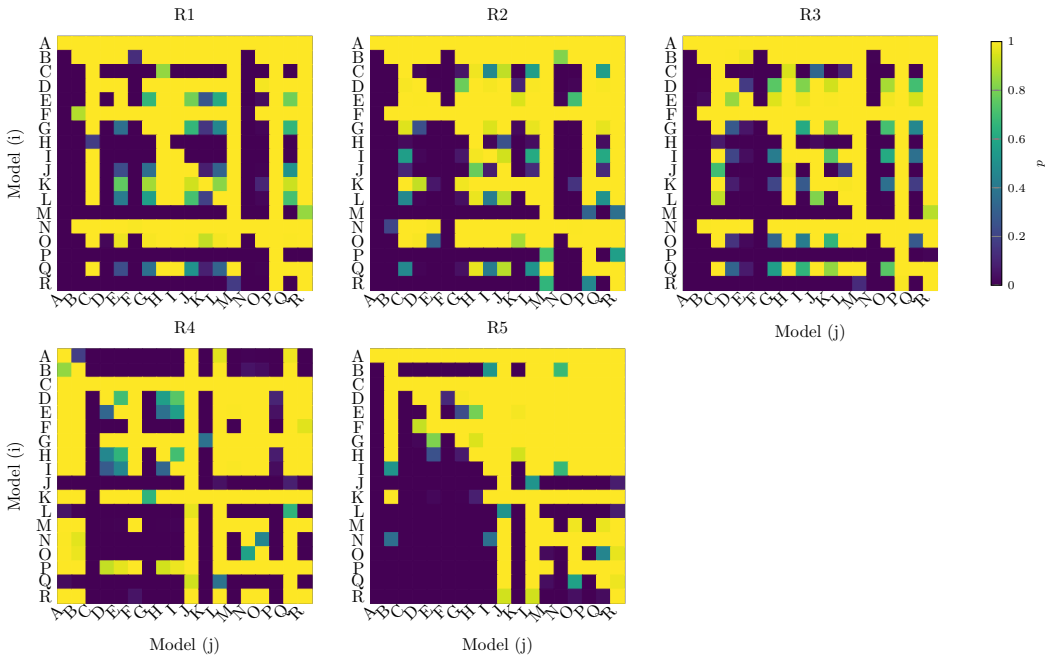

Figure 3: p-value heatmap of LLMs in overall datasets with zero-shot action.

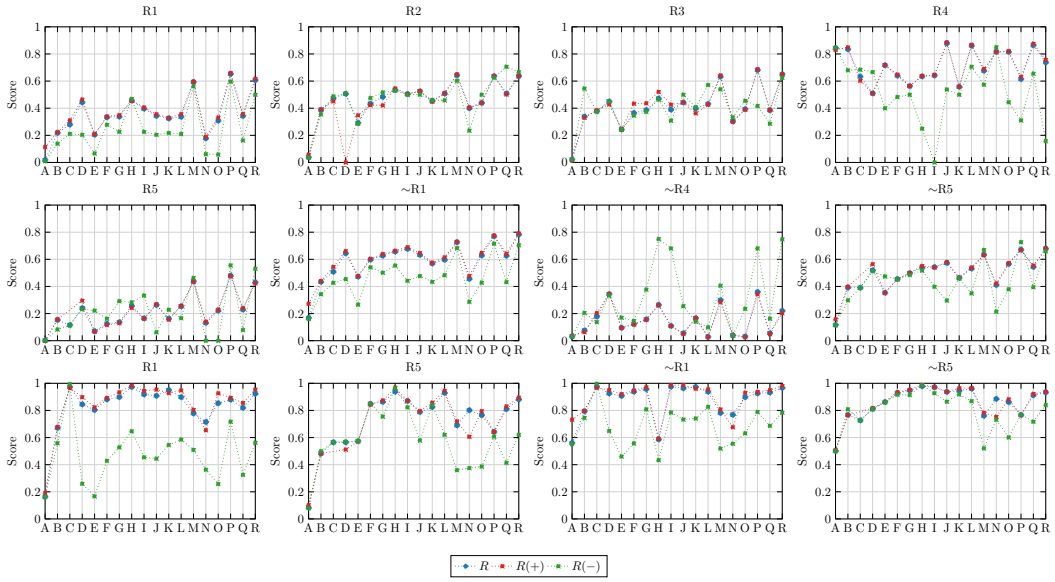

Figure 4: Comparison of R and R(+/–) consistency for Base (1st and 2nd rows) and CtE (3rd row).

satisfied by the model's *own* answers; otherwise a *self-contradiction* occurs. We define the *self-contradiction rate* as the percentage of cases where a model's predicted relation is not satisfied by its own answers. This measure distinguishes between models that are internally consistent but wrong, and those that are internally inconsistent, thereby offering a finer-grained perspective on model reliability. We denote such rates as $c(R)$.

Consistency by relation correctness $R(+/-)$. To further analyze model behaviour, we distinguish consistency depending on whether the logical relation between two questions is correctly identified with respect to the gold standard. We denote this by $R(+/-)$: the $(+)$ condition refers to cases where the predicted relation matches the gold relation, while $(-)$ refers to cases where it does not. For each relation, we report both the percentage of consistent cases (or the average Jaccard score) under $R(+)$ and under $R(-)$. This breakdown reveals how much consistency stems from correctly recognizing the relation versus how much persists even when the relation is misclassified.

F.2 Results

In Table 7, we present the results of the contradiction-free rates for five relations, while Table 9 presents the results for consistency across these five relations. Overall, we see that the models do tend to contradict themselves, i.e., the answers they return for questions do not respect the relation between the questions that they themselves predict. A lot of variance is seen across the models, with large models showing more consistency.

G GenAI Usage Disclosure

GenAI is used for text refinement, assist code debugging, dataset creation.

Table 7: Per-relation contradiction-free rates $c(R_i)$ (0–100 scale).

LLM	Act	$c(R_1)$	$c(R_2)$	$c(R_3)$	$c(R_4)$	$c(R_5)$
Llama-3.1-8b	CtE	81.00	13.83			82.67
	Base	93.83	86.00	87.00	28.67	97.00
GPT-oss-20b	CtE	66.50	89.33			40.33
	Base	26.17	41.00	33.50	80.50	16.17
GPT-4.1-nano	CtE	69.33	36.50			100.00
	Base	45.83	48.83	52.50	49.50	96.17
Mistral-small:24b	CtE	88.50	0.17			56.50
	Base	49.17	49.17	54.83	50.83	87.33
Llama-3.1-70b	CtE	82.33	9.00			53.67
	Base	22.67	68.50	61.83	71.83	4.67
Gemini-2.0-flash	CtE	88.50	46.83			46.27
	Base	37.33	44.33	60.67	64.17	22.17
GPT-4.1-mini	CtE	89.33	44.83			62.50
	Base	38.67	46.17	58.67	56.33	8.17
GPT-4o	CtE	96.50	37.09			77.67
	Base	46.00	49.33	53.33	63.83	38.17
GPT-4.1	CtE	92.33	69.67			74.67
	Base	42.56	50.52	50.69	64.71	10.73
Grok-3-mini	CtE	91.83	95.17			69.33
	Base	39.17	52.50	44.33	87.67	25.83
DeepSeek-V3.1	CtE	89.33	10.00			52.00
	Base	34.67	50.83	54.50	55.83	14.67
Gemini-2.5-flash	CtE	87.50	94.99			85.33
	Base	41.50	51.17	42.67	85.00	26.17
GPT-5-nano	CtE	77.67	89.33			8.83
	Base	58.33	61.33	62.67	66.00	24.67
DeepSeek-reasoner	CtE	65.33	92.00			54.00
	Base	27.24	41.62	31.13	79.19	15.74
Gemini-2.5-pro	CtE	90.67	95.83			74.67
	Base	38.33	44.00	39.83	81.67	22.83
GPT-5-mini	CtE	84.67	96.17			38.00
	Base	63.50	63.00	68.50	63.33	19.00
GPT-o3	CtE	84.33	85.67			77.50
	Base	39.17	49.67	39.50	85.00	25.67
GPT-5	CtE	91.33	98.17			63.00
	Base	60.67	63.17	64.67	76.00	26.50

Table 8: Per-relation consistency for $R(+)$ (0–100 scale).

LLM	Act	$\%R_1(+)$	$\%R_2(+)$	$\%R_3(+)$	$\%R_4(+)$	$\%R_5(+)$	$\sim R_1(+)$	$\sim R_4(+)$	$\sim R_5(+)$
Llama-3.1-8b	CtE	19.23	100.00			10.26	73.28		50.77
	Base	11.43	5.80	1.49	83.05	0.00	27.31	3.60	15.88
GPT-oss-20b	CtE	67.84	100.00			48.26	79.69		76.58
	Base	22.34	39.10	33.04	84.91	15.78	43.99	6.34	39.82
GPT-4.1-nano	CtE	96.51	66.23				96.72		
	Base	31.08	45.11	38.40	60.27		54.44	20.40	
Mistral-small:24b	CtE	89.82	100.00			51.16	95.08		81.67
	Base	46.46	0.00	42.86	50.92	29.58	66.11	34.42	56.45
Llama-3.1-70b	CtE	82.30	100.00			57.27	92.09		86.26
	Base	20.85	34.78	24.54	71.93	6.77	47.74	9.49	35.21
Gemini-2.0-flash	CtE	89.25	100.00			84.51	94.75		93.55
	Base	33.88	42.26	43.41	64.80	11.59	60.34	12.09	45.48
GPT-4.1-mini	CtE	93.42	65.32			87.39	97.17		95.15
	Base	34.92	42.11	43.65	56.40	13.02	63.97	15.52	49.89
GPT-4o	CtE	98.28	100.00			96.88	59.12		98.83
	Base	45.60	54.59	52.05	63.76	24.23	66.28	26.16	55.12
GPT-4.1	CtE	94.53	70.34			87.14	98.65		97.24
	Base	40.59	50.53	42.68	64.52	16.35	68.99	10.81	54.18
Grok-3-mini	CtE	95.42	100.00			79.52	98.67		93.89
	Base	35.57	52.63	44.14	88.59	26.88	64.69	5.06	58.07
DeepSeek-V3.1	CtE	92.81	100.00			85.71	95.98		96.43
	Base	32.93	44.81	36.29	55.87	15.65	57.50	16.67	46.40
Gemini-2.5-flash	CtE	94.79	100.00			94.57	95.60		96.53
	Base	35.65	51.04	42.66	86.62	25.69	61.50	2.68	53.96
GPT-5-nano	CtE	80.55	100.00			72.00	80.81		78.26
	Base	59.60	65.04	64.18	68.98	43.59	72.99	28.51	63.11
DeepSeek-reasoner	CtE	65.47	100.00			60.56	67.65		75.33
	Base	19.01	40.59	30.02	81.44	14.08	47.70	3.98	42.31
Gemini-2.5-pro	CtE	92.70	100.00			79.50	93.07		88.22
	Base	33.15	43.86	38.98	82.06	22.93	64.78	2.79	57.21
GPT-5-mini	CtE	89.35	100.00			64.34	93.71		76.66
	Base	65.93	63.70	68.71	63.05	47.64	77.59	34.29	66.87
GPT-o3	CtE	85.54	88.41			83.01	94.99		92.23
	Base	35.73	50.26	38.74	87.57	24.02	64.18	4.92	55.43
GPT-5	CtE	95.47	100.00			89.49	98.21		93.82
	Base	61.76	63.78	65.03	75.73	42.43	79.14	20.32	68.03

Table 9: Per-relation consistency for $R(-)$ (0–100 scale).

LLM	Act	$\%R_1(-)$	$\%R_2(-)$	$\%R_3(-)$	$\%R_4(-)$	$\%R_5(-)$	$\sim R_1(-)$	$\sim R_4(-)$	$\sim R_5(-)$
Llama-3.1-8b	CtE	16.20	100.00			8.05	55.25		50.09
	Base	1.06	3.58	2.25	84.65	0.17	16.12	3.09	11.50
GPT-oss-20b	CtE	55.88	100.00			50.00	74.50		80.90
	Base	13.89	35.56	54.55	68.00	8.33	34.31	20.51	29.83
GPT-4.1-nano	CtE	99.41	73.50			56.50	99.71		72.72
	Base	21.08	48.77	37.43	68.44	11.50	42.72	13.73	39.07
Mistral-small:24b	CtE	26.00	100.00			56.91	64.90		81.06
	Base	20.41	50.75	45.03	66.67	23.25	45.42	33.33	51.31
Llama-3.1-70b	CtE	16.67	100.00			57.52	46.07		86.09
	Base	6.67	28.70	24.26	40.00	22.22	26.59	17.00	47.39
Gemini-2.0-flash	CtE	42.86	100.00			85.11	55.66		91.69
	Base	27.78	47.54	34.61	48.28	16.24	54.11	14.63	45.43
GPT-4.1-mini	CtE	52.83	63.47			75.47	80.86		91.40
	Base	22.64	51.66	37.34	50.00	29.17	50.16	37.72	48.44
GPT-4o	CtE	64.71	100.00			96.45	43.37		98.65
	Base	46.88	53.01	46.49	25.00	28.30	55.42	75.00	51.75
GPT-4.1	CtE	45.45	70.00			82.35	78.50		92.73
	Base	22.58	50.00	30.86	0.00	33.33	44.15	68.06	39.79
Grok-3-mini	CtE	44.44	100.00			57.89	73.36		86.46
	Base	20.41	50.00	50.00	53.85	6.25	47.69	25.50	29.63
DeepSeek-V3.1	CtE	54.55	100.00			83.92	74.09		91.83
	Base	21.74	45.88	40.76	50.00	22.81	43.39	14.06	46.07
Gemini-2.5-flash	CtE	58.54	100.00			62.07	82.66		86.86
	Base	20.99	45.83	57.14	70.59	16.67	48.31	9.86	34.92
GPT-5-nano	CtE	50.91	100.00			36.00	51.92		52.07
	Base	56.25	60.23	54.00	57.35	46.30	68.30	40.60	66.91
DeepSeek-reasoner	CtE	36.36	100.00			37.50	55.45		73.20
	Base	6.15	23.53	33.33	85.00	0.00	28.64	3.95	21.45
Gemini-2.5-pro	CtE	25.76	100.00			38.64	63.17		60.09
	Base	5.88	50.00	45.45	44.44	0.00	42.85	23.57	37.92
GPT-5-mini	CtE	71.74	100.00			60.71	78.90		77.00
	Base	59.65	62.50	41.67	31.03	55.56	71.61	68.10	72.75
GPT-o3	CtE	32.50	86.36			41.38	68.66		71.69
	Base	16.28	70.59	28.57	65.52	7.89	43.24	16.28	39.50
GPT-5	CtE	56.25	100.00			62.07	78.48		84.04
	Base	50.00	66.67	62.50	15.79	53.12	70.50	74.82	66.04