

# CHAIN-OF-KNOWLEDGE: Integrating Knowledge Reasoning into Large Language Models by Learning from Knowledge Graphs

Anonymous ACL submission

## Abstract

Large Language Models (LLMs) have exhibited impressive proficiency in various natural language processing (NLP) tasks, which involve increasingly complex reasoning. Knowledge reasoning, a primary type of reasoning, aims at deriving new knowledge from existing one. While it has been widely studied in the context of knowledge graphs (KGs), knowledge reasoning in LLMs remains under-explored. In this paper, we introduce CHAIN-OF-KNOWLEDGE, a comprehensive framework for knowledge reasoning, including methodologies for both dataset construction and model learning. For dataset construction, we create KNOWREASON via rule mining on KGs. For model learning, we observe rule overfitting induced by naive training. Hence, we enhance CoK with a trial-and-error mechanism that simulates the human process of internal knowledge exploration. We conduct extensive experiments with KNOWREASON. Our results show the effectiveness of CoK in refining LLMs in not only knowledge reasoning, but also general reasoning benchmarkms.

## 1 Introduction

Large Language models (LLMs) have established new state-of-the-arts across a wide range of natural language processing (NLP) tasks (Brown et al., 2020; Bang et al., 2023). Increasingly, their impressiveness have expanded to complex problems challenging reasoning abilities, including arithmetic reasoning (Cobbe et al., 2021), commonsense reasoning (Talmor et al., 2018), and symbolic reasoning (Srivastava et al., 2022). These reasoning abilities enables LLMs to make informed decisions, solve complex problems, and provide more accurate and relevant responses

Knowledge reasoning represents an indispensable aspect of reasoning, which combines acquired knowledge to derive novel knowledge (Chen et al., 2020), as shown in Figure 1. It shares similarities

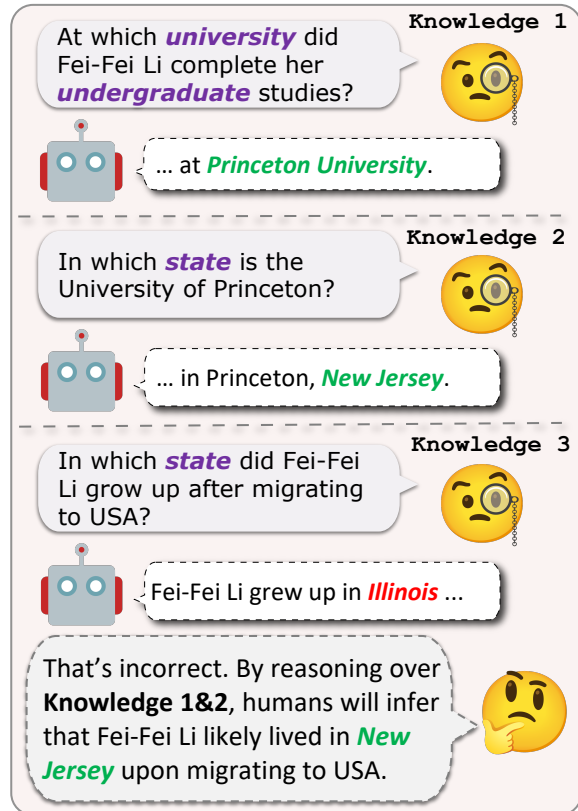


Figure 1: Current LLMs struggle with knowledge reasoning, *i.e.*, combining acquired knowledge to infer new knowledge.

with commonsense reasoning and symbolic reasoning in its reliance on existing knowledge and logical inference to derive new conclusions. Previously, knowledge reasoning has been extensively studied within the context of knowledge graphs (KGs). KGs represent fact knowledge in the form of relational triples, *e.g.*, (*Plato*, *author\_of*, *The Republic*). Knowledge reasoning over KGs is to harness the existing knowledge to infer and derive novel one, typically by explicitly modeling or implicitly learning compositional rules for relational patterns (Sun et al., 2019). This enriches the KGs and supporting downstream tasks such as link predic-

tion (Zhu et al., 2021) and fact classification (Yao et al., 2019). Existing methods for KG reasoning could be distinguished into structured-based methods such as TransE (Bordes et al., 2013) and description-based methods such as LMKE (Wang et al., 2022). However, knowledge reasoning in LLMs remains significantly underexplored, which could serve as a valuable complement to LLM reasoning.

In this paper, we propose to integrate this knowledge reasoning ability into LLMs, leveraging KGs. Specifically, we introduce CHAIN-OF-KNOWLEDGE (CoK), a comprehensive learning framework for knowledge reasoning. CoK includes methodologies for both dataset construction and model learning. The *dataset construction* is based on KGs. As illustrated in Figure 2, it includes three steps: 1) *rule mining*, which mines compositional rules in KGs; 2) *knowledge selection*, which identifies interrelated triples matching those rules; and 3) *sample generation*, which transforms the triples into natural language samples. For *model learning*, we observe that training LLMs via behavior cloning often leads to rule overfitting and consequent hallucination. Hence, we further enhances CoK with a *trial-and-error* mechanism, which simulates humans’ internal process of knowledge exploration for improved generalizability.

We conduct extensive experiments to validate the effectiveness of CoK, which covers both anonymized and regular settings. In the anonymized settings, we replace entity names to ensure analysis uninfluenced by data leakage. In the regular settings, we showcase the value of CoK for not only real-word knowledge reasoning, but also other reasoning benchmarks.

The contributions of this paper are mainly summarized as follows:

- We introduce the knowledge reasoning task to evaluating and enrich LLMs. Our curated dataset, named KNOWREASON, will be released to facilitate future research in this direction.
- We propose CHAIN-OF-KNOWLEDGE (CoK), a comprehensive framework for advancing LLMs’ knowledge reasoning ability. CoK provides a detailed method for dataset construction, as well as two learning methods including behavior cloning and trial-and-error.
- We conduct extensive experiments across var-

ious settings, including anonymized ones and regular ones. Our results validate the effectiveness of CoK, with promising generalizability to novel rules and upgraded challenges. Moreover, we showcase the broad utility of CoK via its efficacy in improving other various tasks.

## 2 Related Works

**LLM reasoning** LLMs have achieved significant success in many NLP tasks and their capabilities have been extended to complex reasoning tasks such as common-sense reasoning (Talmor et al., 2018), arithmetic reasoning (Cobbe et al., 2021), and symbolic reasoning (Srivastava et al., 2022). It has been observed that LLMs perform poorly on reasoning tasks when using standard prompts (Wei et al., 2022). To address this issue, Brown et al. proposed few-shot prompting (Brown et al., 2020), which provides the model with examples of question-answer pairs and has proven effective in reasoning tasks. To further enhance performance, Wei et al. (2022) proposed *Chain-of-Thought* (CoT) prompting (Wei et al., 2022), which provides the model with input-output examples that include explicit reasoning steps. Different from CoT, Program of Thoughts(PoT) uses language models to express the reasoning process as a program, and then executes the generated programs to derive the answer (Chen et al., 2022). Tree-of-Thought (ToT) extends the concept of CoT by incorporating a hierarchical structure into the reasoning process (Yao et al., 2024). This approach is particularly useful for tasks requiring complex decision-making and reasoning, where multiple pathways must be evaluated to reach the correct solution. Although ToT is effective for decision-making and path selection, it requires access to external information such as context. In contrast, our work introduces the CoK framework to enhance the knowledge reasoning abilities of LLMs by utilizing their internal knowledge base.

**Knowledge Reasoning over KGs** Knowledge reasoning is the process of using known knowledge to infer new knowledge (Chen et al., 2020), which is widely used in knowledge graph completion (Zhang et al., 2020). Main approaches to knowledge graph reasoning(KGR) can be broadly classified into four main categories: *embedding-based reasoning* captures the implicit association between entities and relations through mapping

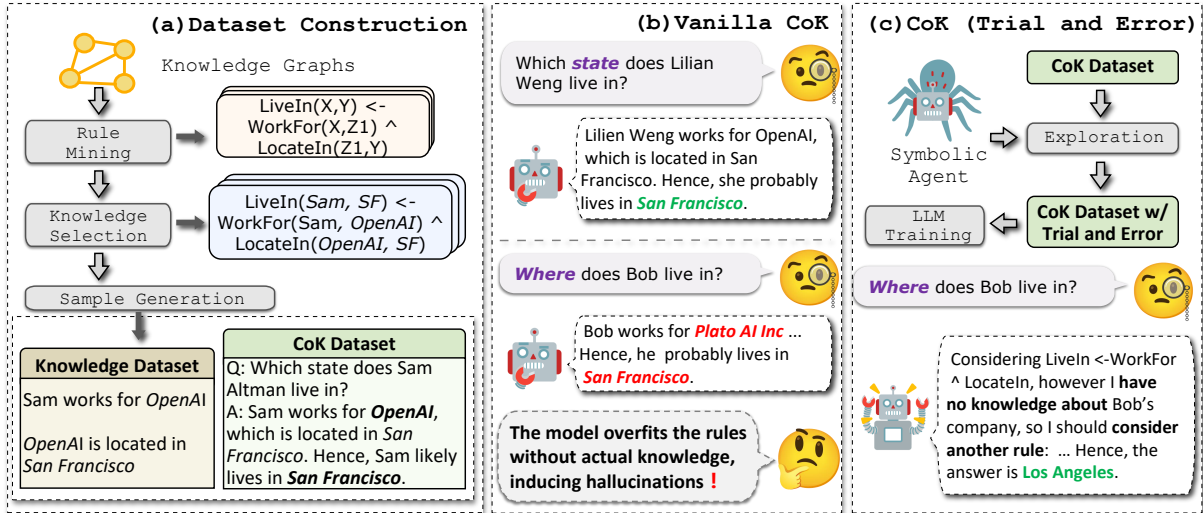


Figure 2: The framework of CHAIN-OF-KNOWLEDGE(CoK). (a) **Dataset Construction** includes three steps, *i.e.*, rule mining, knowledge selection and sample generation. This yields a knowledge dataset and CoK dataset. (b) **Vanilla CoK** trains LLMs in a behavior cloning manner, which may induce rule overfitting and hallucination. (c) **CoK (Trial and Error)** is hence proposed, which enables LLMs to simulate humans’ internal process of knowledge exploration.

a symbolic representation to a vector space for numerical representation (Bordes et al., 2013); *symbolic-based reasoning* uses logical rules to infer new relationships in knowledge graphs, offering interpretable, human-like reasoning (Galárraga et al., 2013); *neural-network-based reasoning* uses neural architectures to predict relationships in knowledge graphs, enabling complex and flexible reasoning (Socher et al., 2013; Schlichtkrull et al., 2018); and *mixed reasoning* combines symbolic-based, embedding-based, and neural network-based reasoning to enhance the accuracy and interpretability of knowledge graph reasoning (Xiong et al., 2017; Guo et al., 2018).

### 3 Methodology

#### 3.1 Preliminaries and Task Formulation

**Knowledge Graphs** KGs store collections of facts as triples, denoted as  $G = \{(e, r, e') \mid e, e' \in \mathcal{E}, r \in \mathcal{R}\}$ , where  $\mathcal{E}$  and  $\mathcal{R}$  represent the sets of entities and relations, respectively.

**Atoms and Rules** KGs contain compositional rules that can be extracted or modeled to infer new knowledge. These rules consist of multiple relational atoms., where an atom can be expressed as  $r(X, Y)$ , where  $r$  is a relation and  $X, Y$  are variables for entities. A *rule* in the KGs can be expressed by the following formula:

$$r_h(X, Y) \leftarrow r_1(X, Z_1) \wedge \dots \wedge r_n(Z_{n-1}, Y) \quad (1)$$

Here,  $r_h(X, Y)$  denotes the *head* atom and denotes the *rule head*, while  $r_1(X, Z_1) \dots \wedge r_n(Z_{n-1}, Y)$  denotes the *rule body*.

For example, in the rule  $LiveIn(X, Y) \leftarrow WorkFor(X, Z_1) \wedge LocateIn(Z_1, Y) \wedge LiveIn(X, Y)$  is the rule head and  $WorkFor(X, Z_1) \wedge LocateIn(Z_1, Y)$  is the rule body.

**Task Formulation** Given an atom  $r_h(X, Y)$ , where  $X$  is known and  $Y$  is unknown, we seek to determine  $Y$ . Knowledge reasoning involves identifying an appropriate rule, and then utilize the facts that support the rule body to determine the value of  $Y$ .

#### 3.2 Chain-of-Knowledge Data Construction

In this section, we will introduce the idea of our CoK method and how we construct the data.

**Rule mining** In this step, we begin by mining 2-hop rules and then combine them to create 3-hop and 4-hop rules.

To derive rules for data construction from triples in the knowledge graph, we utilize a breadth-first approach to sample 2-hop atoms combinations that connect the head entity to the tail entity. The algorithm we use is shown in Appendix A.1 These combinations serve as instances for 2-hop rules. For example, given an atom  $r_3(e1, e3)$ , we can sample the instance  $r_1(e1, e3) \leftarrow r_2(e1, e2) \wedge r_3(e2, e3)$ . The head and tail of this path correspond to the

211 head entity  $e_1$  and the tail entity  $e_3$  of the atom re- 262  
212 spectively. The corresponding rule is  $r_1(X, Y) \leftarrow$  263  
213  $r_2(X, Z) \wedge r_3(Z, Y)$ . 264

214 After sampling across the entire knowledge 265  
215 graph, we obtain a series of rule instances. First, 266  
216 we count the number of instances corresponding 267  
217 to each rule. Rules with fewer than 1000 instances 268  
218 are considered atypical and are removed from the 269  
219 list, serving as the first round of rule filtering. 270

220 For a rule  $r_1(X, Y) \leftarrow r_2(X, Z) \wedge r_3(Z, Y)$ , if 271  
221 the number of instance combinations in the graph 272  
222 that satisfy the *rule body* is  $x$ , and the number of 273  
223 combinations that also satisfy the *rule head* is  $y$ , 274  
224 then the *confidence* formula for the rule is  $y/x$ . 275  
225 Using this formula, we calculate the *confidence* for 276  
226 each rule and set a reasonable threshold of 0.6. If 277  
227 the *confidence* is greater than 0.6, we consider the 278  
228 rule to be valid and retain it; otherwise, we discard 279  
229 it. After applying the above steps, we obtain 203 280  
230 2-hop rules. 281

231 We combine 2-hop rules to create longer rules. 282  
232 Given two rules, if the rule head of one rule is part 283  
233 of the rule body of another rule, we replace that part 284  
234 with the rule body of the first rule. For example, 285  
235 consider the following two rules:

- 236 1)  $BornIn(X, Y) \leftarrow HighSchool(X, Z_1) \wedge$  286  
237  $LocateIn(Z_1, Y)$ ; 287  
238 2)  $CitizenOf(X, Y) \leftarrow BornIn(X, Z_1) \wedge$  288  
239  $CityOf(Z_1, Y)$ . 289

240 The head of Rule1,  $BornIn(X, Y)$ , is part 290  
241 of the body of Rule2, so we replace it 291  
242 with the body of Rule1 to form a 3-hop 292  
243 rule:  $CitizenOf(X, Y) \leftarrow HighSchool(X, Z_1) \wedge$  293  
244  $LocateIn(Z_1, Y) \wedge CityOf(Z_1, Y)$ . 294

245 Using this approach, we generate 159 3-hop 295  
246 rules and 158 4-hop rules. 296

247 **Knowledge Selection** In the process of knowl- 297  
248 edge reasoning, if LLMs do not have access to the 298  
249 facts that support the reasoning path, the reasoning 299  
250 cannot be completed due to the lack of key infor- 300  
251 mation. On the other hand, if the fact we query is 301  
252 already embedded in the LLMs’ internal param- 302  
253 eters, the validity of the reasoning cannot be assured 303  
254 due to potential data leakage. Therefore, it is essen- 304  
255 tial to select knowledge that is appropriate for CoK 305  
256 data construction. 306

257 We construct datasets in both anonymized set- 307  
258 tings and regular settings, and different knowledge 308  
259 selection processes are applied to the two settings. 309

260 For each rule obtained, we first identify all its 310  
261 instances from the knowledge graph. To prevent 311

the model from overfitting to a particular rule dur- 262  
ing training, which could lead to path dependency, 263  
we ensure a balanced quantity of each rule in the 264  
training data. We achieve this by sampling an equal 265  
number of instances for each rule. Next, for each 266  
instance, we gather the involved facts and use them 267  
for knowledge selection. 268

269 In the anonymized setting, we replace all enti- 270  
271 ty names in the facts with random, non-existent 271  
272 strings, making all entities new knowledge to the 272  
273 LLMs. We then collect all the anonymized facts 273  
274 and use them to generate knowledge data for knowl- 274  
275 edge injection. 275

276 In the regular setting, the entities in the facts 276  
277 represent real-world knowledge. For each instance, 277  
278 we gather the relevant facts and use them to perform 278  
279 knowledge probing on the LLM. If the LLM knows 279  
280 all the facts supporting the body of the instance but 280  
281 does not know the fact represented by the instance 281  
282 head, we retain this instance for generating samples 282

283 **Sample Generation** Finally, we apply advanced 283  
284 LLMs <sup>1</sup> to transform the knowledge into natural 284  
285 language sentences. 285

286 For knowledge dataset, we generate a paragraph 286  
287 of natural language description for each entity. For 287  
288 CoK dataset, we generate a sample for each in- 288  
289 stance obtained in the previous step. For the rule 289  
290 head  $r_h(X, Y)$ , we prompt advanced LLMs to gener- 290  
291 ate a natural language question. Given the rela- 291  
292 tionships between entities, if multiple  $Y$ s corre- 292  
293 spond to a single  $X$ , questioning  $Y$  will result in 293  
294 multiple answers, complicating evaluation. There- 294  
295 fore, we choose the unique entity between  $X$  and 295  
296  $Y$  to generate questions. For the rule body, we 296  
297 combine all the facts to form a reasoning chain 297  
298 that describes the reasoning process from  $X$  to  $Y$ , 298  
299 which will be used as the answer to the question. 299  
300 The details of sample generation are shown in Ap- 300  
301 pendix A.3 301

302 Our experiments include both anonymized and 302  
303 regular settings. In each setting, the CoK dataset 303  
304 is used to fine-tune LLMs in a supervised manner. 304  
305 Conversely, the knowledge dataset is employed 305  
306 exclusively during the continuous pretraining stage 306  
307 in anonymized settings to inject knowledge into 307  
308 LLMs. 308

<sup>1</sup>In this paper, we use GPT-3.5 for sample generation.

### 3.3 Chain-of-Knowledge Learning

**Naive Training** First, we directly train LLMs on KNOWREASON in a behavior cloning manner. However, we observe a phenomenon called *rule overfitting*. In this situation, the trained models tend to rely on rules encountered during training, even in the absence of supporting facts.

**Trial and Error** Hence, we introduce a trial-and-error (T&E) mechanism to CoK learning, which simulates the human process of exploring over our internal knowledge.

To simulate the human process of knowledge reasoning, we humans initially select a plausible rule and start reasoning based on it when presented with a question. During this process, if we realize that we lack a crucial fact required by the rule, we switch to an alternative reasoning path instead of continuing without the essential information.

Hence, we integrate the concept of trial and error with our method, incorporating exploration of the LLM’s internal knowledge base into the reasoning process. This approach enables LLMs to discern when to apply a rule and when to backtrack it due to a lack of supporting facts, subsequently switching to a more appropriate rule.

We design a symbolic agent to work in conjunction with LLMs to generate exploration path, employing a trial-and-error approach. For each sample, the symbolic agent first selects a possible rule as a candidate path and then searches for supporting facts for the rule in the internal knowledge base of LLMs. If any part of the rule body lacks supporting facts, the process is recorded as an error, and the symbolic agent switches to another rule as the candidate path. This process repeats until a reasoning path with sufficient supporting facts is found, leading to the desired result. The entire exploration process is captured as a data sample, comprising at least one error and the correct reasoning path. This trial and error process is shown in Algorithm 1.

## 4 Experiments

### 4.1 Settings

**Datasets** We select Wikidata5m (Wang et al., 2021) as our data source, which is a million-scale knowledge graph dataset that is aligned with Wikidata, facilitating data processing and usage.

We construct a dataset KNOWREASON, which includes a knowledge dataset and a CoK dataset. The construction method is detailed in Section 3.

---

#### Algorithm 1: CoK (T&E)

---

**Data:** knowledge graph  $G$ , rule head  $r_h(X, Y)$ , large language model  $M$   
**Result:** exploration process  $P$

```
1  $t \leftarrow 1$ ;  
2 while True do  
   // Select candidate rule  
3    $R_t \leftarrow \text{CandidateRule}(r_h)$ ;  
   // Search for supporting facts  
4   for  $fact \in R_t(\text{rule\_body})$  do  
5     if not  $\text{IsFactExist}(fact, M)$  then  
6        $\text{RecordError}()$ ;  
7        $t \leftarrow t + 1$ ;  
8       continue;  
9   return  $P$ 
```

---

**Anonymized Settings and Regular Settings** We conduct experiments in both the anonymized settings and the regular settings.

In the **anonymized settings**, we conduct the primarily experiments to study knowledge reasoning in LLMs, avoiding the influence of LLMs’ inherent knowledge for this task. In these settings, all entity names are replaced with random, non-existent character names, ensuring that the model parameters contain no prior knowledge of these entities. Consequently, our training data comprises knowledge dataset used during the continuous pre-training stage, as well as CoK dataset used during the instruction fine-tuning stage. The knowledge data includes corpus information related to each entity, which injects the necessary prerequisite knowledge for reasoning into the model. Meanwhile, the CoK and CoK (T&E) data serve as the CoK dataset during supervised fine-tuning stage. The statistics of our training data is showed in Table 1.

In the **regular settings**, we validate the effectiveness of CoK in real-world scenarios. The entities and relationships in the regular settings reflect real-world knowledge. Consequently, we fine-tune LLMs with only the CoK dataset to develop the knowledge reasoning ability, without the knowledge injection step. Besides knowledge reasoning, we further evaluate the benefits of CoK learning for LLMs’ general reasoning abilities in downstream tasks.

**Evaluation Splits** To evaluate the knowledge reasoning ability of the model, we designed two test datasets: In-Domain(ID) and Out-of-

Dataset	#Entity	#Relation	#Rule	#Samples	Avg. Sample Hops	
					CoK	CoK(T&E)
Overall	6611	520	644	2793	2.43	5.34
2-hop	4748	203	326	1993	2	4.4
3-hop	978	159	172	400	3	6.6
4-hop	885	158	146	400	4	8.8

Table 1: Statistics of the training data in anonymized setting of KNOWREASON dataset

Domain(OOD) tests.

**1) In-Domain Tests** The reasoning paths of the samples in ID setting also appear in the training samples. A higher score on the ID setting indicates that LLMs can enhance their knowledge reasoning ability by applying the learned rules.

**2) Out-of-Domain Tests** Unlike the ID setting, the reasoning paths of the samples in the OOD setting do not appear in the training data. A higher score on the OOD setting signifies that the knowledge reasoning capability of LLMs effectively generalizes to previously unseen rules.

Additionally, both ID and OOD settings are divided into three subsets, each corresponding to a different rule length.

**Models** We conduct our experiments on Llama3-8B-instruct(AI@Meta, 2024) and Mistral-7b-instruct-v0.2(Jiang et al., 2023).

During the fine-tuning stage, we employ full fine-tuning for model training. For each training dataset, the model is trained for 4 epochs. After each epoch, it is tested on the evaluation datasets, and the best performance is reported as the result of that training setting.

**Methods** We compare the following four methods in the anonymized settings.

- **Vanilla CoT** In this approach, we prompt the model to answer the question with step-by-step reasoning without any fine-tuning.
- **In-Context-Learning CoK (ICL-CoK)** In this approach, we provide the model with six examples of question and answer pairs from our CoK learning data.
- **CoK** In this approach, we finetune the model with our CoK data.
- **CoK (T&E)** In this approach, we finetune the model with our CoK (T&E) data.

**Metrics** For the knowledge reasoning task, given the rule head  $r_h(X, Y)$ , we pose a question in natural language to identify  $Y$ , where  $Y$  is the *golden entity* for the question.

We use exact match accuracy as our metric. For each subset of our test data, the evaluation formula is as follows:

$$score(T) = \frac{E}{L(T)} \quad (2)$$

where  $T$  represents the test dataset,  $E$  is the number of samples for which the predicted entity matches the golden answer exactly, and  $L(T)$  is the total number of samples in the test dataset  $T$ .

## 4.2 Results in the Anonymized Settings

We conduct experiments using each method mentioned in Section 4.1. For our CoK and CoK (T&E) methods, we construct three versions of data for each method using rules of different lengths. This approach allows us to explore the relationship between rule length and the model’s knowledge reasoning ability. The results of the experiments are shown in Table 2.

**CoK Effectively Improves LLMs’ Knowledge Reasoning Ability.** Our results show that both CoK and CoK (T&E) consistently outperform the baselines on all test datasets. Notable, given some CoK examples, ICL-CoK generally outperforms vanilla CoT. However, it still yield relatively low scores, suggesting that LLMs without fine-tuning struggle with knowledge reasoning based on their internal knowledge, despite having key information within their parameters.

**With Trial & Error, CoK (T&E) Further Improves Performance in OOD Settings.** In CoK, the scores for ID dataset are generally higher than those for OOD dataset, demonstrating the phenomenon of *rule overfitting*, where LLMs rely on reasoning paths encountered during training. This rule overfitting can lead to hallucinations and a loss

Model	Method	Rule Length	ID				OOD			
			2-hop	3-hop	4-hop	all	2-hop	3-hop	4-hop	all
Mistral-7b	Vanilla CoT	-	5.47	6.97	4.48	5.64	4.98	5.47	4.98	5.14
		2	7.46	7.96	7.46	7.63	7.96	6.97	6.97	7.30
	ICL-CoK	2&3-hop	7.96	7.96	6.97	7.63	8.46	5.97	7.46	7.30
		2&3&4-hop	6.47	6.97	7.46	6.97	6.97	4.98	7.46	6.47
	CoK	2	11.94	14.93	12.94	13.27	16.92	7.96	5.97	10.28
		2&3-hop	<u>15.92</u>	<u>19.90</u>	16.9	17.58	16.42	8.96	14.93	13.43
		2&3&4-hop	13.43	15.92	<u>21.89</u>	17.08	12.44	16.92	<u>22.89</u>	17.41
	CoK(T&E)	2-hop	11.94	8.90	9.95	10.28	14.93	11.94	11.94	12.94
		2&3-hop	<b>18.41</b>	<b>24.80</b>	16.92	<b>20.07</b>	<b>20.90</b>	<b>23.88</b>	19.90	<b>21.56</b>
		2&3&4-hop	14.93	17.91	<b>22.89</b>	<u>18.57</u>	<u>18.41</u>	<u>19.90</u>	<b>25.87</b>	<u>21.39</u>
Llama3-8b	Vanilla CoT	-	4.98	6.97	8.96	6.97	5.97	8.96	5.97	6.97
		2-hop	8.46	6.97	6.97	7.46	7.96	6.97	6.97	7.30
	ICL-CoK	2&3-hop	7.46	7.96	6.97	7.46	7.96	7.46	5.97	7.13
		2&3&4-hop	7.46	8.96	8.46	8.29	6.97	8.96	8.96	8.29
	CoK	2-hop	<b>17.41</b>	12.94	13.93	14.76	13.43	8.96	7.96	10.12
		2&3-hop	15.42	<u>19.90</u>	14.93	16.75	<u>19.40</u>	15.92	13.93	16.42
		2&3&4-hop	16.42	18.91	<b>21.89</b>	<u>19.07</u>	10.45	17.91	<u>19.90</u>	16.09
	CoK(T&E)	2	11.94	15.92	15.92	14.59	18.91	16.92	11.94	15.92
		2&3-hop	11.94	<b>20.90</b>	16.92	16.58	<b>21.39</b>	<b>23.88</b>	18.91	<b>21.39</b>
		2&3&4-hop	<u>16.92</u>	19.90	<u>20.90</u>	<b>19.24</b>	12.94	<u>21.89</u>	<b>22.89</b>	<u>19.24</u>

Table 2: Results(%) of the experiments on anonymized setting with different methods and different rule length. The best results are **bolded**, and the second best ones are underlined.

of generalization on OOD dataset. CoK (T&E) surpasses CoK on most datasets, with a particularly significant improvement on OOD dataset. This suggests that CoK (T&E) enables LLMs to consider more appropriate rules for questions, rather than blindly applying previously encountered rules.

**Learning With Long Rules.** To study the impact of rule length, we conduct experiments using datasets with varying rule lengths. We observe that on the 2-hop and 3-hop test datasets, CoK (T&E)-3-hop achieves the highest scores, while CoK (T&E)-4-hop achieves the highest score on the 4-hop test dataset. As the rules in the data lengthen, the model’s performance does not continuously improve; instead, training with rules having a maximum length of 4-hop actually decreases performance.

To find the reason of this, we calculate the length of rules the model uses in the outputs, the result is shown in Table 3. From the results, the model tend to use longer rules when training with longer rules. When a shorter path is available, using a longer reasoning path increases the difficulty of reasoning, as each step in the reasoning process requires the model to explore and make decisions.

Training with longer rules helps LLMs learn to use more complex rules in reasoning. However,

Training Samples	2-hop	3-hop	4-hop
2-hop	100.0	0.0	0.0
2&3-hop	83.7	27.3	0.0
2&3&4-hop	47.2	24.6	28.2

Table 3: Proportion(%) of various rule lengths in the model’s output when trained with data of different rule lengths.

longer rules are not always better. Training with them can lead to a tendency to use longer rules during reasoning, even when shorter or simpler paths are available, potentially decreasing performance on simpler tasks.

**Error Analysis** We conduct additional experiments to analyze the reasons for the model’s incorrect predictions in ID and OOD settings. We categorize the types of errors based on the first incorrect step in the model’s reasoning process. *e.g.* The rule error indicates that the model selects an inappropriate reasoning path, while a fact1 error indicates that the model uses an incorrect  $Y$  in the fact  $r_1(X, Y)$ . The results are presented in Table 4.

The results indicate that on ID dataset, most errors are caused by the model using incorrect facts. This is likely because the reasoning paths in the ID dataset also appear in the training data. In contrast,

	Rule	Fact1	Fact2
ID	34.78	38.89	26.32
OOD	63.15	23.27	13.57

Table 4: Proportion(%) of different types of errors in ID and OOD setting across rule, fact1, and fact2.

on OOD dataset, more errors are attributable to the reasoning paths selected by the model. Regarding fact errors, we note that the model frequently hallucinates during the initial stages of entity selection. We believe this occurs because, in the later stages of reasoning, the model’s knowledge selection scope becomes more restricted as it incorporates additional supporting facts

Specifically, we find that in some error samples, the model selects an appropriate reasoning path and uses correct facts, yet still arrives at an incorrect entity. This occurs because for a fact  $r_h(X, Y)$ , the combination of  $r_h$  and  $X$  can correspond to multiple  $Y$ s. Although the model deduces a result different from the ground truth due to using different facts, the reasoning process remain valid.

### 4.3 Results in the Regular Settings

**Downstream Tasks with Regular Settings** In regular settings, we train the model with regular data that utilizes real-world entities for data construction, and further test the model on downstream tasks.

The results of regular settings are shown in Table 5.

Model	Method	ID	OOD
Mistral-7b	Vanilla CoT	0.00	0.00
	ICL-CoK	5.50	6.20
	CoK	27.00	21.89
	CoK (T&E)	21.33	22.22

Table 5: Performance(%) of different methods on regular setting.

The results from the regular setting validate the conclusions drawn from the anonymized experiments: When prompted with CoK examples, ICL-CoK outperforms vanilla CoT on both ID and OOD dataset. Both CoK and CoK (T&E) enhance the LLM’s Chain-of-Knowledge capabilities. Furthermore, CoK (T&E) further reduces the LLM’s rule dependency, leading to improved performance on OOD dataset.

To further investigate the generalization of CoK,

we tested our CoK exploration method on other popular benchmarks. The results of the downstream tasks are shown in Table 6. The results indicate that CoK (T&E) outperforms the baseline on three commonsense reasoning benchmarks, suggesting that CoK generalizes well to other reasoning tasks that require various types of knowledge, such as world knowledge.

Method	CSQA	BBH	ARC-e	ARC-c
Mistral-7b	66.5	53.2	81.2	<b>72.2</b>
Mistral-7b + CoK (T&E)	<b>68.1</b>	<b>54.7</b>	<b>82.3</b>	69.4

Table 6: Performance comparison of Mistral-7b and Mistral-7b + CoK (T&E) on four reasoning benchmarks.

## 5 Conclusion

In this paper, we propose CHAIN-OF-KNOWLEDGE, a comprehensive learning framework designed to integrate knowledge reasoning abilities into LLMs, encompassing methodologies for both data construction and model learning. We construct the KNOWREASON dataset for model training. While CoK effectively enhances LLM performance on knowledge reasoning tasks, it may also lead to rule overfitting. By employing a trial-and-error approach, CoK (T&E) addresses this issue and further improves model performance. Extensive experiments on two reasoning benchmarks demonstrate the generalization of CoK to other reasoning tasks.

## Limitations

**Evaluation of the Knowledge Reasoning Ability** Because knowledge reasoning in LLMs remains underexplored in previous studies and no public datasets or benchmarks are suitable for our task, the training and testing of the model’s knowledge reasoning ability are conducted using the dataset KNOWREASON that we constructed. To address this concern, we strive to ensure the diversity of our dataset.

## Data of the Regular Setting is Model-Specific

In the regular setting, the entities in the data represent real-world knowledge. To prevent data leakage, we perform knowledge probing on each model, making the regular setting data model-specific. In contrast, for the anonymized setting, we construct data that requires knowledge injection during the continuous pretraining stage but is applicable to all models.



## Ethics Statements

In this paper, we propose the CHAIN-OF-KNOWLEDGE framework, which includes both data construction and a model learning method to enhance the knowledge reasoning ability of LLMs. Our data construction is based on compositional rules from KGs. First, to ensure the reasonableness of our data, we filter the rules based on their confidence. However, there remains a possibility that some rules may still be unreasonable, leading to flawed samples. Second, since we utilize advanced LLMs for sample generation, it is inevitable that biases present in the LLMs may influence the knowledge. To address these ethical concerns, we will optimize the rule mining method and better align the model’s output with human cognition.

## References

AI@Meta. 2024. [Llama 3 model card](#).

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wengliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multi-task, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Antoine Bordes, Nicolas Usunier, A. Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems*.

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. *Advances in neural information processing systems*, 33:1877–1901.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert systems with applications*, 141:112948.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. *Advances in neural information processing systems*, 26.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *Preprint*, arXiv:1902.10197.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

696 Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan  
697 Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021.  
698 Kepler: A unified model for knowledge embedding  
699 and pre-trained language representation. *Transac-  
700 tions of the Association for Computational Linguis-  
701 tics*, 9:176–194.

702 Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua  
703 Xiao. 2022. Language models as knowledge embed-  
704 dings. In *Proceedings of the Thirty-First Interna-  
705 tional Joint Conference on Artificial Intelligence*.

706 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten  
707 Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,  
708 et al. 2022. Chain-of-thought prompting elicits rea-  
709 soning in large language models. *Advances in neural  
710 information processing systems*, 35:24824–24837.

711 Wenhan Xiong, Thien Hoang, and William Yang Wang.  
712 2017. Deeppath: A reinforcement learning method  
713 for knowledge graph reasoning. *arXiv preprint  
714 arXiv:1707.06690*.

715 Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-  
716 bert: Bert for knowledge graph completion. *arXiv  
717 preprint arXiv:1909.03193*.

718 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran,  
719 Tom Griffiths, Yuan Cao, and Karthik Narasimhan.  
720 2024. Tree of thoughts: Deliberate problem solving  
721 with large language models. *Advances in Neural  
722 Information Processing Systems*, 36.

723 Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhiping  
724 Shi, Hui Xiong, and Qing He. 2020. Relational  
725 graph neural network with hierarchical attention for  
726 knowledge graph completion. In *Proceedings of the  
727 AAAI conference on artificial intelligence*, volume 34,  
728 pages 9612–9619.

729 Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhon-  
730 neux, and Jian Tang. 2021. Neural bellman-ford net-  
731 works: A general graph neural network framework  
732 for link prediction. *Advances in Neural Information  
733 Processing Systems*, 34:29476–29490.

## A Details of CoK data construction

### A.1 Rule minig

In the rule mining step, we first use a breadth-first search algorithm to find composite rule instances from raw KGs, the algorithm we use is as Algorithm 2.

**Algorithm 2:** BFS for rule instances

```
1 for each triple  $(A, r1, B)$  do
2   if  $B$  is in triplets then
3     for each  $(r2, C)$  for the value of  $B$ 
4       do
5         if  $A$  is in triplets then
6           for each  $(r3, C')$  for the
7             value of  $A$  do
9               if  $C' == C$  then
10                return combination
```

### A.2 Knowledge Selection

In the anonymized setting, after identifying the supporting facts for the rules, additional processing of the data is required. If the head of one instance is part of another instance’s body, using this instance for sample generation and model training can lead to data leakage, resulting in an unfair evaluation. To address this issue, we separate the head and body parts within the instances. By traversing all instances, we create a set for the body facts. If an instance’s head fact appears in this set, the instance is at risk of data leakage and is therefore discarded.

### A.3 Sample Generation

**Knowledge Dataset** The knowledge dataset is used to inject knowledge into LLMs exclusively in anonymized settings. It is employed during the continuous pretraining stage and, as such, is presented in the form of a corpus.

For each entity in our knowledge base, we establish a mapping to all related facts. Using these facts, we prompt advanced LLMs to generate a descriptive paragraph, encapsulating the knowledge about the entity. The prompt we use is Prompt A.3

Prompt 1: prompt for knowledge corpus generation

Rewrite the following sentence as a paragraph describing {{entity}}, taking care not to change the original meaning of the sentence or lose information: {{fact1}} {{fact2}} ... {{factn}}

To enhance the model’s ability to memorize knowledge, we group related facts of each entity into sets of 10 and input them into the LLMs to generate a corpus. For each entity, we prompt the LLM to generate four different versions of the knowledge corpus. Furthermore, to improve the model’s extraction of knowledge from its internal knowledge base, we integrate the corresponding CoK data into the pretraining corpus. Thus, each subset of the CoK dataset with different rule lengths has a corresponding knowledge dataset for pretraining.

**CoK dataset** The CoK dataset is used for model learning in the supervise finetuning stage, and has both anonymized and regular setting. We have 3 steps to generate samples for CoK dataset:

1) *Relation Template Generation* For each relation, we generate a template sentence which describes the relation of the two entities in the atom. e.g., having an atom *CitizenOf*( $X, Y$ ), the template is  $\{\{X \text{ is a citizen of } Y\}\}$ . The prompt we use to generate the relation templates is as follows:

Prompt 2: prompt for relation template generation

You will be given a triple and you should output a sentence which describes the relation between the two entities in the triple, here are some examples:  
Triple: (<ENT1>, citizen of, <ENT2>)  
Output: <ENT1> is a citizen of <ENT2>.  
...  
Triple: {{triple}}  
Output:

2) *Question Template Generation* For each relation in the rule head, we generate a question template for it. Considering the sufficiency and non-necessity of the rules, and to ensure that each posed question has only one correct answer, we prompt the LLMs to generate a question with possible tone and only question for the unique entity in the atom. The prompt we use to generate question template is as follows:

Prompt 3: prompt for question template generation

You will be provided with a triple, and you should formulate a question that queries the relationship between the two entities. Ensure that the question you generate is in possible tone and has only one correct answer. Here are some examples:  
Triple: (<ENT1>, citizen of, <ENT2>)  
Output: Which country may <ENT1> be a citizen of?  
...  
Triple: {{triple}}  
Output:

3) *Sample Generation* The sample of CoK dataset

797 is in the form of question-answer pairs, so for a  
798 sample, we generate question and answer respec-  
799 tively.

800 For question generation, we substitute the rule  
801 head into the corresponding template. For answer  
802 generation, we replace each fact in the rule body  
803 and rule head, connecting all resulting sentences  
804 into a template answer. We then prompt the LLM to  
805 polish these sentences into natural language, which  
806 serves as the answer for this sample. The prompt  
807 we use to generate the natural language answer is  
808 as follows:

Prompt 3: prompt for answer generation

Rewrite the following sentence into natural language,  
take care not to change the original meaning or lose  
information: {{sentence}}

## 809 B Experiment Settings

### 810 B.1 Details of Datasets

811 **Test Dataset of KNOWREASON** The statistic  
812 of the test dataset of KNOWREASON is shown in  
813 Table 7  
814

### 815 Benchmarks in Downstream Tasks

- 816 • **CommonsenseQA(CSQA)** (Talmor et al.,  
817 2019) CommonsenseQA is a new multiple-  
818 choice question answering dataset that re-  
819 quires different types of commonsense knowl-  
820 edge to predict the correct answers . It con-  
821 tains 12,102 questions with one correct an-  
822 swer and four distractor answers.
- 823 • **AI2 Reasoning Challenge (ARC)** (Clark  
824 et al., 2018) ARC is a dataset of 7,787 genuine  
825 grade-school level, multiple-choice science  
826 questions, assembled to encourage research in  
827 advanced question-answering. The dataset is  
828 partitioned into a Challenge Set(ARC-c) and  
829 an Easy Set(ARC-e).
- 830 • **BIG-Bench Hard(BBH)** (Suzgun et al.,  
831 2022) BBH is a diverse evaluation suite that  
832 focuses on tasks believed to be beyond the ca-  
833 pabilities of current language models. It focus  
834 on a suite of 23 challenging BIG-Bench tasks  
835 for which prior language model evaluations  
836 did not outperform the average human-rater.

### 837 B.2 Details of Methods

838 **Vanilla CoT** In vanilla CoT, we prompt the  
839 model simply with {{let's think step by step}}.  
840 The prompt we use in as follows:

Prompt 4: prompt for Vanilla CoT

**Instruction:** Knowledge reasoning is the process  
of using known knowledge to infer new knowledge.  
You will be given a question of knowledge reasoning  
task, use you internal knowledge to reason out the  
answer. Let's think step by step.  
**Question:** Which state may Lily live in?

841 **ICL-CoK** In ICL-CoK, we prompt the model  
842 with 6 examples from the CoK dataset. In different  
843 rule length setting, we use examples with different  
844 rule length, the detail is shown in Table 8  
845

846 The prompt we use is as follows:

Prompt 5: prompt for ICL-CoK

**Instruction:** Knowledge reasoning is the process  
of using known knowledge to infer new knowledge.  
You will be given a question of knowledge reasoning  
task, use you internal knowledge to reason out the  
answer.  
Here are some examples:  
{{Example1}}  
...  
{{Example6}}  
**Question:** {{question}}

847 **CoK** The prompt we use for CoK is as follows:  
848

Prompt 6: prompt for CoK

**Instruction:** Knowledge reasoning is the process  
of using known knowledge to infer new knowledge.  
You will be given a question of knowledge reasoning  
task, use you internal knowledge to reason out the  
answer.  
**Question:** {{question}}

849 **CoK (T&K)** The prompt we use for CoK (T&E)  
850 is as follows:  
851

Prompt 7: prompt for CoK (T&E)

**Instruction:** Knowledge reasoning is the process  
of using known knowledge to infer new knowledge. You  
will be given a question of knowledge reasoning task,  
use you internal knowledge to reason out the answer.  
If you don't have the knowledge of the supporting fact  
during reasoning, you should backtrack and change to  
another path until you can get the answer.  
**Question:** {{question}}

## 852 C Case Study

### 853 C.1 Examples of Rules

854 Table 9 shows examples of the rules we mine from  
855 KGs.  
856

### 857 C.2 Examples of Knowledge Dataset

858 Table 10 shows examples of the knowledge dataset  
859 we use in continuous pretraining stage.

Setting	Rule Length	#Entity	#Relation	#Rule	#Sample
ID	2-hop	460	135	147	201
	3-hop	307	96	70	201
	4-hop	360	118	73	201
OOD	2-hop	446	43	28	201
	3-hop	267	66	36	201
	4-hop	353	94	41	201

Table 7: Statistics of the test dataset of KNOWREASON

Rule Length	# Example		
	2-hop	2&3-hop	2&3&4-hop
2	6	0	0
3	3	3	0
4	2	2	2

Table 8: #Examples in ICL-CoK’s prompting

### 860 C.3 Examples of CoK Dataset

861 Table 11 shows examples of data in CoK setting.

862 Table 12 shows examples of data in CoK (T&E)  
863 setting.

### 864 C.4 Cases of Different Types of Errors

865 Table 13 shows different types of errors in error  
866 analysis, including rule error, fact1 error and fact2  
867 error.

Rule Length	Rules
2-hop	Country(X,Y)←PlaceOfBirth(Z,X)∧CountryOfCitizenship(Z,Y) Contry(X,Y)←MemeberOfSportsTeam(Z,X)∧CountryOfCitizenship(Z,Y) CastMember(X,Y)←OriginalLanguageOfFilm(X,Z)∧LanguagesSpoken(Y,Z)
3-hop	Country(X,Y) ← PlaceOfBirth(Z,X) ∧ ResidentOf(Z,W) ∧ Country(W,Y) Country(X,Y) ← PlaceOfBirth(Z,X) ∧ EducatedIn(Z,W) ∧ Country(W,Y)
4-hop	Country(X,Y)←PlaceOfBirth(Z,X)∧Spouse(Z,W)∧PlaceOfBirth(W,V)∧CountryOfCitizenship(W,Y) DistantRelative(A,B)←PlaceOfBirth(A,X)∧ParentOf(Y,A)∧CountryOfCitizenship(Y,Z)∧Spouse(A,W)∧ChildOf(B,W)

Table 9: Examples of different length of rules we mine

Knowledge
<p>Excn, a company known for its innovative approach and cutting-edge solutions, has its headquarters situated in Dyxeti. This strategic location serves as the central hub for Excn’s operations, allowing the company to efficiently coordinate its various departments and teams. Dyxeti’s vibrant business environment and access to top talent make it an ideal setting for Excn to thrive and continue its mission of driving progress and success in the industry.</p>
<p>Ccmr is the location where the headquarters of Bplx is situated. The presence of Bplx’s headquarters in Ccmr is significant as it serves as the central hub for the organization’s operations and decision-making. Ccmr, with its strategic position, provides Bplx with easy accessibility to its stakeholders and allows for efficient management of the business. This centralized location in Ccmr ensures effective coordination among various departments and facilitates seamless communication between Bplx and its global network.</p>
<p>Nryxg is the official language of Gexdzjp, serving as the primary means of communication within the region. It is the designated language used for official documents, government proceedings, and educational instruction, reflecting the cultural and linguistic identity of the people of Gexdzjp. Through the use of Nryxg, individuals in Gexdzjp are able to effectively communicate and connect with one another, fostering a sense of unity and shared understanding among its inhabitants.</p>

Table 10: Examples of the knowledge dataset

Rule Length	Question	Answer
2-hop	Which country might Anykid be a citizen of?	Cckqlvy has Anykid as a part of their team. Cckqlvy is from the country Vevedgta. Therefore, Anykid may be a citizen of Vevedgta. Consequently, Vevedgta is the correct answer.
3-hop	Which country may Pzdazwi belong to?	Mjzk is a citizen of Pzdazwi and is a cast member of Kqts, who is from the country Vevedgta. Therefore, it is possible that Pzdazwi is also from the country Vevedgta.
4-hop	What may be the original language of Yvag?	Brq is fluent in speaking, writing, or signing Owiyxpqf languages. Brq is a actor in Vffdgv, which is produced by Lmljpgzx. Yvag is directed by Lmljpgzx. Therefore, it is possible that the original language of Yvag is Owiyxpqf.

Table 11: Examples of CoK method in CoK dataset

<b>Rule Length</b>	<b>Question</b>	<b>Answer</b>
2-hop	To which country does Qzdhsdq belong?	To find the answer, we can follow the reasoning path: $\text{official\_language}(Y, X) \leftarrow \text{language\_of\_work\_or\_name}(A, X), \text{country\_of\_origin}(A, Y)$ . Lzdxfxw is from Vevedgta, but since we are unsure of Lzdxfxw’s military branch, this path is not applicable. Let’s consider a different path: $\text{country}(X, Y) \leftarrow \text{military\_branch}(A, X), \text{country\_of\_citizenship}(A, Y)$ . Bmezedu serves in the military branch Qzdhsdq, which is linked to Vevedgta. Therefore, we can conclude that Qzdhsdq is associated with Vevedgta. Hence, the answer is Vevedgta.
3-hop	What might be the language of Yjuphmb?	If a person has a work X and is an author of a book Y, then the language of book Y could be the same as the language of work X. In this case, Yjuphmb is a work of Kvyyps, Kvyyps is the author of Tkbz, and the language of Tkbz is Xsbf. Therefore, it is possible that the language of Yjuphmb is Xsbf.
4-hop	Which country might Liose from?	To find the answer, we can use the reasoning path: $\text{Country}(X, Y) \leftarrow \text{PlaceOfBirth}(Z, X) \wedge \text{EducatedIn}(Z, W) \wedge \text{Country}(W, Y)$ . However, since we are unsure of where Liose was educated, this path is not viable. Let’s consider an alternative path: $\text{Country}(X, Y) \leftarrow \text{PlaceOfBirth}(Z, X) \wedge \text{Spouse}(Z, W) \wedge \text{PlaceOfBirth}(W, V) \wedge \text{LocateIn}(W, Y)$ . Since Liose is born in Bhsz and Ari is her spouse, Ari was born in Zhww which locates in Vevegde, it is plausible that Liose is a citizen of Vevegde.

Table 12: Examples of CoK (T&E) method in CoK dataset

<b>Error Type</b>	<b>Question</b>	<b>Model Output</b>	<b>Explain</b>
Rule	Can you tell me the country where Nzwscwm is from?	Nzwscwm is the country that the author of Wcsa is a citizen of. Wcsa is from the country Vevedgta, so it’s possible that Nzwscwm is also from Vevedgta. So the answer is Vevedgta.	The reasoning path is not reasonable.
Fact1	To which country does Ztgl possibly belong?	Ztgl is the head coach of Mfqep. Mfqep is from the country Zxdcgh. Therefore, it is possible that Ztgl also belongs to the country Zxdcgh. Thus, Zxdcgh is the answer.	Ztgl is not a coach, the first fact is wrong.
Fact2	What is the language Arstkb use?	Arstkb has cast member Qoztebgc, who speaks the language Crbzovw. This means that Arstkb use Crbzovw language. Thus, Crbzovw is the answer.	Qoztebgc does not speak Crbzovw, the second fact is wrong.

Table 13: Examples of three types of errors