LOGicalThought: Logic-Based Ontological Grounding of LLMs for High-Assurance Reasoning

Navapat Nananukul[1], **Yue Zhang**[2], **Ryan Lee**[1], **Eric Boxer**[1], **Jonathan May**[1], **Vibhav Giridhar Gogate**[2], **Jay Pujara**[1], **Mayank Kejriwal**[1]

[1] University of Southern California
[2] The University of Texas at Dallas

Abstract

High-assurance reasoning, particularly in critical domains such as law and medicine, requires conclusions that are accurate, verifiable, and explicitly grounded in evidence. This reasoning relies on premises codified from rules, statutes, and contracts, inherently involving defeasible or non-monotonic logic due to numerous exceptions, where the introduction of a single fact can invalidate general rules, posing significant challenges. While large language models (LLMs) excel at processing natural language, their capabilities in standard inference tasks do not translate to the rigorous reasoning required over high-assurance text guidelines. Core reasoning challenges within such texts often manifest specific logical structures involving negation, implication, and, most critically, defeasible rules and exceptions. In this paper, we propose a novel neurosymbolically-grounded architecture called LOGicalThought (LogT) that uses an advanced logical language and reasoner in conjunction with an LLM to construct a dual *symbolic graph context* and *logic-based context*. These two context representations transform the problem from inference over long-form guidelines into a compact grounded evaluation. Evaluated on four multi-domain benchmarks against four baselines, LogT improves overall performance by 11.84% across all LLMs. Performance improves significantly across all three modes of reasoning: by up to +10.2% on negation, +13.2% on implication, and +5.5% on defeasible reasoning compared to the strongest baseline.

Introduction

In recent years, the deployment of large language models (LLMs) for tasks demanding complex reasoning has marked a significant advancement in prompting approaches like Thought Generation, Program-Aid, and Self-Criticism (Wei et al. 2022a; Gao et al. 2023; Dhuliawala et al. 2024). However, LLMs have seen limited evaluation in *high-assurance domains* such as tax law and healthcare, where reasoning must be rigorous, verifiable, and explicitly evidence-based. (Groen and Patel 2014; Susskind 1986). While methods like Chain-of-Thought (CoT) prompting can generate reasoning chains in an end-to-end fashion, they lack the rigor of strict logic symbolic derivation, leading to potentially inaccurate conclusions (Lyu et al. 2023; Zhang et al. 2024; Zhou et al. 2021). Also, their reasoning chains are not grounded as formally verifiable claims, making it difficult to assess whether their reasoning was, in fact, correct, even when the answer itself is correct (Turpin et al. 2023; Lanham et al. 2023).

In this work, we introduce a neurosymbolic framework called LOGicalThought (LOGT) aimed at significantly enhancing the high-assurance reasoning capabilities of LLMs. LOGT addresses current challenges with high-assurance reasoning by grounding the original natural language problem into a *dual* neuro-symbolic context extracted from long-form guidelines: an ontologically-grounded *symbolic graph* capturing the high-level relationships in the text, and a machine-readable *logic program* formalizing the underlying defeasible rules and facts. By providing the dual context to the LLM, LOGT enables a more robust and transparent reasoning process. Specific contributions include:

- A formal specification for high-assurance reasoning that decomposes and extends the (originally, natural language) inference task via a *logic program mapping* that explicitly represents non-monotonic logical constructs.
- A novel neurosymbolically guided approach (LOGT) that addresses the challenges of high-assurance reasoning by constructing a *dual context* based on a symbolic graph representation and a non-monotonic logic program.
- A systematic benchmark construction methodology for augmenting ordinary inference datasets with scenarios that enable LLM testing on negation, implication, and defeasibility. Additionally, we introduce a new, handcrafted benchmark based on a complex text game.
- Extensive evaluation of LOGT's accuracy and reasoning performance on four expert-level benchmarks against four competitive LLM-based baselines. LOGT is found to produce higher-assurance reasoning traces compared to CoT, and is also more accurate.

LOGT follows a long tradition in AI of building systems capable of logical deduction that extends significantly beyond ordinary propositional and first-order logic (e.g., through defeasible and non-monotonic reasoning). While much recent success has been achieved with LLMs in informal "common sense" reasoning settings, and problems with clear reasoning pathways (such as high school mathematics), non-monotonic and defeasible reasoning has been less studied (see *Related Work*). We aim to address this

gap through both our proposed neurosymbolic approach and evaluation methodology.

Problem Formulation

Given a set of guidelines X and an associated scenario S providing additional context, the task is to evaluate the truth of a hypothesis H by reasoning from X and the facts stated in S. An example of all three can be found in the left column of Figure 1, based on a high-assurance task (tax law).

In this paper, we consider the following three (non-exhaustive) modalities of hypotheses for evaluating high-assurance reasoning: *negation, implication,* and *defeasible*. We formulate our targeted problem as predicting the logical relation $\hat{L} = \mathrm{M}(f(X, S, H))$ between H and (X, S).

Here, $\hat{L} \in \{\texttt{Entailment}, \texttt{Contradiction}, \texttt{Neutral}\}$ indicates whether H is entailed, contradicted, or neutral given X and S. The function f extracts logic program rules from the natural language inputs (X, S, H), decomposing the problem into three reasoning modes:

$$f_{\text{neg}}(X,S,H) = \{\, r_i\colon\ \neg\, \mathrm{q}_i \leftarrow \mathrm{p}_i \,\}$$

$$f_{\text{imp}}(X,S,H) = \{\, r_i\colon\ \mathrm{q}_i \leftarrow \mathrm{p}_i \,\}$$

$$f_{\text{def}}(X,S,H) = \{\, r_i\colon\ \mathrm{q}_i \leftarrow \mathrm{p}_i,\ r_j\colon\ \neg\, \mathrm{q}_i \leftarrow \text{exception}_j,\ \text{exception overrides default } p_i \,\}$$

Here, p_i is the condition for a rule, q_i is the conclusion, both extracted from (X, S, H), and exception_j is a condition for defeasible reasoning when a natural language exception overrides a default condition. In Appendix D, we provide a table with illustrative examples of all three modes.

In an evaluation setting, a model M is considered successful if its prediction $\hat{L}$ matches the truth-logical relationship L. Note that M and f do not necessarily have to be LLM-based, and also do not need to be explicitly specified. For example, ordinary LLM prompting implicitly models f, whereas a more neurosymbolic approach, such as our proposed approach, would aim to make f more explicit. The presence of logic rules $r_i, r_j \in R$ within a model's generated reasoning trace is considered direct evidence that its conclusion is grounded by the logic program generated from function f.

LogicalThought

The central idea of LOGT is to convert complex, unstructured reasoning tasks into a concise, ontologically grounded neurosymbolic representation, facilitating robust inference. Unlike much of the prior work (such as in the growing literature on graph-based retrieval augmented generation (Peng et al. 2024; Li, Miao, and Li 2024)), where the neurosymbolic context is typically some version of a knowledge graph, the neurosymbolic context extracted by LOGT from raw natural language inputs comprises both knowledge *and* logic. This transformation is essential in high-assurance domains, where reasoning must navigate the subtleties, exceptions, and complex dependencies codified in their guidelines. Such challenges often manifest as specific logical structures involving negation, implication, and defeasible reasoning (Xiu, Xiao, and Liu 2022). LLMs can be brittle in such non-monotonic settings, with their performance notably deteriorating when required to reason through extensive and intricate rule sets. (Li et al. 2023a,b).

Algorithm 1: The LOGT Grounded Reasoning Framework

Require: LLM; Guidelines X; Scenario S; Hypothesis H
Ensure: Predicted Label $\hat{L}$; Reasoning Trace T

— Stage 1: Symbolic Context Generation —
$X' \leftarrow \text{RulesSelection}(X, S, H)$ {Filter rules from X relevant to S, H}
$\mathcal{O}_{\text{rules}}, \mathcal{K}_{\text{instance}}, \mathcal{Q}_{\text{nl}} \leftarrow \text{SymbolicContext}(X', S, H)$ {Generate Ontology, Triples, and NL Queries}
$\mathcal{C}_{\text{sym}} \leftarrow \{\mathcal{O}_{\text{rules}}, \mathcal{K}_{\text{instance}}, \mathcal{Q}_{\text{nl}}\}$
— Stage 2: Logical Context Construction —
$\mathcal{F} \leftarrow \text{SynthesizeFacts}(\mathcal{K}_{\text{instance}})$
$\mathcal{R}_{\text{base}}, \mathcal{R}_{\text{override}} \leftarrow \text{SynthesizeRules}(\mathcal{O}_{\text{rules}})$
$\mathcal{Q}_{\text{Ergo}} \leftarrow \text{SynthesizeQueries}(\mathcal{Q}_{\text{nl}})$ {All synthesis via LLM with ErgoAI syntax template}
$\mathcal{P} \leftarrow \mathcal{F} \cup \mathcal{R}_{\text{base}} \cup \mathcal{R}_{\text{override}}$ {Assemble the full program}
$\mathcal{P} \leftarrow \text{SyntacticCorrection}(\mathcal{P})$ {Apply rule-based script}
$\mathcal{P}' \leftarrow \text{ErgoAI\_CompileAndFilter}(\mathcal{P})$ {Keep only compilable part of the program}
$\mathcal{A} \leftarrow \text{ErgoAI\_ExecuteQueries}(\mathcal{P}', \mathcal{Q}_{\text{Ergo}})$
$\mathcal{C}_{\text{log}} \leftarrow \{\mathcal{P}', \mathcal{A}\}$
— Stage 3: Hypothesis Evaluation —
$(\hat{L}, T^{\text{raw}}) \leftarrow \mathcal{M}(\mathcal{C}_{\text{sym}}, \mathcal{C}_{\text{log}}, H)$ {Grounded LLM call, see prompt in Appendix C.5}
$T_i \leftarrow \text{OrganizeTrace}(T^{\text{raw}}; \mathcal{T}_{\text{templates}})$ {Re-organize reasoning into six trace types according to a pre-defined template}
return $(\hat{L}, T)$

Symbolic graph context. To better guide LLMs, LOGT first constructs a *Symbolic Graph Context* ($\mathcal{C}_{\text{sym}}$) in a two-step process. First, to overcome the challenge of long guidelines, an LLM is prompted to select only the pieces of information needed to resolve the given scenario and hypothesis. Formally, given the raw input consisting of the guideline X, scenario S, and hypothesis H, an LLM first performs *guideline selection* to select only the subset of rules X' from X that are relevant for resolving H given S.

Next, a prompting step performs three related graph-extraction tasks simultaneously:

1. **Guideline Ontology** (O_{rules}): A graph-based representation of the selected rules from the guidelines-set X, capturing concepts, relations and hierarchical dependencies encoded in the selected guidelines.
2. **Knowledge Triples** ($K_{instance}$): A set of structured triples of the form (subject, predicate, object) encoding entities and relations in the scenario S and hypothesis H, constrained and typed according to the ontology O_{rules}.
3. **Queries** (Q_{nl}): The original hypothesis H decomposed into a set of precise, natural language questions that must be resolved to evaluate the hypothesis for each reasoning mode.

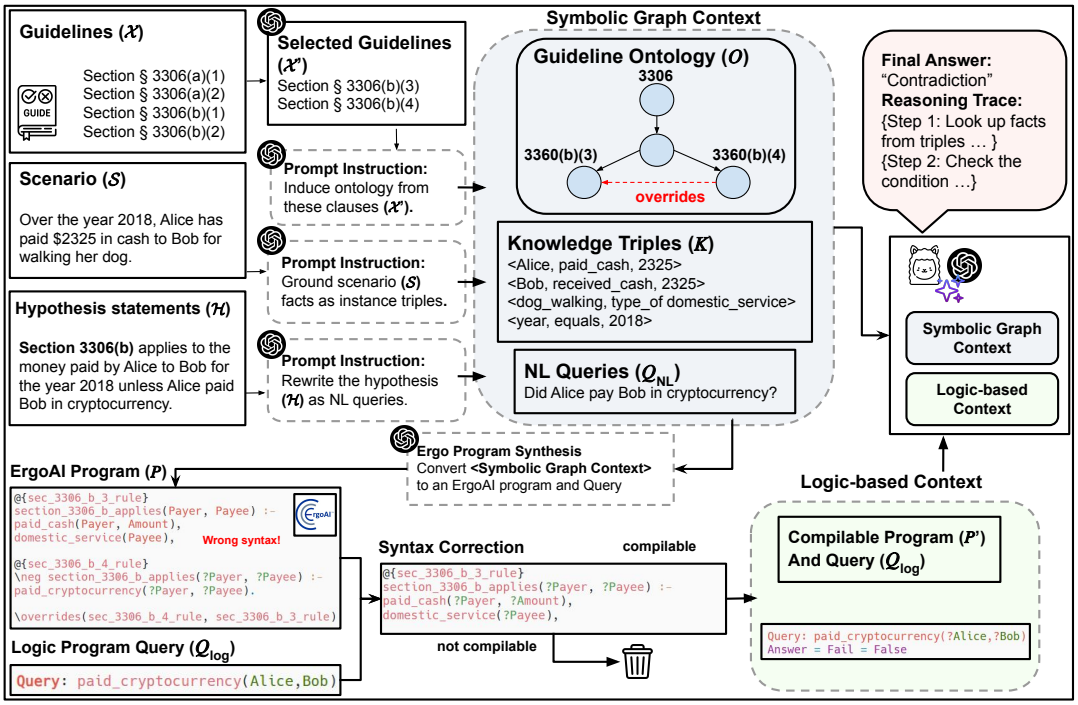


Figure 1: A schematized illustration of LOGT. An LLM initially processes raw guidelines, a scenario, and a hypothesis to create a *symbolic graph context*. This unified structure contains a rule ontology, factual knowledge triples, and a natural language query. This context acts as an input for two distinct reasoning approaches: (1) it provides rich context for guiding LLM reasoning, and (2) it serves as a blueprint for synthesizing a logic program. Compilable logic programs and queries are formalized into *logic-based context*, completing the dual neurosymbolic context underlying the approach.

Together, these three components form the complete *symbolic graph context* for the instance:

$$\mathcal{C}_{\text{sym}} = \{\mathcal{O}_{\text{rules}}, \mathcal{K}_{\text{instance}}, \mathcal{Q}_{\text{nl}}\}$$

Logic-based context. To capture the exceptions required for non-monotonic reasoning, LOGT also synthesizes a formal *logic-based context*. An LLM translates the symbolic graph context $\mathcal{C}_{\text{sym}}$ into a logic program: facts are derived from knowledge fragments $K_{instance}$, rules and (defeasible) overriding rules are generated from the ontology O_{rules}, and the natural-language questions Q_{nl} are converted into executable queries. Ideally, the objective of this stage is to produce a *machine-readable* knowledge base (KB) that stores facts and rules and correctly returns results when queried. We chose ErgoAI, based on RuleLog (Grosof, Kifer, and Fodor 2017), for our logic and reasoning engine because it is designed for higher-order logic and non-monotonic reasoning with built-in features that extend far beyond standard logic programming (Grosof et al. 2023). ErgoAI's native support for defeasible reasoning allows it to directly execute the kind of exception-driven logic that is fundamental to our high-assurance domains.

We utilize ErgoAI's built-in functionality for facts, rules, and defeasible reasoning with exception handling to validate the KB. The resulting ErgoAI program then undergoes a two-stage verification: (i) a rule-based script applies syntactic corrections; and (ii) the corrected program is compiled and executed in the ErgoAI reasoner. We retain only the compilable subset and execute the queries to obtain zero, one, or multiple answers to the queries. Formally, as also shown in Algorithm 1, the LLM maps $(O_{rules}, K_{instance}, Q_{nl})$ into an ErgoAI logic program P using a few-shot ErgoAI syntax rule bank (facts, rules, defeasible rules; illustrative examples are provided in Appendix E). The LLM is then instructed to perform the following mappings:

1. **Facts** (F): Convert $K_{instance}$ into atomic predicates; entity identifiers are canonicalized and constants are typed to match the predicate schema.
2. **Rules and Defeasible Rules** ($R_{base}, R_{override}$): Translate hierarchical relationships from O_{rules} into rules and defeasible (overriding) rules in ErgoAI format, encoding priorities from the ontology's partial order.
3. **Logic Program Queries** (Q_{log}): Convert Q_{nl} into formal logic queries with explicit variables.

Because the LLM-generated program P may contain errors, we apply automated syntactic fixes (e.g., terminators, variable normalization) and then compile the program in the ErgoAI reasoner. We construct the compilable program P' by retaining only facts and rules that successfully compiled and discard the rest. We then execute Q_{Ergo} against P' and store any resulting answers A. The final *logic-based context* is

$$C_{log} = \{P', A\}.$$

If query compilation fails, we fall back to using only the verified program, $\mathcal{P}'$, as the context. Finally, both the *symbolic context* ($\mathcal{C}_{\text{sym}}$) and the *logic-based context* ($\mathcal{C}_{\text{log}}$) are provided to ground the LLM for the final prediction. The prompt templates and configurations, including fixed seeds and temperature for reproducibility, are detailed in Appendix B and C.3. Because discarding non-compilable parts of the original program P could potentially remove important information, we include within the final context the entire verified program, P' rather than relying solely on the query answers A produced by P'.

Hypothesis evaluation using neurosymbolic context The final stage of LOGT involves using the two contexts constructed in the previous steps to ground LLMs for evaluating hypothesis H. We define $\mathcal{M}$ as the LLM being evaluated on the entailment task. $\mathcal{M}$ accepts the two contexts ($\mathcal{C}_{\text{sym}}$ and $\mathcal{C}_{\text{log}}$) and the original hypothesis as an input and is prompted to produce its predicted label (from one of *entailment, contradiction,* and *neutral*), and a reasoning trace:

$$\hat{L}, T^{raw} = \mathcal{M}(C_{sym}, C_{log}, H)$$

By grounding the LLM using both contexts, we hypothesize that it will enable the LLM to conduct a focused semantic evaluation of pre-compiled knowledge *vis-a-vis* open-ended reasoning over long-form documents. We provide the definition of `entailment` if the contexts corresponding to the hypothesis are derivable, `contradiction` if the negated context is derivable, and `neutral` otherwise.

To enable a standard comparison between the reasoning trace T^{raw} output by LOGT and that output by baselines such as CoT, we use an LLM-as-a-judge (Zheng et al. 2023; Gu et al. 2024) framework to re-organize each raw generated reasoning step into one of six types: *fact_lookup*, *apply_rule*, *check_condition*, *resolve_conflict/override*, *contradiction_detected*, and *conclude_label*. Subsequently, we analyze how the reasoning trace from LOGT is aligned with the symbolic and logical contexts by linking the *fact_lookup* and *apply_rule* trace types to their provenance in O_{rules} and $K_{instance}$.

Experimental Setup

Benchmarks. We evaluate LOGT on three established NLI benchmarks as well as a new hand-crafted benchmark incorporating domain expert guidelines, scenarios, and hypotheses. These benchmarks pose significant challenges, primarily requiring identification of pertinent evidence and inference over dense, complex textual guidelines. **ContractNLI** is a legal domain NLI dataset with document-level evidence in contract clauses and three labels: entailment, contradiction, and neutral (Koreeda and Manning 2021). **Statutory Reasoning Assessment (SARA)** offers scenarios for statutory reasoning using the US Internal Revenue Code, requiring the application of complex legal rules (Holzenberger, Blair-Stanek, and Durme 2020). The **Safe Biomedical NLI for Clinical Trials (BioMedNLI)** evaluates reasoning in medical contexts by classifying related statements as entailment or contradiction based on clinical trial reports (Jullien, Valentino, and Freitas 2024). Lastly, **Dungeons & Dragons NLI** is a custom benchmark proposed in this paper that is drawn from the Dungeons & Dragons rulebook, also featuring scenarios, rules, and test hypotheses. Detailed statistics for all benchmarks are provided in Appendix F.

Benchmark Enhancement for Non-Monotonic Reasoning. Existing entailment benchmarks typically contain hypotheses that are relatively simple and are based on associations with prior knowledge acquired during an LLM's pre-training owing to their longstanding status. While some benchmarks include examples involving negation, implication, or defeasible, they usually do so implicitly and in isolation, not as part of a unified categorization. Hence, current NLI benchmarks do not support proper evaluation of how well rival models handle non-monotonic reasoning modes. It remains unclear whether a model can robustly distinguish, for example, between a hypothesis that contradicts the premise due to negation, versus one that is overridden by an exception (defeasibility). Our preliminary analysis shows that current domain-expert NLI benchmarks are insufficiently challenging for advanced LLMs, suggesting potential exposure during model pre-training or inadequately complex tasks[1]. For newer LLMs, likely due to pre-training bias or due to inadequate complexity of the tasks themselves (Yuan et al. 2023).

We introduce a systematic augmentation pipeline (Figure 2) that organizes hypotheses under one of three reasoning modes: *negation, implication*, and *defeasibility*. As alluded to earlier, these modes (while not exhaustive to all high-assurance reasoning) are broadly encountered in human reasoning and decision-making, prove to be challenging for LLMs, and have minimal representation in current NLI benchmarks. For each original hypothesis hyp_i, the methodology can generate three hypothesis according to the following operative definitions:

1. **Negation** (h_i^{neg}) is a claim that directly contradicts a rule-derived fact, typically expressed in the form "[Subject] shall not [Relate to] [Object]," requiring the system to identify when a prohibition overrides an allowance.
2. **Implication** (h_i^{imp}) is a conditional claim that asserts a cause-effect relationship based on the rule, expressed as "If [Condition], then [Result]," requiring the system to reason over hypothetical or contingent outcomes.
3. **Defeasibility** (h_i^{deaf}) is a rule-based statement that includes an exception, typically in the form "[Rule] unless [Exception]," requiring the system to understand when a general rule may not apply due to exceptions.

We instantiate these transformations with eleven prompt templates, controlled by three symbolic reasoning modes and their corresponding target label. For example, as illustrated in Figure 2 , a negation template transforms an original hypothesis labeled as *Entailment* into one classified as *Contradiction*. In contrast, another template applying a double negation increases linguistic complexity while preserving the original label.

[1] In a pilot study, we found newer LLMs to achieve near-perfect accuracy (0.90–0.95) on existing NLI benchmarks just from basic prompting. We provide the preliminary results in Appendix G

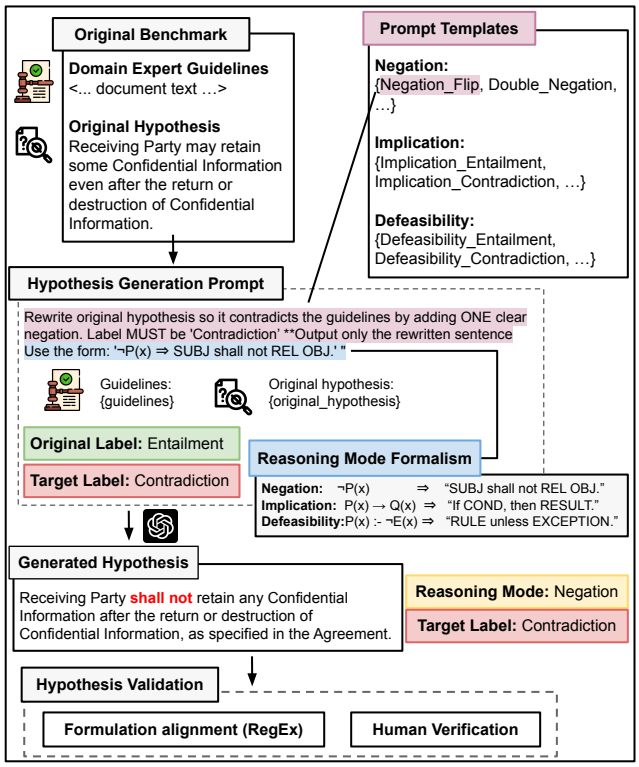


Figure 2: The proposed benchmark enhancement workflow for evaluating three modes of reasoning (*negation, implication*, and *defeasibility*).

Once generated, the hypotheses are verified using two steps. First, we use a heuristic regular expression function with a pre-defined list of keywords to ensure the intended reasoning modes are present. Second, we confirm that the generated hypotheses are more difficult than the original by running them through two pre-trained NLI models. We provide all results and prompt templates in Appendix C and G, together with a detailed exposition on the verification. Finally, to ensure the quality and correctness of this automated process, we conducted a manual audit on a random sample of 100 generated hypotheses per benchmark, verifying that the new hypotheses are consistently well-formed, and correctly aligned with the semantics and ground-truth labels of the targeted reasoning modes.

Baselines. We evaluate LOGT's accuracy against five baselines including: *(1) Basic Prompting with No Document (basic-nd)*, which relies on the LLM to answer the question directly using its own knowledge; *(2) Basic Prompting with Document (basic-d)*, which provides the model with the relevant domain-expert guidelines as added context; *(3) Few-Shot Prompting (FS)* (Brown et al. 2020), where the prompt is constructed with a solved example from a training set for each possible class label (e.g., Entailment, Contradiction) to guide the model via in-context learning ; and (4) *Multi-Step Chain-of-Thought (CoT)* (Wei et al. 2022b), an advanced technique where the model first generates a step-by-step rationale that is then programmatically used in subsequent prompts to synthesize the final answer.

Additionally, we evaluate the quality of LOGT's reasoning traces in comparison to multi-step CoT reasoning by considering cases where the predicted labels contradict those produced by an alternative method. We use the *LLM-as-a-judge* (Zheng et al. 2023; Gu et al. 2024) framework to categorize each generated reasoning step into one of six types: *fact_lookup*, *apply_rule*, *check_condition*, *contradiction_detected*, and *conclude_label*. We provide all prompt template examples in appendix C.1. Each full reasoning trace is then classified as a correct *Reason* if it contains appropriate and coherent use of rules and facts that support the final decision, or as an incorrect *Reason* if the reasoning is flawed, incomplete, or inconsistent.

We use six diverse LLMs including Mistral 7B, LLaMA 3.1-8B, LLaMA 3.1-70B, Claude 3.5 Haiku, GPT-o3 Mini, and DeepSeek R1. A comprehensive table with model details, including their number of parameters, release dates, and developers, is available in Appendix A. To ensure a fair comparison, all methods (LOGT and baselines) are always provided with the same system prompts and in-context examples where applicable. For reproducible and deterministic results, we set the decoding temperature to 0. Performance is evaluated across all three reasoning categories (*implication, negation,* and *defeasibility*).

Results and Discussion

Finding 1: LOGT consistently improves performance across all models, with smaller models exhibiting the most substantial gains. LOGT significantly outperforms standard modes of prompting on all benchmarks, improving performance by an average of +4.41% compared to the best baseline and +11.82% compared to the average of all baselines (Table 1). Promisingly, the largest gains are on the (initially) under-performing models. In ContractNLI, Mistral-7B and LLaMA-8B benefit the most, improving from 45.20% and 44.30% to 53.40% (+8.2%) and 50.40% (+6.1%), respectively. We see similar trends for these two models on other benchmarks. On average, Mistral-7B and LLaMA-8B outperform the (next) best and the average of all baselines by 6.73% and 10.38%, respectively.

Finding 2: LOGT improves the ability of models across all three reasoning modes. As presented in Figure 3 (a), we compared the performance of LOGT to the *best* performance among all prompting baselines (excluding LOGT W/O SGC and LOGT W/O LC). LOGT consistently improves the performance of models across all three reasoning modes (*negation, implication*, and *defeasible*). The gains are most pronounced and consistent in the implication reasoning mode, where LOGT boosts performance by +2.2% on ContractNLI, +5.0% on SARA, +10.8% on Biomed, and a remarkable +13.2% on DnD. Although, the impact of LOGT generally improves the performance across all reasoning modes, we can see small performance drop on some areas, including performance drops of -0.8% on ContractNLI (Defeasible), -3.2% on SARA (Negation), and -0.7% on BiomedNLI (Negation). This indicates that while LOGT provides a strong overall benefit, its efficacy varies

Model	Basic-ND	Basic-D	FS	CoT	LogT (SGC)	LogT (LC)	LogT (Full)
ContractNLI*							
Mistral 0.3-7B	34.00 (±1.48)	44.30 (±1.58)	45.20 (±1.58)	44.40 (±1.58)	47.90 (±1.59)	52.90 (±1.56)	**53.40** (±1.58)
LLaMA 3.1-8B	38.00 (±1.53)	39.00 (±1.53)	40.60 (±1.53)	44.30 (±1.58)	43.00 (±1.55)	47.40 (±1.55)	**50.40** (±1.58)
LLaMA 3.3-70B	48.60 (±1.58)	62.70 (±1.53)	64.30 (±1.53)	62.70 (±1.53)	59.00 (±1.50)	62.90 (±1.53)	**66.90** (±1.48)
Claude 3.5 Haiku	32.40 (±1.48)	62.60 (±1.53)	61.70 (±1.53)	63.90 (±1.53)	64.50 (±1.50)	65.80 (±1.45)	**65.90** (±1.48)
GPT-o3 Mini	42.70 (±1.58)	69.90 (±1.43)	**70.00** (±1.43)	61.60 (±1.53)	64.30 (±1.50)	64.50 (±1.50)	69.20 (±1.48)
DeepSeek R1	54.40 (±1.58)	70.10 (±1.43)	69.90 (±1.43)	61.30 (±1.53)	66.60 (±1.45)	68.70 (±1.44)	**70.20** (±1.43)
SARA*							
Mistral 0.3-7B	65.13 (±2.94)	62.84 (±2.99)	64.75 (±2.96)	63.22 (±2.99)	62.80 (±3.00)	65.51 (±2.95)	**72.41** (±2.77)
LLaMA 3.1-8B	56.70 (±3.07)	65.13 (±2.94)	48.28 (±3.09)	64.75 (±2.96)	67.04 (±2.91)	68.19 (±2.88)	**69.73** (±2.84)
LLaMA 3.3-70B	64.75 (±2.96)	59.39 (±3.04)	62.45 (±3.00)	73.95 (±2.72)	68.19 (±2.88)	70.11 (±2.80)	**74.03** (±2.78)
Claude 3.5 Haiku	60.54 (±3.02)	61.69 (±3.01)	70.11 (±2.83)	72.80 (±2.76)	72.03 (±2.78)	70.49 (±2.83)	**73.56** (±2.73)
GPT-o3 Mini	58.24 (±3.05)	61.30 (±3.02)	**72.41** (±2.77)	64.37 (±2.97)	63.60 (±2.92)	68.20 (±2.88)	69.73 (±2.84)
DeepSeek R1	56.32 (±3.07)	65.52 (±2.94)	65.13 (±2.94)	62.45 (±3.00)	65.52 (±2.91)	67.43 (±2.90)	**72.41** (±2.77)
BioMedNLI*							
Mistral 0.3-7B	57.00 (±3.53)	61.00 (±3.45)	61.50 (±3.44)	63.50 (±3.42)	66.00 (±3.35)	64.00 (±3.39)	**68.00** (±3.29)
LLaMA 3.1-8B	54.00 (±3.53)	62.00 (±3.44)	57.50 (±3.51)	67.50 (±3.37)	64.00 (±3.39)	61.50 (±3.44)	**76.00** (±3.04)
LLaMA 3.3-70B	53.50 (±3.53)	69.00 (±3.30)	65.50 (±3.39)	73.50 (±3.16)	75.50 (±3.06)	75.00 (±3.06)	**78.00** (±2.94)
Claude 3.5 Haiku	55.00 (±3.53)	64.00 (±3.43)	65.00 (±3.39)	72.00 (±3.13)	71.50 (±3.19)	71.50 (±3.19)	**72.50** (±3.11)
GPT-o3 Mini	56.00 (±3.53)	73.50 (±3.16)	74.50 (±3.11)	74.00 (±3.13)	73.50 (±3.12)	74.50 (±3.08)	**77.50** (±2.95)
DeepSeek R1	52.00 (±3.53)	**76.00** (±3.04)	72.00 (±3.22)	73.00 (±3.18)	74.50 (±3.08)	73.00 (±3.14)	73.50 (±3.16)
Dungeons and Dragons							
Mistral 0.3-7B	55.46 (±4.56)	67.79 (±3.83)	73.15 (±3.63)	73.83 (±3.60)	72.48 (±3.66)	76.51 (±3.47)	**79.19** (±3.33)
LLaMA 3.1-8B	57.14 (±4.54)	64.43 (±3.92)	62.42 (±3.97)	61.74 (±3.98)	61.74 (±3.98)	65.10 (±3.90)	**69.79** (±3.76)
LLaMA 3.3-70B	57.98 (±4.52)	84.56 (±2.96)	87.92 (±2.67)	89.26 (±2.54)	89.93 (±2.47)	90.60 (±2.39)	**90.60** (±2.39)
Claude 3.5 Haiku	57.14 (±4.54)	63.76 (±3.94)	75.84 (±3.51)	**87.25** (±2.73)	84.56 (±2.96)	79.87 (±3.29)	86.58 (±2.79)
GPT-o3 Mini	79.83 (±3.68)	88.59 (±2.60)	89.93 (±2.47)	91.94 (±2.06)	92.62 (±2.14)	93.95 (±2.04)	**95.30** (±1.73)
DeepSeek R1	67.23 (±4.30)	92.44 (±2.42)	91.95 (±2.23)	93.29 (±2.05)	94.63 (±2.14)	92.62 (±2.14)	**95.30** (±1.74)

Table 1: Accuracy (with standard error) of LOGT variants and baselines across four benchmarks. "ND" = No Document; "D" = Document; "LC" = Logic-based Context; "SGC" = Symbolic-Graph Context. Best per-row values are in **bold**.

by the specific domains, presenting a clear direction for future refinement. We also provide the same analysis in Appendix H where we compare LOGT performance against the average performance of all baselines instead of the best one. Extra analysis shows that LOGT improves the performance in *all* reasoning modes and benchmarks.

Finding 3: Due to its explicit symbolic grounding, LOGT elicits approximately 21.5% more reasoning steps per example compared to standard CoT methods. We compare the correctness of the reasoning trace between CoT and LOGT. Figure 3 (b) presents statistics and a breakdown of the types of reasoning steps used by each method. On average, LOGT generates 21.5% more reasoning steps compared to Chain-of-Thought (CoT), despite using the same prompt instructions. This increase reflects a shift in the reasoning process rather than mere verbosity. Notably, the number of *apply_rule* steps increases substantially—from 1.08 in CoT to 2.66 in LOGT demonstrating that LOGT reasoning traces emphasize explicit rule-based deduction. Similarly, *fact_lookup* steps increase from 2.73 to 2.90, indicating that LOGT encourages models to retrieve and verify factual information more actively than standard CoT. We provide qualitative evidence that LLMs utilized the provided neuro-symbolic contexts, especially on *apply_rule* and *fact_lookup* types, in Appendix H.

Finding 4: LOGT reasoning traces are more robust than CoT. As shown in Figure 3 (c), the bar chart reveals that LOGT improves the quality of reasoning traces. When the model produces correct reasoning, LOGT leads to more accurate predictions (48%), compared to only (39%) for CoT. Moreover, LOGT reduces the number of false negative cases, where correct reasoning leads to incorrect predictions compared to CoT. False negative cases reduce from 41% (CoT) to 28% (LOGT). Hence, LOGT not only produces more robust reasoning trace but also enhances its alignment with final predictions, resulting in notable accuracy gains (54% vs. 46% on average). The same trend as the overall can also be seen across all three benchmarks. We also provide confusion matrices to show the instance-level analysis in Appendix H.

Finding 5 (ablation study): Both symbolic and logic-based contexts improve performance, but logic-based context contribute the most improvement. We compare the improvement of *LogT (SGC)* and *LogT (LC)* with *CoT* as presented in Table 1. On average, LC improves accuracy by +2.4% relative to CoT, whereas adding only the SGC

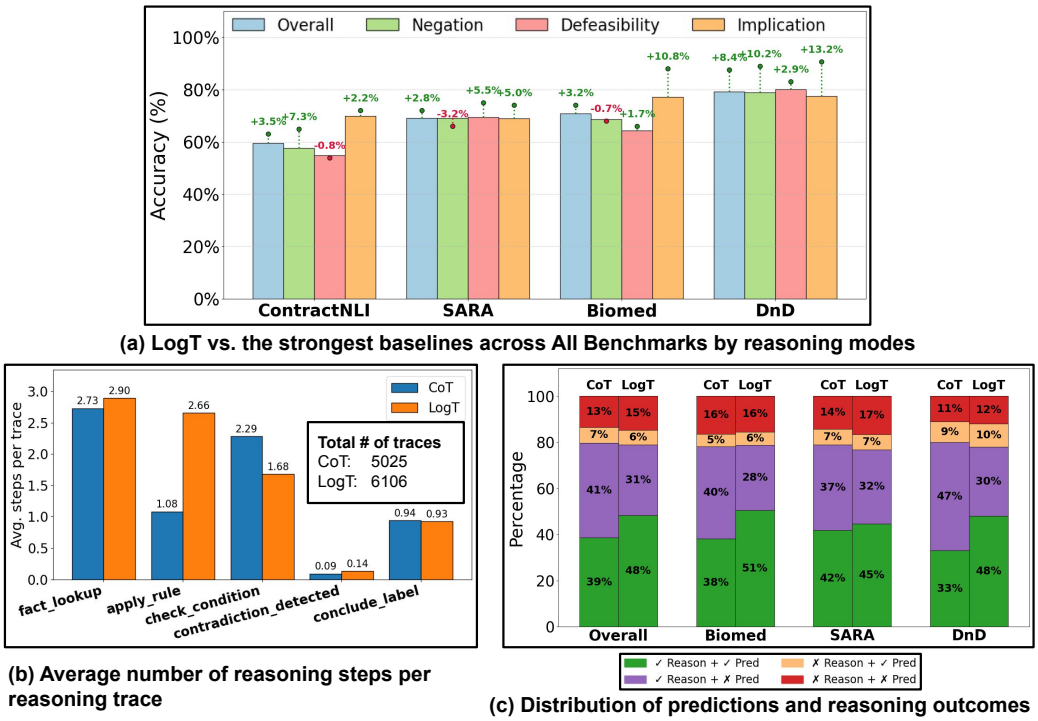


Figure 3: Performance evaluation of LOGT: (a) shows an accuracy comparison between LOGT (indicated by markers) and the strongest baselines (bars) across four benchmarks; (b) details the average number of reasoning steps per trace for both LOGT and the CoT baseline; (c) displays the four outcome distributions for LOGT and CoT, classified by whether the reasoning and final prediction were correct or incorrect.

yields a much smaller gain of +0.5%. When both contexts are used together (LOGT (FULL)), the average improvement is boosted up to +7.4%. The main exceptions appear in BIOMEDNLI and DUNGEONS & DRAGONS, where six cases of SGC alone outperforms LC.

Related Work

High-Assurance Reasoning The integration of neural networks with symbolic reasoning is an important agenda in AI, with foundational neuro-symbolic systems developed in computer vision, rule learning, and knowledge representation (Andreas et al. 2016; Hitzler et al. 2022; Hatzilygeroudis and Prentzas 2004; Hudson and Manning 2019). Contemporary methods within NLP typically take advantage of an LLM to translate natural language into an intermediate symbolic representation, such as program code or logical expressions (Nye et al. 2021; Lyu et al. 2023; Pan et al. 2023). However, existing methods typically focus on general logical formulations, neglecting specific high-assurance or non-monotonic reasoning scenarios such as defeasible logic. In contrast, our work builds on this neurosymbolic tradition but is primarily focused on high-assurance reasoning.

Natural Language Inference While Natural Language Inference (NLI) is a longstanding area of research (Cooper et al. 1994), it has taken on new importance in an era of transformer-based models. Even predating LLMs, recent work has taken on inference tasks of increasingly difficult and complexity (Bowman et al. 2015; Storks, Gao, and Chai 2019; Welleck et al. 2018), with a range of papers exploring inferences tasks that exhibit ambiguity, plausibility, and commonsense reasoning, for which ordinary entailment is insufficient (Nie, Zhou, and Bansal 2020; Meissner et al. 2021). Following this trend, NLI tasks both in domain-specific contexts such as law and health (Koreeda and Manning 2021; Holzenberger, Blair-Stanek, and Durme 2020; Romanov and Shivade; Jullien, Valentino, and Freitas 2024; Kazemi et al. 2023), and those exploring non-monotonic reasoning, have been proposed but are still limited in scope, and do not involve formal evaluation of reasoning traces (Madaan et al. 2021; Rudinger et al. 2020; Yanaka et al. 2019; Zhang, Jing, and Gogate 2025; Zhang et al. 2025; Gubelmann et al. 2024).

Prompt Engineering in LLMs Prompt engineering and optimization has been an active area of research for getting the most of out LLMs, especially on complex problems requiring multi-step reasoning. In-context learning has demonstrated remarkable success, enabling models to tackle various tasks effectively with minimal examples, without updating model parameters (Brown et al. 2020). Chain-of-Thought (CoT), surveyed by Yu et al. (2023); Xia et al. (2024); Chen et al. (2025), and its variants (Wei et al. 2022b; Wang et al. 2022; Long 2023; Yao et al. 2023) are promising, but the lack of a symbolic infrastructure in CoT reason-

ing traces do not make them amenable to verification. To the best of our knowledge, the use of a dual context that involves knowledge and logic is novel.

References

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 39–48.

Blair-Stanek, A.; Holzenberger, N.; and Van Durme, B. 2023. Can gpt-3 perform statutory reasoning? In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, 22–31.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; 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, Q.; Qin, L.; Liu, J.; Peng, D.; Guan, J.; Wang, P.; Hu, M.; Zhou, Y.; Gao, T.; and Che, W. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Cooper, R.; Crouch, R.; Van Eijck, J.; Fox, C.; Van Genabith, J.; Jaspers, J.; Kamp, H.; Pinkal, M.; Poesio, M.; Pulman, S.; et al. 1994. FraCaS–a framework for computational semantics. *Deliverable D6*.

Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.; Celikyilmaz, A.; and Weston, J. 2024. Chain-of-Verification Reduces Hallucination in Large Language Models. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, 3563–3578. Association for Computational Linguistics.

Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; and Neubig, G. 2023. PAL: Program-aided Language Models. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 10764–10799. PMLR.

Groen, G. J.; and Patel, V. L. 2014. The relationship between comprehension and reasoning in medical expertise. In *The nature of expertise*, 287–310. Psychology Press.

Grosof, B.; Kifer, M.; Swift, T.; Fodor, P.; and Bloomfield, J. 2023. Ergo: a quest for declarativity in logic programming. In *Prolog: The Next 50 Years*, 224–236. Springer.

Grosof, B. N.; Kifer, M.; and Fodor, P. 2017. Rulelog: Highly Expressive Semantic Rules with Scalable Deep Reasoning. In *RuleML+ RR (Supplement)*.

Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Gubelmann, R.; Katis, I.; Niklaus, C.; and Handschuh, S. 2024. Capturing the varieties of natural language inference: A systematic survey of existing datasets and two novel benchmarks. *Journal of Logic, Language and Information*, 33(1): 21–48.

Hatzilygeroudis, I.; and Prentzas, J. 2004. Neuro-symbolic approaches for knowledge representation in expert systems. *International Journal of Hybrid Intelligent Systems*, 1(3-4): 111–126.

Hitzler, P.; Eberhart, A.; Ebrahimi, M.; Sarker, M. K.; and Zhou, L. 2022. Neuro-symbolic approaches in artificial intelligence. *National Science Review*, 9(6): nwac035.

Holzenberger, N.; Blair-Stanek, A.; and Durme, B. V. 2020. A Dataset for Statutory Reasoning in Tax Law Entailment and Question Answering. In Aletras, N.; Androutsopoulos, I.; Barrett, L.; Meyers, A.; and Preotiuc-Pietro, D., eds., *Proceedings of the Natural Legal Language Processing Workshop 2020 co-located with the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2020), Virtual Workshop, August 24, 2020*, volume 2645 of *CEUR Workshop Proceedings*, 31–38. CEUR-WS.org.

Holzenberger, N.; and Van Durme, B. 2023. Connecting symbolic statutory reasoning with legal information extraction. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, 113–131. Association for Computational Linguistics.

Hudson, D.; and Manning, C. D. 2019. Learning by Abstraction: The Neural State Machine. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jullien, M.; Valentino, M.; and Freitas, A. 2024. SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. In Ojha, A. K.; Doğruöz, A. S.; Tayyar Madabushi, H.; Da San Martino, G.; Rosenthal, S.; and Rosá, A., eds., *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 1947–1962. Mexico City, Mexico: Association for Computational Linguistics.

Kazemi, M.; Yuan, Q.; Bhatia, D.; Kim, N.; Xu, X.; Imbrasaite, V.; and Ramachandran, D. 2023. Boardgameqa: A dataset for natural language reasoning with contradictory information. *Advances in Neural Information Processing Systems*, 36: 39052–39074.

Koreeda, Y.; and Manning, C. 2021. ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 1907–1919. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Lanham, T.; Chen, A.; Radhakrishnan, A.; Steiner, B.; Denison, C.; Hernandez, D.; Li, D.; Durmus, E.; Hubinger, E.; Kernion, J.; et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Li, D.; Shao, R.; Xie, A.; Sheng, Y.; Zheng, L.; Gonzalez, J.; Stoica, I.; Ma, X.; and Zhang, H. 2023a. How long can context length of open-source llms truly promise? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Li, J.; Wang, M.; Zheng, Z.; and Zhang, M. 2023b. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*.

Li, M.; Miao, S.; and Li, P. 2024. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv preprint arXiv:2410.20724*.

Long, J. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.

Lyu, Q.; Havaldar, S.; Stein, A.; Zhang, L.; Rao, D.; Wong, E.; Apidianaki, M.; and Callison-Burch, C. 2023. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*.

Madaan, A.; Rajagopal, D.; Tandon, N.; Yang, Y.; and Hovy, E. 2021. Could you give me a hint? generating inference graphs for defeasible reasoning. *arXiv preprint arXiv:2105.05418*.

Meissner, J. M.; Thumwanit, N.; Sugawara, S.; and Aizawa, A. 2021. Embracing ambiguity: Shifting the training target of NLI models. *arXiv preprint arXiv:2106.03020*.

Narendra, S.; Shetty, K.; and Ratnaparkhi, A. 2024. Enhancing contract negotiations with LLM-based legal document comparison. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, 143–153.

Nie, Y.; Zhou, X.; and Bansal, M. 2020. What can we learn from collective human opinions on natural language inference data? *arXiv preprint arXiv:2010.03532*.

Nye, M.; Tessler, M.; Tenenbaum, J.; and Lake, B. M. 2021. Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. *Advances in Neural Information Processing Systems*, 34: 25192–25204.

Pan, L.; Albalak, A.; Wang, X.; and Wang, W. Y. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Peng, B.; Zhu, Y.; Liu, Y.; Bo, X.; Shi, H.; Hong, C.; Zhang, Y.; and Tang, S. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Romanov, A.; and Shivade, C. ???? Lessons from Natural Language Inference in the Clinical Domain.

Rudinger, R.; Shwartz, V.; Hwang, J. D.; Bhagavatula, C.; Forbes, M.; Le Bras, R.; Smith, N. A.; and Choi, Y. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4661–4675.

Storks, S.; Gao, Q.; and Chai, J. Y. 2019. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.

Susskind, R. E. 1986. Expert systems in law: A jurisprudential approach to artificial intelligence and legal reasoning. *The modern law review*, 49(2): 168–194.

Turpin, M.; Michael, J.; Perez, E.; and Bowman, S. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36: 74952–74965.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022a. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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

Welleck, S.; Weston, J.; Szlam, A.; and Cho, K. 2018. Dialogue natural language inference. *arXiv preprint arXiv:1811.00671*.

Xia, Y.; Wang, R.; Liu, X.; Li, M.; Yu, T.; Chen, X.; McAuley, J.; and Li, S. 2024. Beyond chain-of-thought: A survey of chain-of-x paradigms for llms. *arXiv preprint arXiv:2404.15676*.

Xiu, Y.; Xiao, Z.; and Liu, Y. 2022. LogicNMR: Probing the non-monotonic reasoning ability of pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 3616–3626.

Yanaka, H.; Mineshima, K.; Bekki, D.; Inui, K.; Sekine, S.; Abzianidze, L.; and Bos, J. 2019. Can neural networks understand monotonicity reasoning? *arXiv preprint arXiv:1906.06448*.

Yao, B.; Chen, G.; Zou, R.; Lu, Y.; Li, J.; Zhang, S.; Sang, Y.; Liu, S.; Hendler, J.; and Wang, D. 2024. More Samples or More Prompts? Exploring Effective Few-Shot In-Context Learning for LLMs with In-Context Sampling. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Findings of the Association for Computational Linguistics: NAACL 2024*, 1772–1790. Mexico City, Mexico: Association for Computational Linguistics.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Yu, Z.; He, L.; Wu, Z.; Dai, X.; and Chen, J. 2023. Towards better chain-of-thought prompting strategies: A survey. *arXiv preprint arXiv:2310.04959*.

Yuan, L.; Chen, Y.; Cui, G.; Gao, H.; Zou, F.; Cheng, X.; Ji, H.; Liu, Z.; and Sun, M. 2023. Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis, and llms evaluations. *Advances in Neural Information Processing Systems*, 36: 58478–58507.

Zhang, X.; Du, C.; Pang, T.; Liu, Q.; Gao, W.; and Lin, M. 2024. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37: 333–356.

Zhang, Y.; Jing, L.; and Gogate, V. 2025. Defeasible Visual Entailment: Benchmark, Evaluator, and Reward-Driven Optimization. In Walsh, T.; Shah, J.; and Kolter, Z., eds., *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, 25976–25984. AAAI Press.

Zhang, Y.; Sun, J.; Guo, Y.; and Gogate, V. 2025. Can Video Large Multimodal Models Think Like Doubters-or Double-Down: A Study on Defeasible Video Entailment. *CoRR*, abs/2506.22385.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36: 46595–46623.

Zhou, P.; Khanna, R.; Lee, S.; Lin, B. Y.; Ho, D.; Pujara, J.; and Ren, X. 2021. RICA: Evaluating Robust Inference Capabilities Based on Commonsense Axioms. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7560–7579. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Technical Appendix Contents

This Technical Appendix, a supplement to the paper *LOGicalThought: Logic-Based Ontological Grounding of LLMs for High-Assurance Reasoning*. This appendix consists of:

Appendix A - Model Summary

We present a summary of the models used in our experiments in Table 2. Our models were selected in order to represent popular LLMs in use today while also covering a range of sizes and providers. We use 4 open source models and 2 closed source models from the two promiment providers (OpenAI and Anthropic). All models were accessed via OpenRouter's Python API[2].

Appendix B - Experiment Configuration

To ensure the reproducibility of our experiments, this section provides the complete details of the language model configurations and the prompt templates used for our primary tasks.

Reproducibility Setup All symbolic reasoning querying was conducted locally on a MacBook Pro (Apple M1 Pro, CPU), with ErgoAI installed natively. This is specifically for compilability check and logic program execution. Additionally, all language model prompts and evaluation scripts are designed to be hardware-agnostic and can be executed on both the local machine (CPU) and Google Colab (CPU). This setup ensures accessibility and ease of replication without requiring GPU acceleration.

Language Models Configuration

To ensure fair comparison and reproducibility, a consistent set of hyperparameters was used for all models listed in Appendix A across all experiments. This standardized approach was applied to all baseline generation strategies and LOGT variants.

[2]https://openrouter.ai/

Prediction Generation. For all final prediction steps, we employed a deterministic configuration with temperature set to 0.0, a fixed seed of 42, and a maximum token limit of 4096. This setup was used for the *Basic-ND*, *Basic-D*, and *Few-shot* baselines, as well as for the final prediction stage of both *CoT* and all LOGT variants.

Reasoning Generation. To encourage more elaborative intermediate reasoning, we increased the temperature slightly to 0.2 during the analysis-generation phase within the *CoT* and LOGT pipelines. However, the final prediction after this reasoning step still used temperature 0.0 for consistency.

Benchmark Enhancement and Hypothesis Generation

To ensure a robust and balanced evaluation, the test sets were curated by generating new hypotheses for some benchmarks. This process was standardized to create consistent and reproducible datasets.

Hypothesis Enhancement Configuration. All supplementary hypotheses were generated using GPT-4o with a deterministic setup: temperature was set to 0.0, and the seed was fixed at 42. This configuration guarantees the generation of a static and perfectly reproducible dataset. Our main objective in this process was to ensure an even distribution of instances across the three core reasoning modes: Negation, Implication, and Defeasible.

Appendix C - Prompt Templates

This appendix provides a comprehensive overview of the prompt templates used throughout our study. These templates form the foundation of our experimental design, supporting various stages of the workflow including benchmark enhancement, context enrichment, reasoning trace generation, and grounded model evaluation. The prompts are designed to enable systematic probing of model reasoning capabilities, generalization, and alignment with symbolic or logical information. To streamline presentation, we consolidate all templates in this section and present them as visual examples in the following pages.

Appendix C.1 Benchmark Enhancement for Nonmonotonic Reasoning. This category includes prompt templates used to generate controlled variations of benchmark hypotheses across different reasoning types. Specifically, we define 11 distinct formats that introduce transformations aligned with negation, causality, or defeasibility. The goal is to ensure a balanced and diverse evaluation set that challenges the model's ability to reason under different conditions. For example, "Negation_Flip" inverts the polarity of the hypothesis, while "Entailment_Defeasibility" introduces an entailment that holds under normal but defeasible assumptions. Table 3 summarizes the full set of formats.

Appendix C.2 Symbolic Graph Context. These prompts instruct the language model to extract symbolic triples from natural language hypotheses and contexts. Each triple captures a relation in the form of *subject–predicate–object*, allowing us to construct a structured semantic graph. This

Weight Type	Model	Platform (Provider)	Released	Training Cut-off	Parameters
Open source	Mistral Instruct v0.3	Together (Mistral)	May 2024	Unknown	7B
	LLaMA 3.1	Together (Meta)	Jul 2024	Dec 2023	8B
	LLaMA 3.3	Together (Meta)	Dec 2024	Dec 2023	70B
	DeepSeek R1	Deepseek	Jan 2025	Jul 2024	67B
Closed source	GPT-o3 Mini	OpenAI	Jul 2024	Oct 2023	?
	Claude 3.5 Haiku	Anthropic	Oct 2024	Jul 2024	?

Table 2: Overview of the models evaluated and utilized in case studies.

Template Format Name	Reasoning Mode
Negation_Flip	Negation
Negation_Maintain	Negation
Causality_Entailment	Implication
Causality_Contradiction	Implication
NotMentioned_Defeasibility	Defeasible
NotMentioned_Negation	Negation
NotMentioned_Causality	Implication
Entailment_Defeasibility	Defeasible
Contradiction_Defeasibility	Defeasible
Defeasibility_Entailment	Defeasible
Defeasibility_Contradiction	Defeasible

Table 3: List of 11 prompt templates categorized by reasoning mode: Negation, Implication, and Defeasible.

symbolic representation is later used as additional context to enhance model reasoning. For instance, given a biomedical statement, the prompt may elicit triples like "(fever, indicates, infection)," providing an interpretable scaffold for downstream inference.

Appendix C.3 ErgoAI Program Generation. To support formal reasoning, we design prompts that generate lightweight logic programs using the ErgoAI language. These logic-based contexts capture conditional rules, exceptions, and defeasible logic extracted from the input. The generated programs serve as abstract representations of background knowledge, enabling structured inference.

Appendix C.4 Reasoning Traces. This class of prompts elicits step-by-step explanations from language models. Rather than asking for direct answers, we guide the model to reason explicitly—identifying relevant facts, linking them, and drawing conclusions in a transparent manner. The output typically begins with a statement like "First, we note that...," and proceeds through a logical narrative before arriving at a final prediction. These traces improve interpretability and allow for trace-level evaluation of model reasoning quality.

Appendix C.5 Grounded LLM Calls. Finally, we develop prompt templates that combine the original input with structured context—either symbolic graphs, logical-based contexts, or both—before querying the model for a prediction. These prompts allow us to test whether and how external structure improves performance. By controlling the form and source of the grounding, we can compare zero-shot, symbolic-only, logic-only, and combined configurations. This setup enables a systematic evaluation of grounding effectiveness under different model architectures and settings.

Together, these prompt templates define a modular and extensible interface between symbolic knowledge, logical reasoning, and language models. The full prompt examples for each category are provided as figures in the remainder of this appendix.

Appendix C.1 Benchmark Enhancement for Non-monotonic Reasoning.

Prompt Template for Benchmark Generation (Negation Flip)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS so that it contradicts the EVIDENCE by adding ONE clear negation following:
¬P(x) ⇒ "SUBJ(x) shall not REL(x) OBJ(x)." (where x is drawn from the EVIDENCE).

Label MUST be **opposite** of the {original_label}
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Negation Maintain)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS by adding double negation that preserves the meaning of the original hypothesis.
¬P(x) ⇒ "SUBJ(x) shall not REL(x) OBJ(x)." (where x is drawn from the EVIDENCE).

Label MUST be **remain** {original_label}
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Negation NotMentioned)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS by adding ONE negation that is not mentioned in the EVIDENCE
¬P(x) ⇒ "SUBJ(x) shall not REL OBJ." (where x is drawn from the EVIDENCE).

Label MUST be **NotMentioned**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Implication Entailment)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS into an implication statement fully supported by the EVIDENCE.
$P(x) \rightarrow Q(x) \Rightarrow$ "If COND(x), then RESULT(x)." (where x is drawn from the EVIDENCE).

Label MUST be **Entailment**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Implication Contradiction)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS into an implication statement whose THEN part is opposite of the EVIDENCE.
$P(x) \rightarrow Q(x) \Rightarrow$ "If COND(x), then RESULT(x)." (where x is drawn from the EVIDENCE).

Label MUST be **Contradiction**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Implication NotMentioned)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS into an implication statement whose THEN part is not mentioned of the EVIDENCE.
$P(x) \rightarrow Q(x) \Rightarrow$ "If COND(x), then RESULT." (where x is drawn from the EVIDENCE).

Label MUST be **NotMentioned**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Defeasible Entailment)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS into a defeasible statement using UNLESS clause that the EXCEPTION is **false** according to the EVIDENCE, so the statement is still **Entailment**
$P(x) :- \neg E(x) \Rightarrow$ "RULE(x) unless EXCEPTION(x)." (where x is drawn from the EVIDENCE).

Label MUST be **Entailment**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Defeasible Contradiction)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS into a defeasible statement using UNLESS clause that the EXCEPTION is **true** according to the EVIDENCE, so the statement is **Contradiction**
P(x) :- ¬E(x) ⇒ "RULE(x) unless EXCEPTION(x)." (where x is drawn from the EVIDENCE).

Label MUST be **Contradiction**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Defeasible NotMentioned)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS into a defeasible statement using UNLESS clause that the EXCEPTION is **not in** the EVIDENCE
P(x) :- ¬E(x) ⇒ "RULE(x) unless EXCEPTION." (where x is drawn from the EVIDENCE).

Label MUST be **NotMentioned**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Defeasible NotMentioned 2)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS into a defeasible statement using UNLESS clause that the EXCEPTION is **not in** the EVIDENCE. The original hypothesis is entailed, but this new unsupported condition makes the relationship unverifiable.
P(x) :- ¬E(x) ⇒ "RULE(x) unless EXCEPTION." (where x is drawn from the EVIDENCE).

Label MUST be **NotMentioned**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Prompt Template for Benchmark Generation (Defeasible NotMentioned 3)

You are a logic expert. Your task is to rewrite ORIGINAL_HYPOTHESIS into a defeasible statement using UNLESS clause that the EXCEPTION is **not in** the EVIDENCE. The original hypothesis is contradicted, but this new unsupported condition makes the relationship unverifiable.
P(x) :- ¬E(x) ⇒ "RULE(x) unless EXCEPTION." (where x is drawn from the EVIDENCE).

Label MUST be **NotMentioned**
Output only the rewritten sentence (no label tag).

INPUT EVIDENCE:
{evidence}

ORIGINAL_HYPOTHESIS (Label = {original_label}):
{original_hypothesis}

OUTPUT:
REWRITTEN_HYPOTHESIS:
[response]

Appendix C.2 Prompt Template for Symbolic Graph Context.

Prompt Template for Guideline Selection

From the provided GUIDELINE, extract ONLY the minimal sufficient clauses to evaluate H under S.
Prefer the most specific clauses; include an exception/override with its base clause if needed.
No explanation or chain-of-thought reasoning is needed.

Return STRICT JSON ONLY. If nothing applies, return { "selected_guidelines": [] }.

GUIDELINE:
{doc_text}

SCENARIO (S):
{scenario}

HYPOTHESIS (H):
{hypothesis}

RETURN (STRICT JSON ONLY; begin with '{{'; top-level key MUST be "selected_guidelines"):
{{ "selected_guidelines": ["...", "..."] }}

Prompt Template for Symbolic Graph Context

Given a legal GUIDELINE (DOC_TEXT), a SCENARIO, a HYPOTHESIS, and an optional REASONING_MODE,
produce a Symbolic Graph Context with THREE components:
1) O_rules_triples : ontology triples induced from the guideline clauses that are needed.
2) K_instance_triples : grounded facts from the scenario/hypothesis.
3) Q_nl : precise natural-language questions whose answers decide the hypothesis.

• Use ONLY DOC_TEXT, SCENARIO, and HYPOTHESIS (no external knowledge).
• If H cites a section/subsection (e.g., "§152(d)(2)(F)", "Section 3306(c)(1)"), ensure O_rules_triples includes that clause if present; if absent, include the nearest present ancestor (e.g., (c)(1)→(c)).
• Prefer the MOST SPECIFIC clauses; include an exception/override clause ("except", "unless", "notwithstanding")
• lower_snake_case for all symbols; integers for numbers; no quotes inside a triple.
• Canonicalize cited clause IDs into identifiers

TRIPLE FORMAT
{format_example}

DOC_TEXT:
{doc_text}

SCENARIO:
{scenario}

HYPOTHESIS:
{hypothesis}

Return one JSON object with EXACTLY these keys:

```
{ "O_rules_triples": ["<s,r,o>", "..."],       // 2–12 triples
  "K_instance_triples": ["<s,r,o>", "..."],    // 2–20 triples (constants only; grounded in S/H)
  "Q_nl": ["question 1", "question 2"]         // 2–5 concise questions }
```

Symbolic Graph Context Example (Legal)

Hypothesis: Alice bears a relationship to Bob under section 152(d)(2)(D) for the year 2018 unless Bob is Charlie's brother.

Selected Guidelines from the selection step: ['(2) Relationship For purposes of paragraph (1)(A), an individual bears a relationship to the taxpayer described in this paragraph if the individual is any of the following with respect to the taxpayer: (D) A stepfather or stepmother.']

Ontology:
['<sec_152_d_2, subsection, sec_152_d>',
'<sec_152_d_2_d, clause, sec_152_d_2>',
'<sec_152_d_2_d, refines, sec_152_d_2>',
'<sec_152_d_2_b, clause, sec_152_d_2>',
'<sec_152_d_2_b, refines, sec_152_d_2>']

Knowledge Triples:
['<alice, child_of, charlie>',
'<bob, sibling_of, charlie>',
'<tax_year, equals, 2018>']

Natural Language Queries:
["Is Bob Charlie's brother?", "Is Alice Charlie's child?"]

Symbolic Graph Context Example (Biomedical)

Hypothesis: Infections were not the most common adverse events recorded in the primary trial

Selected Guidelines from the selection step: ['Adverse Events 1: Total: 5/55 (9.09%)',
'Infection 2/55 (3.64%)',
'Pain * 1/55 (1.82%)',
'Muscle Weakness * 1/55 (1.82%)',
'Dyspnea 1/55 (1.82%)']

Ontology:
['<adverse_events_1, node_type, section>', '<infection, node_type, subsection>',
'<pain, node_type, subsection>', '<muscle_weakness, node_type, subsection>',
'<dyspnea, node_type, subsection>', '<infection, refines, adverse_events_1>',
'<pain, refines, adverse_events_1>', '<muscle_weakness, refines, adverse_events_1>',
'<dyspnea, refines, adverse_events_1>']

Knowledge Triples:
['<infections, count, 2>', '<pain, count, 1>', '<muscle_weakness, count, 1>', '<dyspnea, count, 1>']

Natural Language Queries:
['What was the count of infections recorded in the primary trial?',
'What was the count of pain recorded in the primary trial?',
'What was the count of muscle weakness recorded in the primary trial?',
'What was the count of dyspnea recorded in the primary trial?']

Symbolic Graph Context Example (Dungeons and Dragons)

Hypothesis: Nyssa may end her move in an ally's space unless her Speed is 0.

Selected Guidelines from the selection step: During your move, you can pass through the space of an ally, a creature that has the Incapacitated condition (see the Rules Glossary), a Tiny creature, or a creature that is two sizes larger or smaller than you. You can’t willingly end a move in a space occupied by another creature.

Ontology:
[{'subject': 'nyssa', 'relation': 'is_a', 'object': 'creature'},
{'subject': 'nyssa', 'relation': 'has_size', 'object': 'medium'},
{'subject': 'nyssa', 'relation': 'speed_feet', 'object': '30'},
{'subject': 'q1', 'relation': 'polarity', 'object': 'true'}]

Knowledge Triples:
{'subject': 'sir_aldric', 'relation': 'is_a', 'object': 'creature'},
{'subject': 'sir_aldric', 'relation': 'has_size', 'object': 'medium'},
{'subject': 'sir_aldric', 'relation': 'is_a', 'object': 'ally'},

Natural Language Queries:
['How much speed does Nyssa have?']

Appendix C.3 - Prompt Template for ErgoAI Program Generation.

Prompt Template for Logic-based Context

You write ErgoAI programs for legal (statutory) scenarios.

OBJECTIVE
From a Scenario, a Hypothesis, and ontology/instance triples, output ONLY code for a complete ErgoAI program that can be executed to test the Hypothesis.

OUTPUT SECTIONS (exact order)

% FACTS
% RULES
% DEFEASIBLE
% QUERIES

MAPPING TO ERGOAI SECTIONS
---
1) % FACTS (instance-level from KNOWLEDGE_TRIPLES)
  • Include every explicit INSTANCE triple as a ground fact.
2) % RULES (schema-level from ONTOLOGY)
  • Induce only the rules needed to evaluate the Hypothesis under the Scenario.
  • Use 'label_short' to name head/body predicates when the guideline defines a relation/category.
  • Use 'refines' to encode hierarchical refinements (e.g., specialized categories imply more general ones). You may add lightweight implication rules when necessary for evaluation.
  • Use 'depends_on' to include prerequisite literals in rule bodies when indicated.
  • Add comments '% source: sec_...' to show which clause a rule comes from.
3) % DEFEASIBLE (exceptions & priorities from ONTOLOGY)
  • For each <a, overrides, b>, label the corresponding rules '@{a}' and '@{b}' and add: \overrides(a, b).
  • If an exception is expressed via an ontology clause, implement it as a conflicting rule that derives \neg head(...) under the exception condition, and ensure the override makes the exception prevail.
4) % QUERIES
• Build queries that matches a rule head in % RULES and aligns with NL_QUERIES
• The arguments should be grounded or variables as appropriate to decide the Hypothesis.

ErgoAI Instruction and Syntax Examples: <ergo_examples>
ONTOLOGY: <ontology_triples> | KNOWLEDGE_TRIPLES: <knowledge_triples> | NL_QUERIES: <nl_queries>

Symbolic Graph Context Example (Legal)

% FACTS
child_of(alice, charlie).
sibling_of(bob, charlie).
tax_year(2018).

% RULES
% source: sec_152_d_2_d
sec_152_d_2_d_relationship(?X, ?Y) :- sibling_of(?Y, ?P), child_of(?X, ?P).

% source: sec_152_d_2_b
sec_152_d_2_b_relationship(?X, ?Y) :- sibling_of(?Y, ?P), child_of(?X, ?P).

% source: sec_152_d_2
sec_152_d_2_relationship(?X, ?Y) :- sec_152_d_2_d_relationship(?X, ?Y).
sec_152_d_2_relationship(?X, ?Y) :- sec_152_d_2_b_relationship(?X, ?Y).

% DEFEASIBLE
@{default_rule}
sec_152_d_2_relationship(alice, bob) :- child_of(alice, charlie), sibling_of(bob, charlie).

@{exception_rule}
\neg sec_152_d_2_relationship(alice, bob) :- sibling_of(bob, charlie).

\overrides(exception_rule, default_rule).

% QUERIES
sec_152_d_2_relationship(alice, bob).

Symbolic Graph Context Example (Biomedical)

% FACTS
count(infections, 2).
count(pain, 1).
count(muscle_weakness, 1).
count(dyspnea, 1).

% RULES
adverse_event(?X) :- count(?X, ?_Count).
most_common_adverse_event(?X) :- adverse_event(?X), count(?X, ?MaxCount), \naf higher_count_exists(?MaxCount).
higher_count_exists(?Count) :- count(?_Event, ?HigherCount), ?HigherCount > ?Count.
infections_not_most_common :- \naf most_common_adverse_event(infections).

% DEFEASIBLE

% QUERIES
count(infections, ?Count).
count(pain, ?Count).
count(muscle_weakness, ?Count).
count(dyspnea, ?Count).
infections_not_most_common.

Symbolic Graph Context Example (Dungeons and Dragons)

% FACTS
creature(nyssa).
has_size(nyssa, medium).
speed_feet(nyssa, 30).
creature(sir_aldric).
has_size(sir_aldric, medium).
ally(sir_aldric).

% RULES
can_end_move_in_ally_space(?X) :- creature(?X), speed_feet(?X, ?Speed), ?Speed > 0.

% DEFEASIBLE

% QUERIES
speed_feet(nyssa,?speed).

Appendix C.4 - Prompt Template for Reasoning Trace.

Prompt Template for Reasoning Trace Analysis

You are an expert reasoning-trace extractor.

— OUTPUT RULES —
• Return one *plain* JSON array (it must start with “[” and end with “]”).
• Do NOT wrap it in any other keys, headings, or explanation.

— STEP OBJECT SCHEMA —
Each array element must be an object with **all** of these keys:
• "step" : integer index starting at 1
• "type" : one of
  "fact_lookup" – consult evidence / retrieve fact from the given evidence
  "apply_rule" – fire a logical or statutory rule from the given evidence
  "check_condition" – test a pre-condition or clause
  "contradiction_detected" – detect inconsistency
  "conclude_label" – output final NLI label
• "input" : array of strings (facts, clauses, or prior outputs)
• "rule" : the rule or guideline invoked, or "" if none
• "output" : one concise atomic conclusion from this step

No other keys are allowed.

LogT Reasoning Trace Example (Legal)

[{'step': 1,
'type': 'fact_lookup', 'input': ['isMarriedUnder(alice, section_7703)'], 'output': 'Alice is married under section 7703 for 2017'},
— Look up fact from knowledge triples —
{'step': 2,
'type': 'fact_lookup', 'input': ['filesJointReturnWith(alice, spouse)'], 'output': 'Alice files a joint return with her spouse'},
— Look up fact from knowledge triples —
{'step': 3, 'type': 'fact_lookup', 'input': ["Alice and her spouse's taxable income is $17,330 for 2017"], 'output': 'Taxable income is $17,330 for 2017'},
{'step': 4, 'type': 'check_condition', 'input': ['$17,330', 'first tax bracket of section 1(a)', 'income not over $36,900'],
'output': 'Income falls within the first tax bracket'},
{'step': 5, 'type': 'apply_rule', 'input': ['married individuals filing jointly', 'income ≤ $36,900', 'tax rate is 15%'],
'output': 'Tax rate is 15%'},
{'step': 6, 'type': 'apply_rule', 'input': ['$17,330', '15%'], 'output': 'Tax is $2,599.50 (rounded to $2,600)'},
{'step': 7, 'type': 'check_condition', 'input': ['hypothesis states $2,600 in taxes', 'unless taxable income was $20,000'],
'output': 'Hypothesis states $2,600 unless income is $20,000'},
{'step': 8, 'type': 'apply_rule', 'input': ['$20,000', '15%'], 'output': 'Tax at $20,000 is $3,000'},
{'step': 9, 'type': 'check_condition', 'input': ['$2,600 tax amount', 'current income of $17,330'], 'output': '$2,600 tax amount is accurate for $17,330'},
{'step': 10, 'type': 'contradiction_detected', 'input': ['unless clause', '$20,000 income', '$3,000 tax'],
'output': 'Unless clause is misleading'},
{'step': 11, 'type': 'conclude_label', 'input': ['hypothesis is logically consistent', 'potentially misleading'], 'output': 'Hypothesis is fundamentally sound but misleading'}]

LogT Reasoning Trace Example (Biomedical)

[{'step': 1,
'type': 'fact_lookup',
'input': ['belongsToCohort(nk105Group, cohort1)', 'belongsToCohort(paclitaxelGroup, cohort1)',
'partOfTrial(cohort1, mainTrial)'],
'output': 'cohort1 is the relevant population for evaluating the hypothesis'},
— look up fact from knowledge triples —
{'step': 2,
'type': 'fact_lookup',
'input': ['medianPfs(nk105Group, 8.4Months)', 'medianPfs(paclitaxelGroup, 8.5Months)'],
'output': 'hypothesisConditionMet(cohort1)'},
— look up fact from knowledge triples —
{'step': 3,
'type': 'fact_lookup',
'input': ['7.0_to_9.9_months', '6.9_to_11.5_months'],
'output': 'shortestPfsNotLessThan3Weeks(cohort1)'},
{'step': 4,
'type': 'apply_rule',
'input': ['hypothesisConditionMet(cohort1)', 'shortestPfsNotLessThan3Weeks(cohort1)'],
'output': 'hypothesisValid(cohort1)'},
— apply rules from Ergo program —
{'step': 5,
'type': 'check_condition',
'input': ['participantsAnalyzed: 211', 'days_1_8_15_of_28day_cycle', 'RECIST 1.1'],
'output': 'No conflicting evidence'}]

LogT Reasoning Trace Example (Dungeons and Dragons)

[{'step': 1, 'type': 'fact_lookup', 'input': [],
'output': 'A creature can't willingly end its movement in a space already occupied by another creature. If it does, it suffers the Prone condition unless it is Tiny or larger than the other creature.'},
{'step': 2, 'type': 'check_condition', 'input': ['Alara', 'Brug'],
'output': 'Both Alara and Brug are Medium in size, so neither qualifies as Tiny nor is Alara larger than Brug.'},
{'step': 3, 'type': 'check_condition', 'input': ['Alara', 'Brug'],
'output': 'Alara ends her turn in Brug's square, which is occupied, triggering the rule's consequence.'},
{'step': 4, 'type': 'apply_rule', 'input': ['Alara'],
'output': 'Alara receives the Prone condition.'},
{'step': 5, 'type': 'fact_lookup', 'input': ['has_condition(alara, prone)'],
'output': 'The fact has_condition(alara, prone) aligns with the conclusion drawn from the rule.'},
{'step': 6, 'type': 'check_condition', 'input': ['hypothesis', 'rule', 'scenario outcome'],
'output': 'The hypothesis accurately reflects the rule's conditional structure.'},
{'step': 7,
'type': 'conclude_label', 'input': ['reasoning steps', 'rule's exceptions', 'Alara's case'],
'output': 'The reasoning steps confirm the hypothesis.'}]

Appendix C.5 - Grounded LLM Evaluation

Prompt Template for LLMs Evaluation on Entailment Task

You are an expert legal (SARA) NLI analyst.
Write a numbered, step-by-step analysis using ONLY the provided material.
Do NOT output the final label.

Defeasible guidance
• Consider an exception ("except", "unless", "notwithstanding") only if its pre-conditions are proven and no stronger contrary exception is proven.
• When rules conflict, respect stated priorities / overrides.

<Symbolic Graph Context>
ONTOLOGY: <ontology_triples> | KNOWLEDGE_TRIPLES: <knowledge_triples> | NL_QUERIES: <nl_queries>
</Symbolic Graph Context>

<Logic-based Context>
ERGO_PROGRAM: <ergo_program> | QUERY_AND_RESULTS: <query> <results>
</Logic-based Context>

Begin your detailed, numbered reasoning steps below.

You are an expert legal NLI classifier.

<hypothesis>
{hypothesis}
</hypothesis>

<analysis>
{analysis_from_step1}
</analysis>

What is the correct label? Output just one number – no other text.
1 = Entailment
2 = Contradiction

Appendix D - Hypothesis Examples of Different Reasoning Mode

Hypothesis Examples of All Reasoning Modes

Negation
ContractNLI: Agreement shall not grant Receiving Party any right to Confidential Information unless both parties agree to amend the terms in writing.
SARA: Section 3306(c)(10)(B) does not apply to Alice's employment situation in 2017.
BiomedNLI: Infections were not the most common adverse events recorded in the primary trial.
Dungeons and Dragons: Aric shall not end his movement in Brynn's space.

Implication
ContractNLI: If information is provided to the Investor by or on behalf of the Company in connection with the Transaction, then all Confidential Information shall be expressly identified by the Disclosing Party.
SARA: If Alice is a head of household for the year 2017 with a taxable income of $194512, then she does not have to pay $57509 in taxes for the year 2017 under section 1(b)(v).
BiomedNLI: If the primary and secondary trials recorded the Overall Survival (OS) of patients in their cohorts, then neither trial would have any data on OS.
Dungeons and Dragons: If Arin has 1 square of movement remaining, then Arin can enter Brin's square.

Defeasible
ContractNLI: Receiving Party may independently develop information similar to Confidential Information unless they are contractually prohibited from doing so.
SARA: Section 63(f)(2)(B) applies to Bob in 2017 with Alice as the spouse unless Alice was not blind in 2017.
BiomedNLI: The most common adverse event in the primary trial was skin infection unless anemia occurred more frequently.
Dungeons and Dragons: Another creature's space is Difficult Terrain for Marcus unless that creature is his ally.

Figure 4: Hypothesis examples of all benchmarks and reasoning modes

Appendix E - ErgoAI Program Synthesis

ErgoAI Syntax and Instruction

```
1) % FACTS (instance-level)
   - Include every explicit INSTANCE triple as a ground fact.
   - Do NOT assert any derived conclusion as a fact.
   - Do NOT assert the final query predicate as a fact.
2) % RULES (schema-level from ONTOLOGY)
   - Induce only the rules needed to evaluate the Hypothesis under the Scenario.
   - Use 'label_short' to name head/body predicates when the guideline defines a relation/category.
     Example: if label_short(sec_152_d_2_f, brother_of_father_or_mother) exists, you may define
       uncle_or_aunt_of(?X, ?Y) :- sibling_of(?X, ?P), parent_of(?P, ?Y).
     (Keep predicate names consistent and lower_snake_case.)
   - Use 'refines' to encode hierarchical refinements (e.g., specialized categories imply more general ones).
     You may add lightweight implication rules when necessary for evaluation.
   - Use 'depends_on' to include prerequisite literals in rule bodies when indicated.
3) % DEFEASIBLE (exceptions & priorities from ONTOLOGY)
   - For each <a, overrides, b>, label the corresponding rules '@{a}' and '@{b}' and add:
       \overrides(a, b).
   - If an exception is expressed via an ontology clause, implement it as a conflicting rule that derives
     \neg head(...) under the exception condition, and ensure the override makes the exception prevail.
—SYNTAX EXAMPLE—
% FACTS
man(socrates).
age(socrates, 56) \\and home(socrates, athens).
student(socrates, plato), student(socrates, xenophon).
% RULES
mortal(?X) :- man(?X).
% DEFEASIBLE
@{bird_rule}
worm_eater(?X) :- ?X:bird.
@{penguin_rule}
\\neg worm_eater(?X) :- ?X:penguin.
\\overrides(penguin_rule, bird_rule).
% QUERIES
mortal(?X).
uncle_or_aunt_of(?X, ?Y).
```

Figure 5: ErgoAI syntax and instruction that goes the ErgoAI program generation prompt.

To guide Ergo Program synthesis from knowledge triples and NL queries, we included valid ErgoAI syntax examples in the LLM prompt and specific instructions on how to convert to valid syntax. More specifically, as shown in Fig. 5, we include ErgoAI syntax examples of facts, rules, defeasible rules, and queries. To parse facts into entries into an ErgoAI rule bank, we specifically instruct the model to distinguish between three main categories of logical constructs based on the input knowledge structure.

The mapping follows a structured approach where facts represent instance-level assertions extracted directly from explicit triples without deriving conclusions, rules encode schema-level relationships from the ontology using consistent predicate naming conventions and hierarchical refinements, and defeasible rules handle exceptions and priority relationships through labeled rule conflicts and override declarations. For instance, the bird-penguin example demonstrates this mechanism by establishing a general rule that birds eat worms (@{bird_rule}) and an exception rule that penguins do not (@{penguin_rule}), with the \overrides(penguin_rule, bird_rule) declaration ensuring the exception takes precedence when both rules are applicable.

Appendix F – Benchmark Statistics

Detailed statistics for the four benchmarks used to evaluate LOGT are presented in Table 4. To ensure sufficient representation across the three reasoning modes (Negation, Implication, and Defeasibility), we curated the final test sets.

- **BiomedNLI** evaluates reasoning in medical contexts by classifying statements as entailment or contradiction based on clinical trial reports. We retain the same number of the original test set of 200 instances.
- **ContractNLI** is a legal domain dataset containing document-level evidence within contract clauses, annotated with entailment, contradiction, and neutral labels. We sampled 1,000 diverse examples from the original test set.
- **SARA** consists of scenarios for statutory reasoning over the U.S. Internal Revenue Code, requiring application of complex legal rules. We retain the same number of the original test set of 261 instances.
- **D&D** is our newly introduced benchmark based on the Dungeons & Dragons rulebook, consisting of 149 handcrafted examples reflecting logic-based gameplay reasoning.

Benchmark	Domain	Total	Neg.	Imp.	Deaf.
BiomedNLI	Med.	200	84	67	49
ContractNLI	Legal	1000	463	223	314
SARA	Legal	261	86	86	89
D&D	Games	149	50	50	49

Table 4: Benchmark statistics by domain and reasoning type.

Appendix G - Pilot Study Results and Enhanced Benchmarks Validation Statistics

To assess whether existing NLI benchmarks remain sufficiently challenging for modern LLMs, we conducted a pilot study using the original hypotheses from the **ContractNLI** and **BiomedNLI** datasets. Our analysis covers both findings from prior literature and results from our own baseline prompting experiments.

Findings from Existing Literature. Several studies have already shown that top-tier models perform remarkably well on these benchmarks. Table 5 summarizes key reported results.

Simple Prompting Results. To validate these trends, we ran a basic prompting baseline on the original ContractNLI test set across a range of models. Results are shown in Table 6.

The results suggest that standard NLI benchmarks may no longer pose sufficient challenge for modern large language models. Even with minimal prompt engineering, models such as LLaMA 3.3 and GPT-4o achieve high accuracy, indicating that these datasets may not adequately differentiate models' reasoning capabilities. This highlights a saturation point in benchmark utility and underscores the need for more diagnostically precise evaluations. Motivated by this, we construct a set of enhanced benchmarks that explicitly probe reasoning mode such as negation, implication, and defeasible reasoning.

Dataset	Domain	Reported Performance
SARA	Legal	GPT-4 (0.868) (Blair-Stanek, Holzenberger, and Van Durme 2023; Holzenberger and Van Durme 2023)
ContractNLI	Legal	GPT-4 (0.82–0.96; varies by contract type) (Narendra, Shetty, and Ratnaparkhi 2024; Yao et al. 2024)
BiomedNLI	Medical	LLaMA 70B (0.793)

Table 5: Performance of large models on original hypotheses from standard benchmarks (as reported in prior work).

Model	ContractNLI	BiomedNLI
LLaMA 3.3 70B	0.902	0.812
GPT-4o	0.861	0.901
GPT-4o mini	0.811	0.856

Table 6: Simple prompting accuracy on original hypotheses from ContractNLI and BiomedNLI.

Validation with Standard NLI Models. To further verify the increased difficulty of our newly generated hypotheses, we evaluated both the original and enhanced hypotheses using two standard NLI models: RoBERTa-NLI and BERT-NLI. As shown in Table 7, both models performed worse on the enhanced hypotheses across all benchmarks.

Human Evaluation of Enhanced Benchmarks. To complement automated evaluation and ensure the validity of our enhancement, we sampled 100 examples from all enhanced benchmarks for manual quality check. These samples were assessed for semantic clarity, logical soundness, and consistency with the intended reasoning phenomena. A subset of these examples, particularly those targeting defeasible reasoning, implication, and negation. Illustrative examples from this evaluation process are shown in Figure 4.

Model	ContractNLI	BioMedNLI	SARA
Original Hypotheses			
RoBERTa-NLI	0.681	0.659	0.702
BERT-NLI	0.642	0.633	0.671
Enhanced Hypotheses			
RoBERTa-NLI	0.512	0.486	0.529
BERT-NLI	0.478	0.441	0.501

Table 7: Accuracy of off-the-shelf NLI models on original vs. enhanced hypotheses across three benchmarks.

Appendix H - Additional Evaluation Results

Appendix H.1 - LOGT Performance against the Average Performance of All Baselines

Based on the results shown in Fig. 6, when comparing LogT against the average performance of all baselines rather than the strongest individual baseline, our approach demonstrates consistent and substantial improvements across all benchmarks and reasoning modes. LogT achieves notable gains ranging from +2.8% to +24.0% across different datasets. The improvements are most pronounced on the Biomed dataset, where LogT shows exceptional gains of +24.0% in negation and +23.0% in implication reasoning modes. These results highlight LogT's robustness across diverse reasoning tasks. The consistent improvements in all four reasoning modes demonstrate that LogT's structured knowledge synthesis approach provides systematic advantages over existing methods.

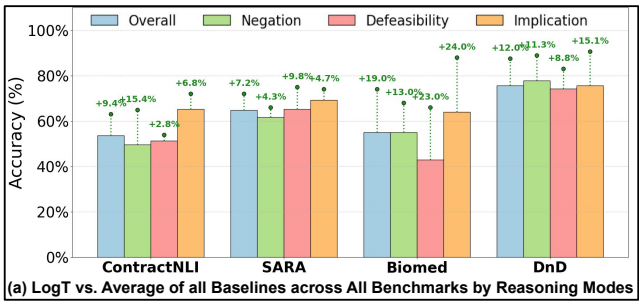


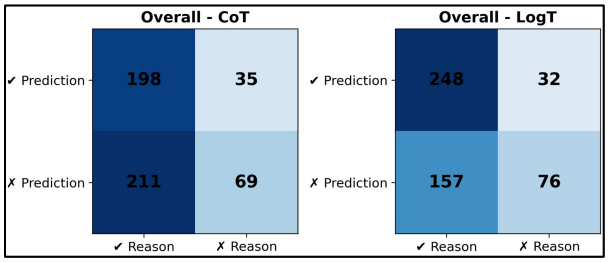
Figure 6: LogT+reasoning accuracy against baseline reasoning accuracy.

Appendix H.2 - Reasoning Trace Instance-level Analysis

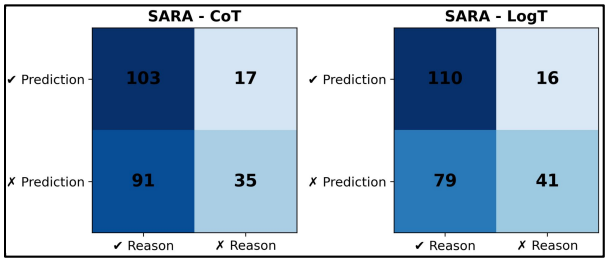
We analyze the reasoning traces of LogT in comparison to baseline CoT reasoning and find improved alignment between the correct predictions produced by LogT with correct reasoning traces. Using our *LLM-as-a-judge* setup to categorize reasoning traces and assign correct or incorrect reasoning labels, we find that LogT has correct reasoning traces for 89% of its correct predictions, improving on CoT's 85% alignment overall (see Fig. 7). We observe similar improvements when we break this down by dataset. 86% alignment for correct prediction to 87% for SARA, 88% to 90% for Biomed, and 79% to 83% for D&D.

We further break down the reasoning traces to look for the qualitative differences between our LogT approach and CoT (see Fig. 8). We find that the greatest difference between the reasoning traces of the two methods is a greater emphasis placed on *apply_rule* steps, which are given 2.88 steps on average by the LogT predictions compared to just 0.81 in the CoT traces. This trend holds when we breakdown the analysis down into individual datasets and suggests that contexts provided from LogT forces models to adhere to the given ontology.

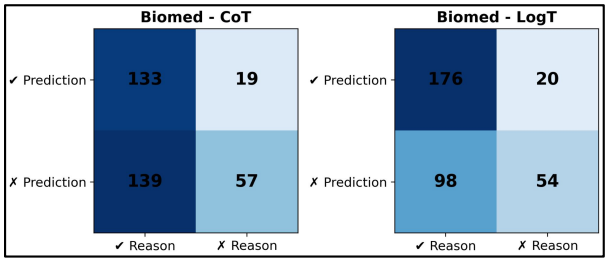
Interestingly, the other big differences between the traces, more steps devoted to *fact_lookup* (with 2.9 steps on average from the CoT approach and 3.24 steps on average from LogT overall) and fewer steps devoted to *check_condition* reasoning (2.14 steps on average for CoT and just 1.58 for LogT) by LogT do not hold for all of our individual datasets. We still observe more *fact_lookup* steps and fewer *check_condition* steps in the SARA and Biomed datasets, but the opposite holds for the D&D dataset. This points to value of our custom D&D dataset, which may require different reasoning for LLMs to successfully solve it.



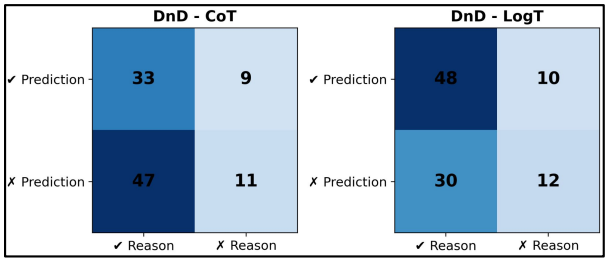
(a) Overall



(b) SARA

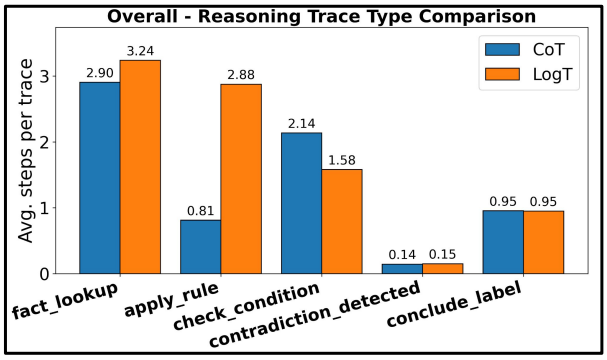


(c) Biomed

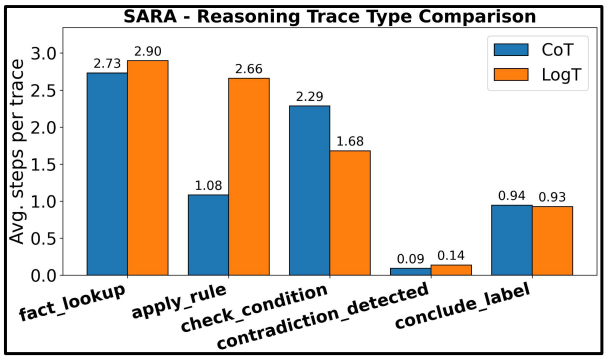


(d) D&D

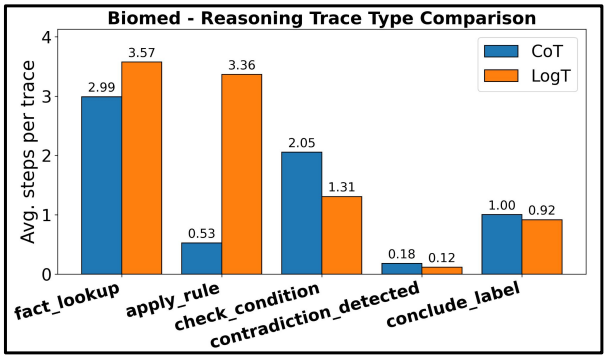
Figure 7: Comparison of confusion matrices for CoT and LogT across three benchmarks; (a) Overall, (b) SARA, (c) BiomedNLI, and (d) D&D. LOGT improves the alignment between correct predictions and correct reasoning traces.



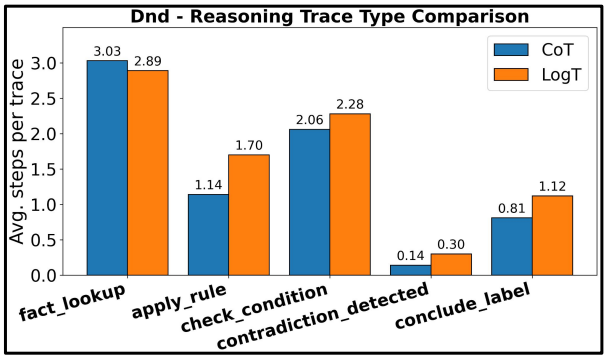
(a) Overall



(b) SARA



(c) Biomed



(d) DnD

Figure 8: Comparison of reasoning trace types and their average steps per trace from our approach LogT and baseline CoT across datasets, based on *LLM-as-a-judge* annotations. We observe that LogT tends to use more *apply_rule* steps irrespective of dataset. *check_condition* steps produced by LogT are shorter for most datasets, except for D&D, where LogT produces slightly more such steps than CoT. Likewise, *fact_lookup* steps are longer for LogT for most datasets, except for D&D.

Appendix H.3 - Reasoning Trace Qualitative Analysis

To understand the qualitative differences between Chain-of-Thought (CoT) prompting and our proposed LOGT, we compare their generated reasoning traces on a set of representative problems. Our analysis focuses on how each model structures its thought process, decomposes complex reasoning steps, and ultimately arrives at a solution. In these examples, LOGT receives additional structured context—either symbolic or logical—while CoT relies solely on natural language reasoning. We highlight key differences in how models approach problem decomposition, symbol grounding, and final answer extraction, particularly in settings that require multi-step inference or domain-specific knowledge.

Across both panels in Table 8, we observe that LOGT consistently produces more structured, faithful reasoning traces compared to Chain-of-Thought (CoT) prompting. While CoT often approximates logical inferences through informal natural language steps, it tends to leave out critical rule applications or misinterpret negations, as seen in examples involving marital status under §7703(b) or eligibility based on HER2 expression. In contrast, LOGT explicitly grounds each step in retrieved facts (e.g., marriedTo(alice, bob) or hasGivenSomethingForServices(alice, bob)) and applies formal inference rules, resulting in accurate conclusions with transparent justification. The rightmost columns of both panels further confirm that CoT frequently fails to predict the correct label, while LOGT succeeds across all examples.

The examples in the second panel of Table 9 further highlight LOGT's superiority in handling negation, conditional logic, and statute interpretation. In the Section 2(b)(2)(C) example, CoT provides a vague eligibility judgment, whereas LOGT correctly falsifies the hypothesis by checking for remarriage and applying conjunctive rule conditions. Similarly, in the Section 3306(b)(7) case, CoT treats the exchange of services imprecisely, while LOGT uses symbolic inputs to formalize the condition and apply the correct logical negation. These examples underscore the importance of integrating structured symbolic reasoning into language model pipelines, particularly for domains requiring legal precision, multi-hop deduction, or verifiable justification.

Model	CoT Trace	LogT Trace	CoT Prediction	LogT Prediction
Claude-Haiku	**Step 1:** Identify Key Conditions of §7703(b) **Step 2:** Alice married Bob in 2018, filed separately → meets first two conditions **Step 3:** Maintains home for Charlie, paid over half costs **Step 4:** Bob lived at home from Sept 2017 to Feb 2018 **Step 5:** Bob was member ¿ 6 months → fails condition **Step 6:** Conclude: §7703(b)(3) does NOT apply to Alice	**Step 1:** Formalize: married + sep return + spouse not present in last 6 months **Step 2:** `marriedTo(alice, bob)` **Step 3:** `maintainsHomeFor(alice, charlie)`, `entitledToDeductionFor` **Step 4:** `notMemberOfHousehold(bob, last_6_months_of_2018)` **Step 5:** Rule fires → sectionApplies(section_7703b3, alice) **Step 6:** Conclude: §7703(b)(3) likely applies to Alice	✗	✓
Llama-3.3-70B	**Step 1:** Alice married Bob (April 5, 2012) → considered married for 2018 **Step 2:** Files separate return **Step 3:** Maintains home for Charlie since Sept 16, 2017 **Step 4:** Furnished over half the cost (assumed) **Step 5:** Bob not in household from Feb 10 – Dec 31, 2018 **Step 6:** Says 3 of 4 conditions met; missing explicit confirmation of cost **Step 7:** Concludes: §7703(b) applies likely but uncertain	**Step 1:** `maintainsHomeForYear(alice, 2018)` via home ¿ half-year **Step 2:** `marriedTo(alice, bob)` **Step 3:** `maintainsHomeFor(alice, charlie)`, `entitledToDeductionFor(alice, charlie)` **Step 4:** `spouseNotMemberOfHousehold(alice, last_6_months_of_2018)` **Step 5:** Fires rule: sectionApplies(section_7703b3, alice) **Step 6:** Conclusion: §7703(b)(3) applies to Alice **Step 7:** Final step confirms hypothesis is supported	✗	✓
Deepseek-R1	**Step 1:** Lookup HER2- breast cancer and ErbB2(HER2) overexpression **Step 2:** HER2 overexpression is a hallmark of HER2+ cancer **Step 3:** HER2- status means no HER2 overexpression **Step 4:** Trial requires HER2 overexpression **Step 5:** Apply logic: HER2- cancer does not satisfy HER2+ criterion **Step 6:** Conclude: HER2- and HER2+ are mutually exclusive → contradiction with eligibility	**Step 1:** Check condition mismatch: HER2- vs HER2+ eligibility **Step 2:** Lookup: overexpression is required for trial eligibility **Step 3:** HER2- patients would not meet inclusion criteria **Step 4:** Apply contradiction rule: HER2- patient with HER2+ trial **Step 5:** Conflict with trial requirements confirmed **Step 6:** Identify mutually exclusive biological states **Step 7:** Conclude: aligns with hypothesis that trial requirements not met	✗	✓
Deepseek-R1	**Step 1–2:** Identify dosing regimen for Letrozole in primary trial **Step 3:** Confirm sentinel lymph node biopsy in secondary trial **Step 4–5:** Eligibility rules link interventions to trials **Step 6–8:** Apply to patients 1–3, showing treatment mappings **Step 9–10:** Verify protocol adherence per trial **Step 11–12:** Each hypothesis clause aligns with relevant info **Step 13:** Final check confirms no contradictions	**Step 1–2:** Identify Letrozole and biopsy interventions **Step 3:** Link eligibility to trial interventions **Step 4:** No explicit eligibility criteria found **Step 5:** Secondary trial intervention lacks specificity **Step 6:** Primary trial aligns with hypothesis **Step 7:** Final output: hypothesis cannot be fully validated due to missing details	✗	✓

Table 8: Qualitative comparison of reasoning traces produced by Chain-of-Thought (CoT) and LOGT for statutory tax reasoning tasks. CoT fails to satisfy or confirm all required conditions in both cases, while LOGT correctly applies formal rules and reaches the correct conclusion.

Model	CoT Trace	LogT Trace	CoT Prediction	LogT Prediction
GPT-4o-mini	**Step 1–3:** Establish family relationships (Charlie is Alice's father, Bob is Charlie's brother) **Step 4–5:** Apply rules to derive that Alice is a descendant of Bob's sibling **Step 6:** Check the negation condition and falsify it **Step 7:** Conclude the hypothesis is supported	**Step 1:** Define section 152(d)(2)(E) **Step 2–3:** Map family structure (Bob–Charlie–Alice) **Step 4:** Misread or test a negated condition **Step 5–6:** Reasoning becomes unclear—confuse relationship alignment **Step 7:** Conclude ambiguity remains in whether Alice qualifies	✗	✓
LLaMA-3.3-70B	**Step 1:** Read section 2(b)(2)(C) → conditions for surviving spouse **Step 2:** Bob has not remarried after Alice's death **Step 3:** Assumed residents, joint return applies **Step 4:** Bob qualifies as surviving spouse in 2014 **Step 5:** Limitation clause likely not violated **Step 6:** Hypothesis judged plausible **Step 7:** Conclusion: hypothesis is supported	**Step 1:** `diedOn(alice, july_9_2014)` and `marriedTo(alice, bob)` **Step 2:** -¿ `spouseDiedDuringTaxableYear(bob, 2014)` **Step 3:** Checks that Bob was not remarried **Step 4:** Fires rule: `jointReturnPossible(bob, 2014)` **Step 5:** All rule inputs met -¿ `section_2b2c_applies(bob, 2014)` **Step 6:** Conclusion: §2(b)(2)(C) applies to Bob **Step 7:** Contradiction check falsified negation	✗	✓
Claude-Haiku	**Step 1:** Defines 'employer' and wage conditions **Step 2:** Alice gave Bob a typewriter valued at $323; Bob painted her house **Step 3:** Interprets exchange as non-cash compensation **Step 4:** Recognizes exchange occurred outside workplace duties **Step 5:** Suggests statutory implications may arise **Step 6:** Concludes that further analysis of Section 3306(b)(7) is needed	**Step 1:** `hasGivenSomethingForServices(alice, bob)` established **Step 2:** `Section 3306(b)(7)` retrieved – wage definitions **Step 3:** Applies special rule (paragraph 4) to non-standard wage **Step 4:** Confirms Alice gave a typewriter in exchange for services **Step 5:** Condition: Section applies unless Alice *never* gave anything **Step 6:** Applies negation logic: Alice has given something **Step 7:** Concludes Section 3306(b)(7) likely applies	✗	✓

Table 9: Qualitative comparison of reasoning traces produced by Chain-of-Thought (CoT) and LOGT for statutory tax reasoning tasks. CoT fails to satisfy or confirm all required conditions in both cases, while LOGT correctly applies formal rules and reaches the correct conclusion.