A Unified Framework for Rule Learning: Integrating Commonsense Knowledge from LLMs with Structured Knowledge from Knowledge Graphs

Qirui Hao
University of Electronic Science and Technology
Chengdu, Sichuan, China
qiruihao@std.uestc.edu.cn

Kewei Cheng
Amazon
Palo Alto, CA, USA
chenkewe@amazon.com

Tongze Zhang
University of Electronic Science and Technology
Chengdu, Sichuan, China
tongzezhang@std.uestc.edu.cn

Hongyuan Liu
University of Electronic Science and Technology
Chengdu, Sichuan, China
hongyuanliu@std.uestc.edu.cn

Junming Shao*
University of Electronic Science and Technology
Chengdu, Sichuan, China
junmshao@uestc.edu.cn

Carl Yang*
Emory University
Atlanta, GA, USA
j.carlyang@emory.edu

Abstract

Unlike many black-box machine learning models, logical rules offer human-understandable explanations for decision-making processes, which is especially important in transparency-critical domains like finance and healthcare. Traditional rule learning methods primarily rely on structured knowledge from Knowledge Graphs (KGs), which cannot align the learned rules with commonsense reasoning, leading to potentially incorrect rules. While Large Language Models (LLMs) offer rich commonsense knowledge, they are prone to hallucinations, which hinder their reliability in learning logical rules. The structured knowledge in KGs can serve as an external reference to mitigate these hallucinations. To leverage the strengths of both approaches, we propose a unified framework CSRL to integrate the commonsense knowledge of LLMs with the structured knowledge from KGs for logical rule learning. CSRL achieves a seamless integration of these two types of knowledge. On one hand, it samples multiple instances of each rule based on the KG structure and exploits LLMs to check the reliability of a rule based on such multiple cases, thereby effectively reducing the effect of hallucinations. On the other hand, CSRL utilizes commonsense knowledge from LLMs to guide dynamic and efficient sampling of useful path instances within KGs. Extensive results from quantitative KG completion experiments and qualitative LLM/human-based semantic assessments demonstrate that our algorithm not only performs well on reasoning tasks but also offers greater reliability and alignment with the real world. Our source code is available at: https://github.com/PerseidsMeteorShower/CSRL.

CCS Concepts

• **Computing methodologies → Reasoning about belief and knowledge**.

Keywords

Logical Rule Learning; Commonsense Learning; Large Language model; Knowledge Graph

ACM Reference Format:
Qirui Hao, Kewei Cheng, Tongze Zhang, Hongyuan Liu, Junming Shao*, and Carl Yang*. 2026. A Unified Framework for Rule Learning: Integrating Commonsense Knowledge from LLMs with Structured Knowledge from Knowledge Graphs. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3774904.3792673

1 Introduction

Knowledge Graphs (KGs) provide an efficient means of storing [29], managing [22], and interpreting [43] large volumes of information in a structured way [9, 11, 14, 21]. This structured organization makes KGs an excellent resource for extracting meaningful patterns, which typically take the form of logical rules that can be applied to infer potential relationships between entities [3, 23, 26]. A logical rule can be represented as $r_h \leftarrow r_{\mathrm{b}}$, expressing that if r_{b} occurs, r_h can be inferred to be true. In this context, r_{b} is referred to as the *rule body* and can comprise multiple atoms, denoted as $r_{b_1} \wedge r_{b_2} \wedge \ldots \wedge r_{b_n}$. Conversely, the relation r_h is the *rule head*. These logical rules are crucial for various applications such as question answering [16, 17], recommendation systems [32, 36], and semantic search [30, 31], by enabling the discovery of new knowledge from existing data.

However, traditional algorithms for rule learning in KGs primarily rely on co-occurrence patterns between the head and body within the graph, and often ignore commonsense knowledge of the real world [12, 18, 40]. While these methods can uncover associations, they often fall short to ensure the rationality and generalizability of the learned rules. Specifically, co-occurrence patterns in KGs do not necessarily imply logical entailment between the head and body, leading to rules that may fail to capture the basic facts and do not align with common sense.

* corresponding authors.

WWW '26, Dubai, United Arab Emirates

ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792673

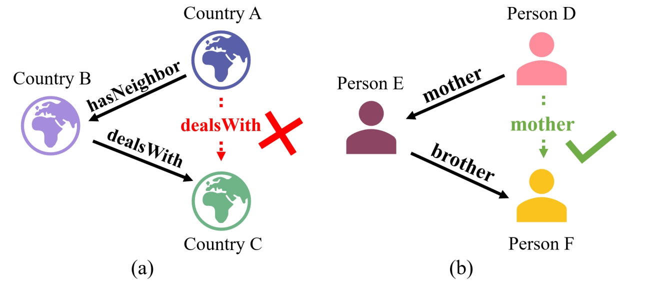


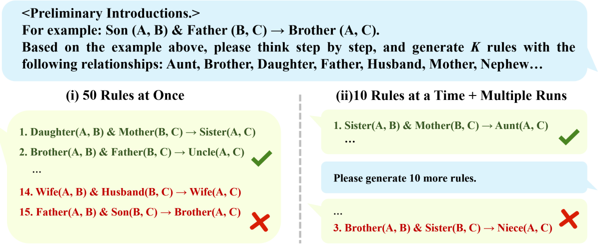
Figure 1: Traditional algorithms often assume that frequently co-occurring relationships in a KG represent valuable logical rules. However, some of these logical inferences do not align with real-world common sense and logical entailment. For example, the inference in (a) does not consider common sense and does not reflect logical entailment. In contrast, the inference in (b) is more reliable and aligns with real-world common sense.

For example, consider the inference in Figure 1 (a): "If country A has neighbor country B, and country B deals with country C, then country A deals with country C". This pattern is frequently observed in KGs, but the occurrence of the rule head does not have a logical entailment with the rule body. The reason for this pattern to be often observed is simply because the rule head itself frequently appears within the KG, leading to its co-occurrence with the rule body. Conversely, the inference in Figure 1 (b), "If person D is the mother of person E, and person E is the brother of person F, then person D is the mother of person F" is a reliable inference. Traditional algorithms fail to distinguish which patterns of frequent co-occurrence are trustworthy. Relying on unreliable rules for further reasoning can be risky, undermining the original purpose of rule learning: extracting rules to enhance the interpretation and reliability of reasoning [6, 25].

Recent advancements in Large Language Models (LLMs) have demonstrated their substantial capacity for commonsense knowledge. Given that LLMs are trained on extensive real-world data, they inherently capture and represent a wide range of commonsense information [39, 45, 46].

Despite the potential of LLMs, existing methods that rely on LLMs for reasoning often struggle with the issue of hallucinations [24, 41, 42]. As Figure 2 shows, if we merely provide an LLM with definitions of rules and relations, and prompt it to generate logical rules based on these relations by itself, it will easily exhibit hallucinations. This will lead to the production of factually incorrect rules. Notably, this phenomenon occurs irrespective of the prompt methods. Additionally, we provide two more kinds of errors that frequently happen in **Appendix F**. Thus, in the absence of external knowledge to provide further guidance, relying solely on LLMs for rule generation is also unreliable.

Addressing hallucinations in LLMs is a challenging task as there is no universally reliable method to verify whether the output is factual. LLM may produce seemingly plausible but factually incorrect output, especially in the absence of explicit external guidance. Structured information including entities and relationships in KGs can be used to provide LLMs with such external guidance, reducing the likelihood of hallucinations. For every abstract rule, KGs contain many different corresponding instances, not only making the rule easier to understand for LLMs but also providing a rich resource to guide the commonsense reasoning of LLMs based on multiple cases, thus reducing the effect of hallucinations.



<Preliminary Introductions.>
For example: Son (A, B) & Father (B, C) → Brother (A, C).
Based on the example above, please think step by step, and generate *K* rules with the following relationships: Aunt, Brother, Daughter, Father, Husband, Mother, Nephew…

(i) 50 Rules at Once

1. Daughter(A, B) & Mother(B, C) → Sister(A, C)
2. Brother(A, B) & Father(B, C) → Uncle(A, C) ✓
…
14. Wife(A, B) & Husband(B, C) → Wife(A, C)
15. Father(A, B) & Son(B, C) → Brother(A, C) ✗

(ii)10 Rules at a Time + Multiple Runs

1. Sister(A, B) & Mother(B, C) → Aunt(A, C) ✓
…

Please generate 10 more rules.

…
3. Brother(A, B) & Sister(B, C) → Niece(A, C) ✗

Figure 2: Generating rules solely based on LLMs is prone to hallucinations. We prompt a typical LLM (GPT-4o) to generate simple 2-hop rules based on given relationships. The prompt provides a comprehensive explanation of logical rule to the LLM, supplemented with examples. The output shows that the LLM easily produces rules with factual errors due to hallucination. The exact prompt we use can be found in Appendix G.2.

Since LLMs and KGs can mitigate the weaknesses of each other, we propose the **C**ommonsense and **S**tructured knowledge integrated **R**ule **L**earning framework (**CSRL**). It can effectively generate rational and trustworthy logical rules by enhancing the synergy between the commonsense knowledge embedded in LLMs and the structured knowledge derived from KGs.

CSRL achieves a close-knit integration of LLMs and KGs. On one hand, CSRL harnesses the structured knowledge from KGs to guide LLMs in a concrete and multiturn manner. Structured knowledge in KG instantiates rules into different specific instances illustrating varying cases. LLMs merely need to perform an easy and concrete task, assessing whether the textual instances align with common sense, consequently mitigating hallucinations. On the other hand, CSRL utilizes the commonsense knowledge from LLMs to guide effective and efficient KG exploration. Rules assessed as potentially correct by LLMs are stored in a candidate set, directing a dynamic sampling mechanism for further exploring useful structured knowledge within KGs.

The rules learned by CSRL demonstrate state-of-the-art inference capabilities in KG completion tasks across multiple datasets. Furthermore, given that the KG completion task is unable to semantically evaluate the commonsense reliability of rules, we propose a novel semantic assessment leveraging advanced LLMs, along with a human assessment, to further evaluate the quality of learned rules. CSRL demonstrates strong performance in both semantic reliability assessments.

In summary, our contributions are three-fold:

- We identify the limitations of using KGs and LLMs independently for logical rule learning. Rules derived solely from co-occurrence patterns in KGs may fail to capture logical entailment, rendering them unreliable. In contrast, relying exclusively on LLMs may suffer from hallucinations, making the learned rules difficult to trust as well. Instead, we propose a unified logical rule learning framework CSRL. By effectively integrating the commonsense knowledge in LLMs with the structured knowledge from KGs, CSRL can leverage their strengths and mitigate each other's weaknesses.

- CSRL seamlessly integrates commonsense knowledge with structured knowledge, allowing them to mutually enhance each other. On one hand, we utilize the structured knowledge inherent in KGs to instantiate rules from multiple cases, thereby leveraging the commonsense knowledge of LLMs and reducing hallucinations. On the other hand, we transform the commonsense assessments from LLM to guide the efficient exploration of structured knowledge within KG.
- We conduct comprehensive experiments on both standard KG completion and LLM/human-based semantic assessments, demonstrating the superior effectiveness of CSRL in reasoning on KG while ensuring the semantic reliability of learned rules.

2 Related Works

2.1 Logical Rule Learning in KGs

Logical rule learning offers a solution to logical reasoning by deriving explicit rules from KGs and utilizing these rules for inference. This approach enhances the interpretability and reliability of reasoning algorithms.

The core idea of rule learning in KGs is to automatically infer logical rules that capture meaningful relational patterns among entities [9, 37, 47]. Traditional methods, such as AMIE [8], define rule scores according to the principle of confidence and further enhance rule quality evaluation by employing the partial completeness assumption to simulate negative examples. The rule mining process is to search over the rule space and select the rules with the highest score.

More recent neural approaches, such as DRUM [27], define logical rules using a differentiable framework and employ low-rank tensor approximations to estimate rule confidence, enabling more efficient and scalable rule inference. RNNLogic [25] treats logical rules as latent variables, simultaneously utilizing them for rule generation and reasoning prediction, thus integrating rule learning and inference into a unified process. NCRL [4] decomposes rule bodies into smaller combinations and recursively merges them to infer the rule head.

While these algorithms employ various methods to model logical rules, they share a fundamental reliance on co-occurrence patterns of relationships in KGs to evaluate rule plausibility. Consequently, they can only measure the co-occurrence relationship, but fall short of capturing true logical entailment. To address this limitation, we incorporate commonsense knowledge from LLMs to better align rule learning with real-world reasoning tasks.

2.2 Logical Reasoning based on LLMs

Recent advancements in LLMs have introduced novel methodologies for logical reasoning [1, 13, 35]. LLM-ERL [2] guides LLMs to perform logical reasoning through two primary strategies: prompting the model to autonomously generate rules, and manually constructing a complex rule dataset. However, manually curating datasets requires significant effort, and these generalized rule datasets may perform poorly in domain-specific tasks. Moreover, the self-generated rules from LLMs often lack reliability.

A more promising approach involves integrating LLMs with KGs. RoG [20] first leverages LLMs to generate relation paths relevant to the problem, identifies these paths in KGs, and then employs LLMs for reasoning based on these paths. While several algorithms follow similar subgraph querying techniques, they do not fully prevent LLMs from producing hallucinations due to a lack of interpretability in the reasoning process. Furthermore, these methods are primarily limited to querying KG and struggle to handle more complex downstream tasks.

Conversely, rule-based reasoning offers better interpretability and can extend beyond simple queries to applications such as KG completion. ChatRule [19] exemplifies this by feeding closed loops from KGs into LLMs, prompting LLMs to generate important rules, and then calculating rule support within KGs. However, relying on LLMs to select important rules is unreliable, as LLMs lack consistent evaluation criteria for evaluating rules, leading to unreliable outputs and an increased risk of hallucination.

Thus, to enable effective logical rule learning with LLMs, it is essential to provide them with external, structured guidance as a reference and design mechanisms that ensure the generated rules are both reliable and well-grounded. In this paper, we propose incorporating structured knowledge from KGs to mitigate hallucinations and enhance the reliability of the reasoning process.

3 Preliminaries

3.1 Knowledge Graph

A knowledge graph can be represented as a mathematical structure denoted by $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{O}\}$. It consists of an entity set $\mathcal{E}$, a relation set $\mathcal{R}$, and the observed fact set $\mathcal{O} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. Each fact in $\mathcal{O}$ can be donated by a triple (e_i, r_k, e_j), where $e_i, e_j \in \mathcal{E}$ and $r_k \in \mathcal{R}$.

3.2 Horn Rule

In first-order logic, a Horn rule can be defined as a combination of the rule body and the rule head [4]:

$$r_h(x, y) \leftarrow r_{b_1}(x, z_1) \wedge \ldots \wedge r_{b_n}(z_{n-1}, y) \tag{1}$$

where $r_\mathbf{b} = r_{b_1}(x, z_1) \wedge \ldots \wedge r_{b_n}(z_{n-1}, y)$ is the rule body and $r_h(x, y)$ is the rule head. It can also be denoted as $\mathbf{r} = (r_h, r_\mathbf{b})$.

However, the rule in symbolic logic is a concept at the schema level, while in KG, we can only observe paths at the instance level. Therefore, to establish a connection between KG and symbolic logic, we define the relation path and the target relation within the KG.

For a path $o_\mathbf{b} = [(e_i, r_{b_1}, e_{b_1}), (e_{b_1}, r_{b_2}, e_{b_2}), \ldots, (e_{b_{n-1}}, r_{b_n}, e_j)]$ between two entities e_i and e_j, we remove all entities contained within it and define $r_\mathbf{b} = [r_{b_1}, r_{b_2}, \ldots, r_{b_n}]$ as the relation path, which corresponds to the rule body in symbolic logic. For the triple $o_h = (e_i, r_h, e_j)$, we eliminate the entities and define the single relation r_h as the target relation, which corresponds to the rule head in symbolic logic.

Consequently, by combining them, we can finally define the rule in KG as $\mathbf{r} = (r_h, r_\mathbf{b})$ and the instance as $\mathbf{l} = (o_h, o_\mathbf{b})$.

3.3 Logical Rule Learning

The objective of logical rule learning is to determine the confidence score, denoted as $\rho(\mathbf{r})$, for each rule $\mathbf{r}$ in the rule space to measure its reliability. Rules with sufficiently high confidence are selected as the final learned rules, which are subsequently employed for downstream reasoning tasks.

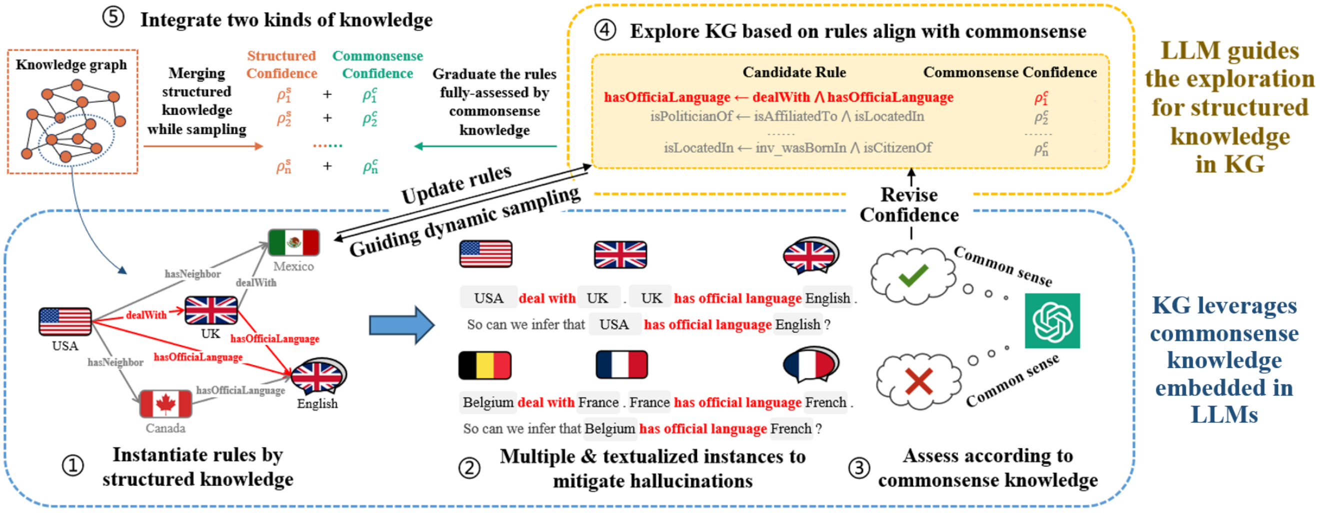

⑤ Integrate two kinds of knowledge
Knowledge graph
Merging structured knowledge while sampling
Structured Confidence
Commonsense Confidence
ρ_1^s + ρ_1^c
ρ_2^s + ρ_2^c
......
ρ_n^s + ρ_n^c
Graduate the rules fully-assessed by commonsense knowledge
④ Explore KG based on rules align with commonsense
Candidate Rule
Commonsense Confidence
hasOfficiaLanguage ← dealWith ∧ hasOfficiaLanguage ρ_1^c
isPoliticianOf ← isAffiliatedTo ∧ isLocatedIn ρ_2^c
isLocatedIn ← inv_wasBornIn ∧ isCitizenOf ρ_n^c
LLM guides the exploration for structured knowledge in KG
Update rules
Guiding dynamic sampling
Revise Confidence
Mexico
hasNeighbor
dealWith
USA
dealWith
UK
hasOfficiaLanguage
hasNeighbor
hasOfficiaLanguage
hasOfficiaLanguage
English
Canada
USA deal with UK . UK has official language English .
So can we infer that USA has official language English ?
Belgium deal with France . France has official language French .
So can we infer that Belgium has official language French ?
Common sense
Common sense
KG leverages commonsense knowledge embedded in LLMs
① Instantiate rules by structured knowledge
② Multiple & textualized instances to mitigate hallucinations
③ Assess according to commonsense knowledge

Figure 3: The overall framework: CSRL systematically fuses LLM commonsense and KG structure. KG instances are translated into natural language for reliable LLM judgment, and LLM feedback dynamically guides efficient KG exploration. By aggregating evidence from both, CSRL produces rules with high confidence in both commonsense and structured validity.

4 Methodology

4.1 Overall Framework

Traditional KG-based rule learning often tends to produce commonsense errors [4, 27], while LLM may output incorrect rules due to hallucinations (as shown in Figure 2). Thus, we propose CSRL, a framework that tightly integrates the strengths of both while compensating for each other’s weaknesses. Specifically, the KG provides structured knowledge to ground LLM reasoning, while LLM filters and guide the exploration of KG for rules that align with commonsense.

The overall structure of CSRL is shown in Figure 3.

Steps 1–3: KG-derived rules are instantiated as varing natural language examples, allowing the LLM to assess their plausibility based on multiple cases. Such varied and repeated interactions help mitigate hallucinations caused by incorrect judgments from the LLM. More details are presented in Section 4.2.

Steps 4: LLM judgments then form a candidate set that dynamically guides KG sampling. Sampling is random but prioritizes rules deemed more reliable in commonsense. This approach leverages LLM’s commonsense knowledge to enable efficient exploration without additional queries. Details are in Section 4.3.

Steps 5: Through iterative cycles, each rule accumulates both structural and commonsense evidence. Its final confidence reflects a blend of commonsense validation and structured pattern support. Structured evidence is not isolated but integrated into the dynamic sampling process, which further enhances the efficiency. Details of this process are discussed in Section 4.4.

4.2 KG Leverages the Commonsense Knowledge Embedded in LLMs

To minimize LLM hallucinations and effectively leverage its commonsense reasoning ability, CSRL instantiates each abstract KG rule into multiple natural language examples which illustrating different cases. By evaluating these natural language examples, the LLM provides a more robust judgment of a rule’s reliability, reducing sensitivity to isolated errors. Besides, unlike some methods that input symbolic triples into LLMs (which LLMs struggle to interpret), CSRL translates rule instances into natural language, aligning with LLM training data and further reducing chance of hallucinations.

For example, as shown in step 2 of Figure 3, an unreliable rule “*hasOfficialLanguage* $(x, z) \longleftarrow$ *dealsWith* $(x, y) \land$ *hasOfficialLanguage* (y, z)” is instantiated with two different sets of entities. Even though LLM judges one case as correct, it can realize the other is incorrect. This case also shows that the LLM uses commonsense reasoning to assess logical validity, not just factual correctness, since each statement is factually true but the logic is flawed. Therefore, the LLM’s responses across different examples offer strong commonsense validation.

4.3 LLM Guides the Efficient Exploration for Structured Knowledge in KG

Traditional KG rule learning algorithms sample paths randomly [5, 8], leading to uninformative patterns. In CSRL, sampling combines randomness with a bias toward instances of rules that more likely to be commonsense reliable. This dynamic sampling method significantly improves efficiency while still exploring new rules. While matching traditional random sampling in speed (2.69×10^{-5} versus 2.79×10^{-5} s/instance), dynamic sampling learns more rules with fewer instances, reducing LLM interactions which are the biggest computation cost. Section 5.4 further shows how dynamic sampling outperforms traditional sampling methods in efficiency.

As dynamic sampling process shown in Figure 4, CSRL maintains two candidate sets: C_T for rules deemed plausible by the LLM, and C_F for implausible ones. Each rule is tracked with the number of positive (n^P) and negative (n^N) LLM judgments. Only when a rule has sufficient evidence is it moved to the result sets S_T and S_F, reducing redundant LLM queries (details on this process are provided in the next subsection).

Specifically, CSRL first samples an equal number of triples from each relation type as anchors. For each anchor, when collecting sequences of subsequent triples, the algorithm prioritizes instantiating rules from C_T. If no instances are found, it then samples random instances using existing paths in the sequence. Each instance is then evaluated by the LLM for reliability. If the rule already exists, the corresponding n^P or n^N is updated. Otherwise, if this rule is new, it will be added to C_T or C_F.

Overall, this dynamic feedback-driven approach quickly focuses sampling on promising rules, while still allowing for the discovery of new candidates.

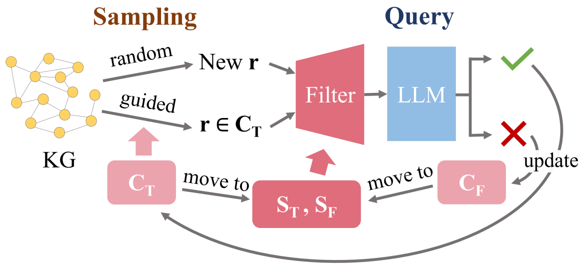


Figure 4: CSRL dynamically adjusts its sampling direction under the guidance of the plausible rule set C_T, enabling it to discover new rules while focusing on those likely to be reliable. Verified rules in S_T and S_F are filtered out to avoid redundant LLM queries. The candidate sets C_T and C_F are updated based on LLM judgments, and once fully verified, their rules are moved to S_T or S_F.

4.4 Integrating Commonsense and Structured Knowledge

4.4.1 Commonsense Confidence. Once a rule accumulates enough evidence (exceeding threshold ϵ), it is moved to the trustworthy (S_T) or unreliable (S_F) set, as detailed in Algorithm 1. This design prevents redundantly LLM query and focuses resources on unresolved rules in two main ways. First, since rules are removed from C_T, they are no longer prioritized for instantiation. Second, as Figure 3, if instances of rules in S_T and S_F are encountered again in random sampling, they are filtered out and will not be sent to the LLM.

When a rule $r_i \in C_T$ is moved, its commonsense confidence ρ_i^c is calculated as the fraction of positive LLM judgments among all evaluated instances:

$$\rho_i^c = n_i^P / (n_i^P + n_i^N). \tag{2}$$

4.4.2 Structured Confidence. To incorporate structured evidence without additional traversal, CSRL records occurrences of rules when filtered out. Specifically, when instance of a rule $r_i \in S_T$ appears during random sampling and is filtered out, its occurrence count $n(r_i)$ is still recorded. This allows structural learning without extra graph traversal. The structured confidence ρ_i^s of a rule $r_i = (r_{h_i}, r_{\mathbf{b}_i}) \in S_T$ measures how often the rule head occurs given the body, reflecting statistical support from the KG.

Algorithm 1 Update Two Candidate Sets and Two Result Sets.

```
Input: Plausible candidate set $C_T$, implausible candidate set $C_F$,
    trustworthy set $S_T$, unreliable set $S_F$ , evaluation limit $\epsilon$
Output: Updated plausible candidate set $C_T$, implausible candidate
    set $C_F$, trustworthy set $S_T$ and unreliable set $S_F$
for rule $\mathbf{r}_i \in C_T$ do
    if $n_i^P \geq \epsilon$ then
        $\rho_i^c$ = CommonsenseConfidence($\mathbf{r}_i$)
        $C_T, S_T, S_F$= Update($\mathbf{r}_i, \rho_i^c$)
    end if
end for
for rule $\mathbf{r}_i \in C_F$ do
    if $n_i^N \geq \epsilon$ then
        $C_F, S_F$= Update($\mathbf{r}_i$)
    end if
end for
```

$$\rho_i^s = \frac{n(r_{h_i}, r_{\mathbf{b}_i})}{n(r_{\mathbf{b}_i})}. \tag{3}$$

The final confidence ρ_i of rule r_i combines both sources, where α balances commonsense and structured evidence.

$$\rho_i = \alpha * \rho_i^c + (1 - \alpha) * \rho_i^s. \tag{4}$$

5 Experiments

To demonstrate the reliability and utility of the logical rules learned by the CSRL algorithm for downstream tasks, we employ three evaluation tasks: (1) Knowledge Graph Completion. This task involves inferring missing entities based on queries such as $(h, r, ?)$ or $(?, r, t)$. (2) Reliability Assessment via Advanced LLMs. Given the lack of tasks for automatically semantically evaluating rule reliability, we propose a new verification task. This task compares the reliability of learned rules against random closed paths using advanced LLMs, thereby assessing the reliability of rules semantically. (3) Human Evaluation. Additionally, we implement manual scoring to verify the reliability of the logical rules, examining the likelihood of their validity in real-world scenarios. Our algorithm achieves SOTA across all these various tasks, underscoring the reliability and practical value of CSRL.

5.1 Knowledge Graph Completion

Knowledge graph completion is frequently utilized by traditional logical rule learning algorithms to evaluate learned rules. This process involves the completion of KG by inferring missing entities based on a given query $(h, r, ?)$ or $(?, r, t)$ using learned rules. Forward-chaining algorithms [28] are commonly employed to infer missing entities based on rules.

Datasets. For a comprehensive comparison with state-of-the-art approaches, we apply our method on four widely-used datasets: FB15k-237 [38], Family [10], WN18RR [7], YAGO3-10 [33]. The statistics of them are shown in **Appendix A**.

Baselines. We compare our methodology with several state-of-the-art algorithms. Traditional rule learning algorithms are solely based on KGs, including Neural-LP [44], DRUM [27], RNNLogic [25], RLogic [5] and NCRL [4]. Given the lack of rule learning

Table 1: KG Completion. Apply MRR, Hit@1, and Hit@10 (%) as evaluation metrics.

Models	FB15K-237			Family			WN18RR			YAGO3-10		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
Neural-LP*	0.24	17.3	36.2	0.88	80.1	98.5	0.38	36.8	40.8	OOM	OOM	OOM
DRUM*	0.23	17.4	36.4	0.89	82.6	99.2	0.38	36.9	41.0	OOM	OOM	OOM
RNNLogic*	0.29	20.8	44.5	0.86	79.2	95.7	0.46	41.4	53.1	OOM	OOM	OOM
RLogic*	0.31	20.3	50.1	0.88	81.3	97.2	0.47	44.3	53.7	0.36	25.2	50.4
NCRL*	0.30	20.9	47.3	0.91	85.2	99.3	0.67	56.3	85.0	0.38	27.4	53.6
GPT-4o solely	-	-	-	0.68	50.0	94.4	0.65	50.6	93.1	0.49	37.4	64.3
ChatRule(GPT-4)	-	-	-	**0.93**	**88.0**	99.8	0.34	30.1	40.0	0.45	35.4	62.7
CSRL(GPT-3.5)	0.42	32.2	**60.7**	0.86	73.6	99.7	0.71	60.1	**95.6**	0.49	37.4	71.5
CSRL(GPT-4o)	**0.44**	**37.0**	59.1	0.87	76.5	**99.9**	**0.72**	**63.2**	90.4	**0.59**	**52.3**	**76.3**

algorithms solely based on LLMs, we develop an experiment to test the reliability of extracted rules with only LLMs. Using the template in **Appendix G.2**, we prompt an advanced LLM, GPT-4o, to generate rules based on relationships from KGs. This method is consistent with the experimental setup illustrated in Figure 2 of Section 1. We also involve ChatRule [19], which is a rule learning algorithms that combine KGs with LLMs.

Evaluation Protocols. For each test triple, either the head or tail entity is masked, and different algorithms are used to predict the masked entity. We utilized the filter setting consistent with existing research during the evaluation. The assessment metrics included Mean Reciprocal Rank (MRR) and Hit@K.

Implementation Details. In the graph completion task, GPT-4o (gpt-4o-2024-08-06 API) and GPT-3.5 (gpt-3.5-turbo-0125 API) are employed as the LLMs to evaluate instance reliability within the CSRL algorithm. hyperparameter settings are obtained through grid search and are the same across datasets. Dynamic sampling is set with a reference degree of $1 - \tau = 1.0$ to the candidate set, and the threshold ρ_0^c for commonsense confidence of rules is 0.8. The weight parameter α for balancing commonsense confidence and structured confidence is 0.5, and the evaluation threshold ϵ is 5. These hyperparameter settings are obtained through grid search and are the same across all datasets. As shown in Section 5.4 and **Appendix B**, they are not overly sensitive and can adapt to different datasets without need of extensive adjustments.

Performance Analysis. The results of graph completion across various datasets are presented in Table 1. [*]means the numbers are taken from [4], OOM means out of memory on experiment machine. For the LLM solely rule generation test, the output from the LLM of FB15K-237 dataset contained numerous textual errors, making it impossible to convert them into standard rules for reasoning. ChatRule also cannot be applied to FB15K-237 since its prompt exceeds the input limit of GPT. The best result of each dataset is highlighted in bold.

CSRL outperforms other algorithms across multiple datasets. On the FB15k-237, WN18RR, and YAGO3-10 datasets, CSRL surpassed the previous state-of-the-art baselines. On the Family dataset, although CSRL does not achieve the highest scores in MRR and Hit@1, it exceeds existing algorithms in Hit@10. This demonstrates that the algorithm effectively integrates commonsense knowledge from LLMs with the structural knowledge of KGs, enabling the learned rules to achieve good performance on reasoning task. Due to the comprehensive integrated framework of CSRL, rules derived by it exhibit strong logical coherence, surpassing not only KG or LLM solely based algorithms but also those combining KG and LLM.

Notably, even when utilizing the less advanced GPT-3.5, CSRL demonstrates impressive performance. This indicates that CSRL can mitigate the hallucination of LLMs by using structured knowledge from KGs to transform a single rule into multiple instances, as discussed in Section 4.2. Consequently, our method can achieve strong performance even with less powerful LLMs, thereby reducing costs.

Hyperparameter Sensitivity. To further evaluate the sensitivity of CSRL to hyperparameters, we conducted experiments on each of them. The results show CSRL remains insensitive to hyperparameter settings. The temperature parameter τ is discussed in Section 5.4, as it relates to the algorithm's efficiency. The detailed results of other hyperparameters are provided in **Appendix B**.

5.2 LLM-based Semantic Assessment

Given the absence of automated methods for evaluating the semantical reliability of rules in past research, we propose a verification task for rule reliability assessment using advanced LLMs. Specifically, we employ the LLM to compare the reliability of algorithmically learned rules against that of random rules extracted from randomly sampled paths within KG. The evaluation algorithm ensures that the random rules are not present in the learned rules. If the algorithm produces reliable rules, learned rules should exhibit greater logical coherence than random rules.

Table 2: LLM-based Semantic Assessment Accuracy (%)

Models	Family	WN18RR	YAGO3-10	FB15K-237
NCRL	15.7	44.3	22.9	41.4
ChatRule (GPT-3.5)	68.6	51.4	28.6	-
CSRL (GPT-3.5)	72.1	67.1	37.4	57.1

We leverage a zero-shot chain-of-thought prompting method, following the suggestions from previous studies [34, 42]. Initially, rules are transformed into textual inference sentences, and the LLM is guided with detailed instructions to deliberate step-by-step, selecting the more reliable inference between two options. If the advanced LLM selects the learned rule as more reliable, we consider this rule to be trustworthy. Detailed prompt for this process is provided in **Appendix G.3**.

Then calculate the proportion of reliable rules among all evaluated rules to get the accuracy, which can serve as the measure of the overall reliability of learned rules:

$$Acc = n(\text{Reliable Rules})/n(\text{All Evaluated Rules}) \qquad (5)$$

Experimental Setup. In this semantic assessment task, we employ the GPT-4o as the advanced LLM to evaluate 70 rules. We compare our CSRL algorithm against the NCRL and ChatRule. We choose these two algorithms for comparison because NCRL represents the best-performing algorithm based solely on KGs, while ChatRule combines both KGs and LLMs.

Given the use of GPT-4o for evaluation, we rerun ChatRule using GPT-3.5 based on its recommended settings. The rules from CSRL are also derived from GPT-3.5.

Comparing with Other Methods. The results of rule reliability assessment across different datasets are presented in Table 2. Our algorithm consistently achieves higher reliability scores comparing other methods across all datasets. This is attributed to the effective integration of commonsense knowledge inherent in LLMs with the structured knowledge from KGs. On one hand, our method successfully identifies factually and logically correct logical rules within the KG rather than mere associative relationships through commonsense knowledge. On the other hand, CSRL's incorporation of structured knowledge mitigates the potential hallucinations that might arise when LLMs learn rules.

5.3 Human-based Semantic Assessment

In our study, we also employ a manual evaluation approach to further verify the reliability of the learned rules. Raters assess the reliability of the logic rules derived from the algorithm by assigning scores. Specifically, this reliability refers to the likelihood of the rule head occurring when the rule body is present.

A scoring scale from 0 to 5 is used, with 0 indicating that the rule head cannot possibly occur. Besides, if the rule body itself is unlikely to occur, the reliability of that rule is also assessed as 0. None of the raters directly engage in this research or have ever seen the original dataset before. The manual assessment is conducted on rules about kinship learned from the Family dataset, which can be relatively easy to assess with general human knowledge.

In evaluating the results, we apply the Kruskal-Wallis test and Dunn's test. Further details on the purpose and method of applying these statistical algorithms can be found in **Appendix C.1**.

Experimental Setup. For the reasons previously discussed in Section 5.2, the NCRL and ChatRule algorithms are used as benchmarks. Based on the Family dataset, rules are learned through NCRL, ChatRule, and our CSRL algorithms. Then 50 rules with confidence greater than 0.5 are randomly selected for manual evaluation. A significance level of 0.05 is applied.

Table 3: Post Hoc Multiple Comparisons on Manual Reliability Evaluation Scores.

Models	P-value	Difference
NCRL - CSRL	7.64×10^{-25}	-144.3375295
NCRL - ChatRule	9.81×10^{-08}	-71.32140047
ChatRule - CSRL	6.61×10^{-10}	-73.01612903

Comparing with Other Methods. Initially, we conduct an assessment to determine whether the scores assigned by different raters to the same algorithm results are consistent, ensuring the objectivity and validity of the ratings. Since the scores may not conform to a normal distribution, we apply the Kruskal-Wallis test rathor than the analysis of variance. As detailed in **Appendix C.2**, the results of the Kruskal-Wallis test indicate no significant differences in the ratings among the three data groups from different raters, suggesting that the scores are derived from samples of the same distribution. So they are reliable and can truthfully reflect the performance of three algorithms.

Then, we examine whether there are significant differences in the scores among the three algorithms. The Kruskal-Wallis test yields a test statistic $H = 117.33$ with a p-value of 3.33×10^{-26}, which is less than 0.05, indicating significant differences in the scores of rules learned by three algorithms.

Given these significant differences, Dunn's test is conducted for post hoc multiple comparisons, as shown in Table 3. Since the pairwise p-values of the scores for the three algorithms are all below 0.05, there are significant differences between their scores. Specifically, the mean rank of the CSRL algorithm exceeds those of NCRL and ChatRule, while ChatRule surpasses NCRL in mean rank. Thus, it can be concluded that CSRL significantly outperforms ChatRule in manual reliability evaluation, and ChatRule significantly outperforms NCRL.

This outcome suggests that our approach effectively integrates the structural knowledge from KG with the common-sense knowledge from LLM, thereby leveraging both strengths while mitigating their weaknesses, leading to more reliable rule generation.

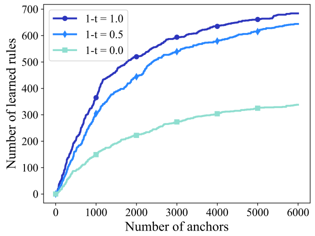


Figure 5: Number of learned rules under varying reference levels $(1 - \tau)$ of C_T.

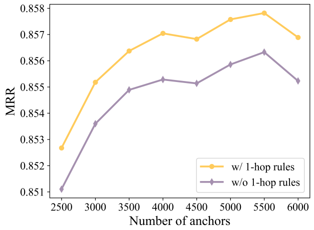


Figure 6: MRR performance of KG completion task with/without 1-hop Rules.

5.4 Efficiency of CSRL

CSRL utilizes candidate set guided dynamic sampling to efficiently explore potentially commonsense-reliable structured knowledge on the graph. As discussed in Section 4.3, dynamic sampling is so fast that its cost is almost the same with traditional random sampling. Consequently, the algorithm's primary computational and time costs come from the LLM itself. Therefore, the efficiency of the algorithm largely depends on how effectively it can interact with the LLM. Dynamic sampling can minimize the waste of sampling associated patterns that are unreliable and merely coincidental. The effectiveness of this mechanism can be observed if the algorithm generates more rules with a given number of anchors when guided by the candidate set C_T.

Figure 5 illustrates the relationship between the number of rules learned by CSRL and the number of sampled anchors based on the Family dataset. Different colored curves represent different temperatures $(1 - \tau)$, which is the degree of dynamic reference to the candidate set C_T during sampling. When $1 - \tau = 0$, the sampling algorithm reduces to traditional random sampling.

The figure shows that with the same amount of sampling, stronger guidance from the candidate set can lead to more rules being learned. This demonstrates the candidate set's ability to leverage the commonsense knowledge from LLMs to guide the efficient exploration of more reliable structured knowledge in KGs.

We further provide the experimental costs of CSRL on each dataset in **Appendix E**, with expenses remaining below $1 for most cases. The low cost is another advantage of CSRL's efficiency.

5.5 Reliability of 1-hop Rules

To verify whether our algorithm has truly learned commonsense reliable rules rather than just frequent co-occurrence patterns in KG, the quality of the 1-hop logical rule can serve as a straightforward metric. By integrating commonsense knowledge from LLMs, CSRL effectively evaluates and learns reliable 1-hop rules (e.g., *husband* ← *inv_wife*) from KGs. Compared to rules of other lengths, it is particularly hard to ensure the reliability of 1-hop rules based on limited structured knowledge within KGs. 1-hop rules learned by CSRL pose a positive impact on inference, as demonstrated by the MRR values for KG completion.

In Figure 6, the x-axis represents the number of sampled anchors, while the y-axis shows the MRR values for KG completion based on the Family dataset. The yellow curve indicates the existence of 1-hop rules, whereas the purple curve represents their absence. We use the Family dataset since it contains many 1-hop spurious relationships, such as $\mathit{Brother}(x, y) \leftarrow \mathit{Sister}(y, x)$. This makes it suitable for evaluating whether our algorithm can effectively learn reliable 1-hop rules. Unreliable 1-hop rules can disrupt the inference of other reliable rules significantly. By contrast, the figure reveals that regardless of the sample size, 1-hop rules learned by CSRL consistently positively contribute to inference. This indicates that the rules derived by integrating commonsense knowledge are reliable and beneficial.

5.6 LLMs Consistency

We conduct experiment on different LLMs also based on the KG completion task with the Family dataset and GPT-4o. The results are shown in Table 4.

Table 4: performance on different llms.

LLM	MRR	Hit@1	Hit@10
GPT-4o	0.87	76.5	99.9
Llama3-70B	0.86	75.3	99.1
Mistral-Large	0.88	78.2	99.0
Qwen2.5-72B	0.89	80.2	99.8

Comparing GPT, CSRL even perform better with Mistral and Qwen2.5, and its performance are always stable. This is because CSRL can reduce hallucination by multiple instantiations, which shows the advance of CSRL compared to traditional methods. It is because CSRL really achieves a seamless integration between LLMs and KGs that we can overcome the challenges faced by traditional methods such as LLM hallucination.

5.7 Prompt Sensitivity

We evaluate the KG completion performance of rules learned using various prompts, as outlined in **Appendix D**. The experiments are conducted on the Family dataset utilizing GPT-4o, with the results summarized in Table 5. The Original prompt refers to the one used in our paper. Prompt 1 modifies the order of the known information and the query; prompt 2 requires the model to output explanations; prompt 3 incorporates directional relationship terms.

The results show minimal variation across prompts, underscoring the consistently of CSRL on different prompt. we believe CSRL can address bias from prompts since it aggregates multiple instances from KG for a single rule.

Table 5: performance on different prompts.

Prompt	MRR	Hit@1	Hit@10
Original	0.87	76.5	99.9
Prompt 1	0.87	76.4	99.6
Prompt 2	0.88	79.0	99.8
Prompt 3	0.86	74.9	99.7

5.8 Case Studies

5.8.1 Rules Learned and Rejected by CSRL. We examine the rules learned by CSRL across various datasets and pick up some representative rules. Both correct and incorrect rules determined by the algorithm are presented in Table 6. CSRL can effectively refute many rules that traditional algorithms fail to identify as lacking logical entailment, such as $\mathit{sister}(x, y) \leftarrow \mathit{inv\_brother}(x, z) \wedge \mathit{brother}(z, y)$ and $\mathit{dealsWith}(x, y) \leftarrow \mathit{dealsWith}\ (x, z) \wedge \mathit{dealsWith}(z, y)$. There is no logical entailment between the rule body and the rule head of these rules, thus rendering the inference unreliable. However, the relationships within these rules frequently co-occur, making traditional algorithms that rely solely on association occurrences within KG may erroneously assign them high confidence. In contrast, CSRL effectively leverages commonsense knowledge of LLMs, enabling it to reject such rules.

Furthermore, the CSRL algorithm is capable of assigning low confidence to dubious rules, such as $\mathit{aunt}(x, y) \leftarrow \mathit{inv\_nephew}(x, z_1) \wedge \mathit{inv\_uncle}(z_1, z_2) \wedge \mathit{father}(z_2, y)$. This is attributed to its integration of structured knowledge from KGs, allowing it to identify unreliable rules based on their appearance patterns within KGs.

Table 6: Rules Learned and Rejected by CSRL.

Datasets	Rules	Confidences
Family	$\text{brother}(x, y) \leftarrow \text{son}(x, z_1) \wedge \text{mother}(z_1, z_2) \wedge \text{brother}(z_2, y)$	High
	$\text{aunt}(x, y) \leftarrow \text{inv\_nephew}(x, z_1) \wedge \text{inv\_uncle}(z_1, z_2) \wedge \text{father}(z_2, y)$	Low
	$\text{sister}(x, y) \leftarrow \text{inv\_brother}(x, z) \wedge \text{brother}(z, y)$	Rejected
YAGO3-10	$\text{isPoliticianOf}(x, y) \leftarrow \text{isPoliticianOf}(x, z) \wedge \text{isLocatedIn}(z, y)$	High
	$\text{dealsWith}(x, y) \leftarrow \text{dealsWith}(x, z) \wedge \text{dealsWith}(z, y)$	Rejected

6 Conclusions

Logical rule learning is essential for reasoning, yet KG-based methods relying on co-occurrence often yield unreliable rules. LLMs provide commonsense knowledge but suffer from hallucinations. We propose CSRL, a framework that integrates KGs' structured knowledge with LLMs' commonsense to reduce hallucinations and enhance rule reliability.

7 Acknowledgments

This work is supported by the National Natural Science Foundation of China (62376054), Sichuan Science and Technology Program (2025YFHZ0236, DQ202411), Shenzhen Science and Technology Program (JCYJ20230807120008016), AVIC United Technology Center for Intelligent Decision-making and Coordinated Control Mechanism Model Research grant (1725-GSYS02000000001-ZB-Z013-0), Guangdong Basic and Applied Basic Research Foundation (2024A1515011634). It has benefitted from the Microsoft Accelerating Foundation Models Research (AFMR) grant program and the OpenAI Researcher Access program. Carl Yang was not supported by any fund from China.

References

[1] Haoting Chen, Sergio José Rodríguez Méndez, and Pouya Ghiasnezhad Omran. 2025. Open Local Knowledge Graph Construction from Academic Papers Using Generative Large Language Models. In *Companion Proceedings of the ACM on Web Conference 2025*. 2551–2559.

[2] Meiqi Chen, Yubo Ma, Kaitao Song, Yixin Cao, Yan Zhang, and Dongsheng Li. 2024. Improving Large Language Models in Event Relation Logical Prediction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 9451–9478.

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

[4] Kewei Cheng, Nesreen K Ahmed, and Yizhou Sun. 2023. Neural compositional rule learning for knowledge graph reasoning. *arXiv preprint arXiv:2303.03581* (2023).

[5] Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. 2022. Rlogic: Recursive logical rule learning from knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 179–189.

[6] Kewei Cheng, Ziqing Yang, Ming Zhang, and Yizhou Sun. 2021. UniKER: A unified framework for combining embedding and definite horn rule reasoning for knowledge graph inference. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 9753–9771.

[7] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[8] 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*. 413–422.

[9] Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919* (2022).

[10] Geoffrey E Hinton et al. 1986. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, Vol. 1. Amherst, MA, 12.

[11] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)* 54, 4 (2021), 1–37.

[12] Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 6384–6392.

[13] Suramya Jadhav, Suki Perumal, Yasmin Tadavi, Bikshita Dash, and Srinivasan Parthiban. 2025. Leveraging Large Language Models for Biomedical Knowledge Graph Construction and Querying: An Advanced NLP Approach. In *Companion Proceedings of the ACM on Web Conference 2025*. 2560–2566.

[14] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. *arXiv preprint arXiv:2106.10502* (2021).

[15] Erich Leo Lehmann, Joseph P Romano, and George Casella. 1986. *Testing statistical hypotheses*. Vol. 3. Springer.

[16] Hong Liu, Zhe Wang, Kewen Wang, Xiaowang Zhang, and Zhiyong Feng. 2025. Transfer Rule Learning over Large Knowledge Graphs. In *Proceedings of the ACM on Web Conference 2025*. 2135–2143.

[17] Lihui Liu, Zihao Wang, Jiaxin Bai, Yangqiu Song, and Hanghang Tong. 2024. New frontiers of knowledge graph reasoning: Recent advances and future trends. In *Companion Proceedings of the ACM Web Conference 2024*. 1294–1297.

[18] Ye Liu, Yao Wan, Lifang He, Hao Peng, and S Yu Philip. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 6418–6425.

[19] Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Chatrule: Mining logical rules with large language models for knowledge graph reasoning. *arXiv preprint arXiv:2309.01538* (2023).

[20] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061* (2023).

[21] Linhao Luo, Thuy-Trang Vu, Dinh Phung, and Gholamreza Haffari. 2023. Systematic assessment of factual knowledge in large language models. *arXiv preprint arXiv:2310.11638* (2023).

[22] Zaiqiao Meng, Fangyu Liu, Ehsan Shareghi, Yixuan Su, Charlotte Collins, and Nigel Collier. 2021. Rewire-then-probe: A contrastive recipe for probing biomedical knowledge of pre-trained language models. *arXiv preprint arXiv:2110.08173* (2021).

[23] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[24] Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 13679–13707.

[25] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2020. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. *arXiv preprint arXiv:2010.04029* (2020).

[26] Meng Qu and Jian Tang. 2019. Probabilistic logic neural networks for reasoning. *Advances in neural information processing systems* 32 (2019).

[27] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems* 32 (2019).

[28] Eric Salvat and Marie-Laure Mugnier. 1996. Sound and complete forward and backward chainings of graph rules. In *International Conference on Conceptual Structures*. Springer, 248–262.

[29] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207* (2021).

[30] Yuval Schwartz, Lavi Ben-Shimol, Dudu Mimran, Yuval Elovici, and Asaf Shabtai. 2025. Llmcloudhunter: Harnessing llms for automated extraction of detection rules from cloud-based cti. In *Proceedings of the ACM on Web Conference 2025*. 1922–1941.

[31] Oshani Seneviratne, Brendan Capuzzo, and William Van Woensel. 2025. Explainability-Driven Quality Assessment for Rule-Based Systems. In *Companion Proceedings of the ACM on Web Conference 2025*. 2133–2140.

[32] Zhenning Shi, Dan Zhao, Yijia Zhu, Guorui Xie, Qing Li, and Yong Jiang. 2025. Helios: Learning and Adaptation of Matching Rules for Continual In-Network Malicious Traffic Detection. In *Proceedings of the ACM on Web Conference 2025*. 2319–2329.

[33] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*. 697–706.

[34] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *arXiv preprint arXiv:2307.07697* (2023).

[35] Qiang Sun, Yuanyi Luo, Wenxiao Zhang, Sirui Li, Jichunyang Li, Kai Niu, Xiangrui Kong, and Wei Liu. 2025. Docs2KG: A Human-LLM Collaborative Approach to Unified Knowledge Graph Construction from Heterogeneous Documents. In *Companion Proceedings of the ACM on Web Conference 2025*. 801–804.

[36] Yanchao Tan, Wanzi Shao, Guofang Ma, and Carl Yang. 2025. Large Language Model Empowered Logical Relations Mining for Personalized Recommendation. In *Companion Proceedings of the ACM on Web Conference 2025*. 1326–1330.

[37] Xiaojuan Tang, Song-chun Zhu, Yitao Liang, and Muhan Zhang. 2024. RulE: Knowledge Graph Reasoning with Rule Embedding. In *Findings of the Association for Computational Linguistics ACL 2024*. 4316–4335.

[38] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*. 57–66.

[39] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[40] Guojia Wan, Shirui Pan, Chen Gong, Chuan Zhou, and Gholamreza Haffari. 2021. Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning. In *International Joint Conference on Artificial Intelligence*. International Joint Conference on Artificial Intelligence.

[41] Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. 2023. Boosting language models reasoning with chain-of-knowledge prompting. *arXiv preprint arXiv:2306.06427* (2023).

[42] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[43] Yilin Wen, Zifeng Wang, and Jimeng Sun. 2023. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729* (2023).

[44] Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems* 30 (2017).

[45] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129* (2019).

[46] Zirui Zhao, Wee Sun Lee, and David Hsu. 2024. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems* 36 (2024).

[47] Yueqin Zhu, Wenwen Zhou, Yang Xu, Ji Liu, and Yongjie Tan. 2017. Intelligent learning for knowledge graph towards geological data. *Scientific Programming* 2017, 1 (2017), 5072427.

A Datasets

We employ FB15K-237, WN18RR, YAGO3-10, and Family datasets in our research. They serve as benchmarks for evaluating KG models. Many state-of-the-art algorithms are currently developed based on these datasets. Statistical details of these datasets are presented in Table 1, including the number of relations, types of relations, and number of entities contained within each.

- FB15K-237 [38] is a refined version of the original FB15k dataset, designed to address issues of redundancy and leakage in training and test sets. It includes a broad range of structured data, derived from various sources, including wiki entries submitted by users. This dataset is widely used for benchmarking due to its extensive coverage of relationships and entities, making it a critical resource for developing and testing knowledge graph embedding models.
- Family [10] exemplifies various familial relationships among individuals. Its simplicity and intuitive nature make it an excellent dataset for interpreting complex relationships in a confined scope. This allows researchers to easily understand and visualize how knowledge graph models learn and predict familial connections.
- WN18RR [7] is an essential dataset used for evaluating knowledge graph models. It's a revised version of WN18, created to eliminate issues related to inverse relations that made the original dataset easier to predict. WN18RR is structured to serve as a dictionary and thesaurus, facilitating automatic text analysis. Its entities represent linguistic concepts known as word senses, and the relationships illustrate the lexical connections among them.
- YAGO3-10 [33] is a subset of YAGO, a comprehensive semantic knowledge base. It is compiled from multiple authoritative sources such as Wikipedia, WordNet, WikiData, and GeoNames, allowing it to integrate diverse types of information into a single descriptive framework. This dataset is widely leveraged to test knowledge graph models due to its rich semantic structure and the vast quantity of included facts, which offers a challenging platform for research in knowledge representation.

Table 1: Statistics of Data.

Datasets	# Data	# Relation	# Entity
FB15K-237	310,116	237	14,541
WN18RR	93,003	11	40,943
YAGO3-10	1,089,040	37	123,182
Family	28,356	12	3,007

B Hyperparameter Sensitivity

We conduct sensitivity tests for hyperparameters on the KG completion task with the Family dataset, based on GPT-4o.

The purpose of weight parameter α is simply to combine the frequency of patterns within the KG with the commonsense from LLM, so it is not sensitive when both confidence components are already present and accounted for. According to Figure 1, we can tell that both commonsense confidence and structured confidence contribute to the final performance, since the MRR of $\alpha = 0$ (0.85) and $\alpha = 1$ (0.84) are not good. Besides, α appears relatively insensitive between 0.25 to 0.75.

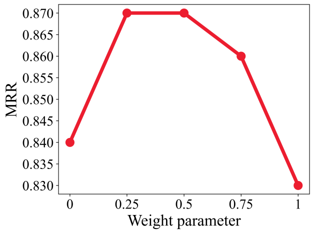


Figure 1: α balances two kind of confidences.

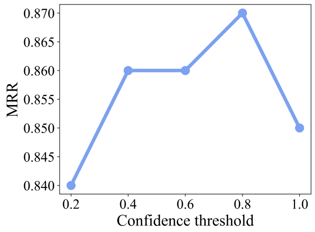


Figure 2: ρ_0^c is the threshold of confidence.

For confidence threshold ρ_0^c, The results are shown in Figure 2. When $\rho_0^c = 0.2$, the threshold is too low, allowing unreliable rules to be learned, which results in a lower MRR (0.84). As ρ_0^c increases, the algorithm's performance improves. However, when $\rho_0^c = 1$, all rules containing instance errors are completely eliminated. We consider this approach overly strict, leading to a certain degree of decline in MRR (0.85). Besides, within the range of ρ_0^c values from 0.4 to 0.8, the performance is relatively stable and not highly sensitive to changes in ρ_0^c.

Evaluation threshold ϵ can balance the trade-off between quality and efficiency. The results in Table 2 show that with $\epsilon = 2$, more rules are learned (369) with fewer queries (3642), but a higher error rate (25 incorrect), resulting in poor performance (MRR = 0.82). With $\epsilon = 10$, fewer rules are identified (116) with more queries (8222), and fewer errors (1), but performance is still suboptimal (MRR = 0.84). $\epsilon = 5$ provides a better balance between rule quantity and reliability, leading to improved performance. So setting ϵ around 5 ensures rule reliability and prevents redundancy.

Table 2: ϵ balances quality and efficiency.

ϵ	Query times	Learned rules	Incorrect rules	MRR
2	3642	369	25	0.82
5	6283	241	3	0.87
10	8222	116	1	0.84

Table 3: prompt for consistency study.

Prompt	Content
Original	Person <entity 1> has a relationship of "<relation 1>" with person <entity 2>. ... . Based on the facts above, can we infer that person <entity 1> has a relationship of "<relation 4>" with person <entity 4>? Please answer with "Yes" or "No". Do not output other words.
Prompt 1	Based on the following information, can we infer that person <entity 1> has a relationship of "<relation 4>" with person <entity 4>? Person <entity 1> has a relationship of "<relation 1>" with person <entity 2>. ... . Please answer with "Yes" or "No". Do not output other words.
Prompt 2	Person <entity 1> has a relationship of "<relation 1>" with person <entity 2>. ... . Based on the facts above, can we infer that person <entity 1> has a relationship of "<relation 4>" with person <entity 4>? Please answer with "Yes" or "No". Please provide the result in JSON format with the following structure: {"explanation": "Your explanation here.", "answer": "Yes" or "No"}
Prompt 3	Person <entity 1> is the <relation 1> of person <entity 2>. ... . Based on the facts above, can we infer that person <entity 1> is the <relation 4> of person <entity 4>? Please answer with "Yes" or "No". Do not output other words.

C Statistics Algorithms

C.1 Purposes and Methods of Statistics Algorithms

In the human-based semantic assessment, we employ the Kruskal-Wallis test to investigate two objectives: first, to demonstrate the consistency of scores given by different raters for the same set of rules; second, to identify whether there are significant differences in scores among the three sets of rules.

The Kruskal-Wallis test is a non-parametric statistical method used to determine if there are significant differences in the medians of multiple independent samples. The test statistic H measures the significance of differences between groups. A larger H value indicates greater differences between groups. For p-value, The null hypothesis posits that the medians of all groups are equal [15].

We utilize Dunn's test for pairwise comparisons of the scores of three rule sets to assess which set achieves the highest overall scores.

Dunn's test is a non-parametric post-hoc analysis used to identify specific differences between pairs of group medians following a significant Kruskal-Wallis test result.The test statistic Z measures the significance of differences between pairs of groups. A larger absolute Z value typically indicates greater differences between the pairs. A negative Z value indicates that the median of the first group is lower than the median of the second group. For p-value, the null hypothesis posits that the medians of the pair of groups being compared are equal [15].

C.2 Result of Kruskal-Wallis Test

We perform the Kruskal-Wallis test on scores given by different raters to the same algorithm, to ensure raters provide consistent scores and the ratings are valid. The test statistic H and the p-value of three algorithms' scores are shown in table 4

Table 4: Results of Kruskal-Wallis Test.

Models	Test Statistic H	P-value
NCRL	2.952340399	0.085753
ChatRule (GPT-4)	0.257477501	0.879204
CSRL (GPT-4o)	3.108556683	0.211342

D Prompt Sensitivity

The detailed prompts we used to test the prompt sensitivity of CSRL are listed in Table 3

E LLM Cost Analysis

CSRL effectively reduces interaction costs with LLMs due to the following advantages: (1) CSRL can dynamically adjust the sampling direction under the guidance of the commonsense knowledge from LLMs to select more meaningful rules, reducing useless interactions. (2) CSRL utilizes the structured knowledge from KGs to textualize rules, thereby minimizing additional explanations in prompts. Table 5 presents the costs of rule learning across various datasets based on GPT-4o and GPT-3.5. For GPT-4o, we utilize the gpt-4o-2024-08-06 API in experiments, priced at $2.50 per million input tokens and $10.00 per million output tokens. As for GPT-3.5, we use the gpt-3.5-turbo-0125 API, with a cost of $0.50 per million input tokens and $1.50 per million output tokens.

Notably, since the algorithm also performs well with GPT-3.5, it is feasible to choose less powerful and cheaper LLMs in practical applications. This approach ensures a cost-effective rule learning process without compromising performance.

Table 5: costs of learning rules on datasets (in us dollars $).

LLMs	YAGO3-10	Family	WN18RR	FB15K-237
GPT-4o	0.226	0.139	0.792	6.705
GPT-3.5	0.114	0.124	0.231	1.620

F Common Errors in Rules Generated Solely by LLMs

There are two more kinds of common errors in rules generated solely by LLMs. Firstly, LLMs tend to generate many relations that do not exist in the KG due to hallucinations, rendering these rules unusable for downstream tasks. For example, based on the Family dataset, the LLM generates the rule $brother(x, y) \leftarrow son(x, z) \wedge parent(z, y)$. Although it is logically valid, the dataset lacks the *parent* relation, having only *father* and *mother* relations, which prevents this rule from being applied to downstream tasks based on this dataset.

Secondly, LLMs may misinterpret the meanings of relations due to hallucinations. For instance, based on the YAGO3-10 dataset, the

LLM generates the rule $\mathit{isConnectedTo}(x, y) \leftarrow \mathit{wasBornIn}(x, z) \wedge \mathit{hasOfficialLanguage}(z, y)$. Here, the LLM assumes *isConnectedTo* means a connection between people and objects, but in fact, it refers to a hierarchical relationship between institutions.

G Prompt Templates

G.1 Prompt for LLMs Evaluation based on Instances

We utilize the prompts below to transform triples into text, guiding LLMs in assessing instances of logical rules. Although the textual content varies slightly across different datasets due to differing contexts, the underlying structure remains consistent. Terms corresponding to the Family, WN18RR, YAGO3-10, and FB15K-237 datasets are People, Concept, <empty>, and Entity. In a relationship where "inv_" is present, the order of the two entities will be reversed.

Prompt for CSRL

<Term> <entity 1> has a relationship of "<relation 1>" with <Term> <entity 2>.
<Term> <entity 2> has a relationship of "<relation 2>" with <Term> <entity 3>.
<Term> <entity 3> has a relationship of "<relation 3>" with <Term> <entity 4>.
Based on the facts above, can we infer that <Term> <entity 1> has a relationship of "<relation 4>" with <Term> <entity 4>?
Please answer with "Yes" or "No". Do not output other words.

G.2 Prompt for Rule Generation by LLMs solely

Here is the prompt for the example we use in Section 1 and Section 5.1 to generate rule based solely on LLMs. To ensure comprehensive guidance, we use a "step-by-step" zero-shot prompting approach and provide a detailed explanation of the rules, along with examples.

Prompt for LLM Generating Rule Solely

Logical rules can be expressed in the form: Rule Body 1 (A, B) & Rule Body 2 (B, C) → Rule Head (A, C). This means that if the conditions in the rule bodies are satisfied, then the rule head will occur.
For example: Son (A, B) & Father (B, C) → Brother (A, C). It means if A is the son of B and B is the son of C, then A is the brother of C.
Based on the example above, please think step by step and generate 100 rules with the following relationships: aunt, brother, daughter, father, husband, mother, nephew, niece, sister, son, uncle, wife, inv_aunt, inv_brother, inv_daughter, inv_father, inv_husband, inv_mother, inv_nephew, inv_niece, inv_sister, inv_son, inv_uncle, inv_wife. 'Inv_' means inverse relationship.

G.3 Prompt for LLM-based Semantic Assessment

Here is the prompt used in our experiments for automated semantic reliability assessment based on high-performance LLMs. The meanings of terms are the same as those in Appendix G.1.

Prompt for Assessment

A, B, C, D, E, F are entities. Which of the following two logical inferences is more reliable?
Option 1:
<Term> A has a relationship of "<relation 1>" with <Term> B.
<Term> B has a relationship of "<relation 2>" with <Term> C.
So we can infer that <Term> A has a relationship of "<relation 4>" with <Term> C.
Option 2:
<Term> D has a relationship of "<relation 5>" with <Term> E.
<Term> E has a relationship of "<relation 6>" with <Term> F.
So we can infer that <Term> D has a relationship of "<relation 8>" with <Term> F.
Think step by step to assess whether these two options are reasonable reasoning. Firstly, for each option, judge whether the inference in the last sentence is reliable based on what is known in previous sentences. Then, compare the reliability of the two options and choose the higher option.
Please answer with "Option 1" or "Option 2". Output in JSON format, for example: "Explanation": "", "Answer (Option 1 or Option 2)": ""