Evaluating Step-by-Step Reasoning through Symbolic Verification

Yi-Fan Zhang[1] **Hanlin Zhang**[2] **Li Erran Li**[3*] **Eric Xing**[1,2,4]
[1]MBZUAI [2]Carnegie Mellon University, [3]AWS AI, Amazon, [4]Petuum Inc.

Abstract

Pre-trained language models (LMs) have shown remarkable reasoning performance using explanations or chain-of-thoughts (CoT)) for in-context learning. On the other hand, these reasoning tasks are usually presumed to be more approachable for symbolic programming. To understand the mechanism of reasoning of LMs, we curate synthetic datasets containing equivalent (natural, symbolic) data pairs, where symbolic examples contain first-order logic rules and predicates from non-parametric knowledge bases (KBs), supporting automated verification of intermediate reasoning results. Then we revisit neuro-symbolic approaches and propose to learn from demonstrations containing logic rules and corresponding examples to iteratively reason over KBs, recovering Prolog's backward chaining algorithm and supporting automated verification of LMs' outputs. Comprehensive experiments are included to systematically compare LMLP with CoT in deductive reasoning settings, showing that LMLP enjoys more than 25% higher accuracy than CoT on length generalization benchmarks even with smaller model sizes.

1 Introduction

There are emerging interests in leveraging LMs to enable planning (Li et al., 2022; Huang et al., 2022), heuristic search (Dahlgren et al., 2021) and symbolic inference (Wei et al., 2022b; Zelikman et al., 2022; Zhang et al., 2022). Among them, *chain of thought* prompting or scratchpads (Wei et al., 2022b; Nye et al., 2021) shows that taking `(input, explanation, output)` as in-context examples for LMs can lead to significant performance gain in reasoning tasks. However, like many fine-tuning approaches, it can be difficult for these models to generalize compositionally (Zhou et al., 2022a), meaning they may struggle to apply their knowledge to solve new problems that

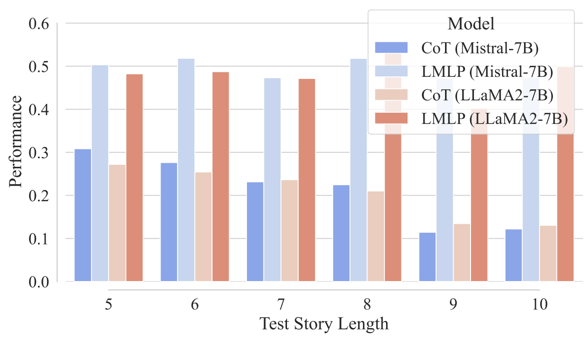


Figure 1: Deductive reasoning performance (human evaluation accuracy) comparisons on the CLUTRR-LP given training data with story length 2, 3, 4.

involve novel combinations of information (Lake and Baroni, 2018; Bahdanau et al., 2018; Keysers et al., 2019). One notable case is that LMs would suffer from catastrophic performance degradation when tested on sequences longer than training ones (Figure 1). As a solution, *least-to-most prompting* (Zhou et al., 2022a) takes inspiration from symbolic programs and proposes to tackle the challenge by modularizing the prompt on the reduced problem. The divide-and-conquer strategy is useful to improve the reasoning ability of language models, but it also presents additional challenges: what are the appropriate representations for factual knowledge and in-context samples that can ensure the correctness of each individual reasoning step? How do natural language explanations compare to symbolic provenance, which is easily verifiable, when used as prompts for reasoning?

Our goal is to evaluate the natural and symbolic paradigms closely in order to answer these questions. To enable fine-grained comparison and gain insight into in-context learning for reasoning tasks, we study relational reasoning over both natural language and knowledge bases (KBs). KBs are particularly useful for this purpose because they are constructed using clear pipelines and strong su-

*Work done outside of Amazon.

Findings of the Association for Computational Linguistics: NAACL 2024, pages 2984–3002

pervision, which makes them reliable and easy to control. This allows us to verify and evaluate reasoning paths and provenances without the need for human-provided rationales or explanations (Camburu et al., 2018; Zhou et al., 2020; Wei et al., 2022b; Nye et al., 2021; Zelikman et al., 2022). We study language models as logic programmers (LMLP) to enable few-shot learning from symbolic demonstrations and simultaneous planning in an explainable and scalable way. LMLP uses logic rule templates, examples, and pre-trained knowledge to iteratively perform in-context learning and answer relational queries.

Specifically, given a goal query as the in-context example that can be interpreted as a question answering (QA) task, LMLP searches or retrieves a related task example with a corresponding logic rule (Figure 2). Then the context and task description are concatenated as the input prompt for an autoregressive planning LM. At each step of generation, we use a masked translation LM to compare the similarity between the generated natural language sentences and encoded it into `(subject, relation, object)` predicates in the KB. In this way, each generated sentence is transformed into the most similar predicate and the reasoning path is confined within the KB. The process is iterated until a predefined maximum iteration or the target of interest is reached (Figure 2) and the generated reasoning path is evaluated manually.

To evaluate the reasoning capability of CoT and LMLP, we curate two datasets and design a series of experiments, aiming to compare two recent in-context learning paradigms and explore both symbolic and naturalistic scenarios. Specifically, we adopt synthetic datasets containing `(natural, symbolic)` data pairs. The symbolic part contains predicates and first-order logic (FOL) rules, which are well-suited for investigating the role of symbolic representations for few-shot reasoning. The natural part of our study includes a story written in natural language that describes a set of entities and relations, as well as the reasoning paths that connect them. These reasoning paths can be seen as explanations for the relationships and events described in the story. Moreover, we create experimental settings that are unfavorable for LMLP since (i) we use GPT-2 and SentenceBERT as its backbones, which is known to be of much smaller scale compared to CoT which is usually based on GPT-3 (Brown et al., 2020) or PaLM (Chowdhery et al., 2022a); (ii) LMs are pre-trained over natural language sentences as opposed to KBs, which creates substantial gaps in semantics and representations, thus posing a grounding challenge where LMs are known to be ineffective (Bisk et al., 2020).

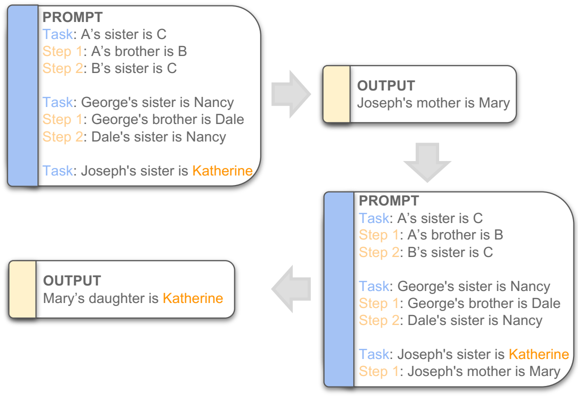


Figure 2: Illustration of a deductive reasoning example and iterative prompting of LMLP. LMLP retrieves a first-order logic rule and an associated grounded example to answer the question. It stops when predefined maximum iterations or the target entity of interest is reached. The reasoning path explains the *sister* concept.

Controlled experiments on relational reasoning have shown that (i) CoT prompting struggles to solve the compositionality challenge (Sinha et al., 2019), while with explicit verification, LMLP can work more reliably as reasoning length increases by taking symbolic inputs that explicitly separate logic and control (Kowalski, 1979). (ii) While it is commonly believed that large pre-trained language models (LMs) are not grounded in contexts that require rich experiences, experimental results suggest that in-context learning, which maps the conceptual structure of a space learned from text onto a new structured space, is sufficient to solve some challenging reasoning tasks over knowledge bases (KBs). (iii) LMs struggle to effectively solve relational reasoning tasks without proper demonstrations containing the target relation and correct input-label mappings. This is supported by evidence in in-context examples, which are poorly understood and have many intricate design choices (Zhao et al., 2021; Liu et al., 2021; Min et al., 2022).

2 Related Works

In-context learning concerns feeding input texts describing a task with some examples to the black-box model for learning the task (Brown et al., 2020). Many works show that there are intricate design choices like prompt formats (Jiang et al., 2020; Liu et al., 2021; Zhao et al., 2021; Min et al., 2022), example choices and their ordering (Zhao et al., 2021; Lu et al., 2021b), pretraining data distribution (Xie et al., 2021; Shin et al., 2022; Chan et al., 2022) and model architectures (Chan et al., 2022) to improve the LMs' powerful and versatile in-context learning ability. Recent work focuses on bootstrapping LM with natural language explanations, intermediate steps, or rationales for reasoning (Camburu et al., 2018; Zhou et al., 2020; Nye et al., 2021; Wei et al., 2022b; Nye et al., 2021; Zelikman et al., 2022). Recent works showcase both some positive (Clark et al., 2021) and negative results (Kassner et al., 2020; Helwe et al., 2021; Talmor et al., 2020) in adapting LMs for symbolic or logical reasoning. The length generalization challenge is echoed in a few recent works (Zhang et al., 2022; Anil et al., 2022; Liu et al., 2022; Zhou et al., 2022b; Press et al., 2022). Though there are some encouraging progress (Clark et al., 2021; Wei et al., 2022b; Chowdhery et al., 2022a; Zelikman et al., 2022), they require a significant amount of computation for re-training and human annotations about reasoning paths or explanations (Wei et al., 2022b; Nye et al., 2021). Moreover, their entangled nature with natural language makes them hard to make robust inferences over symbolic factual knowledge. However, our goal is fundamentally different from theirs in investigating the role of symbolic representations on few-shot reasoning using in-context learning. LMLP that bootstraps the reasoning process from the LMs in a few-shot manner (Figure 2) is in contrast to popular methods that need expensive human annotations and retraining (Camburu et al., 2018; Zhou et al., 2020; Wei et al., 2022b; Zelikman et al., 2022) or uncontrollable using only pre-trained knowledge (Kojima et al., 2022). Moreover, related works typically *finetune* the model using rationales or explanations (Camburu et al., 2018; Zhou et al., 2020) or focus on natural language based reasoning such as commonsense reasoning, arithmetic reasoning, open domain question answering (Wei et al., 2022b), concept grounding (Patel and Pavlick, 2021) etc. Synthetic ontology datasets are constructed in (Saparov and He, 2022) to understand the failure modes of CoT reasoning, but they are in natural language forms instead of investigating the reasoning done over interpretable symbolic structures as we do. Huang et al. (2022) uses a mechanism for constraining the LLM output to feasible action sequences, which we adopt in this work. LMLP can be conceptually understood as a realization of recency biases (Press et al., 2021), which has been shown effective in scratchpad-based reasoning (Liu et al., 2022). Therefore, all the above works are different from our goal of exploring the representations of prompts in-context learning.

Retrieval-augmented Generation. Our study is also related to retrieval-augmented generation (Lewis et al., 2020) like kNN-LM (Khandelwal et al., 2019), DPR (Karpukhin et al., 2020), RALM (Guu et al., 2020), and RETRO (Borgeaud et al., 2022), which integrates parametric models with non-parametric KBs to address key LM challenges like knowledge staleness (Roberts et al., 2020) and hallucination (Shuster et al., 2021), reasoning (Shao et al., 2023). We explore more controllable environments where the evaluation of intermediate reasoning can be automated, demonstrating that this verification process helps filter out incorrect reasoning paths. This, in turn, enhances reasoning performance by assessing how effectively language models can reason when instances of hallucination are minimized.

3 Methodology Overview

We consider the reasoning task with an SRL query as the question and some background knowledge as the context. The relational information in the query and context can be expressed either using natural language or a `(subject, relation, object)` predicate/triplet. There is a KB with facts $\mathcal{F}$ and (FOL) rules $\mathcal{R}$ to support the above QA. There are two equivalent ways for representing the problem, symbolic or natural language, which leads to the designs below.

Datasets construction. To ensure that the natural and symbolic data are equivalent, we keep the ground truth facts the same in natural language stories and knowledge bases. We construct natural language story datasets following the method described in (Sinha et al., 2019). As shown in Table 1, we seek to curate new symbolic datasets from the original ones into (i) *A query subset* containing predicates needed for proving. (ii) *A set of facts* $\mathcal{F}$ containing all the available facts/predicates, which composes a KB, and (iii) *A set of rules* $\mathcal{R}$ containing examples (A task and its proofs) extracted from the training subset using backward chaining based neuro-symbolic reasoners (Rocktäschel and Riedel, 2017). See appendix B.1 for more details.

Dataset	Natural Language Samples	Query	Facts $\mathcal{F}$	Logic rules $\mathcal{R}$
CLUTRR	Task: What's the relationship between Ashley and Nicholas? Story: Ashley told her daughter Lillian to wash up. Dinner was ready. Lillian called her brother, Nicholas up to see how he was doing after surgery.	(Ashley, son, Nicholas)	(Ashley, daughter, Lillian) (Lillian, brother, Nicholas) ...	Task: Ashley's son is Nicholas Step 1: Ashley's daughter is Lillian Step 2: Lillian's brother is Nicholas
Countries	Task: Is palau located in oceania?	(palau, locatedIn, oceania)	(palau, locatedIn, micronesia) ...	Task: palau locatedIn oceania Step 1: palau locatedIn micronesia Step 2: micronesia locatedIn oceania

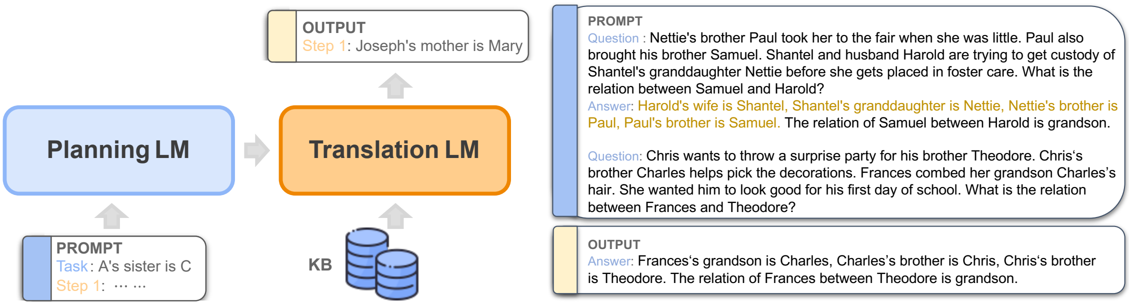
Table 1: Examples of data processing and curation.

Task. Given a query `Task: Joseph's sister is Katherine`, which consists of two entities `Joseph, Katherine` and a target relation `sisiter`. Our task is to find a proof path from `Joseph` to `Katherine` where the relationship `sisiter` can be correctly inferred. On a high level, we need to leverage an abstract logic rule `Sister(A,C) ← Brother(A,B) ∧ Sister(B,C)` and its grounded example `Sister(George, Nancy) ← Brother(George, Dale) ∧ Sister(Dale, Nancy)` to derive the answer for the query `Sister(Joseph, Katherine)` (Figure 3(a)).

Language Models as Logic Programmers achieves this goal using in-context learning. At first, examples and logic rules r in $\mathcal{R}$ are selected. For example, in Figure 2, LMLP samples one logic rule and its grounded example, which is concatenated with the query q `Task: Joseph's sister is Katherine` as a prompt $r' = [r, q]$. The prompt is fed into a **Planning LM** $\mathcal{P}_\theta$, which is an autoregressive LM such as GPT-3 for proof generation. Multiple sentences x are generated using temperature sampling from $\mathcal{P}_\theta(r')$. However, these sentences are in free-form language and often not in the `(subject, relation, object)` predicate format. In LMLP, the generated output is converted to the most similar fact in KB $\mathcal{F}$ using the cosine similarity of the embedding from a **Translation LM** $\mathcal{T}_\phi$, implemented as a sentence-specific Masked LM. Specifically, $\mathcal{T}_\phi$ embed the output sentence from $\mathcal{P}_\theta$: $\mathcal{T}_\phi(x)$ and all predicates f from $\mathcal{F}$: $\mathcal{T}_\phi(f)$, calculating their cosine similarity. The most similar f to x is chosen as the conversion results f'. By translating the output space of $\mathcal{P}_\theta$ into an external KB this way, LMLP is expected to produce a more plausible provenance to explain the reasoning process of a final prediction. Given frozen $\mathcal{P}_\theta$ and $\mathcal{T}_\phi$, we then repeatedly generate proofs by prompting $\mathcal{P}_\theta$ using $r' = [r', f']$, projecting the generated sentences to the KB by the $\mathcal{T}_\phi$, attaching the output to the prompt (Figure 2). The model terminates when the predefined maximum number of iterations or the target entity of interest is reached. To improve coherency, we enforce the chain rule transition constraints: the tail entity of the previous predicate should be the same as the head entity of the next predicate for each output step. Specifically, during the translation phase, we only select the predicates satisfying the requirement to compare similarity with $\mathcal{T}_\phi(x)$. The faithfulness of the reasoning path is governed by post-hoc human evaluations. The overall algorithm is described in Algorithm 1 in Appendix B. Using the prompt supported by the KBs, we bootstrap the reasoning process from the LMs in a few-shot manner (Figure 2).

Chain-of-Thought prompting. CoT (Wei et al., 2022b) solves complicated multi-step reasoning tasks by providing explanations, which is also intuitive for our multi-hop SRL tasks since we can take intermediate reasoning paths as explanations.

Figure 3(b) shows an example of applying CoT to solve an SRL task from the CLUTRR dataset (Sinha et al., 2019): given an in-context sample in the form of `(input, explanation, output)`. LMs are expected to imitate the reasoning process of the given explanation to generalize to a new query. The explanation of each question is generated just the same as the rule set $\mathcal{R}$, which is extracted from the training set using a neuro-symbolic reasoners and converted to natural language forms. Specifically, the in-context exemplar adapts LMs to another sample containing multiple relations and a query for the relation between two entities *"What is the relation between Theodore and Frances?"*, CoT first generates a reasoning path from *Frances* to *Theodore*, namely *"France's grandson is Charles, . . . , Chris's brother is Theodore."*, and finally answers the query: *"The relation of Frances between Theodore is grandson"*. With such a prompt, LMs are expected to generate both the reasoning paths and the resulting queried relation. For a fair comparison with LMLP, human judgments on the reasoning path are included to calculate the accuracy. Note that the explanation in CoT is extracted from the story in the question, which contains much clearer information than the logic rules for LMLP.



(a) LMLP (b) CoT

Figure 3: Schematic overview of (a) LMLP and (b) CoT.

4 Experiments

We now describe the experimental setups, empirically evaluate LMLP and compare it with existing methods. See Appendix C for full details of data preprocessing and performance evaluation.

Settings. We curate two datasets for evaluating the in-context learning capability of LMs for reasoning: CLUTRR-LP and Countries-LP, which are based on CLUTRR (Sinha et al., 2019) and Countries (Bouchard et al., 2015) datasets respectively. CLUTRR (Sinha et al., 2019) contains a group of KBs, where each node denotes a family member and edges are family relations. The target of CLUTRR dataset is to infer a two-family members' relationship that is not explicitly mentioned. The training set of CLUTRR consists of graphs that the target relation can be inferred by traversing a limited number of edges while the relation in the test set needs more traversing steps for inference, which allows controlled studies on compositionality. Another intriguing property of CLUTRR is that there are ground truth **one-to-one correspondances** between KBs and natural language stories, which exactly suits our needs. Countries (Bouchard et al., 2015) concerns link prediction, where countries, regions, and sub-regions are entities and relations containing *LocatedIn* and *NeighborOf*. Countries has three tasks, *R1*,*R2*, and *R3*, each requiring reasoning skills of increasing complexity (Rocktäschel and Riedel, 2017).

Implementation details. For LMLP, we implement the planning LM $\mathcal{P}_\theta$ as GPT-2 (Radford et al., 2019), the translation LM $\mathcal{T}_\phi$ as Sentence BERT (Sent-BERT) (Reimers and Gurevych, 2019) based on Hugging Face Transformers (Wolf et al., 2019). The default model for Translation LM is Sentence-RoBERTa-Large and for Planning LM is GPT2-Large (Radford et al., 2019) pretrained on large corpora by default. For CoT, we follow the original paper (Wei et al., 2022b) to sample in-context samples. We conduct all the experiments on a machine with four Nvidia TITAN XP (10GB) GPU cards.

Since prompt formats lead to significant performance variations (Liu et al., 2021), we propose to explore two simple design choices for LMLP and find that they can further boost the reasoning capacity. (i) Multiple examples for prompting. Denote N the number of examples we used in one proof task. Table 9 shows two examples with $N = 1$ and $N = 2$ are supplied respectively. The intuition is that, getting more examples in the prompt can make LMs better recognize the proof task and thus produce more reliable reasoning paths. See the experimental section for empirical verification. (ii) Prompts Ensembling. Table 10 shows the results of different prompts for the same task. We can see the influence of prompts on the generated proof path. The first few proof steps are largely similar to the provided example. If the provided example supplies a wrong direction, the proof is likely to be wrong. To study and exploit the benefit brought by different prompts, during experiments, we propose to use K prompts alternatively for one task, where one task is marked to be successfully proved if any of these K prompts gets the right result. Namely, a larger K means that we have a higher probability of picking a good prompt. The default hyper-parameters N, K are set to one.

Evaluation metrics In Table 2 and Table 3, where LMLP is compared to various baselines, the correctness of the proven reasoning path is evaluated manually. For each reasoning path, we ask annotators to answer "Yes" or "No" to whether the generated proof path is plausible to human commonsense and the target relation can be induced from it. We include 5 participants to reduce randomness and observe that their answers are almost the same. Because of resource Limitations, for other simple ablation studies of LMLP, the metric is **proven accuracy or success rate**. For example, for query "Task: palau locatedIn oceania", we begin with entity "palau" and select facts from the $\mathcal{F}$. If the chosen triplet ends with entity "oceania", the proven path is correct, e.g., "micronesia locatedIn oceania" in Table 1. For LMLP, if there is no chosen triplet ends with entity "oceania", the prediction is incorrect.

4.1 Comparisons of LMLP and CoT

The goal of this part is to systematically compare LMLP with CoT both quantitatively and qualitatively on SRL tasks to better understand the reasoning of LMs using in-context learning.

In Figure 1 and Table. 4, we compare LMLP to CoT and the reported performances are all human evaluation results. Qualitatively, CoT can get positive results on some query examples, for example, in Table 12, we showcase two examples where CoT can generate a correct proof path and predict the target relation at the same time. However, compared to LMLP, CoT achieves inferior results in all query sets with test reasoning length $5, 6, 7, 8, 9, 10$ with different LLMs for text generation. In addition, as the reasoning length increases, the performance of CoT shows a clear downward trend. Table 12 shows two negative examples where the story contains sophisticated relations and the model cannot get the right reasoning path or just generate a wrong relation. In contrast, LMLP can consistently achieves a high human evaluation score (Table 2), which again verifies the systematic generalization capability of LMLP. Table 7 in the appendix shows examples with the same task but processed by the two methods respectively, where CoT cannot get deduce a right relation path from *Margaret* to *Charles* but LMLP can extract a simple yet right relation path. The reason why LMLP is better than CoT can be that, although CoT decomposes complex multi-hop relation reasoning tasks into a multi-step reasoning process and then predict the final results, the proof path is all generated by LMs at once. The decomposition of LMLP to multi-hop reasoning tasks is more thorough, where the generation of a proof path is divided into multi-steps and each step will be projected into the KB, which is a much stronger inductive bias. Therefore, the decomposed tasks in each step are easier to solve and the knowledge in the KB can be well exploited. See appendix for results on Countries-LP.

4.2 Analysis of LMLP

Given the above observations that LMLP outperforms CoT by a large margin, we systematically analyze LMLP with extensive experiments below.

Ablation Studies on prompting strategies. As illustrated in Table 2, **No Prompt** means that we only feed the target directly and generate each step, prompts in the **Only Rule** baseline is one proof example with entities replaced by some symbols. We also compare LMLP to Language Planner (Huang et al., 2022), which first finds the most similar target in the $\mathcal{R}$ and uses such an example as the prompt. **LMLP-reverse** swaps the position of the abstract logic rule and its grounded example in the prompt of LMLP. For example, in Figure 2, the in-context prompt of LMLP-reverse will place `Sister(George, Nancy) ← Brother(George, Dale) ∧ Sister(Dale, Nancy)` before its abstract logic rule `Sister(A,C) ← Brother(A,B) ∧ Sister(B,C)`. Examples for all baselines are shown in Appendix Table 9.

Table 2 shows that directly applying Language Planner for relational reasoning does not work and using only facts or no prompt attain inferior performance. The possible reason for the inferior performance of Planner can be that it finds the example from $\mathcal{R}$ with the most similar task as the prompt, which usually retrieves rules with the same entities of the goal task. However, for reasoning tasks over KBs, relation contains much more information of the task than the entity. As shown in Table 9, for the task "Patricia's uncle is Donald", Planner finds the example with task "David's nephew is Don", whose following proofs do not make sense for the relation "uncle". LMLP in contrast finds an example whose task has the same relation as the goal predicate, which is more informative.

LMLP can be robust to large search space. We may wonder if the superior results of LMLP are an artifact for datasets with a small search space. To control the confounding, we progressively inject $5,000$ random noisy facts/predicates into the facts set $\mathcal{F}$. With more noisy facts, at each decoding step, it will be more difficult for LMLP to choose the correct proof path as the search space is enlarged. Figure 4(b) shows the results when we vary the number of noisy facts, where the noisy rate is 0.5

Test Story Length	Baseline		Ablation				Ours	
	Planner	**CoT**	**No Prompt**	**Only Rule**	**Random**	**Entity-based**	**LMLP-reverse**	**LMLP**
5	0.0973	0.173	0.1514	0.1622	0.2919	0.2000	**0.3730**	0.3297
6	0.1810	0.1365	0.1238	0.1524	0.2095	0.1429	**0.3048**	0.2476
7	0.2258	0.1032	0.2000	0.2129	0.2323	0.1742	**0.3742**	0.2581
8	0.1037	0.1506	0.2222	0.2000	0.3111	0.2370	**0.3556**	**0.3556**
9	0.1048	0.0914	0.1935	0.2177	0.1613	0.1855	**0.3548**	0.2984
10	0.1230	0.123	0.2869	0.2131	0.3934	0.2705	0.**5246**	0.4754
Average	0.1393	0.1296	0.1963	0.1931	0.2666	0.2017	**0.3812**	0.3275

Table 2: Numerical results and ablation on the length of test samples on CLUTRR-LP.

Tasks	Baseline		Ablation				Ours	
	Planner	**CoT**	**No Prompt**	**Only Rule**	**Random**	**Entity-based**	**LMLP-reverse**	**LMLP**
S1	0.7500	0.3333	0.8542	0.7708	0.6042	**0.8958**	0.8333	0.7917
S2	0.7917	0.3750	0.6667	0.4583	0.6750	0.7500	**0.8333**	0.6250
S3	0.7500	0.2500	0.7292	0.7083	0.6458	0.6667	0.7500	**0.8333**
Average	0.7639	0.3194	0.7500	0.6458	0.6417	0.7708	**0.8055**	0.7500

Table 3: Human evaluation results in various settings of Countries-LP. *S1, S2, S3* (Minervini et al., 2020) are three different tasks with different $\mathcal{F}$ (see the experimental setting for details).

Test Story Length	GPT-2		Mistral-7B-v0.1		LLaMA2-7B	
	CoT	**LMLP**	**CoT**	**LMLP**	**CoT**	**LMLP**
5	0.1730	0.3297	0.3083	0.5032	0.2721	0.4823
6	0.1365	0.2476	0.2762	0.5182	0.2543	0.4872
7	0.1032	0.2581	0.2314	0.4732	0.2364	0.4715
8	0.1506	0.3556	0.2247	0.5181	0.2102	0.5323
9	0.0914	0.2984	0.1143	0.4723	0.1345	0.4021
10	0.1230	0.4754	0.1220	0.4741	0.1305	0.4992
Average	0.1296	**0.3275**	0.2128	**0.4932**	0.2063	**0.4791**

Table 4: Numerical results considering different backbone models.

	K=1	K=3	K=5	K=10	A Long Example
S1	0.7083	0.9583	1.0000	1.0000	Task: A locatedIn C Step 1: A neighborOf B Step 2: B locatedIn C
S2	0.5000	0.8750	0.9583	1.0000	Task: uruguay locatedIn south_america Step 1: uruguay neighborOf argentina Step 2: argentina locatedIn south_america
S3	0.7500	0.9167	0.9167	1.0000	Task: sudan locatedIn africa Step 1: sudan neighborOf central african republic Step 2: central african republic neighborOf chad Step 3: chad neighborOf south sudan Step 4: south sudan neighborOf dr congo Step 5: dr congo neighborOf republic of the congo Step 6: republic of the congo locatedIn middle africa Step 7: middle africa locatedIn africa

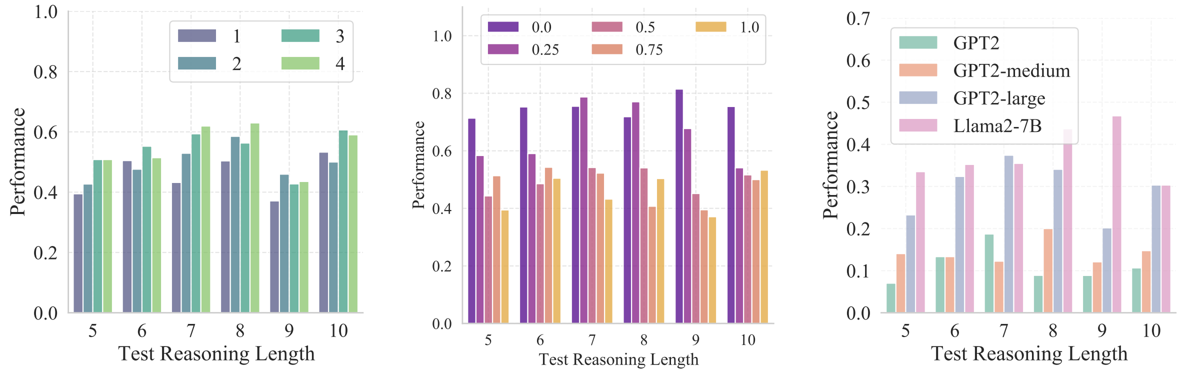
Table 5: Results of LMLP on Countries-LP. *S1, S2, S3* (Minervini et al., 2020) are three different tasks with different $\mathcal{F}$ (see the experimental setting for details).

means that we add $5000 * 0.5$ random facts to the $\mathcal{F}$ during evaluation and noisy rate 0 means $\mathcal{F}$ only contains query-relevant facts. We see that enlarging the search space generally decreases the performance. However, even though when all the noisy facts are injected into $\mathcal{F}$, i.e. more than 95% facts are noisy, the performance is still favorable (more than 38% success rate), showing that LMLP can produce robust reasoning performance.

Effects of model size. Figure 4(c) shows the impact of the size of the planning LM model: larger GPT models generally attain better performance; using GPT2-large and LlaMA2-7B (Touvron et al., 2023) can dramatically improve model performance, which aligns with the findings that reasoning performance can emerge in larger models (Wei et al., 2022a; Saparov and He, 2022).

Prompts ensembling boosts the reasoning capability. For each test example, we sample K in-context examples and count as correct if any one of them can solve the task. We show the evaluation results on CLUTRR-LP in Table 6 and the proposed method can generate realistic and correct proof paths. A large K can further boost performance, which also verifies the importance of prompt ensembling: Table 5 shows the performance on Countries-LP where almost all the query samples can be proved correctly with a large K. One interesting phenomenon is that LMLP can generate a much longer proof path even though the proof path length in the rule set $\mathcal{R}$ is less than 3. This manifests a potential improvement with respect to the significant weakness in systematic generalization of fine-tuning or re-training of LMs (Sinha et al., 2019). The $\mathcal{R}$ of CLUTRR-LP contains only examples whose proof paths are less than five. However, during testing, our model can produce proof paths much longer than five steps and perform well on all query sets.

Prompting using multiple examples boosts the



(a) (b) (c)

Figure 4: (a) Effect of the number of templates for LMLP on CLUTRR-LP. (b) The effects of noisy facts for LMLP on CLUTRR-LP. Ablation on the scaling of (c) Planning LMs.

Test Reasoning Length	K=1	K=3	K=5	K=10	Avg
5 Hops	0.3946	0.6865	0.7838	1.0000	0.7162
6 Hops	0.5048	0.7143	0.7619	1.0000	0.7452
7 Hops	0.4323	0.8065	0.8774	1.0000	0.7790
8 Hops	0.5037	0.8000	0.8593	1.0000	0.7907
9 Hops	0.3710	0.6452	0.7500	1.0000	0.6915
10 Hops	0.5328	0.8279	0.8525	0.9180	0.7828

Table 6: Ablation of LMLP on CLUTRR-LP.

reasoning capability. N denotes the number of in-context examples used in one proof task. Results show that a larger N can generally produce performance gains (Figure 4(a)). However, longer prompts require larger GPU memories, so there is a trade-off between memory and performance.

4.3 Analysis of Demonstrations of ICL

Besides results in Appendix Table 7, we conduct qualitative analysis of demonstrations of in-context learning.

Failure cases analysis of baselines. Since the generated sentences are closely related to the prompt, Table 11 in Appendix shows that if we randomly choose prompts, the generated proof path has relations similar to the prompt, but is wrong for the given task. For entity-based prompts, since the task has the same start entity as the in-context exemplar, the generated steps 1 in this setting are very similar, leading to many wrong proof paths. Language Planner, without chain rule constraint, the generated triplets are chaos, e.g., in Example 1, the generated proof does even not contain the subject "Jon" and thus exactly wrong. Although the proposed LMLP attains a high success rate, there are also some failure cases. As shown in Appendix Table 10, an appropriate prompt needs to be chosen for the right proof paths.

Takeaways. Similar to previous work (Liu et al., 2021; Min et al., 2022), we find that in-context learning performance varies greatly with choices of exemplars (Table 6). One of the key findings in (Min et al., 2022) is that even without any labeled data, LMs can achieve k-shot performance by simply prompting with demonstrations containing unlabeled inputs. Our findings are generally in-line is in line with the importance of input-label formats highlighted in the work. However, we show in Table 8 and 9 that the correct mapping of rule-example pairs is important since giving only rules with symbols like X, Y, Z rather than concrete entities like $China$ makes LMLP fail catastrophically.

5 Concluding Remarks

In this study, we systematically examine in-context learning of language models (LMs) from a symbolic reasoning perspective, demonstrating that LMs can be prompted with logical demonstrations to generate plausible explanations for reasoning tasks over knowledge bases (KBs). Our evaluation results show that constraining outputs of LMs and ensuring intermediate reasoning correctness are important for reasoning performance, providing new insights into in-context learning and a mechanism to reduce incorrect reasoning through symbolic verification.

Limitations

Like previous works, we study reasoning empirically without theoretical justifications and focus specifically on synthetic data. Therefore, our results serve as a proof of concept on investigating how ensuring and reducing hallucination can improve overall reasoning, and might not transfer to more complex reasoning tasks. Moreover, due to access and computation restrictions, we are not able to conduct experiments with the latest LMs like PaLM (Chowdhery et al., 2022b).

Acknowledgement

This research has been generously funded by NSF IIS1955532, NSF HM04762010002, NIH R01GM140467.

References

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. *NeurIPS*.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2018. Systematic generalization: What is required and can it be learned? In *ICLR*.

Emily M Bender and Alexander Koller. 2020. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *ACL*.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, et al. 2020. Experience grounds language. In *EMNLP*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Guillaume Bouchard, Sameer Singh, and Theo Trouillon. 2015. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI*.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *AAAI*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *NeurIPS*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *NeurIPS*.

Stephanie CY Chan, Adam Santoro, Andrew K Lampinen, Jane X Wang, Aaditya Singh, Pierre H Richemond, Jay McClelland, and Felix Hill. 2022. Data distributional properties drive emergent few-shot learning in transformers. *arXiv preprint arXiv:2205.05055*.

Swarat Chaudhuri, Kevin Ellis, Oleksandr Polozov, Rishabh Singh, Armando Solar-Lezama, Yisong Yue, et al. 2021. *Neurosymbolic Programming*. Now Publishers.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022a. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022b. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *IJCAI*.

Adam Dahlgren, Johanna Björklund, and Frank Drewes. 2021. Perception, memory, and inference: The trinity of machine learning. In *Is Neuro-Symbolic SOTA still a myth for Natural Language Inference? The first workshop*.

Arthur M Glenberg and Michael P Kaschak. 2002. Grounding language in action. *Psychonomic bulletin & review*.

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

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Chadi Helwe, Chloé Clavel, and Fabian M. Suchanek. 2021. Reasoning with transformer-based models: Deep learning, but shallow reasoning. In *AKBC*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *TACL*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? *arXiv preprint arXiv:2006.10413*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. In *ICLR*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners.

Robert Kowalski. 1979. Algorithm= logic+ control. *Communications of the ACM*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Shuang Li, Xavier Puig, Yilun Du, Clinton Wang, Ekin Akyurek, Antonio Torralba, Jacob Andreas, and Igor Mordatch. 2022. Pre-trained language models for interactive decision-making. *arXiv preprint arXiv:2202.01771*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*.

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

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. 2021a. Pretrained transformers as universal computation engines. *arXiv preprint arXiv:2103.05247*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. 2022. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. *NAACL*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021b. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *PNAS*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Pasquale Minervini, Matko Bosnjak, Tim Rocktäschel, and Sebastian Riedel. 2018. Towards neural theorem proving at scale. *arXiv preprint arXiv:1807.08204*.

Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. 2020. Learning reasoning strategies in end-to-end differentiable proving. In *ICML*.

Stephen Muggleton and Luc De Raedt. 1994. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Roma Patel and Ellie Pavlick. 2021. Mapping language models to grounded conceptual spaces. In *ICLR*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *EMNLP-IJCNLP*.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2020. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *ICLR*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *EMNLP*.

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

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. Clutrr: A diagnostic benchmark for inductive reasoning from text. *EMNLP*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics-on what language model pre-training captures. *TACL*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *TMLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yuhuai Wu, Markus N Rabe, Wenda Li, Jimmy Ba, Roger B Grosse, and Christian Szegedy. 2021. Lime: Learning inductive bias for primitives of mathematical reasoning. In *ICML*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *ICLR*.

Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. *NeurIPS*.

Yuan Yang and Le Song. 2020. Learn to explain efficiently via neural logic inductive learning. In *ICLR*.

Eric Zelikman, Yuhuai Wu, and Noah D Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*.

Hanlin Zhang, Ziyang Li, Jiani Huang, Mayur Naik, and Eric Xing. 2022. Improved logical reasoning of language models via differentiable symbolic programming. In *First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022a. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Hattie Zhou, Azade Nova, Hugo Larochelle, Aaron Courville, Behnam Neyshabur, and Hanie Sedghi. 2022b. Teaching algorithmic reasoning via in-context learning. *arXiv preprint arXiv:2211.09066*.

Wangchunshu Zhou, Jinyi Hu, Hanlin Zhang, Xiaodan Liang, Maosong Sun, Chenyan Xiong, and Jian Tang. 2020. Towards interpretable natural language understanding with explanations as latent variables. *NeurIPS*.

Appendix

A Extended Related Work

Neuro-Symbolic Reasoning. ILP (Muggleton and De Raedt, 1994) and its neural version (Yang and Song, 2020) are unable to reason about disjoint relations in confront of missing links when KBs are noisy like in FreeBase, which means ILP only synthesizes rules based on existing relations. Methods like Neural-LP (Yang et al., 2017) and RNNLogic (Qu et al., 2020) require enumeration of all possible rules given a max rule length T. Thus the complexity of these models grows exponentially as maximum rule length increases, which is a significant disadvantage for systematicity problems. For deductive reasoning, NTP (Rocktäschel and Riedel, 2017) and its improved versions (Minervini et al., 2018, 2020) require hand-crafted templates to imitate backward chaining for deductive reasoning. This belies the considerable user burden of authoring the templates which then fundamentally biases the tool toward a specific subset of programs that the author has in mind. Moreover, the performance and efficiency of NTP is far from satisfactory: the performance usually lags far behind its neural counterparts like knowledge graph embedding methods (Lin et al., 2015); during both training and inference, NTPs need to compute all possible proof trees needed for proving a query, relying on the continuous unification of the query with all the rules and facts in the KB. The search space of existing works is exponentially large, which makes them hard to scale up in general (Minervini et al., 2018; Chaudhuri et al., 2021).

LMs for Theorem Proving. Most works focus on proving formal mathematical theorems: GPT-f (Polu and Sutskever, 2020) shows promising results by generative language modeling over mathematical formulas. Systematicity of LMs when training on proofs is evaluated in (Gontier et al., 2020) but shows negative results in generalizing to unseen proof steps in extrapolation and complex language. Three synthetic tasks inspired by three reasoning primitives of deduction, induction, and abduction are demonstrated in (Wu et al., 2021). The above works provide insights into understanding LMs' reasoning capabilities. Though they share similar problem structures like compositionality with ours, they fundamentally require large-scale pre-training and fine-tuning due to the mismatch between Wikipedia pre-training corpora and mathematical formulas. Such a re-training requirement not only results in computational inefficiency but lacking in compositional generalization to longer proof steps unseen during training (Gontier et al., 2020).

Symbolic Reasoning with LMs. Large LMs pre-trained on open-domain text corpora have achieved impressive advances in natural language generation and understanding tasks (Kenton and Toutanova, 2019; Brown et al., 2020). By self-supervised imitation on human-generated texts, LMs contain rich factual knowledge (Petroni et al., 2019; Bouraoui et al., 2020; Roberts et al., 2020) and linguistic structures (Manning et al., 2020), serving as a versatile inference regime for various downstream tasks (Brown et al., 2020; Lu et al., 2021a). Among them, GPT-3 stands out by its few-shot generalization to unseen cases without further fine-tuning given in-context samples as demonstrations (Brown et al., 2020). Constraint decoding is shown to be effective in incorporating logical constraints into natural language generation (Lu et al., 2022). However, it is a common belief that LMs have not yet enjoyed a comparable success in tasks that require extensive planning and grounding (Glenberg and Kaschak, 2002; Bender and Koller, 2020; Bisk et al., 2020) as well as symbolic reasoning (Kassner et al., 2020; Helwe et al., 2021; Razeghi et al., 2022).

B Algorithm Description

Algorithm 1 describes the procedure or LMLP. It can also be illustrated in Figure 3(a).

B.1 Data Generation.

CLUTRR-LP. CLUTRR has 9 subsets with difference story length, named $l_2, l_3, \ldots, l_{10}$. Following (Minervini et al., 2020), we convert l_2, l_3, l_4 to the $\mathcal{R}$ and use $l_5, \ldots, l_{10}$ to the **query sets**. As illustrated in Table. 1, data samples in CLUTRR consist of a story and a target, where the target contains two entities and the relation that is needed to be inferred, the story contains available triplets. Each sample in the l_2, l_3, l_4 will be converted to the format "Task: ..., Step i: ..." and added to the $\mathcal{R}$. Note that all examples in the $\mathcal{R}$ have a story

Algorithm 1 Generate proof path from Pre-Trained Language Models.

Require: Planning LM $\mathcal{P}_\theta$, Translation LM $\mathcal{T}_\phi$, Query set $\mathcal{Q}$ that contains all query triplets, $\mathcal{F}$ that contains all available facts, $\mathcal{R}$ that contains all the available logic rules or proof examples.

for $q = (s, p, o) \in \mathcal{Q}$ **do** *// s, p, o denote subject entity, predicate (relation) and object entity respectively.*
  Find $r \in \mathcal{R}$, whose task relation is p.
  Construct prompt $r' = [r, q]$. *// $[r, q]$ means the concatenation of two strings.*
  while Max step is not reached **do**
    Sample 10 sentences $\{x_i\}_{i=1}^{10}$ from $\mathcal{P}_\theta(f')$.
    Set $\mathcal{F}' \in \mathcal{F}$ whose first entity are s.
    if $|\mathcal{F}'| == 0$ **then**
      Break *// No available facts in the $\mathcal{F}$ start with entity s.*
    for $x \in \{x_i\}_{i=1}^{10}$ **do**
      $score_i = \max_{\forall r \in \mathcal{F}'} cosine(\mathcal{T}_\phi(x), \mathcal{T}_\phi(r))$;*// Cosine similarities of s to facts in $\mathcal{F}'$.*
      $idx = \arg\max_{\forall r \in \mathcal{F}'} cosine(\mathcal{T}_\phi(x), \mathcal{T}_\phi(r))$;*// Select $r \in \mathcal{F}'$ with the highest similarity to x.*
      $x' = \mathcal{F}'[idx]$
    Choose the highest score rule x^* as the next proof step and append it to the prompt $f' = [f', x^*]$.
    if $o' == o$ **then**
      Break *// The object entity converges to the target entity o.*

length of less than five, which enables us to test the systematic generalization ability of LMLP. For CLUTRR, the story triplets in the $\mathcal{R}$ are not useful for test target proving, because they are all from different relation graphs. For example, story triplets in the l_2, l_3, l_4 contain "(William's brother is Steve)" while one test story on l_5 contains "(William's uncle is Steve)". During the evaluation, if the model chooses "(William's brother is Steve)", the proof path will be wrong. However, the similarity of these two triplets is high, the model is then easy to make errors and these noisy facts increase proof difficulties. We hence evaluate our methods in two settings considering the number of noisy facts. The simplest setting (**Test Facts Setting**) is that, when queries are from $l_i, i \in [5, \dots, 10]$, the $\mathcal{F}$ only contains facts in l_i. In this case, the $\mathcal{F}_{5\sim10}$ have 251,222,275,279,285,304 facts respectively. The most difficult setting is termed **All Facts Setting**. We first extract facts in the $\mathcal{F}$ with length l_2, l_3, l_4 and get totally $5,210$ facts. When queries are from $l_i, i \in [5, \dots, 10]$, the $\mathcal{F}$ contains triplets in l_i, l_2, l_3, l_4, where the additional $5,210$ facts are not useful for the proof path and are noisy facts. The All Facts Setting is set as our default setting and experimental results of the Test Facts Setting are mainly in the Appendix. For CoT, the $\mathcal{F}$ is needless and the construction of prompt examples is slightly different from the procedure above. Specifically, as shown in Figure 3(b), for each target in the training samples, we need to preserve the story and extract a proof path for the target.

Countries-LP. Training samples in Countries are triplets that describe the *neighbor of* relation or *located in* relation of two regions/subregions/countries and can thus be directly used as $\mathcal{F}$. Because the three tasks (*S1, S2, S3*) (Minervini et al., 2020) have different training sets and thus have different $\mathcal{F}$. Test samples in Countries are also triplets with specific entities and relations, hence the **query set** is just the test set of the original Countries dataset. One main difficulty in applying the proposed method to Countries is the lack of off-the-shelf proof paths ($\mathcal{R}$). The CTP (Minervini et al., 2020) model is trained and used for proving each triplet in the training set. CTP returns the scores of the possible proof path and the proof with the maximum score is iteratively searched as added into the $\mathcal{R}$. After that, $924, 906, 705$ available examples are found for *S1, S2, S3* tasks respectively.

C Additional Experimental Setups and Results

Human evaluation results on the Countries-LP benchmark. Because Countries-LPis a simple reasoning benchmark, which has only two predicates, namely `LocatedIN`, and `NeighborOf`. Table. 3 shows that simple baselines with entity-based prompts or no prompt can attain comparable or higher results than LMLPon the simplest task $S1$. However, LMLPattains better reasoning capability

CoT Prompting	LMLP
Question: Wilhelmina took her uncle Hugh to the grocery store. Francisco and his brother Wesley were wrestling. Wilhelmina, Francisco's daughter, was cheering on the competition. What is the relation between Hugh and Wesley? Answer: Wesley's brother is Francisco, Francisco's daughter is Wilhelmina, Wilhelmina's uncle is Hugh. The relation of Hugh between Wesley is brother. Question: Constance went shoe shopping with her sister Ellen. Elsie had a daughter named Constance. Elsie had picked her daughter Margaret out the cutest new dress to wear on her birthday. Charles and his sister Kathleen have been best friends ever since childhood. Nadia and her father, James, went to the marina. James's daughter, Mabel, had purchased a boat, and they were eager to see it. Mabel bought her mother, Ellen, a puppy for her birthday. James hung his son Charles's finger paintings on the refrigerator. The paintings were right next to the paintings of Nadia, Charles's sister. Kathleen wasn't old enough to make any paintings for her father, James. What is the relation between Margaret and Charles? Answer: charles' mother is nancy, nancy's daughter is elizabeth, elizabeth's husband is john, john's wife is mary, mary's brother is george, david's nephews are william, william's nephews are robert.✗	Task: A's aunt is B Step 1: A's sister is C Step 2: C's mother is D Step 3: D's sister is B Task: Carrie's aunt is Lynn Step 1: Carrie's sister is Natasha Step 2: Natasha's mother is Cindy Step 3: Cindy's sister is Lynn Task: Charles's aunt is Margaret Step 1: Charles's sister is Nadia Step 2: Nadia's aunt is Margaret ✓

Table 7: Qualitative comparison of CoT and LMLP over the same example on CLUTRR-LP.

on harder tasks. Besides, All baselines outperform COT by a large margin, which verifies the importance of generating a reasoning path step by step than proofing the query predicate at once.

Method	Prompt	Output
Facts w/o Provenance	Task: Kazakhstan locatedIn Asia	Step 1: Kazakhstan locatedIn CentralAsia Step 2: CentralAsia locatedIn Asia
Facts w/ Provenance	Task: Kazakhstan locatedIn CentralAsia Step 1: Kazakhstan neighborOf Uzbekistan Step 2: Uzbekistan locatedIn CentralAsia Task: Kazakhstan locatedIn Asia	Step 1: Kazakhstan neighborOf China Step 2: China neighborOf India Step 3: India locatedIn SouthernAsia Step 4: SouthernAsia locatedIn Asia
LMLP (N=1)	Task: X is located in Z Step 1: X is located in Y Step 2: Y is located in M Step 3: M is located in Z Task: Kazakhstan locatedIn Asia	Step 1: Kazakhstan neighborOf kyrgyzstan Step 2: Kyrgyzstan neighborOf Uzbekistan Step 3: Uzbekistan neighborOf Afghanistan Step 4: Afghanistan neighborOf china Step 5: China neighborOf India Step 6: India neighborOf Nepal Step 7: Nepal locatedIn Asia
	Task: X locatedIn Y Step 1: X neighborOf M Step 2: M neighborOf N Step 3: N locatedIn Y Task: Kazakhstan locatedIn Asia	Step 1: Kazakhstan locatedIn CentralAsia Step 2: CentralAsia locatedIn Asia
	Task: X locatedIn Y Step 1: X neighborOf M Step 2: M locatedIn N Step 3: N locatedIn Y Task: Kazakhstan locatedIn Asia	Step 1: Kazakhstan locatedIn CentralAsia Step 2: CentralAsia locatedIn Asia
	Task: X locatedIn Y Step 1: X neighborOf Z Step 2: Z locatedIn Y Task: Kazakhstan locatedIn Asia	Step 1: Kazakhstan neighborOf kyrgyzstan Step 2: Kyrgyzstan neighborOf Uzbekistan Step 3: Uzbekistan neighborOf Afghanistan Step 4: Afghanistan neighborOf Tajikistan Step 5: Tajikistan neighborOf China Step 6: China neighborOf India Step 7: India neighborOf Nepal Step 8: Nepal locatedIn Asia

Table 8: Sensitivity analysis of prompt formats using LMLP in the Countries dataset (Rocktäschel and Riedel, 2017).

Method	Prompt	Output
No Prompt	Task: Richard's father is Donald	Step 1: Richard's grandmother is Elizabeth Step 2: Elizabeth's son is Donald
Random Prompt	Task: C's father is B Step 1: C's brother is A Step 2: A's father is B Task: Janet's father is George Step 1: Janet's brother is John Step 2: John's father is George Task: Pat's grandfather is Jose	Step 1: Pat's uncle is Samuel Step 2: Samuel's father is Jose
Rule Only	Task: B's father is C Step 1: B's mother is D Step 2: D's son is A Step 3: A's father is C Task: Richard's father is Donald	Step 1: Richard's grandmother is Elizabeth Step 2: Elizabeth's son is Donald
Entity-based Prompt	Task: C's brother is A Step 1: C's brother is B Step 2: B's brother is A Task: Milton's brother is Glen Step 1: Milton's brother is Michael Step 2: Michael's brother is Glen Task: Milton's mother is Antonia	Step 1: Milton's daughter is Margaretta Step 2: Margaretta's grandmother is Antonia
Language Planner	Task: David's nephew is Don ✗ Step 1: David's mother is Frances Step 2: Frances's daughter is Lynn Step 3: Lynn's son is Don Task: Patricia's uncle is Donald	Step 1: Richard's father is Donald
LMLP (N=1)	Task: B's father is C Step 1: B's mother is D Step 2: D's daughter is A Step 3: A's father is C Task: Bobby's father is Hugh Step 1: Bobby's mother is David Step 2: David's daughter is Marie Step 3: Marie's father is Hugh Task: Richard's father is Donald	Step 1: Richard's mother is Maria Step 2: Maria's husband is Donald
LMLP (N=2)	Task: E's father is B Step 1: E's daughter is A Step 2: A's sister is D Step 3: D's brother is C Step 4: C's grandfather is B Task: Louis's father is Harold Step 1: Louis's daughter is Jennifer Step 2: Jennifer's sister is Nettie Step 3: Nettie's brother is Samuel Step 4: Samuel's grandfather is Harold Task: D's father is C Step 1: D's husband is A Step 2: A's son is B Step 3: B's grandfather is C Task: Lena's father is James Step 1: Lena's husband is Wesley Step 2: Wesley's son is Theodore Step 3: Theodore's grandfather is James Task: Richard's father is Donald	Step 1: Richard's grandmother is Elizabeth Step 2: Elizabeth's son is Donald

Table 9: Failure cases analysis - the results of different prompts for the same task.

Wrong			Right
Task: C's nephew is A Step 1: C's son is D Step 2: D's aunt is B Step 3: B's son is A	Task: A's nephew is B Step 1: A's brother is C Step 2: C's son is B	Task: C's nephew is A Step 1: C's son is D Step 2: D's aunt is B Step 3: B's son is A	Task: A's nephew is E Step 1: A's son is D Step 2: D's grandmother is B Step 3: B's son is C Step 4: C's son is E
Task: Lena's nephew is Richard Step 1: Lena's son is Theodore Step 2: Theodore's aunt is Cindy Step 3: Cindy's son is Richard	Task: Theresa's nephew is John Step 1: Theresa's brother is George Step 2: George's son is John	Task: Lena's nephew is Richard Step 1: Lena's son is Theodore Step 2: Theodore's aunt is Cindy Step 3: Cindy's son is Richard	Task: Antonio's nephew is Allan Step 1: Antonio's son is Bobby Step 2: Bobby's grandmother is Shantel Step 3: Shantel's son is Robert Step 4: Robert's son is Allan
Task: Donald's nephew is Gilbert Step 1: Donald's son is Edgar Step 2: Edgar's mother is Marge ✗ Step 3: Marge's son is John	Task: Donald's nephew is Gilbert Step 1: Donald's brother is James Step 2: James's grandson is Johnson Step 3: Johnson's father is Hugh Step 4: Hugh's brother is Wesley ✗	Task: Donald's nephew is Gilbert Step 1: Donald's son is Edgar Step 2: Edgar's mother is Marge ✗	Task: Donald's nephew is Gilbert Step 1: Donald's son is Jeremy Step 2: Jeremy's grandfather is Jose Step 3: Jose's son is Samuel Step 4: Samuel's son is Gilbert

Table 10: The comparison of different demonstrations on results.

Method	Example 1	Example 2
Random Prompt	Task: D's mother is B Step 1: D's father is E Step 2: E's daughter is C Step 3: C's brother is A Step 4: A's mother is B Task: Allan's mother is Marie Step 1: Allan's father is Robert Step 2: Robert's daughter is Michelle Step 3: Michelle's brother is Ronald Step 4: Ronald's mother is Marie Task: Jose's granddaughter is Margaretta Step 1: Jose's father is William ✗ Step 2: William's niece is Margaretta	Task: A's grandson is B Step 1: A's granddaughter is C Step 2:C's brother is B Task: Clarence's grandson is James Step 1: Clarence's granddaughter is Charlotte Step 2: Charlotte's brother is James Task: Samuel's nephew is Charles Step 1: Samuel's aunt is Marie ✗ Step 2: Marie's grandfather is Charles
Entity-based Prompt	Task: B's granddaughter is A Step 1: B's daughter is D Step 2: D's brother is C Step 3: C's daughter is A Task: James's granddaughter is Juanita Step 1: James's daughter is David Step 2: David's brother is Joshua Step 3: Joshua's daughter is Juanita Task: James's niece is Mary ✗ Step 1: James's daughter is Mary	Task: B's granddaughter is D Step 1: B's grandson is C Step 2: C's brother is A Step 3: A's sister is D Task: James's granddaughter is Andrea Step 1: James's grandson is Thomas ✗ Step 2: Thomas's brother is Don Step 3: Don's sister is Andrea Task: James's nephew is Don Step 1: James's grandson is Thomas Step 2: Thomas's brother is Don
Language Planner	Task: James's father is Dale Step 1: James's sister is Charlotte Step 2:Charlotte's father is Dale Task: Jon's grandfather is Daniel Step 1: James's grandmother is Mary Step 2: Dorothy's mother is Mary Step 3: Mary's husband is Daniel ✗	Task: Lynn's nephew is Michael Step 1: Lynn's brother is Hugh Step 2:Hugh's son is Johnson Step 3:Johnson's brother is Michael Task: Bonnie's nephew is Michael Step 1: Ross's brother is Michael ✗
LMLP (N=1)	Task: B's father is C Step 1: B's son is D Step 2: D's mother is A Step 3: A's father is C Task: Hugh's father is James Step 1: Hugh's son is Bobby Step 2: Bobby's mother is David Step 3: David's father is James Task: Irene's father is Milton Step 1: Irene's grandfather is Jose Step 2: Jose's mother is Mary ✗ Step 3: Mary's husband is Milton	Task: Task: D's nephew is A Step 1: D's brother is B Step 2: B's son is C Step 3: C's brother is A Task: Francisco's nephew is Clarence Step 1: Francisco's brother is Joshua Step 2: Joshua's son is Joseph Step 3: Joseph's brother is Clarence Task: Melanie's nephew is Charles Step 1: Melanie's husband is William Step 2: William's nephew is Kyle Step 3: Kyle's father is Charles ✗

Table 11: Failure cases analysis of different baselines. Examples for each method are all from the same setting.

Right	
Question: Nettie's brother Paul took her to the fair when she was little. Paul also brought his brother Samuel. Shantel and husband Harold are trying to get custody of Shantel's granddaughter Nettie before she gets placed in foster care. What is the relation between Samuel and Harold? Answer: Harold's wife is Shantel, Shantel's granddaughter is Nettie, Nettie's brother is Paul, Paul's brother is Samuel. The relation of Samuel between Harold is grandson.	Question: Kevin went to the park with his sister, Heidi, and his father, Wayne. The three of them had a wonderful time there. What is the relation between Wayne and Heidi? Answer: Heidi's brother is Kevin, Kevin's father is Wayne. The relation of Wayne between Heidi is father.
Question: Chris wants to throw a surprise party for his brother Theodore. Chris's brother Charles helps pick the decorations. Frances combed her grandson Charles's hair. She wanted him to look good for his first day of school. What is the relation between Theodore and Frances? Answer: Frances's grandson is Charles, Charles's brother is Chris, Chris's brother is Theodore. The relation of Theodore with Frances is grandson. ✓	Question: Harold was so excited to surprise his son, Louis, with the tickets to the playoffs. Shantel enjoyed a homemade dinner with her son Louis Tracy was making dinner for her son, Kevin. Shantel, Kevin's grandmother, stopped by, so Shantel made a bit more so they all had enough to eat. What is the relation between Harold and Tracy? Answer: tracy's mother is shantel, tracy's father is harold. the relation of harold and tracy is father. ✓
Wrong	
Question: Wilhelmina took her uncle Hugh to the grocery store. Francisco and his brother Wesley were wrestling. Wilhelmina, Francisco's daughter, was cheering on the competition. What is the relation between Hugh and Wesley? Answer: Wesley's brother is Francisco, Francisco's daughter is Wilhelmina, Wilhelmina's uncle is Hugh. The relation of Hugh between Wesley is brother.	Question: Nicholas bought his brother Wayne a present. It was to congratulate him on becoming a father to his new son, Lorraine. What is the relation between Nicholas and Lorraine? Answer: Lorraine's father is Wayne, Wayne's brother is Nicholas. The relation of Nicholas between Lorraine is uncle.
Question: Constance went shoe shopping with her sister Ellen. Elsie had a daughter named Constance. Elsie had picked her daughter Margaret out the cutest new dress to wear on her birthday. Charles and his sister Kathleen have been best friends ever since childhood. Nadia and her father, James, went to the marina. James's daughter, Mabel, had purchased a boat, and they were eager to see it. Mabel bought her mother, Ellen, a puppy for her birthday. James hung his son Charles's finger paintings on the refrigerator. The paintings were right next to the paintings of Nadia, Charles's sister. Kathleen was n't old enough to make any paintings for her father, James. What is the relation between Margaret and Charles? Answer: charles' mother is nancy, nancy's daughter is elizabeth, elizabeth's husband is john, john's wife is mary, mary's brother is george, george's nephews are david, david's nephews are william, william's nephews are robert.✗	Question: William wanted to have a family cookout so he invited his brother James. James wanted to invite his other brother Cesar. Darryl's mother, Patrice, waited impatiently for him at the diner. Dan played basketball with his brother Eric. Darryl took his brother Eric to the baseball game with Nora's father Cesar. Cesar took his son Eric to go get nachos during the game. Patrice fixed her husband Cesar dinner and then they watched a movie they rented. Cesar rushed to the hospital to find out that his wife and already given birth to a boy and had named him Dan. Cesar was so excited to surprise his son, Eric, with the tickets to the playoffs. What is the relation between William and Nora? Answer: nora has two brothers, one of them being william. william is the only one who knows about nora's pregnancy. he is also the only one who knew about nora's pregnancy.✗

Table 12: Showcase of results and failure cases analysis of CoT.