ROBUSTLR: A Diagnostic Benchmark for Evaluating Logical Robustness of Deductive Reasoners

Anonymous EMNLP submission

Abstract

Transformers have been shown to be able to perform deductive reasoning on inputs containing rules and statements written in English natural language. However, it is unclear if these models indeed follow rigours logical reasoning to arrive at the prediction, or rely on spurious correlation patterns in making decision. A strong deductive reasoning model should consistently understand the semantics of different logical operators. To this end, we present ROBUSTLR, a diagnostic benchmark that evaluates the robustness of language models to minimal logical edits in the inputs and different logical equivalence conditions. In our experiments with RoBERTa and T5, we show that the models trained on deductive reasoning datasets do not perform consistently on the ROBUSTLR test set, thus showing that the models are not robust to our proposed logical perturbations. Further, we observe that the models find it especially hard to learn logical negation operator. Our results demonstrate the shortcomings of current language models in logical reasoning, and call for the development of better inductive biases to teach the logical semantics to language models. All the datasets and code base will be made publicly available.

1 Introduction

Building systems that can automatically reason over a given context to generate valid logical inferences is a long pursued goal within the field of AI (McCarthy, 1959; Rocktäschel and Riedel, 2017; Manhaeve et al., 2019). Recently, Clark et al. (2020) have shown that language models (Liu et al., 2019; Raffel et al., 2020) are able to emulate deductive reasoning on a logical rulebase (*theory*) containing rules and declarative statements written in natural language. While this is impressive, it is unclear if these models are able to perform logical reasoning robustly by understanding the semantics of the logical operators and the different logical conditions involving such operators.

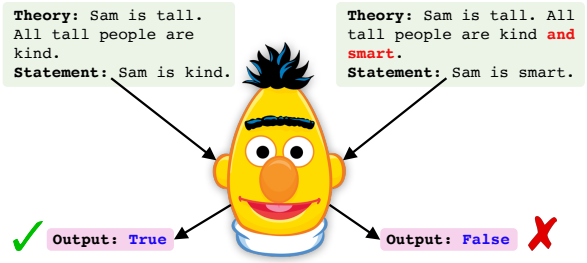


Figure 1: **Overview of ROBUSTLR.** We expect a strong deductive reasoning model should be robust to logical variations in the input. Here, the model fails to understand the logical conjunction in second example and predicts the wrong entailment of the statement.

Logical reasoning is an important skill required in various NLP tasks such as NLI (Dagan et al., 2006), Question Answering (Yang et al., 2018a), Multi-turn Dialogue Reasoning (Cui et al., 2020), etc. Models used to solve such tasks may use spurious patterns to reach to the predictions, rather than following the intended logical reasoning process. Additionally, these models might only understand certain ways of expressing the inference knowledge (e.g., rules) and not possess systematic generalization (Gontier et al., 2020). Hence, it is important to ensure that language models can consistently use the logical operators when described in natural language. Prior works Gururangan et al. (2018); Chen and Durrett (2019); McCoy et al. (2019) have found that models solving different reasoning tasks tend to exploit spurious correlations between the context/question and the label. But logical reasoning needs special considerations as there are very well-defined relationships on how different logical operators modify any given context. Hence, it is important to understand if models use these logical relationships consistently to solve a task. To the best of our knowledge, a study evaluating a language model's logical consistency on different logical operations is currently missing.

A key desirable property of a strong deductive reasoning model is *logical robustness*. This is the ability to make consistent predictions on inputs that

have some logical modifications. In Figure 1, we show how the lack of logical robustness can lead to wrong inferences in a model. Thus, to test this, we develop ROBUSTLR, a diagnostic benchmark for evaluating logical robustness across two main aspects. First, we aim to evaluate how robust these models are when tested on the three logical operators: conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$). Inspired by the idea of contrast sets (Gardner et al., 2020), we design the Logical Contrast set, where theories are minimally modified so that we can test the model's robustness across logical operators. Examples of this are shown in Figure 3(b) and 3(c). Next, we study the model's ability of reasoning consistently across *logical paraphrases*. A logical paraphrase uses equivalence conditions in logic to replace a rule with another equivalent form, essentially rewriting the surface form of the rule. This poses a different challenge than the more common language paraphrases such as synonym changes, voice modifications, style changes, etc., because the model needs to understand that the underlying logical structure of the two paraphrased sentences mean the same thing. An example of the equivalence perturbation is shown in Figure 3(d). Based on this, we design the Logical Equivalence set containing three logical equivalences.

To test the model performance on ROBUSTLR, we first fine-tune them on different training datasets and then evaluate on the test sets. We study three language models: RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2020), and GPT-3 (Brown et al., 2020). Overall, we find that language models (LMs) fine-tuned on different deductive reasoning datasets are not sufficiently robust to the Logical Contrast and Logical Equivalence sets. Specifically, we find that models are more inconsistent with logical negations in sentences. We also find that using larger models such as T5-11B improves the performance to an extent, but they still perform worse compared to human performance on ROBUSTLR. We show that it is partly due to spurious correlations in the data and the inherent difficulty of the task. Thus, using ROBUSTLR, we demonstrate some key limitations of the language models trained for deductive reasoning. We hope that this research will encourage as a test bed to evaluate robustness of deductive reasoning models.

2 Deductive Reasoning

In deductive reasoning, we predict whether a given theory T supports a statement s or not. We define a theory T as a set of facts $F = \{f_1, f_2, \ldots, f_n\}$ and rules $R = \{r_1, r_2, \ldots, r_m\}$ expressed in natural language (See Figure 3 for an example). For a given theory, a statement can be either provably supported, provably unsupported (i.e., the negation of the statement is provable), or not provable at all. This is a 3-class classification problem, with the labels *True*, *False*, and *Unknown*, respectively. In this work, we focus on this task, where we expect the model to correctly predict the entailment of a statement for a given theory. In Figure 3, the statement is entailed by the theory, leading to the label True. It can be proved by simply using fact f_1 and rule r_1 to derive the statement. Formally, we define the proof set of a statement s, denoted by $G(T, s)$, as the set of rules and facts that are required to obtain the statement s from the theory.

3 Evaluating LMs for Logical Robustness

3.1 Logical Robustness

We consider a deductive reasoner (language model) to be *logically robust* if the model behavior is consistent across various logical perturbations, as illustrated in Figure 1. Specifically, we evaluate logical robustness on two types of perturbations.

Logical contrastive edits Here, we test the model's ability to correctly capture the semantics of different logical operators, when presented in minimally edited contrast inputs. A contrast set (Gardner et al., 2020) is one where the input is changed minimally, but meaningfully, such that there is (typically) some change in label. These probes test the LM's robustness to conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$).

Logical paraphrases Here, we evaluate whether the model performs consistently when shown the same input with different *logical paraphrases*. A theory can be logically paraphrased by modifying the rules using standard logical equivalence conditions [1]. These probes evaluate the model's consistency in solving logically equivalent theories, when logical conditions are used to rewrite the input.

A strong deductive reasoning model should be robust to both the minimally edited contrast inputs and logical paraphrases. Overall, these evaluation sets probe a deductive reasoning model to check whether it indeed learns the semantics of the logical operators and their underlying working principles.

[1] https://en.wikipedia.org/wiki/Logical_equivalence

```
f1: Charlie is tall.
f2: Erin is not the brother of Gary.
r1: If Charlie is tall or smart, then
Gary is kind.
r2: Charlie is round if Gary is kind.
statement: Charlie is not round.
label: False
```

(a) Natural Language Form

```
f1: tall(Charlie)
f2: ¬brother(Erin, Gary)
r1: tall(Charlie) ∨ smart(Charlie) → kind(Gary)
r2: kind(Gary) → round(Charlie)
statement: ¬round(Charlie)
label: False
```

(b) Logical Form

Figure 2: **Logical Form of a Theory.** (a) A theory in natural language. (b). The corresponding logical form of the theory. Refer to Section 3.2 for more details.

Modified Rule	Facts	Statement	Label	Group
$p \implies q$	$\{p\}$	q	*True*	BASE
$p \wedge t \implies q$	$\{p\}$	q	*Unknown*	CONJ
$p \wedge t \implies q$	$\{p, t\}$	q	*True*	CONJ
$p \wedge t \implies q$	$\{p, \neg t\}$	q	*Unknown*	CONJ+NEG
$p \wedge t \implies \neg q$	$\{p\}$	q	*Unknown*	CONJ+NEG
$p \wedge t \implies \neg q$	$\{p, t\}$	q	*False*	CONJ+NEG
$p \wedge t \implies \neg q$	$\{p, \neg t\}$	q	*Unknown*	CONJ+NEG

Table 1: **Conjunction Contrast Perturbations**. The minimal edits done to a base theory (first row) for testing the conjunction and negation reasoning abilities. The group reflects the overall change in theory w.r.t. the base theory.

3.2 Notations

In this work, we consider two predicate forms: unary and binary. A unary predicate contains one argument and is denoted by $X(a)$. Similarly, a binary predicate is represented as $X(a, b)$. Here, X is the predicate relation and a, b are the variables. An *atomic* predicate is defined as either a predicate or the negation of the predicate (denoted as $\neg X(a)$). A *complex* predicate can contain multiple predicates (or their negated forms) combined using logical operators conjunction and disjunction.

Internally, we maintain a symbolic representation of these facts and rules, enabling us to later create the different evaluation sets of ROBUSTLR. A fact is symbolically represented by a predicate. In this work, we consider all facts as atomic predicates. A rule is symbolically represented by a logical connection between predicates, separated by the "*implies that*" logical symbol ($\implies$). Thus, a rule can be defined as $p \implies q$, where the LHS p and RHS q are atomic or complex predicates. If both p and q consist of atomic predicates, then the rule is called a *simple* rule. A *compound* rule is one where p and/or q contain some complex predicates connected by the conjunction or disjunction operator. An example of a natural language theory and its corresponding logical form is shown in Figure 2. Here, facts f_1 and f_2 are unary atomic and binary predicates, respectively. Rule r_1 is a compound rule, with the LHS p of the rule being a complex predicate. Rule r_2 is a simple rule.

3.3 Logical Contrast Sets

In this evaluation set, we probe the ability of the model to robustly understand the three different logical operators ($\wedge, \vee, \neg$). For this, we develop different contrast sets (Gardner et al., 2020) with minimal editing of the theory, probing specific reasoning abilities of different operators. The key intuition is to evaluate if the model is able to understand the minor changes in the theory brought by the addition of logical operators, and predict the change in label accordingly.

For a given theory T and statement s, we first select a rule to be modified such that it is part of the proof set $G(T, s)$. This ensures that our perturbation would influence the model's reasoning process while predicting entailment of the statement s. Next, we add an unseen predicate t to the rule LHS p of one of the rules using conjunction ($\wedge$) or disjunction ($\vee$). In some further variants of perturbations, we include the predicate t (or the negated $\neg t$) as a fact in the theory, leading to different labels. Lastly, we also negate the rule RHS q to introduce the logical negation ($\neg$) perturbations. Based on the logical operator in the perturbation, we broadly divide the Logical Contrast set into three types: Conjunction Contrast Set (C-CS), Disjunction Contrast Set (D-CS), and Negation Contrast Set (N-CS). Examples of these perturbations are shown in Figure 3. The perturbations for C-CS are listed in Table 1. Please refer to Appendix F for the other perturbations. We categorize these perturbations into groups based on the logical operators involved in the perturbation with respect to the base theory. For e.g., the three groups for C-CS are BASE, CONJ, CONJ+NEG, as shown in Table 1. If a model performs accurately on the Logical Contrast set, we expect that the model understands the semantics of the logical operators robustly.

Evaluation Protocol To evaluate the logical robustness to contrast perturbations, we first fine-tune the language model on a deductive reasoning dataset containing different combination of logical operators. Then, we report the model performance on these evaluation datasets as the weighted-F1 score from the Scikit-learn (Pedregosa et al., 2011). The weighted-F1 score modifies the macro-F1 to take any label imbalance into account. For instance,

f1: Charlie is tall.
r1: Erin is kind, if Charlie is tall.
statement: Erin is kind.
Label: *True*

(a) Original Theory

f1: Charlie is tall.
r1: Erin is kind, if Charlie is tall ***and round***.
statement: Erin is kind.
Label: *Unknown*

(b) Conjunction Perturbation

f1: Charlie is tall.
r1: Erin is kind, if Charlie is tall ***or round***.
statement: Erin is kind.
Label: *True*

(c) Disjunction Perturbation

f1: Charlie is tall.
r1: ***If Erin is not kind, then Charlie is not tall.***
statement: Erin is kind.
Label: *True*

(d) Contraposition Perturbation

Figure 3: **Examples of perturbations in ROBUSTLR.** (a) An original theory contains facts, rules, a statement, and the entailment label. The Logical Contrast set perturbations using conjunction and disjunction are shown in bold in (b) and (c), respectively. In (d), we show one of the Logical Equivalence perturbations where the rule is paraphrased using logical contraposition. Please refer to Sections 3.3 and 3.4 for more details.

we have label imbalance by design of the perturbations in the Logical Contrast sets shown in Tables 1, 9, and 10. We use the model's prediction for the base theory and all its perturbations to compute the F1-score at a theory level, and then average this score across all theories in the evaluation set.

3.4 Logical Equivalence Sets

The Logical Equivalence set contain theories where the underlying symbolic representation of a rule is replaced by another representation that is logically equivalent. The logical equivalent form of a rule can be derived from standard logical equivalence conditions, as defined below:

- **Contrapositive**: $p \implies q \equiv \neg q \implies \neg p$
- **Distributive 1**: $(p \implies q) \wedge (p \implies r) \equiv p \implies (q \wedge r)$
- **Distributive 2**: $(p \implies q) \wedge (r \implies q) \equiv (p \vee r) \implies q$

Here p, q, r can be both atomic predicates or complex predicates. Based on the above conditions, the Logical Equivalence set is divided into three types: Contrapositive Equivalence Set (C-ES), Distributive 1 Equivalence Set (D1-ES), and Distributive 2 Equivalence Set (D2-ES). For the (C-ES) set, every rule r_i in the theory T is replaced by the logically equivalent form to create a new logically equivalent theory T'. An example of this perturbation is shown in Figure 3 (d). Similarly, for the D1-ES and D2-ES sets, a pair of rules in T are merged according to the equivalence to create a new theory T'.

In both instances, the theory T' still has the same label for a given statement, as the logical steps required to solve the task remains the same. These modifications are more challenging than traditional surface-level paraphrases of the natural language text, as it requires the model to understand the equivalence of different symbolic representations.

Evaluation Protocol Similar to the Logical Contrast set evaluation, we finetune a language model on a deductive reasoning dataset and report the weighted-F1 score for the base theory and the corresponding logical paraphrase, averaged across all theories in the evaluation set.

4 The ROBUSTLR Dataset

In this section, we describe details about the ROBUSTLR dataset sampling and filtering procedure. Please refer to Appendix A for details about the domains of rules and facts.

4.1 Dataset Sampling

For sampling the theories in ROBUSTLR, we use a modified version of the Label-Priority sampling (Zhang et al., 2022). The detailed algorithm is described in Algorithm 1 in Appendix. At a high-level, we sample different predicates and assign the value 0 or 1 to them. After that, we divide the predicate set into multiple levels. This helps us in sampling theories with multi-hop reasoning depths. After that, rules are derived by connecting predicates with the same label between two different levels. Finally, the predicates with value 1 at the 0th level form the facts in the theory, the connections denote the rules, and the predicates in the last level denote some candidate statements.

4.2 Filtering Statistical Features

In a contemporary work, Zhang et al. (2022) find that LMs are specifically prone to pick up any existing *statistical features* that can be present in the training datasets. These are described as certain statistic of an instance that has a strong correlation with the label. Examples of statistical features are #facts, #rules, #negation op, #facts with negation, etc.

We introduce some changes in our sampling algorithm to minimize the influence of such statistical features. We define a check to ensure that the number of statements with the different labels are similar for any given theory in the training dataset. Since we have negations in the dataset, it allows us to exactly control the *True* and *False* label distribution per theory instance systematically. We control the *Unknown* label by oversampling theories

and discarding ones with skewed label distribution. Please refer to Appendix E for more details.

5 Experimental Setup

Training Data Details We use four different training datasets to fine-tune baselines, described as follows: **NOT**: In this dataset, we allow negations in both facts and rules, but restrict to only using simple rules. Note that it is not possible to create a dataset without any operators (i.e., no negation, conjunction, and disjunction) as it would not be possible to have the *False* label in that dataset. **AND+NOT**: Here, we restrict the connector for compound rules to AND ($\wedge$). As before, we allow negations in facts and rules. **OR+NOT**: Similar to AND+NOT, we restrict the connector of the compound rules to OR ($\vee$). We allow negations in facts and rules as before. **All**: This dataset has all the three logical operators (AND, OR, NOT) present.

The dataset All contains all the logical operators we consider in the Logical Contrast set. Thus, instances from the All dataset cover all forms of rules seen in these test sets. We aim to understand the effect of these training datasets on the evaluation sets by fine-tuning the model on each dataset separately. Please refer to Appendix D for more details on the training and evaluation data statistics.

Models and Experiment Details Following prior works (Clark et al., 2020; Tafjord et al., 2021; Sanyal et al., 2022), we evaluate the performance of three language models: RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2020), and GPT-3 (Brown et al., 2020). Specifically, we evaluate the model checkpoints RoBERTa-Large, T5-Large, T5-3B, T5-11B, and GPT-3. To evaluate a model, we first fine-tune it on one of the training dataset mentioned above, and then evaluate on the Logical Contrast (Section 6.2) and Logical Equivalence (Section 6.3) evaluation sets. For T5-11B, we only finetune it on the All dataset due to compute constraints. For GPT-3, we evaluate its performance on a subset of the test sets using demonstrations. Please refer to Appendix B for details on the input formats for each model and Appendix C for the hyperparameter settings and other implementation details.

6 Results

6.1 In-domain Performance

The performance of the LMs on the in-distribution held-out data are shown in Table 3. We note that the models are able to solve the in-distribution test dataset almost perfectly in all cases. This either means the model understands the logical reasoning task perfectly (which is unlikely) or it learns some spurious features to solve the task using shortcuts. Now, we evaluate these models on our test sets to check the logical robustness.

6.2 Performance on Logical Contrast set

Overall Result We finetune RoBERTa-Large, T5-Large, and T5-3B models on different training datasets and evaluate them on the three types of Logical Contrast set. The results are shown in Table 2. Based on the **Avg** performance, we find that the models perform significantly worse on the Logical Contrast set, compared to the almost perfect performance on the in-distribution test sets. This shows that even after finetuning on the logical deductive reasoning datasets, these models do not learn the semantics of the logical operators in a robust manner, but likely use spurious correlations.

Additionally, we find that on average, model performance is similar for different training datasets, except for NOT dataset. While this result seems a bit surprising at first, we find the reason behind this in the performance breakdown by perturbation type. We observe a trend that training on related operators as the perturbation in the evaluation subset usually leads to better performance. For instance, we find that models trained on the OR+NOT training data perform better on the D-CS compared to when trained on NOT or AND+NOT training data. Similar trend is observed for C-CS. This is intuitive as training on the related operators helps the model to understand the semantics of the logic relatively better than just relying on having seen these at pre-training time. This shows that the model indeed requires some training data that is strongly aligned with the operators being evaluated the test set.

Variation with logical operators Next, we want to understand which among the three operators are more challenging for the models to learn. To better understand this, we evaluate the models after finetuning on the All dataset and plot the model performance for different perturbation groups in Figure 4. These groups (defined in Section 3.3) contain perturbations of a specific operator, as suggested by their names. We find that the most challenging operator is negation. This is evident from the lowest scores on the NEG perturbation group among CONJ, DISJ, and NEG. Further, this is also

Data	RoBERTa-Large				T5-Large				T5-3B			
	Avg	C-CS	D-CS	N-CS	**Avg**	C-CS	D-CS	N-CS	**Avg**	C-CS	D-CS	N-CS
NOT	0.39	0.39	0.45	0.34	0.44	0.36	0.55	0.41	0.59	0.58	0.60	0.60
AND+NOT	0.52	0.56	0.53	0.48	0.47	0.43	0.55	0.42	0.57	0.58	0.57	0.55
OR+NOT	0.47	0.39	0.61	0.42	0.46	0.36	0.60	0.43	0.56	0.44	0.67	0.57
All	0.47	0.44	0.61	0.37	0.46	0.37	0.61	0.40	0.58	0.54	0.65	0.54

Table 2: Performance of RoBERTa-Large, T5-Large, and T5-3B on Logical Contrast sets. We report the weighted-F1 score for each subset, and average that for the **Avg** column. Please refer to Section 6.2 for more details.

Training Dataset	RoBERTa-Large	T5-Large	T5-3B	T5-11B
NOT	1.00	1.00	0.98	-
AND+NOT	1.00	0.99	0.97	-
OR+NOT	1.00	0.99	0.97	-
All	1.00	0.99	0.92	0.94

Table 3: Performance of RoBERTa-Large, T5-Large, T5-3B, and T5-11B on in-domain held-out set. Please refer to Section 6.1 for more details.

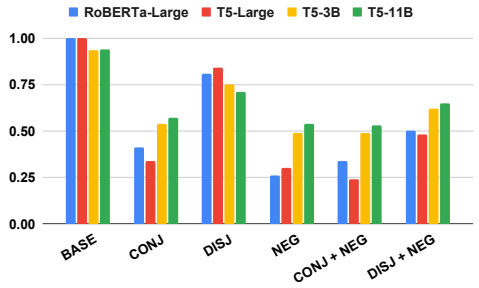


Figure 4: Performance comparison of RoBERTa-Large, T5-Large, T5-3B, and T5-11B across different groups of contrast perturbations. Negations are hardest to learn across all settings. Refer to Section 6.2 for more details.

observed from the fact that performance generally drops when negation perturbations are introduced along with any other perturbations. For instance, we see an average drop of around 25% between DISJ and DISJ+NEG. This demonstrates that the model not able to learn the negation semantics very well. Lastly, we find that models find conjunction relatively harder than disjunction. Please refer to Appendix G for more details.

6.3 Performance on Logical Equivalence set

Results on Contrapositive Equivalence Next, we evaluate the fine-tuned LMs on the Logical Equivalence sets. We observe that the model performance degrades by approximately 20% for the C-ES, compared to the in-distribution performance in Table 3. Contraposition involves changing the rule into a format that has two negations, thus testing the limits of the model on understanding negations. From the experiments on Logical Contrast sets, we know that negations are not well understood by the model. Thus, these results reinforce our previous findings. We do not find any significant changes in performance when trained on different operators. Thus, we conclude that, for this test set, it is not sufficient to just understand the semantics of the logical operators, but it rather requires a higher order understanding about the interactions between the logical operators and implications. Including such knowledge in deductive reasoning models is an interesting direction for future works.

Results on Distributive Equivalence For D1-ES and D2-ES test sets, we see a higher performance compared to C-ES as these equivalence conditions are relatively easier than the contraposition rule. This is because the distributive rules are similar to having compound rules in the dataset (that is already present in AND+NOT, OR+NOT, and All datasets). Between the two sets, we observe that D1-ES is more challenging for the model. This indicates that models find conjunction operator harder than the disjunction, similar to our observations from Figure 4. One reason for this can be the strictness involved in the conjunction operation. A rule with conjunction is true only if the individual parts are independently true.

6.4 Human Evaluation

To better understand the upper limit of ROBUSTLR evaluation sets, we ask 3 Computer Science graduate students to annotate 30 randomly sampled theories from each subset of Logical Contrast and Logical Equivalence sets. The results are shown in the last row of Table 5. We find that humans are significantly better than other baselines, performing around 27% higher on the Logical Contrast set compared to T5-11B on average. Additionally, we see similar trends that humans find negation based perturbations (N-CS) hardest, followed by conjunction (C-CS), and disjunction (D-CS). Please refer to Appendix I for further details.

[2]Fintuned for two epochs on the All training set.
[3]Evaluated on 500 samples for each test subset.

Training Dataset	RoBERTa-Large				T5-Large				T5-3B			
	Avg	C-ES	D1-ES	D2-ES	**Avg**	C-ES	D1-ES	D2-ES	**Avg**	C-ES	D1-ES	D2-ES
NOT	0.81	0.79	0.91	0.74	0.88	0.76	0.90	0.97	0.70	0.78	0.79	0.54
AND+NOT	0.87	0.80	0.88	0.94	0.88	0.77	0.90	0.96	0.83	0.76	0.81	0.93
OR+NOT	0.87	0.78	0.87	0.95	0.86	0.76	0.86	0.97	0.86	0.77	0.82	0.98
All	0.89	0.79	0.93	0.94	0.89	0.77	0.93	0.98	0.84	0.72	0.83	0.98

Table 4: Performance of RoBERTa-Large, T5-Large, and T5-3B on Logical Equivalence sets. We report the weighted-F1 score, and average that for the **Avg** column. Please refer to Section 6.3 for more details.

Models	Logical Contrast Set				Logical Equivalence Set			
	Avg	C-CS	D-CS	N-CS	**Avg**	C-ES	D1-ES	D2-ES
From scratch	0.14	0.10	0.21	0.10	0.45	0.33	0.50	0.51
RoBERTa	0.47	0.44	0.61	0.37	0.76	0.40	0.93	0.94
T5-Large	0.46	0.37	0.61	0.40	0.89	0.77	0.93	0.98
T5-3B	0.58	0.54	0.65	0.54	0.84	0.72	0.83	0.98
T5-11B [2]	0.61	0.57	0.67	0.58	0.83	0.76	0.76	0.97
GPT-3 [3]	0.36	0.34	0.50	0.25	0.67	0.36	0.78	0.87
human	0.88	0.87	0.94	0.84	0.91	0.81	0.97	0.96

Table 5: Comparisons between training a model from scratch, finetuning a pre-trained checkpoint at scale (RoBERTa, T5-Large, T5-3B, and T5-11B), using in-context learning (GPT-3), and human performance, on the Logical Contrast and Logical Equivalence sets. Please refer to Sections 6.4 and 6.5 for more details.

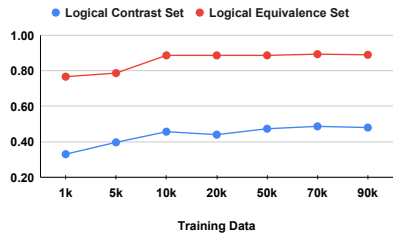


Figure 5: Average performance of RoBERTa-Large on Logical Contrast and Logical Equivalence sets when trained on varying amount of All dataset.

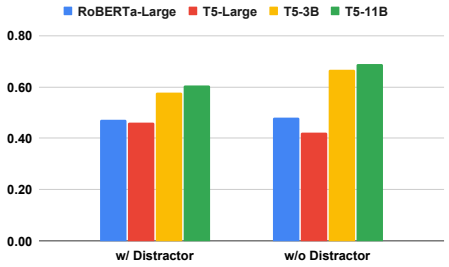


Figure 6: Performance comparison of RoBERTa-Large, T5-Large, T5-3B, and T5-11B for two variants of Logical Contrast sets: with and without distractors. Refer to Section 6.5 for more details.

6.5 Analysis

Performance of Larger LMs Here we evaluate the performance of some larger models on ROBUSTLR. The goal is to estimate the possible gains with scaling to large LMs. For this, we finetune a T5-11B using the All training dataset for two epochs. Additionally, we evaluate GPT3 (Brown et al., 2020) on a subset of 500 samples per test set, using demonstration-based in-context learning. The results are shown rows 5-6 in Table 5. We observe that increasing the model size from T5-Large to T5-11B indeed leads to a significant performance gain on most datasets. But it is still quite far compared to human performance. This suggests that scaling can potentially help with learning robust logical operations to an extent. Additionally, we find that GPT-3 is not able to perform well on these test sets. We hypothesize this is likely due to the mismatch of the training distribution of the GPT-3 model, versus our synthetic datasets, as we do not finetune the GPT-3 model on our training sets.

Effect of LM Pre-training In this part, we evaluate the usefulness of using a pre-trained checkpoint in our experiments, in comparison with training a RoBERTa-Large architecture from scratch. We train both RoBERTa-Large pre-trained checkpoint and a similar model from scratch using the All dataset, and evaluate on the ROBUSTLR sets. The results are shown in rows 1-2 in Table 5. We observe a significant drop in performance, demonstrating that knowledge learned during pre-training is crucial for this task.

Effect of size of training data Next, in Figure 5, we plot the overall performance of RoBERTa-Large model finetuned on a varying amount of All training data. We observe that the model performance increases with increasing amount of training data, as expected, and then saturates at a fixed level. This shows that there is no significant effect of using larger training datasets on model performance. We use 50k training samples in all datasets.

Effect of distractors We define distractors as the facts and rules that are not part of the proof set for a given theory and statement. Figure 6 depicts the effect of distractors on the model performance, when finetuned on the All dataset and evaluated on the Logical Contrast set. We observe that the

all Dataset	Logical Contrast Set				Logical Equivalence Set			
	Avg	C-CS	D-CS	N-CS	**Avg**	C-ES	D1-ES	D2-ES
Original	0.47	0.44	0.61	0.37	0.89	0.79	0.93	0.94
w/o *Unknown*	0.83	0.83	0.77	0.90	0.86	0.72	0.87	1.00

Table 6: Performance of RoBERTa-Large checkpoint when finetuned on a subset of the training dataset without *Unknown* label. We observe that the performance is still below the in-domain performances.

performance generally improves (or stays similar) on the variant without distractors. This shows that retrieving the relevant facts and rules in the theory from a given set of sentences is a non-trivial challenge. Thus, performing both retrieval and entailment prediction in a single model can lead to some performance degradation.

Effect of Statistical Features In a contemporary work, Zhang et al. (2022) claims that deductive reasoning models inherently learn to use statistical features in the training data such as #rules, #facts, etc. Here, we demonstrate that it is not the complete reason for failure using the following control study. We finetune the RoBERTa-Large model on a subset of the `All` dataset, where we restrict to two labels: *True* and *False*, by filtering out the *Unknown* label. In our sampling algorithm, we ensure that each theory has the exact same number of *True* and *False* labeled statements in the training set. Thus, it is not possible to learn any statistic of the data, as the label distribution per theory is exactly uniform [4]. Next, we evaluate the model on the ROBUSTLR evaluation sets, with the *Unknown* label filtered in each set. The results are shown in Table 6. We observe that, although the performance on this reduced test set is improved, there is still around 15% gap on average with respect to performance on in-distribution data. This gap suggests that the model is not able to learn the logical operators robustly, even without any scope of spurious statistical features in the training set. Thus, this shows that although spurious correlation can lead to non-robust model behavior, it is not the sole reason for failure of these deductive reasoning models. This calls for the development of better inductive biases to teach the logical semantics more robustly to the language models.

7 Related Works

Reasoning in natural language has been a prevalent problem in NLP. There are multiple reasoning datasets, studying different aspects of reasoning over textual inputs. Natural Language Inference (NLI) (Dagan et al., 2006) is a prominent dataset that requires reasoning over text to answer if a statement is entailed, contradicted, or neutral given a hypothesis. HotpotQA (Yang et al., 2018b) tests multi-hop reasoning abilities that require comparisons and inferring missing bridge between sentences. CLUTRR (Sinha et al., 2019) tests whether models can infer biological relationships between entities in a context. RICA (Zhou et al., 2021) requires the model to employ commonsense reasoning to answer questions based on a context.

Recently, there has been an increasing focus on evaluating the logical reasoning abilities of LMs. ReClor (Yu et al., 2020) and LogiQA (Liu et al., 2021) are logical reasoning datasets derived from examinations. RuleTaker (Clark et al., 2020) proposes synthetic deductive reasoning datasets that uses only the knowledge in the context. There are very limited works that probe the logical reasoning abilities of language models (LMs). FaiRR (Sanyal et al., 2022) tests the robustness of logical reasoning models when the subjects and attributes in the context are altered to out-of-distribution terms. In a contemporary work, Zhang et al. (2022) show that language models can learn to use statistical features that can be present in deductive reasoning datasets. To the best of our knowledge, ROBUSTLR is the first dataset that tests how robust these LMs are to different *logical* perturbations.

8 Conclusion

In this paper, we proposed ROBUSTLR, a diagnostic benchmark to test the logical robustness of deductive reasoning models. In ROBUSTLR, we propose two evaluation sets, Logical Contrast and Logical Equivalence, each probing different logical reasoning abilities. Overall, we find that fine-tuning LMs such as RoBERTa and T5 on deductive reasoning datasets is not sufficient to learn the semantics of the logical operators conjunction, disjunction, and negation. Although well-aligned training dataset improves model performance, the models still find it challenging to understand negations, both in Logical Contrast and Logical Equivalence sets. We demonstrate some interesting shortcoming of LMs designed for logical reasoning, that can eventually enable building better reasoning models.

[4] Zhang et al. (2022) do not consider negations in the theory, and thus cannot ensure this property.

9 Limitation

A key limitation of the work is the synthetic nature of the dataset. While it is ideal to explore more natural theories, it makes the systematic logical perturbation process very challenging. Thus, in this work, we resort to using synthetic datasets, but aim to bridge this gap in future works. Another limitation is the complexity of the datasets we explore. We use fairly simple logical rules and constructs for ROBUSTLR. Some more complex forms of logical reasoning-based theories can potentially reveal even more limitations of deductive reasoning models. Another interesting aspect we do not explore in this scope is potential techniques to improve these models on deductive reasoning tasks. This might involve trying different inductive biases in the form of architectural designs, more specialized datasets, etc.

References

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, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. 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)*, pages 4026–4032, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization. Main track.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. MuTual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Nicolas Gontier, Koustuv Sinha, Siva Reddy, and C. Pal. 2020. Measuring systematic generalization in neural proof generation with transformers. In *NeurIPS'20*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Yinhan Liu, Myle Ott, Naman Goyal, and Jingfei Du an. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Robin Manhaeve, Sebastijan Dumancic, A. Kimmig, T. Demeester, and L. D. Raedt. 2019. Deepproblog: Neural probabilistic logic programming. In *BNAIC/BENELEARN*.

John W. McCarthy. 1959. Programs with common sense. In *Proc. Tedding Conf. on the Mechanization of Thought Processes*, pages 75–91.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Tim Rocktäschel and S. Riedel. 2017. End-to-end differentiable proving. In *NeurIPS*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022. FaiRR: Faithful and robust deductive reasoning over natural language. *arXiv preprint arXiv:2203.10261*.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations (ICLR)*.

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

Pei Zhou, Rahul Khanna, Seyeon Lee, Bill Yuchen Lin, Daniel Ho, Jay Pujara, and Xiang Ren. 2021. RICA: Evaluating robust inference capabilities based on commonsense axioms. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7560–7579, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A ROBUSTLR Dataset Domain

Facts The domains of the predicate relation X and variable a in the unary predicate $X(a)$ are the simple English adjectives and the proper names, respectively. Examples of this predicate form are "green(Alex)", "kind(John)", etc. Each predicate is associated to the English template sentence form "$\{a\}$ is $\{X\}$.". For the binary predicate $X(a, b)$, we consider family relationships and proper names as the domain of X and a respectively. Some examples of this predicate form are "daughter(Mary, Gary)","father(Bob, John)", etc. Each predicate is associated with template sentences such as "$\{a\}$ is the $\{X\}$ of $\{b\}$.", "The $\{X\}$ of $\{b\}$ is $\{a\}$.", etc. Note that, currently, we do not enforce any gender constraints on the names, thus allowing predicates such as "daughter(Bob, Gary)", which might be unlikely based on the genders associated statistically to names in English.

Rules For the rules, we follow the same domain as mentioned above for facts. We allow rules containing unary predicates, binary predicates, or a combination of both. Examples of some simple rules consisting of atomic predicates are "green(Alex) $\implies$ daughter(Bob, Gary)", "$\neg$ father(Bob, John) $\implies$ kind(John)", etc. Similarly, examples of some compound rules containing complex predicates are "green(Alex) $\vee$ smart(Bob) $\implies$ daughter(Bob, Gary) $\wedge$ $\neg$ kind(John)", etc. We note that, for the sake of keeping the theories deterministic, we do not allow the disjunction operator in the RHS of a compound rule. A rule of the form $p \implies q$ is associated with templates such as "If $\{p\}$ then $\{q\}$.", "$\{q\}$ if $\{p\}$.", etc., where the p's and q's can be recursively resolved to their own templates as defined in the predicates.

B Model implementation details

In this section, we describe the implementation details of the language models used to evaluate ROBUSTLR.

- **RoBERTa-Large**: Following Rule-Taker (Clark et al., 2020), we use a pre-trained RoBERTa-Large (Liu et al., 2019) model to perform the classification task. Specifically, we input in the format $[CLS]$ T $[SEP]$ s $[SEP]$ to the RoBERTa-Large model, and extract the $[CLS]$ embedding to predict the label. The schematics of the RoBERTa model input is shown in Figure 7. Here, T is the theory which is the concatenation of the facts and rules, and s is the statement. We use Cross Entropy loss to fine-tune the model on the training dataset.

- **T5-Large**: Following ProofWriter (Tafjord et al., 2021), we train a T5-Large (Raffel et al., 2020) model for the deductive reasoning task. For this, we add a prefix to the T5-Large's input and generate the output in a fixed format. Specifically, we give the input in the format: $\$answer\$$; $\$question\$ = s$; $\$context\$ = T$. Here, T is the theory which is the concatenation of the facts and rules, and s is the statement. And the output is defined to be in format: $\$answer\$ = True/False/Unknown$. The model is trained on the default language modeling loss to match the output format. At evaluation time, we match the output template with the above description and generate the model's predicted label accordingly.

- **T5-3B**: Similar to T5-Large above, we use the T5-3B checkpoint.

- **T5-11B**: Similar to T5-Large above, we use the T5-11B checkpoint.

- **GPT3**: We use GPT-3 (Brown et al., 2020) for model evaluation to check its performance on our C-CS,D-CS,N-CS. Following (Sanh et al., 2021), we experiment with all the prompts for the NLI task and select a prompt which performs the best on the evaluation datasets. We experiment with inserting 3 demonstrations and 10 demonstrations before the sentence and find that the performances are nearly same. So, we finally use the prompt named "based on the previous passage" which will give the input in a format as shown below:

  Input Template:
  `{{premise}} Based on the previous passage, is it true that "{{hypothesis}}"? Yes or no?`

  Output Template:
  `{{ answer_choices[label] }}`

  Answer Choices Template:
  `Yes ||| Maybe ||| No`

  We use 3 demonstrations for each sample to

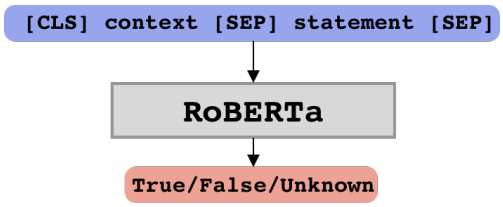


Figure 7: **Overview of the RoBERTa-Large model** - The *context* (containing the facts and rules) and the *statement* are concatenated together as input and passed into a RoBERTa-Large model. The model is trained on cross entropy loss for a 3-class classification task.

limit the total tokens evaluated using the OpenAI GPT-3 API [5].

C Hyperparameter Details

Here we use RoBERTa-Large (Liu et al., 2019)and T5-Large,T5-3B,T5-11B (Raffel et al., 2020) models for the 3-class deductive reasoning classification task. Only use GPT-3 (Brown et al., 2020) for evaluation. We train the pre-trained checkpoints available in the Hugging Face (Wolf et al., 2020) Transformers library. For RoBERTa-Large model, we use AdamW (Loshchilov and Hutter, 2019) with learning rate $1e$-5. For T5-Large T5-3B and T5-11B, we use AdamW with learning rate $1e$-4, adamw_epsilon $1e$-6, warmup_ratio $1e$-1, weight_decay $1e$-2. All these models are trained with batch size 8 on Nvidia Quadro RTX 8000 GPUs. For RoBERTa-Large and T5-Large,training a single task on one GPU costs nearly 8 hours on average. For T5-3B and T5-11B, we use 4 GPUs to train the model and averagely need 5 and 10 hours for one epoch.

D Dataset Statistics

In this section, we describe the training and evaluation dataset statistics. We first train the model on the datasets in Table 7. Each dataset comprises of different types of logical operators to help us in understanding the effect of different logical operators. Then we evaluate the trained models on evaluation datasets mentioned in Table 8. For evaluation, we test the model on two subsets of ROBUSTLR: Logical Contrast set and Logical Equivalence set. Each set is further sub-categorized into three different parts, based on the type of perturbations.

[5] https://openai.com/api/

Training Dataset	Train	Dev	Test
NOT	50000	10000	10000
AND+NOT	50000	10000	10000
OR+NOT	50000	10000	10000
All	50000	10000	10000

Table 7: Training dataset statistics. Please refer to Section D for more details.

Evaluation Set	Number of instances
C-CS	20000
D-CS	20000
N-CS	20000
C-ES	20000
D1-ES	20000
D2-ES	20000

Table 8: Evaluation dataset statistics. Refer to Section D for more details.

E Filtering Statistical Features

In Figure 8, we show the plots of the label distribution for the following statistical features in the input theory and statement: #rules, #facts, #facts with negation, #rules with negation, #rules with conjunction, #rules with disjunction, and #statements with negations. We observe that there is no significant bias between any of these features and the task label. Additionally, we show the count histogram of the instances in blue. Overall, our dataset filtering is able to remove some of the count-based statistical features.

F Contrast Perturbations

Following Section 3.3, we show the conjunction, disjunction, and negation contrast perturbations for the case when base theory's label is *False* in Tables 11, 12, and 13, respectively.

For the conjunction contrast set perturbations in Table 1 and 11, the first row is a base theory which is used to generate these contrast sets. In the next set of triads, the rule is modified to have an unseen predicate t in conjunction with the existing rule LHS. Here t is a predicate that is not part of the existing facts and inferences in the theory (hence, referred to as unseen predicate). Additionally, we add t (or $\neg t$) as part of the facts in the theory. This lead to modification of the label as shown in rows 3-4. For the next set of triads, we modify the base rule to have a negated rule RHS $\neg q$. The corresponding label changes are shown in rows 5-7. In Table 11, we assume the label of the statement is

Modified Rule	Facts	Statement	Label	Group
$p \implies q$	$\{p\}$	q	*True*	BASE
$p \vee t \implies q$	$\{p\}$	q	*True*	DISJ
$p \vee t \implies q$	$\{p, t\}$	q	*True*	DISJ
$p \vee t \implies q$	$\{\neg p, \neg t\}$	q	*Unknown*	DISJ+NEG
$p \vee t \implies \neg q$	$\{p\}$	q	*False*	DISJ+NEG
$p \vee t \implies \neg q$	$\{p, t\}$	q	*False*	DISJ+NEG
$p \vee t \implies \neg q$	$\{\neg p, \neg t\}$	q	*Unknown*	DISJ+NEG

Table 9: **Disjunction Contrast Perturbations**. The minimal edits done to a base theory (first row) for testing the disjunction and negation reasoning abilities. The group reflects the overall change in theory w.r.t. the base theory.

Modified Rule	Facts	Statement	Label	Group
$p \implies q$	$\{p\}$	q	*True*	BASE
$p \implies \neg q$	$\{p\}$	q	*False*	NEG
$\neg p \implies q$	$\{p\}$	q	*Unknown*	NEG
$\neg p \implies \neg q$	$\{p\}$	q	*Unknown*	NEG

Table 10: **Negation Contrast Perturbations**. The minimal edits done to a base theory (first row) for testing the negation reasoning abilities.

False for the base theory in row 1. Similar perturbations are possible for the label *True*, and is shown in Table 1 in 3.3. We group these perturbations into three classes as shown in Table 11: BASE, CONJ, CONJ+NEG. These groups are based on which logical operator is the new addition with respect to the base theory. If a model performs accurately on this contrast set, we expect that the model understands the semantics of conjunction and negation logical operators reasonably well.

Similar to the C-CS above, we show the perturbations considered in the D-CS in Table 9 and 12, where the distractor is added to the rule LHS using disjunction ($\vee$). Lastly, we show the N-CS perturbations in Tables 10 and 13, where negations are added to the rule LHS and/or RHS.

G Logical Contrast set breakdown

In this section, we further discuss the performance of the LMs on each group of the Logical Contrast set. From Tables 14, 15 and 16 we can say that the models generally perform worse when they need to handle more complicated compound rules (CONJ + NEG > CONJ > BASE (where > means harder)). Additionally, we find that when we add more compound rules in the training dataset, the performance is generally better. Giving more complex rules can lead to further drops in performance, as noted by performance on the CONJ+NEG and DISJ+NEG. Models trained on the dataset with aligned operators instead of `All` dataset is better e.g., model trained on `AND+NOT` get best result at CONJ+NEG.

Modified Rule	Facts	Statement	Label	Group
$p \implies \neg q$	$\{p\}$	q	*False*	BASE
$p \wedge t \implies \neg q$	$\{p\}$	q	*Unknown*	CONJ
$p \wedge t \implies \neg q$	$\{p, t\}$	q	*False*	CONJ
$p \wedge t \implies \neg q$	$\{p, \neg t\}$	q	*Unknown*	CONJ+NEG
$p \wedge t \implies q$	$\{p\}$	q	*Unknown*	CONJ+NEG
$p \wedge t \implies q$	$\{p, t\}$	q	*True*	CONJ+NEG
$p \wedge t \implies q$	$\{p, \neg t\}$	q	*Unknown*	CONJ+NEG

Table 11: **Conjunction Contrast Perturbations**. These are perturbations for testing conjunction and negation reasoning abilities. First row is the base theory being perturbed. Please refer to Appendix F for more details.

Modified Rule	Facts	Statement	Label	Group
$p \implies q$	$\{p\}$	q	*False*	BASE
$p \vee t \implies \neg q$	$\{p\}$	q	*False*	DISJ
$p \vee t \implies \neg q$	$\{p, t\}$	q	*False*	DISJ
$p \vee t \implies \neg q$	$\{\neg p, \neg t\}$	q	*Unknown*	DISJ+NEG
$p \vee t \implies q$	$\{p\}$	q	*True*	DISJ+NEG
$p \vee t \implies q$	$\{p, t\}$	q	*True*	DISJ+NEG
$p \vee t \implies q$	$\{\neg p, \neg t\}$	q	*Unknown*	DISJ+NEG

Table 12: **Disjunction Contrast Perturbations**. These are perturbations for testing disjunction and negation reasoning abilities. First row is the base theory being perturbed. Please refer to Appendix F for more details.

It is easy to see that model with larger amount of parameters give more consistent and better result at C-CS, D-CS, N-CS which means the model learned more semantics of logic from language and is more robust.

H Result breakdown by label

We report the performance for each label in Tables 17 to 22, for both the Logical Contrast and Logical Equivalence sets. We find that T5-3B model, the largest model among the three models, get a good result for *Unknown* while other two models are not good at it. It shows that models with large amount of parameters can better learn to predict the *Unknown* label, which is relatively harder than the other two labels. Also, we find that Logical Equivalence set is an easier task in general than Logical Contrast set and the performances are stable across three models.

I Human Evaluation

We recruit three Computer Science graduates to annotate the datasets. To keep the annotation realistic, we sample 30 instances from each test subset of ROBUSTLR and ask the annotators to mark a label from *True*, *False*, and *Unknown*. The question asked is: "Does the theory entail or contradict the statement, or we cannot say anything about it?". Overall, we find the average inter-annotator agree-

Modified Rule	Facts	Statement	Label	Group
$p \implies q$	$\{p\}$	q	*False*	BASE
$p \implies \neg q$	$\{p\}$	q	*True*	NEG
$\neg p \implies q$	$\{p\}$	q	*Unknown*	NEG
$\neg p \implies \neg q$	$\{p\}$	q	*Unknown*	NEG

Table 13: **Negation Contrast Perturbations**. The minimal edits done to a base theory (first row) for testing the negation reasoning abilities.

ment to be around 0.79, evaluated using the Fleiss' kappa score [6].

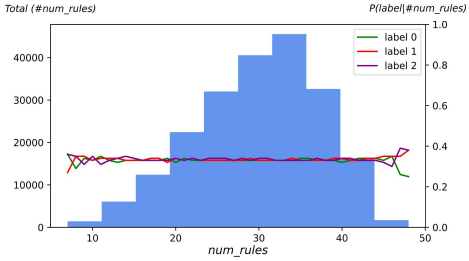
Algorithm 1: Sampling Algorithm

Input : $vocab$ containing the corpus of all predicates, $ruleset$ containing the set of valid rules, predicate negation probability n_1, statement negation probability n_2, max reasoning depth d.

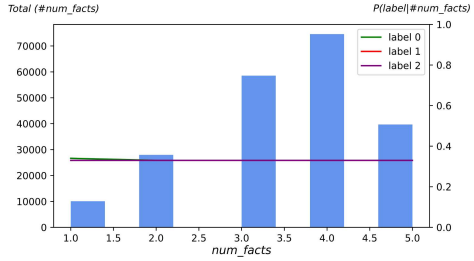
Output : A theory containing a set of $facts$ and $rules$, a $statement$, and a corresponding $label \in \{0, 1, 2\}$

```
pred_num ~ U[10, 30]
preds ← Sample(vocab, pred_num)
set l ~ U[1, d] and group preds into l layers
rules ← [ ]
for predicate p in layer 1 ≤ i ≤ l do
    Negate p with probability n_1
    q ~ U[0, 1]
    assign label q to predicate p
    if i ≥ 1 then
        k ~ U[1, 2]
        cand ← p in layer i − 1 with label q
        body ← Sample(cand, k)
        if len(body) > 1 then
            operator ← Sample([∧, ∨], 1)
            Compose the predicates in the body
            using operator
        end if
        r ← (body ⟹ p)
        if Validate(ruleset, r) then
            add r to rules
        else
            /* Rule r does not match any
               valid rule forms, so the
               predicate is not provable */
            assign label 0 to predicate p
        end if
    end if
end for
facts ← predicates in layer 1 with label 1
statement ← Sample(preds, 1)
label ← pre-assigned label for statement
if label == 1 then
    Negate the statement with probability n_2
    label ← 2
end if
return (facts, rules, statement, label)
```

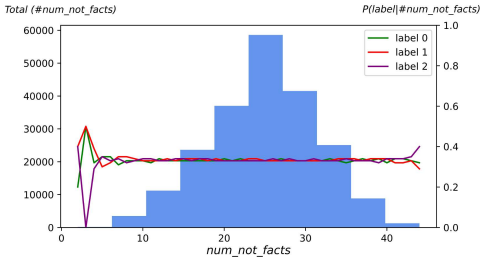
[6] https://www.statsmodels.org/dev/generated/statsmodels.stats.inter_rater.fleiss_kappa.html



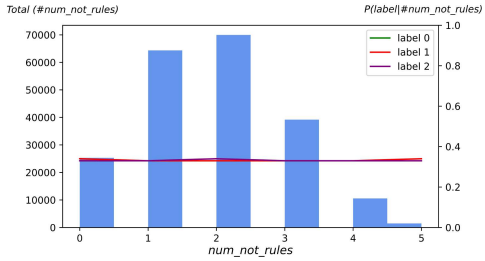
(a) # Rules



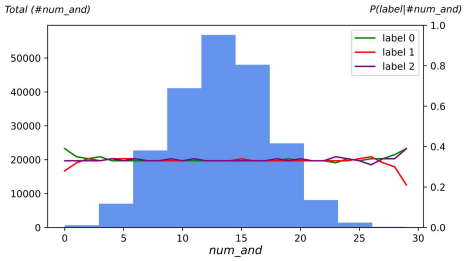
(b) # Facts



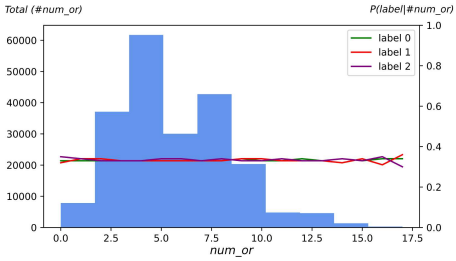
(c) # Facts with Negations



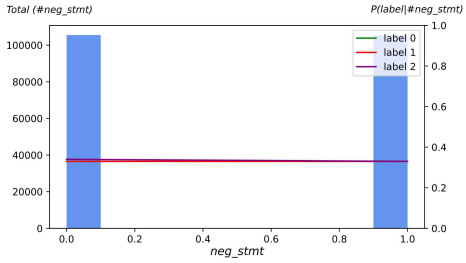
(d) # Rules with Negations



(e) # Rules with Conjunctions



(f) # Rules with Disjunctions



(g) # Statements with Negations

Figure 8: Plots of label distribution with respect to different statistical features for the `All` dataset after our filtering techniques are used. Additionally, we plot the histogram of the count of instances for each feature value.

Logical Contrast set breakdown	C-CS			D-CS			N-CS	
	BASE	CONJ	CONJ + NEG	BASE	DISJ	DISJ + NEG	BASE	NEG
NOT	1.00	0.28	0.28	1.00	0.62	0.28	1.00	0.22
AND+NOT	1.00	0.50	0.48	1.00	0.56	0.43	1.00	0.37
OR+NOT	1.00	0.32	0.29	1.00	0.74	0.52	1.00	0.31
All	1.00	0.41	0.34	1.00	0.81	0.50	1.00	0.26

Table 14: Performance breakdown of RoBERTa-Large with different groups of Logical Contrast set. Please refer to Appendix G for more details.

Logical Contrast set breakdown	C-CS			D-CS			N-CS	
	BASE	CONJ	CONJ + NEG	BASE	DISJ	DISJ + NEG	BASE	NEG
NOT	1.00	0.30	0.23	1.00	0.68	0.43	1.00	0.30
AND+NOT	1.00	0.40	0.31	1.00	0.69	0.43	1.00	0.32
OR+NOT	1.00	0.31	0.23	1.00	0.77	0.49	1.00	0.32
All	1.00	0.34	0.24	1.00	0.84	0.48	1.00	0.30

Table 15: Performance breakdown of T5-Large with different groups of Logical Contrast set. Please refer to Appendix G for more details.

Logical Contrast set breakdown	C-CS			D-CS			N-CS	
	BASE	CONJ	CONJ + NEG	BASE	DISJ	DISJ + NEG	BASE	NEG
NOT	0.96	0.56	0.53	0.96	0.61	0.57	0.96	0.55
AND+NOT	0.91	0.58	0.53	0.90	0.62	0.54	0.90	0.51
OR+NOT	0.95	0.37	0.39	0.93	0.74	0.65	0.93	0.53
All	0.93	0.54	0.49	0.93	0.75	0.62	0.94	0.49

Table 16: Performance breakdown of T5-3B with different groups of Logical Contrast set. Please refer to Appendix G for more details.

Training Dataset	C-CS			D-CS			N-CS		
	False	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*
NOT	0.77	0.76	0.24	0.68	0.66	0.14	0.73	0.74	0.18
AND+NOT	0.89	0.87	0.48	0.71	0.71	0.38	0.67	0.69	0.49
OR+NOT	0.93	0.94	0.19	0.90	0.92	0.19	0.73	0.72	0.33
All	0.88	0.89	0.28	0.91	0.93	0.14	0.76	0.76	0.23

Table 17: Performance breakdown of RoBERTa-Large with different labels for Logical Contrast set. Please refer to Appendix H for more details.

Training Dataset	**Contrapositive**			**Distributive 1**			**Distributive 2**		
	False	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*
NOT	0.72	0.72	0.92	0.91	0.90	-	0.73	0.75	-
AND+NOT	0.73	0.74	0.92	0.88	0.89	-	0.94	0.94	-
OR+NOT	0.72	0.73	0.89	0.85	0.89	-	0.95	0.94	-
All	0.73	0.75	0.90	0.91	0.94	-	0.93	0.94	-

Table 18: Performance breakdown of RoBERTa-Large with different labels for Logical Equivalence set. Please refer to Appendix H for more details.

Training Dataset	C-CS			D-CS			N-CS		
	False	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*
NOT	0.91	0.90	0.13	0.85	0.85	0.10	0.91	0.90	0.15
AND+NOT	0.95	0.93	0.23	0.87	0.83	0.11	0.89	0.88	0.20
OR+NOT	0.94	0.95	0.12	0.93	0.94	0.10	0.83	0.86	0.24
All	0.95	0.93	0.13	0.95	0.94	0.07	0.92	0.90	0.14

Table 19: Performance breakdown of T5-Large with different labels for Logical Contrast set. Please refer to Appendix H for more details.

Training Dataset	**Contrapositive**			**Distributive 1**			**Distributive 2**		
	False	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*
NOT	0.72	0.72	0.83	0.90	0.90	-	0.97	0.97	-
AND+NOT	0.72	0.72	0.86	0.92	0.88	-	0.96	0.97	-
OR+NOT	0.71	0.72	0.87	0.85	0.86	-	0.97	0.98	-
All	0.73	0.72	0.85	0.94	0.91	-	0.98	0.98	-

Table 20: Performance breakdown of T5-Large with different labels for Logical Equivalence set. Please refer to Appendix H for more details.

Training Dataset	C-CS			D-CS			N-CS		
	False	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*
NOT	0.93	0.94	0.53	0.79	0.82	0.43	0.91	0.94	0.46
AND+NOT	0.91	0.86	0.53	0.80	0.76	0.41	0.90	0.84	0.44
OR+NOT	0.95	0.95	0.30	0.92	0.92	0.37	0.93	0.93	0.41
All	0.91	0.94	0.43	0.91	0.93	0.30	0.93	0.96	0.33

Table 21: Performance breakdown of T5-3B with different labels for Logical Contrast set. Please refer to Appendix H for more details.

Training Dataset	**Contrapositive**			**Distributive 1**			**Distributive 2**		
	False	*True*	*Unknown*	*False*	*True*	*Unknown*	*False*	*True*	*Unknown*
NOT	0.71	0.72	0.92	0.77	0.81	-	0.95	0.96	-
AND+NOT	0.71	0.69	0.88	0.83	0.78	-	0.95	0.90	-
OR+NOT	0.70	0.70	0.92	0.81	0.82	-	0.98	0.98	-
All	0.70	0.69	0.76	0.84	0.81	-	0.99	0.98	-

Table 22: Performance breakdown of T5-3B with different labels for Logical Equivalence set. Please refer to Appendix H for more details.