CAUSAL REASONING FAVORS ENCODERS: LIMITS OF DECODER-ONLY MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

In-context learning (ICL) underpins recent advances in large language models (LLMs), yet its role in causal reasoning remains unclear. Causal reasoning demands multi-hop composition and strict conjunctive control, and reliance on spurious lexical relations of the input could provide misleading results. We hypothesize that, due to their ability to project the input into a latent space, encoder- and encoder–decoder architectures are better suited for said multi-hop conjunctive reasoning versus decoder-only models. To do this, we compare fine-tuned versions of all the aforementioned architectures with zero- and few-shot ICL in both natural-language and non-natural language scenarios. We find that ICL alone is insufficient for reliable causal reasoning, often overfocusing on irrelevant input features. In particular, decoder-only models are noticeably brittle to distributional shifts, while fine-tuned encoder and encoder–decoder models can generalize more robustly across our tests, including the non-natural language split. Both architectures are only matched or surpassed by decoder-only architectures at large scales. We conclude by noting that for cost-effective, short-horizon robust causal reasoning, encoder or encoder-decoder architectures with targeted fine-tuning are preferable.

1 INTRODUCTION

In-context learning (ICL) is central to the success of large language models (LLMs) because it enables generalization to unseen tasks without parameter updates. Yet, its performance on *causal* reasoning remains unsettled. Causal reasoning imposes two stringent requirements: *multi-hop composition* and *strict conjunctive control*. The former is the ability to chain several elementary implications or constraints to derive a conclusion across multiple intermediate steps. The latter requires the issuance of a positive decision only when all relevant premises, guards, and boundary conditions are simultaneously satisfied; and rejecting otherwise.

Understanding whether LLMs truly perform causal reasoning matters beyond their widespread adoption. Causal competence underpins scientific inference, reliable decision making under interventions, and safety-critical applications where spurious correlations could lead to inaccurate predictions, and thus brittle and unsafe behavior. Causal judgments must remain stable under representation changes, as well as under all types of distributional and representational shifts–that is, those that preserve the underlying structure and relations of the data, but may alter their "surface form", such as the words used to represent said structure.

It is known that LLMs exhibit limitations in causal reasoning, including difficulty identifying causal relations between variables (Jin et al., 2023b) and reliance on spurious lexical features (Zečević et al., 2023). These shortcomings are particularly visible in natural language settings (Jin et al., 2023a). For example, Nezhurina et al. (2024) demonstrate that LLMs fail on simple, human-solvable commonsense math problems, exposing a sharp breakdown in basic reasoning and generalization. Similarly, Liu et al. (2024) find that even the strongest LLMs perform poorly on tasks requiring data analysis and causal reasoning, underscoring persistent gaps in their reasoning abilities. Further, Chi et al. (2024) show that LLMs often rely on shallow, correlation-based patterns rather than engaging in genuine human-like causal reasoning, with performance degrading substantially on novel causal tasks

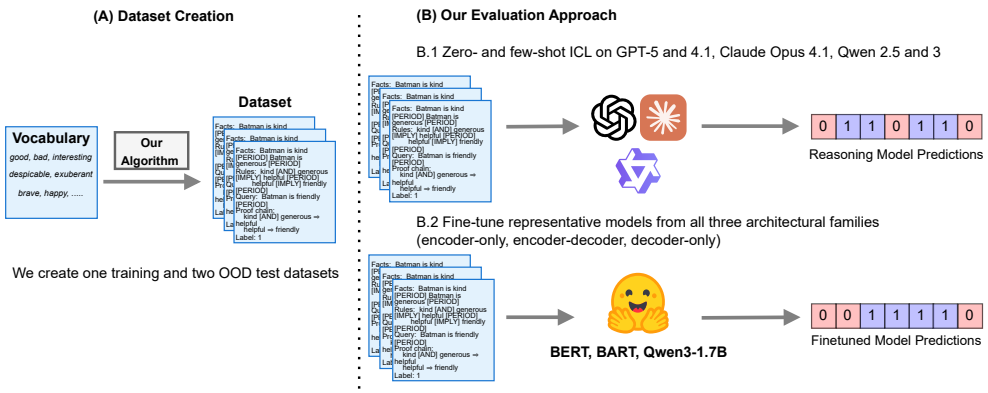


Figure 1: Overview of our approach. *Left, (A)*: Dataset creation. From a fixed set of propositions, we generate one training set, and two out-of-distribution (OOD) test sets. *Right, (B)*: For our evaluation we conduct zero- and few-shot inference with decoder-only reasoning and non-reasoning models. We also fine-tune models from all three architectural families (encoder-only, encoder–decoder, and decoder-only); namely, variants of BERT and BART, and Qwen3-1.7B (Yang et al., 2025)

and benchmarks. However, there has not been a systematic comparison between decoder-only LMs[1] and encoder-enabled architectures such as BERT (Devlin et al., 2019) and BART (Lewis et al., 2019).

In this work we compare encoder and encoder–decoder architectures with decoder-only architectures. Encoder-based architectures have the innate ability to project the full input into a latent space; while decoder-only architectures perform inference recursively, in a token-by-token fashion. Our hypothesis is that an encoder's projective ability will allow them to perform more reliable causal composition, especially under distributional shifts, when compared to decoders. To test this we evaluate a series of encoder, decoder, and encoder-decoder architectures under various out-of-distribution (OOD) scenarios, with synthetic data especially designed to evaluate their robustness to distributional shifts.[2] We test these distributional shifts by (1) progressively deeper reasoning chains in a subset of first-order logic (FOL), and (2) the same dataset with randomized characters to ablate out lexical relations (e.g., "Batman is kind" in the first would be "Batman is a#d}" in the second).

We find that, under our setup, encoder-only models are able to generalize better than decoders. This means that they are able to have better accuracies at deeper reasoning chains, as well as learn to focus on the underlying logical structures of the data. This latter part holds both for non-finetuned (i.e., zero- and few-shot) and finetuned decoder-only models. Moreover, this performance gap becomes wider as the reasoning chains in both scenarios become deeper. The only outlier to our findings is a recent, state-of-the-art (SOTA) model, GPT-5 (OpenAI, 2025b), which attained near-perfect accuracy in all our tests. However, this incurred substantial latency and–presumably–higher compute cost. We thus conclude that **ICL alone is not an effective mechanism for causal reasoning**. Given the gaps in accuracy and compute cost, we argue that robust, cost-effective performance may be achieved with conventional encoder or encoder–decoder architectures and minimal fine-tuning.

2 Related Works

The intersection of causality and large language models (LLMs) has attracted significant attention in recent years. Several surveys provide broad entry points into this emerging area. Ma (2025) and Liu et al. (2025) review how LLMs have been applied to causal inference tasks such as discovery (Jin et al., 2023b), intervention, and counterfactual reasoning, while Li et al. (2025) offers an overview of current challenges and future directions in causal reasoning of LLMs.

[1]Throughout, when the architecture is undocumented we assume the model is decoder-only.

[2]All code and artifacts are available at https://anonymous.4open.science/r/causality_grammar-DB41/README.md.

Early investigations have explored causal reasoning in encoder-only and encoder–decoder architectures. For example, Pirozelli et al. (2024) study whether encoder-based models can handle logical reasoning tasks such as propositional and FOL, including validity checking and theorem proving. Chen et al. (2023) introduce a Causality-Inspired Sequence-to-Sequence model (CI-Seq2Seq) designed to learn causal representations that mimic underlying causal factors for abstractive text summarization. Similarly, Zhou et al. (2024) evaluate models from the BERT family on a range of causal inference tasks, contrasting their performance with that of decoder-only LLMs and traditional causal learning algorithms.

Yao & Koller (2022), Liu et al. (2022) have shown limitations of attention based models on casual reasoning, more recent work has shifted attention to decoder-only LLMs and their reasoning capabilities. Recent studies consistently show that LLMs struggle with fundamental reasoning: they fail on simple commonsense math (Nezhurina et al., 2024), underperform on data analysis and causal inference (Liu et al., 2024), and often default to shallow correlation-based heuristics (Chi et al., 2024; Mondorf & Plank, 2024). Kiciman et al. (2023) show that some causal capabilities of LLMs can generalize beyond memorized data—for instance, to datasets introduced after the training cutoff—yet performance still exhibits unpredictable failure modes and notable limitations, especially when reasoning requires non-textual data or explicit causal structures. In a related direction, several recent studies have proposed benchmarks to evaluate the reasoning capabilities of LLMs: (Goyal & Dan, 2025; Bean et al., 2024) introduce benchmarks to assess LLMs on complex linguistic reasoning tasks. Their evaluations reveal consistent limitations of current models, particularly in compositional generalization, rule abstraction, and multi-step out-of-domain reasoning. Complementing these, Wang (2024) targets causal reasoning across text, math, and code, probing four directions—including interventions—to separate true causal understanding from guesswork and relate it to hallucination propensity.

3 Background

3.1 Causal Reasoning and Our Dataset

Causal reasoning is the process of determining how individual facts or conditions combine to produce an overall outcome (Penn & Povinelli, 2007; Pearl, 2009). Unlike simple associative prediction, it requires models to identify dependencies between propositions and to compute how local truths influence global conclusions. In practice, this almost always entails *multi-hop reasoning*, where intermediate inferences must be chained together across several steps. For example, given clauses X, Y, and Z, a model must first verify whether each clause holds (local checks), then combine them through logical connectives (e.g., $X \vee Y$ at the clause level), and finally apply strict conjunctive control across all clauses to decide whether the full formula is satisfied (e.g., $(X \vee Y) \wedge Z$). This process highlights that accuracy depends not only on local classification but also on the correct *composition* of results across depth.

In the context of our work, we focus on testing specifically these requirements. By stratifying instances according to compositional depth, we enforce tasks where solving the problem requires multiple reasoning hops: at shallow depths, decisions can often be made with one or two local checks, but at greater depths, models must aggregate larger sets of clauses under global conjunctions. From our framing's perspective, reasoning over the structure (clause relations) is more important than the lexical relations encoded in them, and hence we make this the central aspect of our work.

3.2 Architectural Considerations for Causal Reasoning

In this section we provide an informal argument on how encoder layers could have an advantage over decoder-only architectures when dealing with causal reasoning.

Remark that logical classification requires aggregating dispersed evidence across an input sequence. Encoder architectures are well-suited to this task because each layer allows every token to integrate information from the entire sequence. This means that it will be able to express, in its own latent space, every element of an input sequence as a linear combination of the learned features and the other inputs. In other words, this allows for instant global information sharing.

Formally, let s be an n-token input sequence $s = \langle x_1, x_2, \ldots x_n \rangle$, where every x_i is represented by a d_{in}-dimensional vector; $x_i \in \mathbb{R}^{d_{\text{in}}}$. This sequence may be then rewritten as a matrix $X \in \mathbb{R}^{n \times d_{\text{in}}}$.

In the context of encoder layers, a encoder layer ℓ of hidden dimension d, for $\ell \colon \mathbb{R}^{n \times d_{\text{in}}} \to \mathbb{R}^{n \times d}$, transforms X into a contextualized hidden state H. A classification decision would then be the result of pooling h into a single vector $z = \text{pool}(h)$, for some pooling (aggregation) function $\text{pool} \colon \mathbb{R}^{n \times d} \to \mathbb{R}^{d}$. From this perspective, z may be viewed as the output of a projection onto $\mathbb{R}^{d}$. Informally, this projection collapses the information from all tokens into a global representation. In other words, logical programs of the form

$$\text{(literals)} \Rightarrow \text{(clause-level disjunction)} \Rightarrow \text{(global conjunction)} \tag{1}$$

are encoded onto the projection. Given sufficient observations, this mechanism could allow a model to evaluate, in a single pass, such programs by repeatedly projecting and composing the *full* sequence.

In contrast, decoder-only architectures are recursive. To read and aggregate information distributed across the sequence, the model must propagate it step-by-step from left to right. There is still a projective step, but, algorithmically, the output at position t depends only on the previously-observed and generated tokens. If the input clauses are not given in implication order, a solver will require some backtracking to fully consider and evaluate all given clauses. While reasoning models are able to do this to an extent through their "baked-in" chain-of-thought, it comes at the cost of a non-controllable inference process and multiple calls to the same model.

4 Methods

4.1 Dataset

From 3.1, it follows that a–comparatively–simple evaluation of causal reasoning capabilities could be carried out on FOL. Hence, we base our work on a benchmark known as SimpleLogic (Zhang et al., 2022). SimpleLogic is designed to evaluate deductive reasoning skills in a subset of FOL that excludes disjunctions. Every example from SimpleLogic is an algorithmically-generated tuple ⟨facts, rules, query, explanation, label⟩, where the facts are given atoms; the rules are definite clauses; the query is a single atom; and the label indicates whether the query can be deduced. All atoms are drawn from a vocabulary that leverages natural language (e.g., "Amy is sad"). The resulting entries may not be logical from a commonsense perspective, but they are valid within FOL. SimpleLogic uses a templatized language, which allows for controllable input length, linguistic variability, and reasoning depth–that is, the minimum number of reasoning steps needed to derive the truth value of the query. We refer to the reasoning steps as "Proof Chain". In turn this ensures that difficulty is governed by logical complexity, rather than linguistic features.

In this work we create a base training set, and two OOD test datasets. The *training set* is akin to the original SimpleLogic work, with full natural-language strings generated by the base algorithm. It has 40,000 samples from depths 0 to 7, with 5,000 samples per depth. The *NL dataset* is a test set analogous to the training set, but manifests OOD by including deeper (up to 11) sequences. Finally, the *NNL dataset* is constructed by sampling random characters to form an ungrammatical, likely unseen by the tokenizers, vocabulary; and then continue generating the dataset as before. Both test sets have 3,600 samples each, from depths 0 to 11, and 300 samples per depth. Since each dataset spans 12 depths in total, we henceforth refer to them as the Natural Language (NL) Depth-12 Test Dataset and the Non-Natural Language (NNL) Depth-12 Test Dataset. See Figure 2 for examples of entries in our corpora, and Appendix A for in-depth details.

4.2 Models Used

For our encoder-only evaluation, we used two BERT variants (base and large), and for encoder-decoders we used BART variants (base and large). For the decoder-only models we evaluated two non-reasoning models, GPT-4.1 (OpenAI, 2025a) and Qwen 2.5 (Bai et al., 2025); and three reasoning models, GPT-5, Claude Opus 4.1 (Anthropic, 2025), and Qwen3-1.7B. All of these models are considered state-of-the-art LLMs, albeit only the Qwen-line of LLMs have open weights. We finetuned BERT, BART, and Qwen3. See Appendix B for further details on our methodology.

```
Facts:  Batman is kind [PERIOD] Batman is generous
Rules:  kind [AND] generous [IMPLY] helpful [PERIOD]
        helpful [IMPLY] friendly [PERIOD]
Query:  Batman is friendly [PERIOD]
Proof chain:
    kind [AND] generous ⇒ helpful
    helpful ⇒ friendly
Label: 1
Depth: 2
```

```
Facts:  Batman is a#d} [PERIOD] Batman is pqrs
Rules:  a#d} [AND] y_hu] [IMPLY] u&^ho [PERIOD]
        u&^ho [IMPLY] {hu]? [PERIOD]
Query:  Batman is {hu]? [PERIOD]
Proof chain:
    Cannot apply rule a#d} ∧ y_hu] ⇒ u&^ho
    because missing: y_hu]
Label: 0
Depth: 1
```

Figure 2: Sample datapoints from our corpora. *Left*: an NL entry with depth 2. Remark that it will not always be a natural sentence, although it is guaranteed to be a valid set of clauses in SimpleLogic. The proof chain for this example contains the sequence of reasoning steps which solves it. *Right*: an NNL entry with depth 1. The proof chain in this example indicates that it is an unsolvable problem, given that the atom "y_hu]" is not in the derivation. In all corpora we use the separators [AND], [IMPLY], and [PERIOD] to separate elements of SimpleLogic from the atoms.

4.3 Evaluation metrics

We evaluate models using three complementary views: overall accuracy as a point estimate of correctness; *per–depth* precision, recall, and F_1-score to determine how performance changes with reasoning complexity; and threshold–swept discrimination via ROC curves and area under the curve (AUROC), which assesses ranking quality independent of a fixed decision threshold. We utilize the latter as our measure of statistical significance. Unless stated otherwise, depth–wise results are summarized with *macro* averages (the unweighted mean across depths) so that each depth contributes equally. Our setting is binary; accordingly, we report AUROC for the positive class. We describe our prompt methods, including how outputs are requested for each model family, in Appendix B.5.

5 Results

5.1 Non-Finetuned Results

In the non-finetuned setting—where decoder-only models are evaluated via in-context learning (ICL)—we test encoder, encoder–decoder, and decoder-only architectures in 0- and 5-shot configurations. GPT-4.1 improves by 0.01 absolute (0.5%) and Qwen-2.5 by 0.002 absolute (0.48%) (Figure 3). On the Natural-Language depth-12 test set, GPT-4.1 attains 0.64 (0-shot) and 0.65 (5-shot); Qwen-2.5 reaches 0.471 (0-shot) and 0.473 (5-shot); Qwen3-1.7B (0-shot) achieves 0.65. On the Non-Natural-Language depth-12 test set, GPT-4.1 is 0.65 in both 0- and 5-shot; Qwen-2.5 is 0.53 (0-shot) and 0.54 (5-shot); Qwen3-1.7B (0-shot) achieves 0.61.

For all models, except the reasoning models, there was no clear trend in performance differences between the NL and NNL datasets: GPT 4.1 performs roughly the same, Qwen 2.5 performs better in NNL compared to its performance in NL and Qwen3-1.7B performs better in NL compared to its performance in NNL.

These results are supported by our measures of statistical significance: (refer Table 1)

Dataset	**Model**	**AUC (Non-Finetuned)**
Natural Language Test	BART-Large	0.525
Natural Language Test	BERT-Large	0.376
Natural Language Test	Qwen3-1.7B	0.424
Non-Natural Language Test	BART-Base	0.510
Non-Natural Language Test	BERT-Base	0.511
Non-Natural Language Test	Qwen3-1.7B	0.492

Table 1: AUC values of non-finetuned models across Natural Language and Non-Natural Language test sets.

In the state-of-the-art reasoning LLMs, we compared GPT-5 (0-shot) with Claude Opus 4.1 under both 0-shot and 5-shot in-context learning (ICL) setups. Using template-identical prompts across the NL and NNL test suites and averaging metrics over depths 0–11, we observed a clear performance hierarchy (detailed results are in Tables 6, 7, 10 and 11). On the Natural Language (NL) data, GPT-5 (0-shot) achieved near-perfect accuracy (1.00), while Claude Opus 4.1 scored similarly in both 5-shot (0.93) and 0-shot (0.93) conditions. This performance gap was more pronounced on the Non-Natural Language (NNL) data, where GPT-5 (0-shot) again reached perfect accuracy (1.00), whereas Claude Opus 4.1 saw a notable drop, scoring 0.66 with 5-shots and 0.65 with 0-shots.

In terms of output compliance, the models were skewed. In the NL split, BERT-Large output the positive label for all 3,600 items (Figure 4a). The same pattern could be reproduced with BERT-Base in the NNL split (Figure 5a). Other models did not reply with the appropriate label to begin with: BART-Large had 98.5% non-compliant outputs, BART-Base -100%, and Qwen3-1.7B - 94.2% (NL) and 82.3% (NNL).

5.2 Finetuned Results

In the finetuned setting (on the depth-12 test sets,) depth-wise average accuracy differs across models and splits (Figure 6). For the NL split, BART-Large attains **0.75**, followed by Qwen3-1.7B 0.73 and BERT-Large 0.71 (Figure 6a). For the NNL split, BERT-Base reaches **0.61**, BART-Base 0.55, and Qwen3-1.7B 0.53 (Figure 6b). These figures align with the tabulated results in Tables 5 and 9.

These results are supported by our measures of statistical significance: (refer Table 2)

Dataset	**Model**	**AUC (Finetuned)**
Natural Language Test	BART-Large	0.623
Natural Language Test	BERT-Large	0.759
Natural Language Test	Qwen3-1.7B	0.499
Non-Natural Language Test	BART-Base	0.534
Non-Natural Language Test	BERT-Base	0.604
Non-Natural Language Test	Qwen3-1.7B	0.596

Table 2: AUC values of finetuned models across Natural Language and Non-Natural Language test sets.

Model	**Inference Time** (hours)
BART-Large	0.2
BART-Base	0.1
BERT-Large	0.24
BERT-Base	0.17
Qwen3-1.7B	4.91
GPT-5	90.3
GPT-4.1	68.7
Claude Opus 4.1	30

Table 3: Average inference time (hr) for zero-shot, averaged across two test datasets (3600 smaples each) with the same hardware. (See Appendix B.3). Average inference time is much higher for GPT-5 and Claude Opus 4.1 comapred to BERT and BART

Finally, the analysis highlights different class-balance behaviors; Qwen3-1.7B, for example, attains very high Label-0 recall on NNL data but sacrifices Label-1 recall, which lowers its overall accuracy. In contrast, BART-Base is more balanced but weaker in absolute terms.

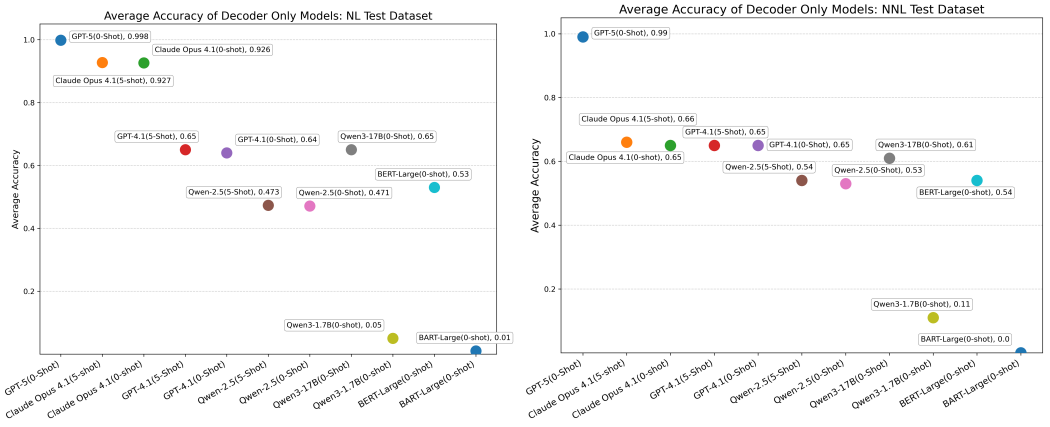
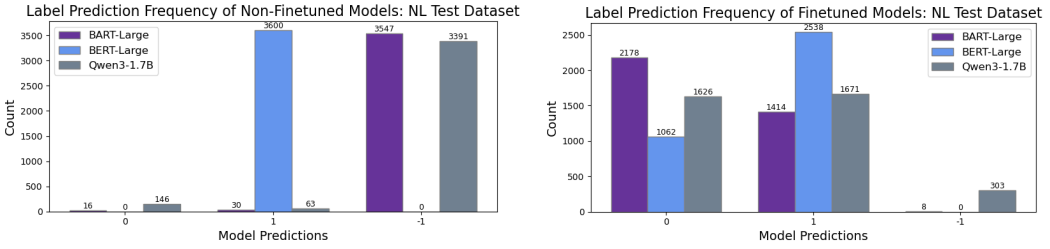
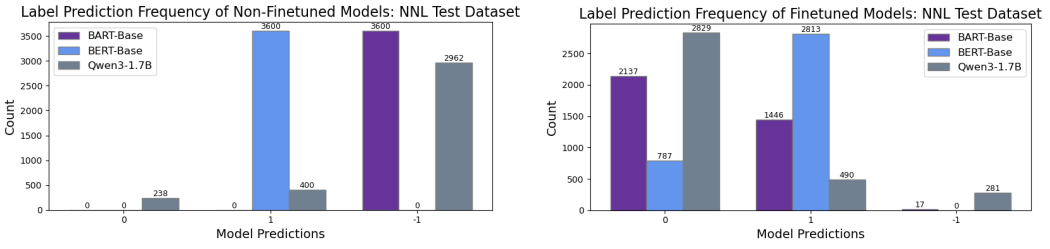


Figure 3: Comparison of average accuracy all non-finetuned models, showing their performance in the NL (*left*) and NNL (*right*) depth 12 test datasets. Increasing the number of exemplars for in-context learning (ICL) does not substantially improve accuracy for non-reasoning models, with only marginal gains observed: Claude Opus 4.1 (+0.05%), GPT-4.1 (+1%), and Qwen-2.5 (+0.5%) in both the datasets.



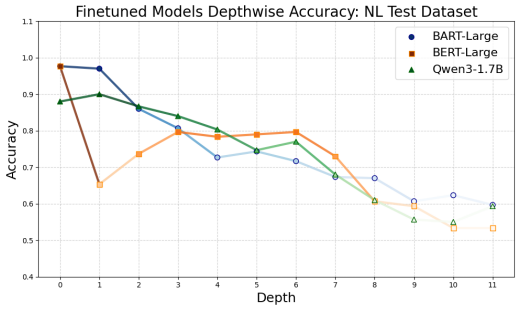
(a) Non-Finetuned Model (b) Finetuned Model

Figure 4: Comparison of label prediction frequencies of non-finetuned and finetuned models on the Natural Language Depth 12 Test Dataset. Each bar indicates the frequency with which a model assigns one of the possible labels (-1, 0, 1), where -1 indicates a parsing error. From the graphs it is clear that the lack of finetuning negatively affects all models. Label compliance increases from 1.3% to 99.8% in BART-Large, and from 5.8% to 91.6% in Qwen3-1.7B. After finetuning, label distributions are balanced for Qwen3-1.7B and BART-Large, whereas BERT-Large shows a skew toward label 1.

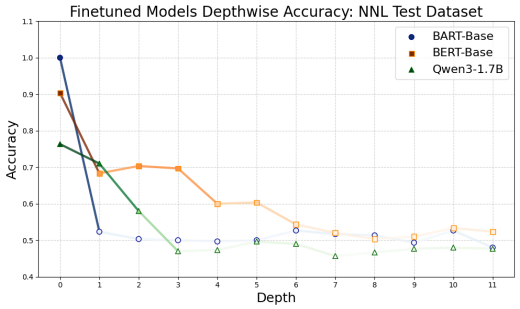


(a) Non-Finetuned Model (b) Finetuned Model

Figure 5: Comparison of label prediction frequencies of non-finetuned and finetuned models on the Non-Natural Language Depth 12 Test Dataset. Each bar indicates the frequency with which a model assigns one of the possible labels (-1, 0, 1), where -1 indicates a parsing error. Label compliance increases from 0% to 91.2% in BART-Base, and from 17.7% to 92.2% in Qwen3-1.7B. BART-Base and Qwen3-1.7B exhibit a skew toward label 0, whereas BERT-Base is skewed toward label 1.



(a) Average Depthwise Accuracy(Natural Language Depth 12 Test dataset)

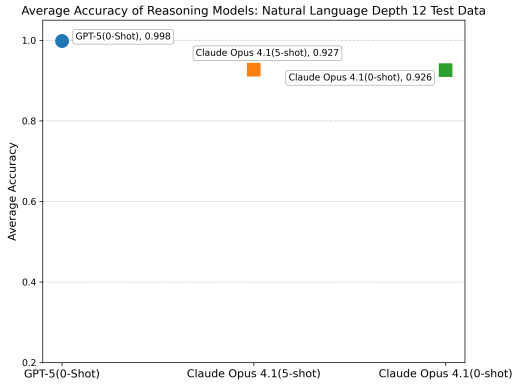


(b) Average Depthwise Accuracy (Non-Natural Language Depth 12 Test dataset)

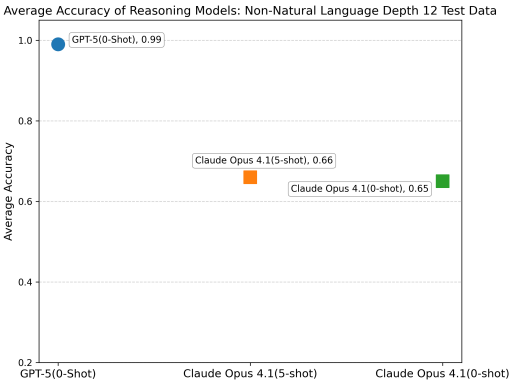
Figure 6: Comparison of average depthwise accuracy across finetuned models. Average model accuracy decreases with depth: on the Natural Language dataset, mean accuracy gradually declines from 0.9 to 0.5, whereas on the Non-Natural Language dataset it drops sharply from .89 to 0.55 and then plateaus. On average, encoder-based models outperform decoder-only models across most depths.

5.3 Study of SOTA Reasoning LLMs (In-context Learning)

Mechanistic Analysis. The depthwise results highlight a key architectural difference in how information is aggregated. Encoder-only and encoder–decoder models broadcast information across the entire sequence in each layer, so reasoning over multiple clauses can be achieved in a small, constant number of steps. In contrast, decoder-only architectures are constrained by cross attention: the hidden state at position t only attends to $\{1{:}t\}$, which forces information to be propagated step by step toward the output position. This sequential bottleneck implies that the "information horizon" scales linearly with the number of layers L, so deeper reasoning tasks require either more depth or additional inference-time compute (e.g., chain-of-thought). The observed patterns—where GPT-5 attains near-perfect accuracy while Claude Opus 4.1 lags on the non-natural language (NNL) split—are consistent with this mechanism: *massive decoders can overcome the bottleneck by sheer capacity and iterative reasoning, but moderate-sized decoders remain brittle when clause-wise aggregation is required.*



(a) Average Accuracy vs. Reasoning Models (Natural Language Depth 12 Test dataset)



(b) Average Accuracy vs. Reasoning Models (Non-Natural Language Depth 12 Test dataset)

Figure 7: Side-by-side comparison of average accuracy across Reasoning models. GPT-5 achieves near perfect accuracy for both the Test Datasets. Claude Opus 4.1 performs worse in the Non-Natural Language Test Dataset

6 Discussion

Non-finetuned models. The non-finetuned setting highlights clear architectural limitations. Encoder-only models degenerate into label collapse, predicting a single class across all examples;

encoder–decoders and small decoder-only models suffer from label-emission failures, frequently omitting the required decision token. Even where outputs are compliant, performance gains from in-context learning are negligible: for example, GPT-4.1 and Qwen-2.5 show changes of less than 1% when moving from zero- to five-shot inference. These drawbacks demonstrate that, without finetuning, neither encoder-only nor moderately sized decoder-only models reliably aggregate logical facts, and autoregression alone does not suffice for robust causal reasoning.

Finetuned models. Finetuning exposes the strengths and sensitivities of each architecture. ROC curves in Figures 10 and 11 quantify the improvements, showing that encoder-based models are both more data-efficient and more robust to distributional shifts. In the natural-language setting, sequence-to-sequence models such as BART-Large reach the highest accuracies, while in the non-natural-language setting the strongest model is the smaller encoder-only BERT-Base. These results align with our hypothesis in §3.2, where encoders can project clause-level evidence in parallel while decoders must recurse across positions. The distribution sensitivity of sequence-to-sequence and decoder-only families further underscores the relative stability of encoders.

Large decoders. A notable exception to this trend is GPT-5, which reaches near-perfect accuracy in both natural-language and non-natural-language depth-12 test sets without finetuning or in-context examples. This performance ceiling reflects the dominance of powerful pretrained priors: massive decoder-only models can internally approximate the multi-step aggregation otherwise difficult for smaller architectures. However, the costs are non-trivial: inference with GPT-5 requires significant compute, longer runtimes (See Table 3), and consequently higher financial and environmental costs. Our findings suggest that while very large decoders can overcome the recursion bottleneck by leveraging capacity and scratchpad-style reasoning, their efficiency trade-offs are stark when compared to encoder-based models.

Implications. Taken together, the results indicate that encoder and encoder–decoder models remain the most resource-efficient choice for causal reasoning benchmarks, particularly under distributional shift. Decoder-only models can match or exceed their performance only at very large scales and with substantial inference-time compute. Thus, while scaling can close the gap, relying on autoregression alone for logical reasoning carries both architectural and practical drawbacks.

7 Conclusion

While in-context learning (ICL) has propelled large language models to the forefront of AI research, our study shows that their ability to perform causal reasoning remains limited. We systematically compared encoder-based architectures with decoder-only LLMs under the hypothesis that the recursive nature of ICL is a hindrance rather than a benefit for reasoning over structured logical forms. Our results support this view: most LLMs—including state-of-the-art reasoning models—struggled to match the efficiency and robustness of encoder-only models such as BERT, particularly at greater reasoning depths and under lexical perturbations. Fine-tuned encoders and encoder–decoders demonstrated superior stability under distributional shifts, validating our mechanistic account of global projection versus recursive aggregation. The sole exception was GPT-5, which attained near-perfect accuracy across both natural-language and symbolic test sets. We hypothesize that this outlier performance stems from a combination of immense capacity and built-in chain-of-thought priors, though at the cost of substantially higher inference-time compute. Taken together, these findings suggest a practical trade-off: encoders and encoder–decoders remain the most resource-efficient and reliable choice for causal reasoning tasks, while decoder-only models can only close the gap at massive scale and cost.

Causal reasoning is important in many contemporary applications of LLMs, ranging from explainable AI to its applications to scientific discovery. Although LLMs are convenient and easy to use, the results shown here illustrate that their application must be done with caution. Our work also suggests that ICL is limited in its ability to properly capture causal compositionality. While further mathematical development is required to formally show the bounds and limits to which this occurs, empirical work could explore the development of architectures that merge the convenience of ICL with the capabilities of encoder-based architectures.

8 ETHICS

The datasets used for our experiments were generated synthetically for the specific purpose of evaluating logical reasoning and do not contain any personally identifiable information (PII) or sensitive user data. The language models evaluated, such as BERT, BART, Qwen, and GPT, were used in accordance with their intended research licenses. While the primary goal of this work is to advance the scientific understanding of AI reasoning, we acknowledge that enhancing logical capabilities in models could have dual-use applications. We encourage the responsible development and deployment of such technologies. The environmental impact associated with training and evaluating these models was considered, and our findings highlight the efficiency of smaller, encoder-based models for specific tasks, which can guide more sustainable model selection in practice.

9 REPRODUCIBILITY STATEMENT

All code developed for data generation, model fine-tuning, and evaluation, along with the complete NL and NNL datasets, will be made publicly available in a GitHub repository upon publication. The pre-trained models used in this study are publicly available and were accessed from the Hugging Face Hub. Detailed configurations, including all hyperparameters, training scripts, and library versions (e.g., PyTorch, Transformers), will be provided in the repository to allow for the full replication of our experiments and to facilitate future research building upon this work. Hyperparameters, call parameters for API-based LLMs, and in-depth methodology are also reported in Appendix B

REFERENCES

Anthropic. Claude Opus 4.1, 2025. URL https://www.anthropic.com/news/claude-opus-4-1.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Andrew M Bean, Simi Hellsten, Harry Mayne, Jabez Magomere, Ethan Chi, Ryan Chi, Scott Hale, and Hannah Rose Kirk. Lingoly: A benchmark of olympiad-level linguistic reasoning puzzles in low resource and extinct languages. *Advances in Neural Information Processing Systems*, 37: 26224–26237, 2024.

Lu Chen, Ruqing Zhang, Wei Huang, Wei Chen, Jiafeng Guo, and Xueqi Cheng. Inducing causal structure for abstractive text summarization. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 213–223, 2023.

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. Unveiling causal reasoning in large language models: Reality or mirage? In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 96640–96670. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/af2bb2b2280d36f8842e440b4e275152-Paper-Conference.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Satyam Goyal and Soham Dan. Iolbench: Benchmarking llms on linguistic reasoning. *arXiv preprint arXiv:2501.04249*, 2025.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. Cladder: assessing causal reasoning in language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023a. Curran Associates Inc.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. Can large language models infer causation from correlation? *arXiv preprint arXiv:2306.05836*, 2023b.

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2023.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Xin Li, Zhuo Cai, Shoujin Wang, Kun Yu, and Fang Chen. A survey on enhancing causal reasoning ability of large language models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 399–416. Springer, 2025.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. Are llms capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. *arXiv preprint arXiv:2402.17644*, 2024.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, Julian McAuley, Wei Ai, and Furong Huang. Large language models and causal inference in collaboration: A comprehensive survey. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7668–7684, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.427. URL https://aclanthology.org/2025.findings-naacl.427/.

Jing Ma. Causal inference with large language model: A survey. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5886–5898, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.327. URL https://aclanthology.org/2025.findings-naacl.327/.

Philipp Mondorf and Barbara Plank. Beyond accuracy: evaluating the reasoning behavior of large language models–a survey. *arXiv preprint arXiv:2404.01869*, 2024.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models. *arXiv preprint arXiv:2406.02061*, 2024.

OpenAI. Introducing GPT-4.1 in the API, 2025a. URL https://openai.com/index/gpt-4-1/.

OpenAI. Introducing GPT-5, 2025b. URL https://openai.com/index/introducing-gpt-5/.

Judea Pearl. *Causality*. Cambridge University Press, 2 edition, 2009.

Derek C Penn and Daniel J Povinelli. Causal cognition in human and nonhuman animals: A comparative, critical review. *Annu. Rev. Psychol.*, 58(1):97–118, 2007.

Paulo Pirozelli, Marcos M José, Paulo de Tarso P. Filho, Anarosa AF Brandão, and Fabio G Cozman. Assessing logical reasoning capabilities of encoder-only transformer models. In *International Conference on Neural-Symbolic Learning and Reasoning*, pp. 29–46. Springer, 2024.

Zeyu Wang. CausalBench: A comprehensive benchmark for evaluating causal reasoning capabilities of large language models. In Kam-Fai Wong, Min Zhang, Ruifeng Xu, Jing Li, Zhongyu Wei, Lin Gui, Bin Liang, and Runcong Zhao (eds.), *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pp. 143–151, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.sighan-1.17/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yuekun Yao and Alexander Koller. Structural generalization is hard for sequence-to-sequence models. *arXiv preprint arXiv:2210.13050*, 2022.

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. Causal parrots: Large language models may talk causality but are not causal. *arXiv preprint arXiv:2308.13067*, 2023.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. On the paradox of learning to reason from data. *arXiv preprint arXiv:2205.11502*, 2022.

Yu Zhou, Xingyu Wu, Beicheng Huang, Jibin Wu, Liang Feng, and Kay Chen Tan. Causalbench: A comprehensive benchmark for causal learning capability of llms. *arXiv preprint arXiv:2404.06349*, 2024.

A Dataset Generation

We construct a supervised dataset of logical reasoning instances with grounded facts, Horn-style rules, and a target query. Each instance is automatically labeled as true or false ($\{0, 1\}$); and annotated with a minimal proof chain (or a failure trace) by running an algorithm based on Dijkstra's algorithm, which is available in the repository.

A.1 Problem Schema and DSL

Each example $\mathcal{E}$ comprises (i) a set of unary atoms (predicates) $\mathcal{A}$ over a single entity e (e.g., `aggressive`, `uptight`), (ii) a set of Horn rules $\mathcal{R} = \{(\Pi_i \Rightarrow c_i)\}$ with $\Pi_i \subseteq \mathcal{A}$ and $c_i \in \mathcal{A}$, and (iii) a unary query $q \in \mathcal{A}$. We serialize $\mathcal{E}$ using a compact DSL:

```
prem_1 [AND] ...[AND] prem_k [IMPLY] concl [PERIOD]
```

We explicitly encode the logical connectives from SimpleLogic and the clause separator as tokens ([AND], [IMPLY], [PERIOD]) otherwise not present in $\mathcal{A}$. Facts are written as `entityis atom [PERIOD]` and queries as `query : entity is atom [PERIOD]`. This DSL maps deterministically to the internal graph representation described next.

A.2 Labeling and Annotation

For each instance $\mathcal{E}$, we evaluate all premises in $\Pi \Rightarrow c$. If q is reached, we backtrack to extract a minimal proof sequence (the proof chain), and otherwise we emit a structured failure trace that lists available premises (with their own proofs) and missing atoms.

Depth Control and Curriculum We control reasoning depth by constraining the longest path length to the query:

$$\text{depth}(q) \;=\; \min \#\text{rule applications to derive } q.$$

We build a curriculum over depths $d \in \{1, \ldots, D_{\max}\}$, balancing the dataset across depths to prevent shallow-pattern overfitting.

Negative Sampling Without Artifacts To avoid annotation shortcuts, we introduce three strategies: premise-missing negatives, where a random premise is removed from a q-concluding rule; distractor chains, where additional rules derive atoms unrelated to q; and adversarial swaps, where one premise is replaced with a semantically close predicate (e.g., `hot` vs. `uptight`) that already appears in the facts.

Quality Checks We enforce the following invariants:

- **Well-formedness:** every rule has non-empty premises; all atoms occur in the vocabulary.
- **Sound labels:** recomputing our search algorithm reproduces $(y, \text{explanation})$ exactly.
- **Bounded depth:** $\text{depth}(q) \leq D_{\max}$.

Failed instances are discarded or repaired by re-sampling.

B Detailed Methods

B.1 Model Specifications

For the decoder-only models we used GPT-4.1 (version SHORTCO-2025-04-14), GPT-5 (2025-08-07), Claude Opus 4.1 (20250805), Qwen-2.5 (vl-7B), Qwen3-17B, and Qwen3-1.7B. For encoder-only we used BERT (Large and Base). For encoder-decoder we used BART (Large and Base). Further details are in Table 4.

Model	Parameters	Type
GPT-4.1$^{\times}$	-	Non-reasoning, ?
GPT-5$^{\times}$	-	Reasoning, ?
Claude Opus 4.1$^{\times}$	-	Reasoning, ?
Qwen-2.5	7B	Non-reasoning, decoder-only
Qwen3-17B	17B	Reasoning, decoder-only
Qwen3-1.7B	1.7B	Reasoning, decoder-only
BERT-Large	340M	Non-reasoning,encoder-only
BERT-Base	110M	Non-reasoning,encoder-only
BERT-Large	406M	Non-reasoning,encoder-decoder
BERT-Base	139M	Non-reasoning,encoder-decoder

Table 4: Models evaluated. For the models marked with $\times$, details regarding architecture, parameter size, or pretraining strategies have not been disclosed. We mark with ? models that we conjecture are decoder-only.

B.2 Call Parameters

For the LLM calls, we set the temperature to zero whenever possible. The requested completion tokens were 128 for the non-reasoning models, and 5,000 tokens for the reasoning models. All our calls were made through the Azure OpenAI API.

B.3 Finetuning

All models were finetuned for 3 epochs on a single NVIDIA RTX 6000 GPU with 48 GB of VRAM. A batch size of 8 was employed due to computational constraints, with a learning rate of 5×10^{-5} yielding the best performance.

B.4 Dataset Creation

Most of the dataset creation is covered in Appendix A. Throughout our generation, we fix random seeds at every stage and log the canonical form of each generated hypergraph, enabling exact regeneration of splits for future work.

B.5 Output format

We use the following prompt for BERT (Large and Base), BART (Base)

```
Answer the question based on the provided context.
Your Final answer should be 0 (if False) and 1 (if True)
Question: {DATA SAMPLE (FACTS, RULES and QUERY)}
Answer:
```

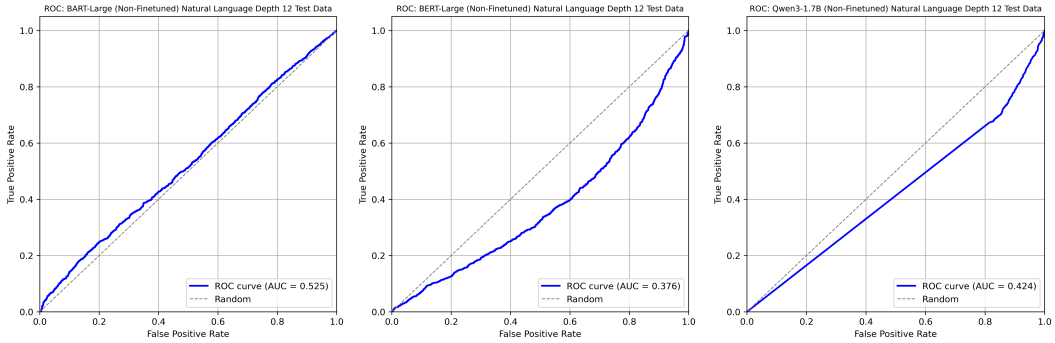
For all the other models (GPT-5, GPT-4.1, Claude Opus 4.1, Qwen 2.5, Qwen3-17B, Qwen3-1.7B BART-Large) we use the following pormpt

```
You are evaluating a subset of first-order logic. In this subset, conjunctions are given by [AND], implications by [IMPLY], and separations between clauses as [PERIOD]
You will be given Facts, and Rules. Based on these, determine the truth value of the Query. Your final answer should be 0 (if the Query is false) or 1 (if true).
Give your final answer in the following format: Label: The label from the criterion. Only use the numbers 0 or 1.
{DATA SAMPLE (FACTS, RULES and QUERY)}
```

C Non-Finetuned Models ROC Curves

C.1 Natural Languge Depth 12 Test Dataset

We plot Receiver Operating Characteristic (ROC) curves for three non-finetuned language models (BART-Large, BERT-Large, and Qwen3-1.7B) on the Natural Language Depth 12 test data. The Area Under the Curve (AUC) values for all models are near 0.5 (ranging from 0.376 to 0.525). Visually, the ROC curves closely follow the dashed "Random" line, confirming that the models' performance on this task, without finetuning, is near random across the board.

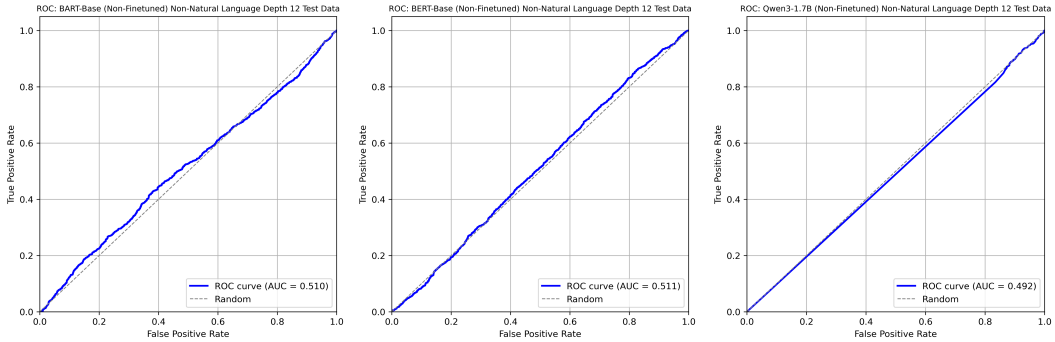


(a) BART-Large (b) BERT-Large (c) Qwen3-1.7B

Figure 8: ROC curves for non-finetuned models on Natural Language Depth 12 test data. Performance is near random across all models.

C.2 Non-Natural Languge Depth 12 Test Dataset

The ROC curves for BART-Base (AUC 0.510), BERT-Base (AUC 0.511), and Qwen3-1.7B (AUC 0.492) on a non-natural language task. All three model curves are visually nearly indistinguishable from the dashed "Random" baseline (AUC 0.5), as indicated by their respective Area Under the Curve (AUC) values being extremely close to 0.5. This demonstrates that the performance of these non-finetuned models on the Non-Natural Language Depth 12 test data is effectively random.



(a) BART-Base (b) BERT-Base (c) Qwen3-1.7B

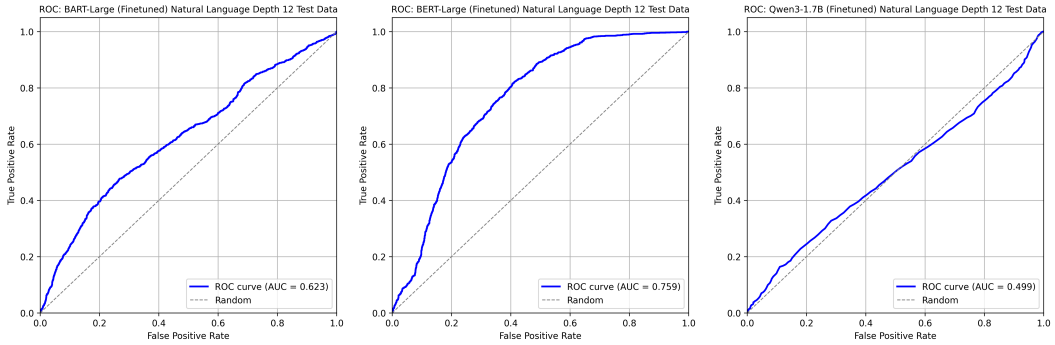
Figure 9: ROC curves for non-finetuned models on the Non-Natural Language Depth 12 test data. The curves for BART-Base, BERT-Base, and Qwen3-1.7B are nearly indistinguishable from the random baseline.

D Test Results on the Natural Language Dataset

We evaluate three finetuned models—BERT-Large, BART-large, and Qwen-3-1.7B—on the Natural Language benchmark across depths 0–11 (See Table 5). Averaged over depths, BART-large achieves the highest accuracy (0.75), followed by Qwen-3-1.7B (0.73) and BERT-Large (0.71). Per-class

analysis shows that BART-large generally balances precision and recall across both labels, while Qwen-3-1.7B excels in F1 for Label 1, and BERT-Large combines strong precision on Label 0 with high recall on Label 1. Figure 10 shows the ROC Curves for the finetuned models on the Natural Language test dataset.

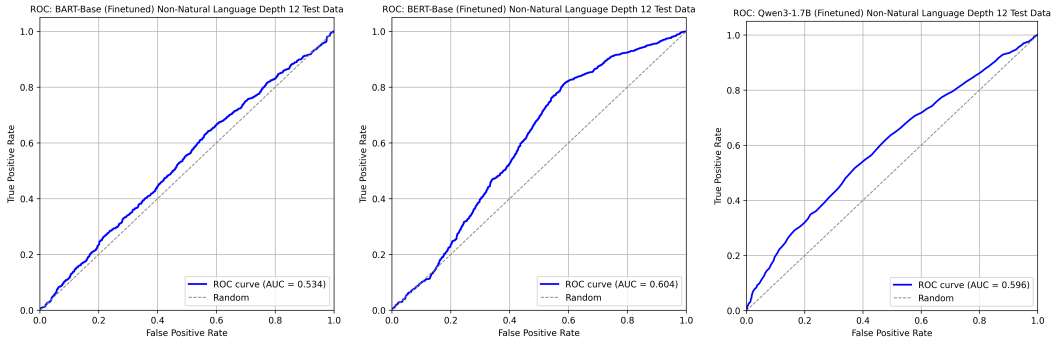
We also evaluate current reasoning and non-reasoning models (See Tables 6 to 8). GPT-5, Claude Opus 4 (Zero and five shot) have very strong performance across all depths, with GPT-5 achieving near perfect accuracy across all depths. The non-reasoning models (GPT-4 Qwen 2.5, Qwen3-17B) struggle to generalize at higher depths. GPT 4.1 (zero and five shot) and Qwen3-17B (Zero shot) have roughly similar accuracy (≈ 0.65). Qwen 2.5 (zero and five shot) performs worse than average (0.47)(Table 8).



(a) BART-Large (b) BERT-Large (c) Qwen3-1.7B

Figure 10: Finetuned Models Receiver Operating Characteristic for Natural Language Depth 12 Test Data. Finetuning markedly boosts discrimination on natural-language data.

E Test Results on the Non-Natural Language Dataset



(a) BART-Base (b) BERT-Base (c) Qwen3-1.7B

Figure 11: Finetuned Models Receiver Operating Characteristic for Non-Natural Language Depth 12 Test Data. Finetuned models illustrate strong cross-domain generalization with smaller absolute gains compared to natural language test data.

On the Non-Natural Language benchmark, overall accuracies are lower than in the Natural Language setting for the finetuned models. BERT-base performing best (0.61), followed by BART-base (0.55) and Qwen-3-1.7B (0.53). Per-class results reveal complementary strengths: BERT-base balances Label 0 precision and Label 1 recall/F1, Qwen-3-1.7B strongly favors Label 0 (high recall/F1) but suffers from poor Label 1 recall, while BART-base remains intermediate across metrics. Figure 11 shows the ROC Curves for the finetuned models on the Non-Natural Language test dataset.

Performance on the non-natural language test set varies considerably across models. GPT-5 achieves near-perfect accuracy (Table 10), clearly outperforming others. Claude Opus 4.1 attains 0.66 accuracy in the five-shot setting and 0.65 in zero-shot (Table 11), marking a substantial drop compared to its performance on the natural language test dataset. GPT-4.1, by contrast, maintains consistent accuracy

Table 5: Performance of finetuned models on Natural Language Depth 12 test data. We highlight the best (**bold**) and second-best (underline) values. All numeric values are rounded to two decimal places.

Model	Depth	Metrics						
		precision		recall		f1		accuracy
		Label 0	Label 1	Label 0	Label 1	Label 0	Label 1	
BERT-Large	0	0.00	1.00	0.00	0.98	0.00	0.99	0.98
	1	0.93	0.59	0.35	0.97	0.51	0.73	0.65
	2	0.88	0.66	0.58	0.92	0.70	0.77	0.74
	3	0.90	0.73	0.67	0.93	0.77	0.82	0.80
	4	0.84	0.75	0.66	0.89	0.74	0.81	0.78
	5	0.88	0.73	0.68	0.91	0.76	0.81	0.79
	6	0.87	0.75	0.68	0.90	0.76	0.82	0.80
	7	0.80	0.69	0.63	0.84	0.70	0.75	0.73
	8	0.73	0.55	0.44	0.80	0.55	0.65	0.61
	9	0.66	0.57	0.36	0.82	0.46	0.67	0.59
	10	0.64	0.50	0.28	0.82	0.39	0.62	0.53
	11	0.60	0.51	0.28	0.80	0.39	0.62	0.53
	Average	**0.80**	0.67	0.51	**0.89**	0.62	0.77	0.71
BART-large	0	0.00	1.00	0.00	0.98	0.00	0.99	0.98
	1	0.98	0.96	0.96	0.98	0.97	0.97	0.97
	2	0.82	0.95	0.96	0.75	0.88	0.84	0.86
	3	0.74	0.92	0.95	0.66	0.83	0.77	0.81
	4	0.65	0.91	0.94	0.54	0.76	0.68	0.73
	5	0.69	0.87	0.91	0.58	0.78	0.69	0.74
	6	0.64	0.89	0.93	0.52	0.76	0.65	0.72
	7	0.63	0.79	0.87	0.47	0.73	0.59	0.67
	8	0.66	0.71	0.83	0.48	0.73	0.57	0.67
	9	0.58	0.67	0.78	0.44	0.66	0.53	0.61
	10	0.62	0.64	0.78	0.44	0.69	0.52	0.62
	11	0.58	0.63	0.79	0.39	0.67	0.49	0.60
	Average	0.68	**0.86**	**0.88**	0.63	**0.77**	0.73	**0.75**
Qwen-3-1.7B	0	0.00	1.00	0.00	0.88	0.00	0.94	0.88
	1	1.00	0.96	0.88	0.92	0.93	0.94	0.90
	2	0.98	0.91	0.83	0.91	0.90	0.91	0.87
	3	0.95	0.90	0.82	0.86	0.88	0.88	0.84
	4	0.85	0.90	0.83	0.78	0.84	0.84	0.80
	5	0.80	0.82	0.79	0.70	0.79	0.76	0.75
	6	0.77	0.87	0.81	0.74	0.79	0.80	0.77
	7	0.71	0.77	0.78	0.57	0.74	0.66	0.68
	8	0.67	0.65	0.67	0.54	0.67	0.59	0.61
	9	0.60	0.63	0.58	0.53	0.59	0.58	0.56
	10	0.65	0.61	0.58	0.51	0.61	0.56	0.55
	11	0.63	0.62	0.69	0.49	0.66	0.55	0.59
	Average	0.77	0.83	0.75	0.72	0.76	**0.77**	0.73

Table 6: Performance of GPT models on Natural Language Depth 12 test data. We highlight the best (**bold**) and second-best (underline) values. All numeric values are rounded to two decimal places.

Model	Depth	Metrics						
		precision		recall		f1		accuracy
		Label 0	Label 1	Label 0	Label 1	Label 0	Label 1	
GPT 5 (Zero Shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	1.00	0.99	0.99	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	4	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	5	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	9	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	10	1.00	0.99	0.99	1.00	1.00	1.00	1.00
	11	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Average	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
GPT 4.1 (Five Shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	0.95	0.96	0.96	0.95	0.96	0.96	0.96
	2	0.72	0.84	0.89	0.61	0.79	0.70	0.76
	3	0.60	0.77	0.90	0.37	0.72	0.50	0.64
	4	0.50	0.67	0.86	0.26	0.63	0.37	0.54
	5	0.56	0.69	0.86	0.32	0.68	0.43	0.59
	6	0.54	0.71	0.86	0.33	0.66	0.45	0.58
	7	0.54	0.60	0.81	0.29	0.65	0.39	0.55
	8	0.57	0.54	0.82	0.26	0.67	0.35	0.56
	9	0.55	0.65	0.81	0.35	0.65	0.45	0.58
	10	0.53	0.43	0.79	0.19	0.63	0.26	0.51
	11	0.50	0.41	0.78	0.17	0.61	0.24	0.48
	Average	0.58	0.78	0.85	0.47	0.69	0.58	0.65
GPT 4.1 (Zero Shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	0.93	0.96	0.96	0.92	0.95	0.94	0.94
	2	0.70	0.83	0.89	0.58	0.79	0.68	0.74
	3	0.59	0.77	0.90	0.36	0.71	0.49	0.63
	4	0.49	0.62	0.84	0.23	0.62	0.34	0.51
	5	0.56	0.69	0.85	0.33	0.68	0.45	0.59
	6	0.56	0.78	0.90	0.32	0.69	0.46	0.60
	7	0.52	0.54	0.76	0.28	0.62	0.37	0.53
	8	0.57	0.55	0.83	0.24	0.67	0.34	0.56
	9	0.53	0.60	0.78	0.32	0.63	0.42	0.55
	10	0.51	0.39	0.76	0.18	0.61	0.25	0.49
	11	0.50	0.43	0.76	0.19	0.61	0.27	0.49
	Average	0.58	0.77	0.84	0.46	0.68	0.57	0.64

Table 7: Performance of Claude Opus 4.1 models on Natural Language Depth 12 test data. We highlight the best (**bold**) and second-best (underline) values. All numeric values are rounded to two decimal places.

Model	Depth	Metrics						
		precision		recall		f1		accuracy
		Label 0	Label 1	Label 0	Label 1	Label 0	Label 1	
Claude Opus 4.1 (Five shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	1.00	0.97	0.97	1.00	0.99	0.99	0.99
	2	1.00	0.90	0.91	1.00	0.95	0.95	0.95
	3	1.00	0.90	0.89	1.00	0.94	0.95	0.94
	4	0.99	0.91	0.89	0.99	0.94	0.95	0.95
	5	0.98	0.87	0.85	0.98	0.91	0.92	0.92
	6	0.95	0.89	0.88	0.96	0.91	0.93	0.92
	7	0.94	0.86	0.86	0.94	0.89	0.90	0.90
	8	0.93	0.89	0.91	0.91	0.92	0.90	0.91
	9	0.93	0.87	0.86	0.94	0.89	0.91	0.90
	10	0.93	0.82	0.83	0.93	0.87	0.87	0.87
	11	0.89	0.89	0.90	0.88	0.89	0.89	0.89
	Average	0.96	**0.91**	**0.89**	0.97	**0.92**	**0.93**	**0.93**
Claude Opus 4.1 (Zero shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	1.00	0.97	0.97	1.00	0.99	0.99	0.99
	2	1.00	0.88	0.87	1.00	0.93	0.93	0.93
	3	1.00	0.91	0.91	1.00	0.95	0.95	0.95
	4	0.99	0.89	0.86	0.99	0.92	0.94	0.93
	5	0.96	0.87	0.86	0.97	0.91	0.92	0.91
	6	0.98	0.84	0.81	0.98	0.88	0.91	0.90
	7	0.94	0.88	0.88	0.94	0.90	0.91	0.91
	8	0.93	0.86	0.87	0.92	0.90	0.89	0.89
	9	0.97	0.91	0.91	0.97	0.94	0.94	0.94
	10	0.88	0.82	0.83	0.86	0.85	0.84	0.85
	11	0.93	0.90	0.90	0.92	0.92	0.91	0.91
	Average	**0.96**	0.91	0.8791	**0.97**	0.92	0.93	0.93

(0.65–0.66) across both test datasets. The Qwen family shows a different trend (Table 12): Qwen 2.5 (zero and five shots) improves by 6% over its performance on the natural language test dataset while Qwen3-17B shows performs slightly worse (0.61).

Table 8: Performance of Qwen 2.5 models on Natural Language Depth 12 test data. We highlight the best (**bold**) and second-best (underline) values. All numeric values are rounded to two decimal places.

Model	Depth	Metrics						
		precision		recall		f1		accuracy
		Label 0	Label 1	Label 0	Label 1	Label 0	Label 1	
Qwen 2.5 (Five Shot)	0	0.00	1.00	0.00	0.99	0.00	0.99	0.99
	1	0.74	0.66	0.64	0.76	0.68	0.71	0.70
	2	0.46	0.44	0.31	0.60	0.37	0.51	0.45
	3	0.50	0.48	0.35	0.64	0.41	0.55	0.49
	4	0.31	0.45	0.23	0.56	0.26	0.50	0.41
	5	0.36	0.42	0.26	0.54	0.30	0.47	0.40
	6	0.35	0.43	0.30	0.49	0.32	0.46	0.40
	7	0.33	0.37	0.27	0.44	0.30	0.40	0.35
	8	0.34	0.31	0.27	0.39	0.30	0.34	0.32
	9	0.39	0.44	0.32	0.51	0.35	0.47	0.42
	10	0.37	0.34	0.30	0.41	0.33	0.37	0.35
	11	0.42	0.39	0.38	0.43	0.40	0.41	0.40
	Average	<u>0.42</u>	<u>0.50</u>	0.33	**0.60**	0.37	<u>0.55</u>	<u>0.47</u>
Qwen 2.5 (Zero Shot)	0	0.00	1.00	0.00	0.99	0.00	1.00	0.99
	1	0.72	0.66	0.64	0.73	0.67	0.69	0.68
	2	0.46	0.43	0.32	0.58	0.38	0.50	0.44
	3	0.50	0.49	0.34	0.65	0.41	0.56	0.49
	4	0.32	0.45	0.24	0.54	0.28	0.49	0.40
	5	0.37	0.42	0.28	0.52	0.32	0.46	0.40
	6	0.34	0.42	0.29	0.48	0.31	0.45	0.39
	7	0.35	0.39	0.28	0.47	0.31	0.42	0.37
	8	0.36	0.32	0.28	0.39	0.32	0.35	0.33
	9	0.38	0.42	0.32	0.48	0.35	0.45	0.40
	10	0.34	0.33	0.28	0.40	0.31	0.36	0.33
	11	0.42	0.39	0.39	0.43	0.40	0.41	0.41
	Average	0.42	0.50	<u>0.33</u>	<u>0.59</u>	<u>0.37</u>	0.54	0.47
Qwen3-17B (Zero Shot)	0	0.00	1.00	0.00	0.96	0.00	0.98	0.96
	1	0.83	0.99	0.99	0.78	0.90	0.87	0.89
	2	0.75	0.95	0.97	0.63	0.84	0.76	0.81
	3	0.63	0.94	0.97	0.40	0.76	0.56	0.69
	4	0.55	0.83	0.92	0.34	0.69	0.49	0.61
	5	0.55	0.72	0.90	0.26	0.69	0.38	0.58
	6	0.52	0.78	0.94	0.20	0.67	0.32	0.56
	7	0.54	0.65	0.89	0.22	0.67	0.32	0.56
	8	0.55	0.51	0.88	0.15	0.68	0.24	0.55
	9	0.51	0.62	0.90	0.16	0.65	0.25	0.52
	10	0.55	0.53	0.87	0.17	0.67	0.26	0.54
	11	0.50	0.42	0.82	0.14	0.62	0.21	0.49
	Average	**0.58**	**0.85**	**0.91**	0.41	**0.71**	**0.56**	**0.65**

Table 9: Performance of finetuned models on Non-Natural Language Depth 12 test data. We highlight the best (**bold**) and second-best (underline) values. All numeric values are rounded to two decimal places.

Model	Depth	Metrics						
		precision		recall		f1		accuracy
		Label 0	Label 1	Label 0	Label 1	Label 0	Label 1	
BERT-base	0	0.00	1.00	0.00	0.90	0.00	0.95	0.90
	1	0.80	0.63	0.49	0.88	0.61	0.74	0.68
	2	0.78	0.66	0.57	0.84	0.66	0.74	0.70
	3	0.77	0.65	0.56	0.83	0.65	0.73	0.70
	4	0.69	0.57	0.36	0.84	0.47	0.68	0.60
	5	0.75	0.57	0.31	0.89	0.44	0.69	0.60
	6	0.62	0.53	0.23	0.86	0.33	0.65	0.54
	7	0.56	0.51	0.19	0.85	0.29	0.64	0.52
	8	0.51	0.50	0.15	0.85	0.24	0.63	0.50
	9	0.53	0.51	0.18	0.84	0.27	0.63	0.51
	10	0.59	0.52	0.21	0.85	0.31	0.65	0.53
	11	0.57	0.51	0.19	0.85	0.29	0.64	0.52
	Average	**0.66**	0.60	0.31	**0.86**	0.42	**0.71**	**0.61**
BART-base	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	0.52	0.54	0.73	0.31	0.61	0.40	0.52
	2	0.50	0.51	0.74	0.27	0.60	0.35	0.50
	3	0.50	0.51	0.69	0.31	0.58	0.39	0.50
	4	0.50	0.50	0.68	0.31	0.58	0.39	0.50
	5	0.50	0.52	0.65	0.35	0.56	0.42	0.50
	6	0.52	0.53	0.63	0.43	0.57	0.47	0.53
	7	0.52	0.53	0.65	0.38	0.58	0.44	0.52
	8	0.51	0.52	0.63	0.39	0.57	0.45	0.51
	9	0.49	0.50	0.59	0.40	0.54	0.44	0.49
	10	0.52	0.54	0.65	0.41	0.58	0.46	0.53
	11	0.48	0.48	0.59	0.37	0.53	0.42	0.48
	Average	0.51	0.62	0.66	0.46	0.57	0.52	0.55
Qwen-3-1.7B	0	0.00	1.00	0.00	0.76	0.00	0.87	0.76
	1	0.75	0.94	0.77	0.65	0.76	0.77	0.71
	2	0.59	0.83	0.87	0.29	0.70	0.43	0.58
	3	0.51	0.68	0.83	0.11	0.63	0.19	0.47
	4	0.50	0.59	0.88	0.07	0.64	0.12	0.47
	5	0.53	0.75	0.93	0.06	0.67	0.11	0.50
	6	0.52	0.64	0.93	0.05	0.67	0.09	0.49
	7	0.49	0.46	0.87	0.04	0.63	0.07	0.46
	8	0.50	0.43	0.91	0.02	0.64	0.04	0.47
	9	0.51	0.78	0.91	0.05	0.65	0.09	0.48
	10	0.50	0.60	0.94	0.02	0.65	0.04	0.48
	11	0.50	0.33	0.94	0.01	0.65	0.03	0.48
	Average	0.52	**0.8857**	**0.89**	0.22	**0.66**	0.36	0.53

Table 10: Performance of GPT models on Non-Natural Language Depth 12 test data. We highlight the best (**bold**) and second-best (underline) values. All numeric values are rounded to two decimal places.

Model	Depth	Metrics						
		precision		recall		f1		accuracy
		Label 0	Label 1	Label 0	Label 1	Label 0	Label 1	
GPT 5 (Zero Shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	4	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	5	1.00	0.99	0.99	1.00	1.00	1.00	1.00
	6	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	7	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	8	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	9	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	10	1.00	0.98	0.98	1.00	0.99	0.99	0.99
	11	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Average	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
GPT 4.1 (Five Shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	0.95	0.90	0.90	0.95	0.92	0.93	0.93
	2	0.70	0.71	0.72	0.69	0.71	0.70	0.71
	3	0.65	0.67	0.70	0.62	0.67	0.65	0.66
	4	0.63	0.66	0.69	0.59	0.66	0.62	0.64
	5	0.62	0.61	0.60	0.63	0.61	0.62	0.62
	6	0.54	0.54	0.53	0.55	0.54	0.54	0.54
	7	0.57	0.55	0.47	0.65	0.52	0.60	0.56
	8	0.54	0.53	0.43	0.64	0.48	0.58	0.54
	9	0.58	0.55	0.47	0.66	0.52	0.60	0.57
	10	0.52	0.51	0.32	0.71	0.40	0.59	0.52
	11	0.58	0.55	0.45	0.67	0.51	0.60	0.56
	Average	0.63	0.66	<u>0.57</u>	0.72	0.60	0.69	0.65
GPT 4.1 (Zero Shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	0.95	0.91	0.91	0.95	0.93	0.93	0.93
	2	0.70	0.73	0.75	0.68	0.72	0.70	0.72
	3	0.64	0.65	0.66	0.63	0.65	0.64	0.65
	4	0.64	0.67	0.70	0.61	0.67	0.64	0.66
	5	0.64	0.63	0.62	0.65	0.63	0.64	0.64
	6	0.58	0.59	0.59	0.58	0.59	0.58	0.59
	7	0.57	0.55	0.46	0.65	0.51	0.59	0.56
	8	0.53	0.52	0.41	0.63	0.46	0.57	0.52
	9	0.57	0.54	0.41	0.69	0.48	0.61	0.55
	10	0.53	0.51	0.33	0.71	0.41	0.60	0.52
	11	0.57	0.54	0.43	0.67	0.49	0.60	0.55
	Average	<u>0.64</u>	<u>0.67</u>	0.57	<u>0.73</u>	<u>0.60</u>	<u>0.70</u>	<u>0.66</u>

Table 11: Performance of Claude Opus 4.1 model on Non-Natural Language Depth 12 test data. We highlight the best (**bold**) and second-best (underline) values. All numeric values are rounded to two decimal places.

Model	Depth	Metrics						
		precision		recall		f1		accuracy
		Label 0	Label 1	Label 0	Label 1	Label 0	Label 1	
Claude Opus 4.1 (Five shot)	0	0.00	1.00	0.00	0.47	0.00	0.64	0.47
	1	0.68	1.00	1.00	0.52	0.81	0.68	0.76
	2	0.65	0.88	0.93	0.50	0.77	0.64	0.72
	3	0.67	0.87	0.92	0.55	0.78	0.67	0.74
	4	0.64	0.83	0.90	0.49	0.75	0.62	0.70
	5	0.65	0.85	0.91	0.52	0.76	0.65	0.72
	6	0.61	0.78	0.87	0.45	0.72	0.57	0.66
	7	0.64	0.82	0.89	0.49	0.74	0.61	0.69
	8	0.59	0.77	0.88	0.40	0.71	0.53	0.64
	9	0.57	0.69	0.83	0.37	0.67	0.48	0.60
	10	0.59	0.74	0.85	0.42	0.70	0.54	0.64
	11	0.60	0.73	0.84	0.44	0.70	0.55	0.64
	Average	**0.59**	0.84	0.90	**0.47**	**0.71**	**0.60**	**0.66**
Claude Opus 4.1 (Zero shot)	0	0.00	1.00	0.00	0.45	0.00	0.62	0.45
	1	0.65	0.92	0.96	0.49	0.78	0.64	0.73
	2	0.62	0.82	0.90	0.45	0.73	0.58	0.68
	3	0.67	0.87	0.92	0.54	0.77	0.67	0.73
	4	0.63	0.90	0.95	0.45	0.76	0.60	0.70
	5	0.65	0.88	0.93	0.49	0.76	0.63	0.71
	6	0.64	0.84	0.91	0.49	0.75	0.62	0.70
	7	0.61	0.77	0.87	0.44	0.72	0.56	0.66
	8	0.61	0.82	0.91	0.41	0.73	0.55	0.66
	9	0.58	0.73	0.87	0.36	0.69	0.48	0.62
	10	0.59	0.78	0.89	0.39	0.71	0.52	0.64
	11	0.57	0.66	0.80	0.39	0.66	0.49	0.60
	Average	0.58	**0.84**	**0.90**	0.45	0.70	0.58	0.65

Table 12: Performance of Qwen 2.5 models on Non-Natural Language Depth 12 test data. We highlight the best (**bold**) and second-best (underline) values. All numeric values are rounded to two decimal places.

Model	Depth	Metrics						
		precision		recall		f1		accuracy
		Label 0	Label 1	Label 0	Label 1	Label 0	Label 1	
Qwen 2.5 (Five Shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	0.72	0.67	0.63	0.76	0.67	0.71	0.70
	2	0.58	0.54	0.41	0.70	0.48	0.61	0.56
	3	0.58	0.56	0.50	0.64	0.54	0.60	0.57
	4	0.51	0.51	0.41	0.61	0.46	0.55	0.51
	5	0.45	0.47	0.36	0.56	0.40	0.51	0.46
	6	0.43	0.45	0.34	0.55	0.38	0.50	0.45
	7	0.48	0.49	0.41	0.56	0.44	0.52	0.49
	8	0.43	0.45	0.34	0.54	0.38	0.49	0.44
	9	0.48	0.49	0.43	0.54	0.46	0.51	0.49
	10	0.37	0.38	0.35	0.40	0.36	0.39	0.38
	11	0.47	0.47	0.45	0.49	0.46	0.48	0.47
	Average	0.50	0.57	0.42	**0.64**	0.46	**0.60**	0.54
Qwen 2.5 (Zero Shot)	0	0.00	1.00	0.00	1.00	0.00	1.00	1.00
	1	0.73	0.68	0.64	0.76	0.68	0.72	0.70
	2	0.53	0.52	0.40	0.65	0.46	0.58	0.53
	3	0.55	0.54	0.48	0.61	0.51	0.57	0.55
	4	0.49	0.49	0.38	0.60	0.43	0.54	0.49
	5	0.45	0.46	0.39	0.53	0.42	0.50	0.46
	6	0.47	0.48	0.36	0.59	0.41	0.53	0.48
	7	0.46	0.46	0.42	0.50	0.44	0.48	0.46
	8	0.45	0.47	0.38	0.54	0.41	0.50	0.46
	9	0.46	0.46	0.41	0.51	0.43	0.49	0.46
	10	0.39	0.41	0.36	0.44	0.38	0.42	0.40
	11	0.41	0.43	0.38	0.46	0.40	0.44	0.42
	Average	0.49	0.56	0.42	0.63	0.45	0.59	0.53
Qwen3-17B (Zero Shot)	0	0.00	1.00	0.00	0.95	0.00	0.97	0.95
	1	0.78	0.96	0.97	0.72	0.86	0.82	0.85
	2	0.62	0.91	0.96	0.41	0.75	0.57	0.69
	3	0.56	0.77	0.92	0.27	0.69	0.40	0.60
	4	0.54	0.70	0.90	0.23	0.67	0.35	0.57
	5	0.56	0.73	0.89	0.29	0.68	0.41	0.59
	6	0.50	0.52	0.86	0.15	0.63	0.23	0.51
	7	0.52	0.61	0.87	0.20	0.65	0.30	0.54
	8	0.54	0.75	0.93	0.21	0.68	0.33	0.57
	9	0.49	0.45	0.84	0.13	0.62	0.20	0.49
	10	0.50	0.50	0.83	0.17	0.62	0.25	0.50
	11	0.47	0.35	0.80	0.11	0.59	0.17	0.46
	Average	**0.54**	**0.80**	**0.89**	0.37	**0.67**	0.50	**0.61**