# Robustness of Neurosymbolic Reasoners on First-Order Logic Problems

**Hannah Bansal**[1,*] **Kemal Kurniawan**[2] **and Lea Frermann**[2]
[1] School of Computing Technologies, RMIT University, Melbourne, Australia
[2] School of Computing and Information Systems
The University of Melbourne, Melbourne, Australia
hannah.bansal@rmit.edu.au
{kurniawan.k,lea.frermann}@unimelb.edu.au

## Abstract

Recent trends in NLP aim to improve reasoning capabilities in Large Language Models (LLMs), with key focus on generalization and robustness to variations in tasks. Counterfactual task variants introduce minimal but semantically meaningful changes to otherwise valid first-order logic (FOL) problem instances altering a single predicate or swapping roles of constants to probe whether a reasoning system can maintain logical consistency under perturbation. Previous studies showed that LLMs becomes brittle on counterfactual variations, suggesting that they often rely on spurious surface patterns to generate responses. In this work, we explore if a neurosymbolic (NS) approach that integrates an LLM and a symbolic logical solver could mitigate this problem. Experiments across LLMs of varying sizes show that NS methods are more robust but perform worse overall that purely neural methods. We then propose NSCoT that combines an NS method and Chain-of-Thought (CoT) prompting and demonstrate that while it improves performance, NSCoT still lags behind standard CoT. Our analysis opens research directions for future work. The code for this work is available at https://github.com/hannahhb/counterfactual_NS_eval

## 1 Introduction

LLMs have shown remarkable success on a range of different tasks including logical reasoning (Wei et al., 2022; DeepSeek-AI, 2024), mathematics (Lewkowycz et al., 2022), coding (Du et al., 2024), and creative tasks (Ramesh et al., 2021). These models have up to trillions of parameters and are pretrained to predict the most likely next word given the preceding words (Radford et al., 2018) on vast amounts of digitized data (Brown et al., 2020; Chowdhery et al., 2023). However, they cannot inherently perform formal rule based inference how a symbolic solver would.

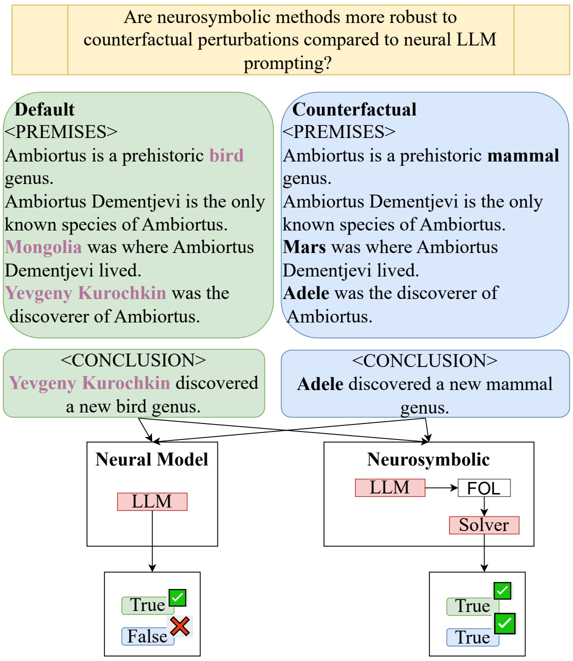
Are neurosymbolic methods more robust to counterfactual perturbations compared to neural LLM prompting?
Default
<PREMISES>
Ambiortus is a prehistoric bird genus.
Ambiortus Dementjevi is the only known species of Ambiortus.
Mongolia was where Ambiortus Dementjevi lived.
Yevgeny Kurochkin was the discoverer of Ambiortus.
Counterfactual
<PREMISES>
Ambiortus is a prehistoric mammal genus.
Ambiortus Dementjevi is the only known species of Ambiortus.
Mars was where Ambiortus Dementjevi lived.
Adele was the discoverer of Ambiortus.
<CONCLUSION>
Yevgeny Kurochkin discovered a new bird genus.
<CONCLUSION>
Adele discovered a new mammal genus.
Neural Model
LLM
Neurosymbolic
LLM
FOL
Solver
True ✓
False ✗
True ✓
True ✓

Figure 1: Illustration of our data and models. We test models in their ability to reason over default and counterfactual inputs, where key nouns were swapped (top). We compare fully neural models (LLMs) with neurosymbolic methods that combine LLMs with logical solvers. In our example the neural model fails on the counterfactual input but the neurosymbolic method makes correct predictions (bottom), suggesting higher robustness. Example taken from (Wu et al., 2024).

Wu et al. (2024) showed that these models suffer in test conditions that systematically differ from the inputs they observed during training by leveraging "counterfactual" (CF) tasks. CF tasks are carefully constructed to require the same reasoning as the original problem but with different assumptions. For example, for logical reasoning, Wu et al. (2024) demonstrated that replacing nouns and adjectives with less plausible alternatives in the input

[*] Work done while at The University of Melbourne.

data to render the statements incompatible with any observed data can lead to a performance drop of 20%. This massive drop suggests that LLMs memorize their training data rather than learning how to reason logically. Figure 1 illustrates this process.

In this work, we explore whether a neurosymbolic (NS) approach could mitigate this problem for logical reasoning tasks. In such an approach, a neural network is used to translate natural language (NL) premises and conclusions to first-order logic (FOL) statements. Then, an FOL solver is used to determine if the conclusion logically follows from the premises. Intuitively, delegating the reasoning process to an external tool—a symbolic FOL solver—should make the process less sensitive to "counterfactual" perturbations.

Our main research question is: *are NS methods more robust to counterfactual variation than purely neural approaches?* To answer this question, we employ LINC (Olausson et al., 2023), an NS method for logical reasoning that uses an LLM to translate NL into FOL, and compare against the LLM alone. Our experiments across LLMs with 7B to 32B parameters answer the question in the positive. To the best of our knowledge, we are the first to test the sensitivity of NS methods on counterfactual tasks.

However, we also find that LINC performs worse overall. We hypothesise that the model requires more explicit guidance to correctly convert NL statements to FOL. To address this, we then propose NSCoT which combines Chain-of-Thought (CoT) prompting (Wei et al., 2022) and NS approaches to improve the overall performance of NS methods. Specifically, we include example CoT reasoning in the LLM's in-context learning examples and prompt the LLM to generate its CoT reasoning when translating into FOL. We find that NSCoT substantially outperforms LINC but still lags behind a purely LLM approach with CoT.

In sum, our contributions are:

1. We provide the first rigorous exploration of the robustness of neurosymbolic methods for logical reasoning on counterfactual inputs. We show that neurosymbolic methods are more robust but achieve lower performance than purely neural methods.
2. We then propose NSCoT that integrates neurosymbolic methods and CoT. We demonstrate that NSCoT outperforms standard neurosymbolic methods, but still lags behind CoT.

## 2 Related work

### 2.1 Testing LLMs on perturbed data

Prior work has studied the sensitivity of LLMs to data perturbations. Jiang et al. (2024) demonstrated that simply replacing the primary noun in the prompt (e.g., from "Linda" to "Bob") causes the model to fail, despite the logical structure of the task remaining unchanged. The paper concluded that LLMs suffer from the *token bias problem*, a phenomenon wherein LLMs exhibited a disproportionate reliance on frequently occurring lexical items such as specific nouns or structural cues to guide their reasoning process. Wu et al. (2024) introduced perturbations for a variety of tasks including arithmetic, code execution, logic, drawing, chord fingering, and chess. Their perturbations make the tasks deviate from the default, generally assumed conditions, which they called as "counterfactual" (CF). These CFs were manually constructed and carefully controlled to fix the difficulty levels of items and keep comparisons fair. They hypothesised that LLMs simply memorise their training examples rather than actually reasoning about problems. They found that although CoT reasoning and few-shot learning can reduce the gap in performance between default and CFs, a significant performance drop remains.

The token bias problem has also been studied in the mathematical domain. The GSM-Symbolic benchmark by Mirzadeh et al. (2025) systematically tests the impact of token bias by creating parsable templates and sampling different proper names and numerical values in mathematical problems. They showed that compared to default settings, numerical perturbations lead to about a 4% performance drop. A similar effect is found with proper names, further showing that the accuracy difference compounds when combined with the numerical perturbations.

Prior work has also shown that even when tasks remain within the reasoning capacity of humans, LLMs exhibit significant failures when problem complexity increases, presumably because such problems are rare in their pretraining data. For example, Shojaee et al. (2025) used the Tower of Hanoi problem as an example where they showed that while LLMs solve the problem with a small number of disks, their reasoning fails with larger number of disks.

Similar to prior work, we apply LLMs to a perturbed dataset to test their sensitivity to data perturbations. In contrast, however, we use LLMs in a neurosymbolic approach where we delegate the reasoning step to a symbolic solver. We note that in the literature, terms such as “counterfactual” are also used to describe hypothetical conditions that are false in the real world (Li et al., 2023). We use the term “counterfactual” hereinafter to be consistent with Wu et al. (2024), i.e., perturbed data samples.

### 2.2 Neurosymbolic reasoning

Recent work has explored the integration between LLMs and symbolic systems to improving the reasoning capabilities of LLMs. Such neurosymbolic methods introduce a two-stage pipeline where natural language is first translated into FOL statements, which are then passed into a symbolic solver for resolution. This positions the LLM to perform a more abstract role of semantic parsing rather than direct reasoning. For logical reasoning, recent neurosymbolic methods include LINC (Olausson et al., 2023), Logic-LM (Pan et al., 2023), and SatLM (Ye et al., 2023).

In this work, we use LINC as a representative of neurosymbolic reasoning approaches and test its robustness to counterfactual perturbations. To the best of our knowledge, we are the first to test LINC in the context of counterfactual examples.

## 3 Methods

### 3.1 Dataset

We mainly work with the data from Wu et al. (2024), a subset of the FOLIO dataset (Han et al., 2024) which has been turned into a ‘counterfactual’ data set. In FOLIO, ‘premises’ are paired with different ‘conclusions’ which either logically follow from the premises (True), or they don’t (False), or a conclusion cannot be drawn given the information in the premises (Uncertain). The task of a model is to classify the given the natural language premises and conclusion into one of these 3 labels.

Wu et al. (2024) manually swapped core noun variables in the premises with semantically implausible nouns which however do not alter the logical conclusion. Figure 1 shows an example. Intuitively, a robust reasoner would not be confused by this, while a brittle reasoning model which relies on surface cues would. There are 81 examples in this dataset, which we will refer to as RR (Reasoning or Reciting, from Wu et al.’s paper title).

Due to limited examples and low representation of more complex reasoning problems in RR we also compare the performance of our methods on the full FOLIO validation set of 204 samples, even though this does not have a counterfactual variant. We note that this is an in-distribution task since we pass examples from the train split of the same dataset as in-context learning input to the LLM.

### 3.2 Neurosymbolic Methods

As our neurosymbolic method, we use LINC (Olausson et al., 2023) where an LLM acts as a semantic parser to translate natural language premises and conclusions into FOL statements. These statements are then passed into a logic solver called Prover9 (McCune, 2005–2010) to predict the classification label. We use 8 in-context learning examples following Olausson et al. (2023).

The solver raises an error if an input cannot be parsed (i.e., if the LLM generate FOL statements that do not comply with Prover9’s format). To handle this, we follow Olausson et al. (2023): we prompt the LLM 10 times to obtain 10 generations, pass each of them to Prover9 to get a predicted label, and perform majority voting to get the final predicted label excluding the error cases. If all generations are errors, we count the prediction as wrong in performance evaluation.

### 3.3 Neural Approaches

We compare our neurosymbolic method **LINC** against three fully neural approaches of varying complexity following Olausson et al. (2023). The input prompt for each model contains 8 in-context learning examples followed by the given premises and conclusion to be evaluated:

1. **Naïve** where we directly prompt an LLM to generate the True/False/Uncertain label. The 8-shot examples consist of premise-conclusion pair along with the label.
2. **Chain of Thought (CoT)**. Here, our 8-shot examples contain of premises, conclusion and the label, together with a human-created reasoning chain explaining why the label follows from the premises and conclusion pair. We use the reasoning chains given by Olausson et al. (2023). We lead the prompt with “Let’s think step by step”. The output consists of a generated reasoning chain and a final label.
3. **ScratchPad**. The LLM is prompted to generate both FOL statements and the True/False/Uncertain label. The scratchpad

| Model | | Naïve | ScratchPad | CoT | LINC | NSCoT |
|---|---|---|---|---|---|---|
| Mistral0.3 7B | Default | 85.19 | **87.65** | **87.65** | 60.49 | 54.32 |
| | CF | 44.44 | **65.43** | 61.73 | 56.79 | 49.38 |
| | $\Delta$ | -40.75$^*$ | -21.99$^*$ | -25.92$^*$ | **-3.70** | -4.94 |
| Qwen2.5 7B | Default | 83.95 | 86.42 | **87.65** | 66.67 | 62.96 |
| | CF | 65.43 | 76.54 | **86.42** | 66.67 | 62.96 |
| | $\Delta$ | -20.99$^*$ | -9.92$^*$ | -1.23 | **0.00** | +2.47 |
| Qwen2.5 32B | Default | **92.59** | 91.37 | 88.89 | 70.37 | 75.31 |
| | CF | 81.48 | 86.42 | **90.12** | 74.07 | 74.07 |
| | $\Delta$ | -11.11$^*$ | -4.94 | **+1.23** | +4.30 | **-1.23** |
| Gemma3 12B | Default | 90.12 | 87.65 | **92.59** | 69.14 | 66.67 |
| | CF | 72.84 | 75.31 | **90.12** | 66.67 | 62.96 |
| | $\Delta$ | -17.32$^*$ | -12.34$^*$ | **-2.47** | **-2.47** | -3.71 |
| Llama3.1 8B | Default | 86.42 | 87.65 | **92.59** | 60.49 | 60.49 |
| | CF | 48.15 | **72.84** | **72.84** | 55.56 | 59.26 |
| | $\Delta$ | -29.63$^*$ | -14.81$^*$ | -19.75$^*$ | -4.94 | **-1.23** |

Table 1: Accuracies on the default and the counterfactual data (CF) as well as their differences ($\Delta$; 0 is best) on RR. For a robust model, we expect a non-significant difference ($\Delta$) between the Default and CF condition. We mark *brittle* models for which this difference *is* significant ($p < 0.05$; McNemar's text (McNemar, 1947)) with an asterisk. The best result per model and metric is marked in bold.

baseline is included to test whether querying the LLM to generate formal FOL statements can impact its performance in comparision to CoT where we ask for a more ambiguous "reasoning".

To keep the comparison with LINC fair, for each method we prompt the LLM 10 times to get 10 generated labels and perform majority voting to obtain the final predicted label. We note that LINC is most likely to benefit from a high number of generations due to its susceptibility to Prover9 errors.

### 3.4 Models

We test instruction-tuned open-source models from four families and of varying sizes: Mistral0.3 (Jiang et al., 2023), Qwen2.5 (Bai et al., 2023), Llama3.1 (Llama Team, 2024) and Gemma3 (Gemma Team, 2024). These span a representative set of models, with the Qwen and Mistral family are chosen for their focus on reasoning tasks while Llama, and Gemma signify more general-purpose language models. We test on model sizes between 7 billion and 32 billion parameters where available.

### 3.5 Metrics

To measure robustness to counterfactual perturbations, we simply calculate the difference between the accuracy on the default data and the accuracy on the counterfactual data in the RR dataset. An ideal model would not be impacted by counterfactual perturbations as these do not impact the logical validity of the inputs. The ideal value of this accuracy difference is thus zero.

## 4 Main Results

In our main results, we compare the neurosymbolic approach LINC against our three neural baselines on the counterfactually manipulated RR dataset. These results are shown in the left part of Table 1, which shows the accuracies on the default and the counterfactual data along with their differences for these methods. We make a number of observations.

**Robustness** First, Table 1 shows that for LINC, the accuracy differences between the default and the counterfactual data are less than 5% across all models. This difference is not statistically significant, indicating robustness of LINC against counterfactual manipulation. In contrast, the fully neural methods generally show larger (over 10%) and statistically significant accuracy differences. One exception is CoT which shows good robustness albeit inconsistently, with only 3 out of 5 underlying LLMs. These findings suggest that the neurosymbolic LINC approach enhances robustness to counterfactual perturbations, compared to fully neural methods.

**Overall performance** Second, we observe that the neural methods generally outperform LINC in terms of overall performance on both the default and the counterfactual data. However, there are exceptions to this trend. For instance, LINC out-

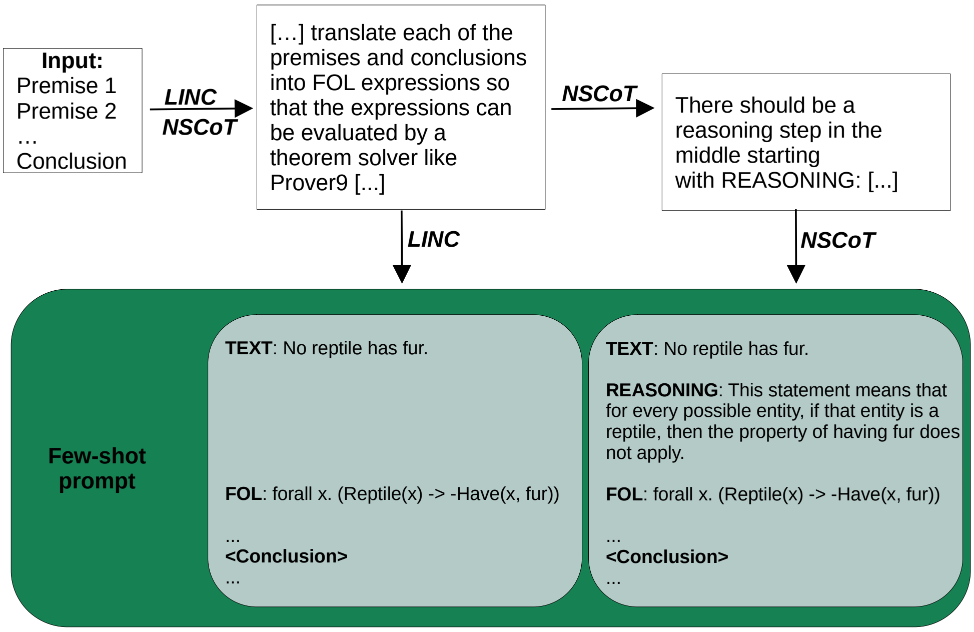
Input:
Premise 1
Premise 2
...
Conclusion
LINC
NSCoT
[...] translate each of the premises and conclusions into FOL expressions so that the expressions can be evaluated by a theorem solver like Prover9 [...]
NSCoT
There should be a reasoning step in the middle starting with REASONING: [...]
LINC
NSCoT
Few-shot prompt
TEXT: No reptile has fur.
FOL: forall x. (Reptile(x) -> -Have(x, fur))
...
<Conclusion>
...
TEXT: No reptile has fur.
REASONING: This statement means that for every possible entity, if that entity is a reptile, then the property of having fur does not apply.
FOL: forall x. (Reptile(x) -> -Have(x, fur))
...
<Conclusion>
...

Figure 2: Comparison of the few-shot prompts in LINC (left) and NSCoT (right). In contrast to LINC, for NSCoT we pass examples that include reasoning chains between the language input and FOL translations; and instruct the model to produce a reasoning chain during generation. After this step, we pass in the generated FOLs to Prover9 for both models.

performs Naïve on the counterfactual data with Mistral0.3, Qwen2.5 7B, and Llama3.1. These mixed results suggest that further analysis is necessary. In the next section, we propose a new method designed to improve the overall performance of LINC.

## 5 Enhancing LINC with CoT

We hypothesise that LINC struggles when the natural language statements are convoluted and thus requires more explicit guidance for NL conversion to FOL statements. Thus, we propose to include an intermediary "reasoning" chain in each few-shot learning example so that the LLM can acquire extra context of how an NL statement should be converted into FOL. We call this approach NSCoT (short for Neuro-symbolic Chain-of-Thought).

### 5.1 Method

We use the ChatGPT o3 reasoning model to generate reasoning chains for our examples, and insert them between the NL and FOL in the prompt. We obtain one reasoning chain for each in-context learning example for a total of 8 reasoning chains (an abbreviated example, with only 1-shot is shown in Section A). We manually verify the reasoning chains to ensure their correctness. By inserting these reasoning chains, we aim for LLMs not to be confused by examples where the inferred FOL does not directly follow from the text (cf., Figure 2). To handle Prover9 errors, we follow a similar approach to LINC where we obtain 10 generations and perform majority voting to get the final predicted label.

We note here that the reasoning chain in each in-context learning example is generated using ChatGPT o3 model. This is different from the CoT approach we included in our baseline, where we included human-generated reasoning chains following Olausson et al. (2023). In addition, we prompt NSCoT to perform reasoning in response to each premise individually. Our CoT baseline performs reasoning over all premises at once, in one contiguous block.

### 5.2 Evaluation

We evaluate NSCoT on RR (N=81) under the same conditions as our main model (Table 1, right). In addition, we also validate NSCoT, LINC, and selected baseline methods on the full FOLIO validation data set of *default* premises (N=204). This is more than double the size of RR and contains examples that were excluded by Wu et al. (2024) in creating RR. Due to its increased size and diversity, we expect this dataset to be a more representative testbed for reasoning accuracy on default premises than RR.

| Model | Naïve | CoT | LINC | NSCoT |
|---|---|---|---|---|
| M0.3 7B | 53.43 | 56.37 | 52.94 | 53.92 |
| Q2.5 7B | 59.31 | 70.59 | 58.33 | 66.67 |
| Q2.5 32B | 66.18 | 75.49 | 68.14 | 71.08 |
| L3.1 8B | 33.33 | 70.59 | 58.33 | 68.14 |
| G3 12B | 64.22 | 77.45 | 57.35 | 63.24 |

Table 2: Accuracy on the FOLIO validation set (Han et al., 2024) which has more than double the size of RR. M=Mistral, Q=Qwen, L=Llama, G=Gemma.

We report the accuracy numbers in Table 2.

### 5.3 Results

From Table 2, we note several observations. First, the performance of Naïve, CoT, and LINC substantially drops across all models relative to their performance on the default RR data in Table 1. This finding suggests that this larger dataset contains more diverse and challenging examples than RR. Second, Naïve outperforms LINC across three out of five models.[1] Moreover, the CoT baseline consistently outperforms LINC across all five models. This trend is similar to that of Table 1 where the baseline methods generally outperform LINC. Third, NSCoT consistently outperforms LINC across all models and outperforms Naïve across 4 out of 5 models.[2] This finding highlights the strength of our proposed incorporation of reasoning chains for FOL conversion. That said, NSCoT consistently lags behind the CoT baseline, which underlines the strength of purely neural approaches in terms of overall performance.

Looking at Table 1 to compare LINC and NSCoT on the smaller RR dataset, we observe that, like LINC, NSCoT shows small and non-significant accuracy differences between the default and the counterfactual data for all models. Moreover, these differences are similar in magnitude to those of LINC. This finding suggests that NSCoT is as robust as LINC to counterfactual perturbations.

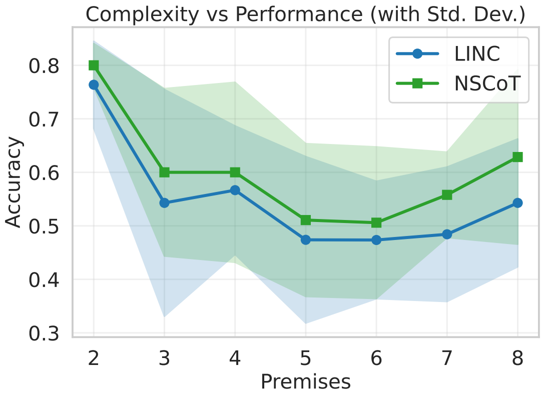

Figure 3: This plot shows the accuracy of LINC (blue) and NSCoT (green) on inputs with different numbers of premises (2 to 8) on the full FOLIO data. The presented results are averaged over all LLMs (as listed in Table 2). LINC suffers a sharper decline in performance than NSCoT.

Instances in the RR data set have an average of 4.3 premises (average length = 283 words), while the FOLIO validation instances have an average of 5.3 premises (average length = 386 words). We checked, using the FOLIO data set, the decline in performance of LINC and NSCoT over instances of increasing complexity, as approximated by the number of premises. Figure 3 shows the accuracy of each model on a subset of instances with a fixed number of premises (x-axis; varying from 2 to 8). The result is averaged over all tested LLMs in Table 2 (variance is shown as shaded areas). We observe that the gap between NSCoT and LINC increases slightly as the number of premises increases. This suggests that NSCoT effectively leverages the intermediate reasoning step to deal with more complex sets of premises.

From Table 1, we observe that NSCoT outperforms (or is on par with) LINC on the default data of the smaller RR dataset only with Qwen2.5 32B and Llama 3.1. This is in contrast with the results we observe on the full FOLIO validation set of default premises in Table 2 where NSCoT consistently outperforms LINC. It also conflicts with our results in Figure 3 where NSCoT slightly outperforms LINC for all levels of complexity. We suspect that this discrepancy is due to the sample selection heuristics that Wu et al. (2024) used in creating RR and leave this investigation to future work. We contend that the results in Table 2 and fig. 3 provide stronger evidence for the advantages of NSCoT due to the larger number and more complex examples in the full FOLIO validation set.

To sum up, our experiments showed that: (1) Neurosymbolic methods outperform purely neural methods in terms of robustness; (2) Pure neural methods, particularly with CoT reasoning, are stronger in terms of accuracy; and (3) the accuracy of neurosymbolic methods can be improved with additional CoT reasoning steps while maintaining

[1] All models except for Qwen2.5 32B and Gemma3 12B.

[2] Negative case: Gemma3 12B.

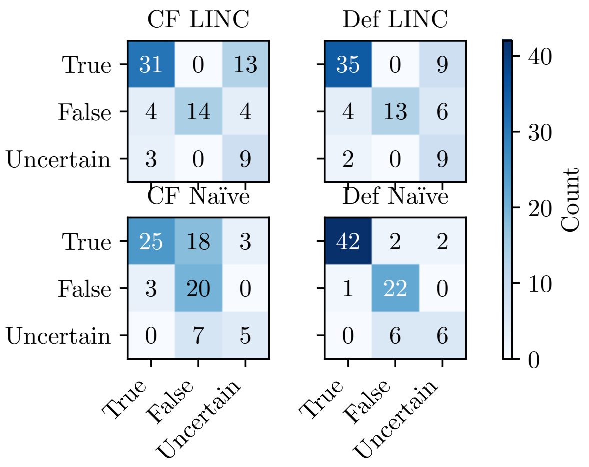

Figure 4: Confusion matrices for the predicted vs gold labels on the CF (left) vs Default (right) versions of RR for LINC (top) and Naïve (bottom). Predicted and ground truth labels are on the x- and y-axis respectively. The underlying LLM is Qwen2.5 7B.

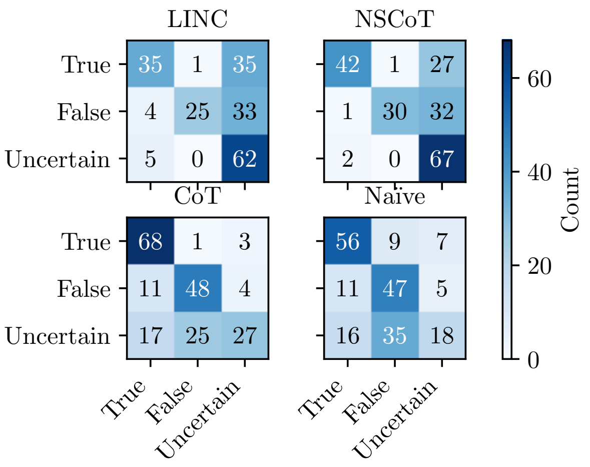

Figure 5: Confusion matrices comparing LINC, NSCoT, CoT and Naïve for the FOLIO validation set. Predicted and ground truth labels are on the x- and y-axis respectively. The underlying LLM is Qwen2.5 7B.

strong robustness, albeit not to the level of neural CoT methods. The remainder of this paper presents and error analysis and an in-depth discussion of our results, with an eye to future research directions.

## 6 Discussion

### 6.1 Class Distributions for Default vs Counterfactual Predictions

Recall that the ground truth label distribution in the default and counterfactual (CF) versions of RR are identical, as the perturbations had no bearing on the logical conclusion of the premises. We thus compare the confusion matrices between predicted and ground truth class distributions of several models. We start by inspecting the predicted class distribution shift for LINC and the Naïve method on RR (Figure 4), and subsequently compare the label distributions of all tested methods on the larger FOLIO validation data set (Figure 5). All results are based on Qwen2.5 7B.

Figure 4 shows that while LINC maintains a nearly identical confusion matrix profile across both settings, the Naïve method shows a noticeable shift in the distribution of labels. Particularly, around 20% of samples flip from the True class to the False class, leading to a substantial reduction in accuracy. This behavior reflects the Naïve model's tendency to rely on surface-level token associations, which collapse when predicates or constants are perturbed. In contrast, LINC's symbolic pipeline ensures that perturbations are more likely to lead to either consistent or *Uncertain* predictions.

Figure 5 compares the confusion matrices for Naive, CoT, NSCoT and LINC based on the Qwen7B instruction fine-tuned model and on the larger FOLIO validation data. We can observe that both the CoT and Naïve models show a higher False Negative (FN) rate for *Uncertain* class instances i.e. Naïve and CoT methods both tend to under-predict *Uncertain*. This suggests that neural methods overfit to surface regularities, confidently outputting categorical answers even when evidence is ambiguous.

In contrast, the neurosymbolic methods (LINC and NSCoT) produce more *Uncertain* predictions. However, this comes with a trade-off: some of these are false positives because the LLM produces predicates with overlapping meaning, and Prover9 as a symbolic solver cannot detect this, thus predicting Uncertain for otherwise resolvable cases. Overlapping meaning refers to instances containing distinct predicates with shared denotation that form a disconnect in the logical flow. For example, if the LLM output contains the predicates "Dog" and "CuteDog" then Prover9 will not be able to be resolve them (i.e. that "CuteDog" implies "Dog"), causing the logical reasoning process to fail. A more elaborate example is included in Section A.2. The quantity of these errors is captured by the Uncertain false positives in the respective confusion matrices of LINC and NSCoT (Figure 5). This error class is an instance of the more general issue of implicit information in NL statements (e.g., that "CuteDog" implies "Dog"), which has also been noted in prior work Olausson et al. (2023).

### 6.2 Error analysis for NSCoT and LINC

To better understand the FOL conversion errors of both LINC and NSCoT, we manually classified the observed errors on all the examples from the FOLIO validation set where the methods generated FOL statements that did not comply with the Prover9 syntax. Here, "error" is generations in which Prover9 was not able to resolve the given FOL statements due to the specified error classes. These were 341 cases for LINC and 366 cases for NSCoT, out of a total of 2040 queries (204 examples × 10 generations). We found two common classes of erros. The first are *arity mismatches* where predicates are used with inconsistent numbers of arguments across premises in the same instance (e.g. Likes(x,y) vs. Likes(x)). The second common error class pertains to *unexpected tokens*. This typically arises based on malformed or incomplete FOL strings (e.g., missing parentheses, unbalanced connectives), which cause Prover9 to throw parsing errors.

Figure 6 shows the relative prevalence of both error classes for LINC and NSCoT with Qwen2.5 7B. NSCoT produces more arity mismatch errors compared to LINC. However, LINC produces more unexpected token errors than NSCoT.

We conducted preliminary tests with a verification module to refine generations which did not execute due to Prover9 errors on both methods. In this setup, the Prover9 error messages and a few examples of common syntax corrections were put into a new prompt, and the model was re-queried in a loop until the FOL expressions executed successfully or a maximum of 3 retries was reached. However we had similar findings to Pan et al. (2023) in which they showed that the execution rate of the symbolic prover increases using a refiner but at the same time the accuracy decreases due to more semantic errors.

### 6.3 Faithfulness of CoT

While CoT outperforms NS methods in terms of accuracy and achieves comparable robustness on three out of six models (Figure 4), the underlying reasoning can be “unfaithful”, introducing hallucinated steps or logical inconsistencies. CoT’s high false negative rate for the Uncertain label fits with a concern about unfaithfulness: the models CoT text output is not actually representing internal logical reasoning, but rather reproducing the most frequently observed labels during training which was either a True or False label.

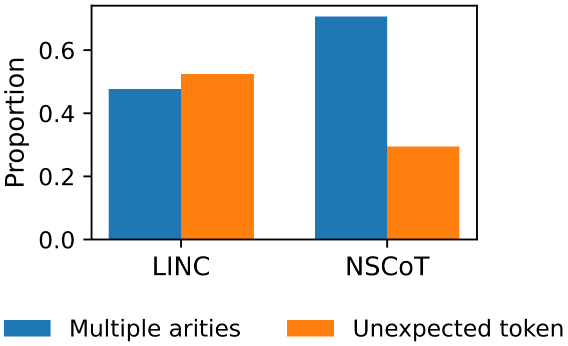

Figure 6: Proportion of the two most common FOL conversion errors of both LINC and NSCoT: arity mismatch and unexpected token. The underlying LLM is Qwen2.5 7B.

The limits of CoT prompting have come under scrutiny by Paul et al. (2024) who showed that CoT rationales often reflect post-hoc justifications rather than the true decision process of the model. They modified the generated CoT reasoning such as making it incomplete, masking some tokens, or introducing some mistakes. They then re-queried the model with the modified reasoning and found that the model still gives the correct output, suggesting that the output does not actually depend on the printed reasoning chains, and that the reasoning chains are produced post-hoc after the final class prediction has already been computed.

An important and intriguing path for future work could be to leverage this method to test whether CoT faithfulness changes between CF and default samples. If the method genuinely reasons over the given input, we expect to not observe a drop in faithfulness since the complexity of the problems stays fixed due to the design of the CF examples.

## 7 Conclusion

We have presented the first rigorous comparison of strong neural methods with the neurosymbolic method LINC — which combines LLM-based natural language to FOL parsing with an FOL solver — on the task of logical reasoning. We showed that while LINC shows stronger robustness results, it falls short of the neural methods in terms of performance. We then extended LINC with CoT reasoning steps showing that reasoning accuracy is enhanced while maintaining robustness. However, the fully neural methods still achieve the strongest results based on performance.

This paper addresses the timely and relevant question of neurosymbolic approaches in AI which are desirable due to a promise of a decreased carbon footprint due to the outsourcing of part of the reasoning to efficient external modules (such as logical reasoners). Furthermore, neurosymbolic approaches promise a tighter control on interpretability and faithfulness of the results. Our results present a step in this direction, by carefully evaluating NS methods on logical reasoning and proposing steps for future research.

We note that further optimization is most critical for improving the **accuracy** of the neurosymbolic methods, rather than robustness, as the main source of performance degradation comes from inconsistent NL–FOL translations. Optimizing this stage would therefore allow the symbolic reasoning component compute on correct logical forms, yielding more faithful and accurate deductions.

## Limitations

Our work is limited to one type of reasoning namely first-order logic and should in the future be expanded to tasks such as math word problems (Huang et al., 2025) (using SymPy as a symbolic solver), coding, and planning. There is a need for systems which are more robust to variation and faithful under hard problems. Other limitations of this study include the small test data size and exclusion of larger-sized models due to computational resources.

## Acknowledgments

This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

## References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and et al. 2023. Qwen technical report.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sashank Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2023. Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1).

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Xueying Du, Mingwei Liu, Kaixin Wang, Hanlin Wang, Junwei Liu, Yixuan Chen, Jiayi Feng, Chaofeng Sha, Xin Peng, and Yiling Lou. 2024. Evaluating large language models in class-level code generation. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA. Association for Computing Machinery.

Gemma Team. 2024. Gemma: Open models based on gemini research and technology.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. 2025. MATH-Perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. *arXiv preprint arXiv:2502.06453*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint*. ArXiv:2310.06825 [cs].

Bowen Jiang, Yangxinyu Xie, Zhuoqun Hao, Xiaomeng Wang, Tanwi Mallick, Weijie J Su, Camillo Jose Taylor, and Dan Roth. 2024. A peek into token bias: Large language models are not yet genuine reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4722–4756, Miami, Florida, USA. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem

Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*.

Jiaxuan Li, Lang Yu, and Allyson Ettinger. 2023. Counterfactual reasoning: Testing language models' understanding of hypothetical scenarios. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 804–815, Toronto, Canada. Association for Computational Linguistics.

AI @ Meta1 Llama Team. 2024. The llama 3 herd of models.

W. McCune. 2005–2010. Prover9 and mace4. `http://www.cs.unm.edu/~mccune/prover9/`.

Q. McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

L. Pan, Alon Albalak, X. Wang, and W. Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15012–15032, Miami, Florida, USA. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2024. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1819–1862, Mexico City, Mexico. Association for Computational Linguistics.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. SatLM: Satisfiability-aided language models using declarative prompting. In *Advances in Neural Information Processing Systems*, volume 36, pages 45548–45580.

# A Appendix

## A.1 1-shot example for NSCoT

**Task.** The following is a first-order logic (FOL) problem. The problem is to determine whether the conclusion follows from the premises. The premises are given in the form of a set of first-order logic sentences. The conclusion is given in the form of a single first-order logic sentence. The task is to translate each of the premises and conclusions into FOL expressions so that the expressions can be evaluated by a theorem solver like Prover9 to determine whether the conclusion follows from the premises. There should be a reasoning step in the middle starting with REASONING:.. The output should follow format TEXT: REASONING: FOL: TEXT: REASONING: FOL:...Expressions should adhere to the format of the Python NLTK package logic module.Remember to never the FOL predicates too long or overlapping meanings with other predicates. Also you must not have empty predicates like Conductor(). Symbols like <, >, = are not allowed.

**Premises.**

- All dispensable things are environment-friendly.
- All woodware is dispensable.
- All paper is woodware.
- No good things are bad.
- All environment-friendly things are good.
- A worksheet is either paper or is environment-friendly.

**Conclusion.** A worksheet is not dispensable.

**Evaluate.**
**TEXT:** All dispensable things are environment-friendly.
**REASONING:** We universally quantify over every entity, asserting that being dispensable implies being environment-friendly.
**FOL:** all x. (Dispensable(x) -> EnvironmentFriendly(x))

**TEXT:** All woodware is dispensable.
**REASONING:** Every instance of woodware falls under the dispensable category.
**FOL:** all x. (Woodware(x) -> Dispensable(x))

**TEXT:** All paper is woodware.
**REASONING:** Paper is a subset of woodware, so paper implies woodware.
**FOL:** all x. (Paper(x) -> Woodware(x))

**TEXT:** No good things are bad.
**REASONING:** For every x, if x is good then x cannot be bad.
**FOL:** all x. (Good(x) -> -Bad(x))

**TEXT:** All environment-friendly things are good.
**REASONING:** Environment-friendly implies good.
**FOL:** all x. (EnvironmentFriendly(x) -> Good(x))

**TEXT:** A worksheet is either paper or is environment-friendly.
**REASONING:** Worksheet must be in exactly one of those two categories.
**FOL:** ((Paper(Worksheet) &
-EnvironmentFriendly(Worksheet)) |
(-Paper(Worksheet) &
EnvironmentFriendly(Worksheet)))

**TEXT:** A worksheet is not dispensable.
**REASONING:** We assert the negation of dispensability for the specific constant Worksheet.
**FOL:** -Dispensable(Worksheet)

## A.2 Comparison between consistent and ambiguous FOL generations

| # | **Generation 1 (Ambiguous)** | **Generation 2 (Consistent)** |
|---|---|---|
| 1 | $\forall x, y.\ (\mathrm{WildTurkey}(x) \land \mathrm{WildTurkey}(y) \land (x = \mathrm{EasternWildTurkey} \lor x = \mathrm{OsceolaWildTurkey} \lor x = \mathrm{GouldsWildTurkey} \lor x = \mathrm{MerriamsWildTurkey} \lor x = \mathrm{RioGrandeWildTurkey} \lor x = \mathrm{OcellatedWildTurkey}) \land x \neq y \to \bot)$ | $\forall x.\ (\mathrm{WildTurkey}(x) \to (\mathrm{EasternWildTurkey}(x) \lor \mathrm{OsceolaWildTurkey}(x) \lor \mathrm{GouldsWildTurkey}(x) \lor \mathrm{MerriamsWildTurkey}(x) \lor \mathrm{RioGrandeWildTurkey}(x) \lor \mathrm{OcellatedWildTurkey}(x)))$ |
| 2 | $\neg\mathrm{WildTurkeyType}(\mathrm{Tom}, \mathrm{EasternWildTurkey})$ | $\neg\mathrm{EasternWildTurkey}(\mathrm{Tom})$ |
| 3 | $\neg\mathrm{WildTurkeyType}(\mathrm{Tom}, \mathrm{OsceolaWildTurkey})$ | $\neg\mathrm{OsceolaWildTurkey}(\mathrm{Tom})$ |
| 4 | $\neg\mathrm{WildTurkeyType}(\mathrm{Tom}, \mathrm{GouldsWildTurkey}) \land \neg\mathrm{WildTurkeyType}(\mathrm{Tom}, \mathrm{MerriamsWildTurkey}) \land \neg\mathrm{WildTurkeyType}(\mathrm{Tom}, \mathrm{RioGrandeWildTurkey})$ | $\neg\mathrm{GouldsWildTurkey}(\mathrm{Tom}) \land \neg\mathrm{MerriamsWildTurkey}(\mathrm{Tom}) \land \neg\mathrm{RioGrandeWildTurkey}(\mathrm{Tom})$ |
| 5 | $\mathrm{WildTurkey}(\mathrm{Tom})$ | $\mathrm{WildTurkey}(\mathrm{Tom})$ |
| 6 | $\mathrm{WildTurkeyType}(\mathrm{Tom}, \mathrm{OcellatedWildTurkey})$ | $\mathrm{OcellatedWildTurkey}(\mathrm{Tom})$ |
|  | *Problem: The predicate WildTurkeyType is never linked to WildTurkey, creating ambiguity between types and individuals.* | *Correct: All predicates share the same unary form, so Tom's type is inferred successfully.* |

Table 3: The correct inference is **True**, but ambiguous predicate names in "Generation 1" lead to **Uncertain**. We compare ambiguous (left) and consistent (right) FOL statements with predicate numbering. Red indicates inconsistent predicate forms causing uncertainty; Teal indicates consistent unary naming that yields a correct inference.