Iterative Natural Logic Verification for Multi-hop Question Answering

Jihao Shi, Liang Yang*, Hongfei Lin
School of Computer Science and Technology, Dalian University of Technology, China
{jhshi,liang}@dlut.edu.cn

Abstract

Multi-hop question answering requires composing multiple pieces of evidence to derive the correct answer, making it particularly vulnerable to hallucinated intermediate reasoning steps. A common failure mode arises when neural generators produce intermediate sentences that are fluent but logically inconsistent with the underlying facts, which can subsequently mislead downstream reasoning. In this paper, we propose ITERLOGIC, an iterative natural logic verification framework for multi-hop question answering, which explicitly integrates symbolic verification into the generation process. Specifically, a neural generator composes candidate intermediate sentences from retrieved facts, while a natural logic verifier evaluates their logical consistency with the source evidence. Low-scoring candidates are iteratively rejected and regenerated, allowing the system to progressively mitigate hallucinated compositions. This lightweight interaction upgrades natural logic from a passive scoring component to an active verification signal that guides intermediate sentence generation. Experiments on QASC, WorldTree, and WikiHop demonstrate that ITERLOGIC improves answer accuracy, highlighting the effectiveness of iterative verification for reliable multi-hop reasoning.

1 Introduction

Multi-hop question answering (QA) (Jansen et al., 2018; Welbl et al., 2018; Khot et al., 2020) aims to answer questions by composing information from multiple pieces of evidence (Jhamtani and Clark, 2020; Tafjord et al., 2021; Dalvi et al., 2021; Hong et al., 2022; Fan et al., 2024; Yuan et al., 2024; Zhang et al., 2024; Chen et al., 2024b; Wang et al., 2025b), rather than relying on a single supporting fact. This setting covers a broad range of benchmarks, including two-hop compositional QA such as QASC (Khot et al., 2020), explanation-based QA such as WorldTree (Jansen et al., 2018), and document-level multi-hop reading comprehension such as WikiHop (Welbl et al., 2018). While recent neural models have achieved strong performance on these tasks (Islam et al., 2024; Shi et al., 2025a,c), multi-hop QA remains challenging due to the need to explicitly construct intermediate reasoning steps that bridge disparate facts in a logically coherent manner.

A central difficulty in this process lies in the generation of intermediate reasoning sentences. Modern neural generators are often capable of producing fluent and plausible intermediate sentences; however, such sentences may be logically inconsistent with the underlying evidence or introduce unsupported concepts (Farquhar et al., 2024; Huang et al., 2025a; Pandit et al., 2025; Zhang et al., 2025). These hallucinated intermediate steps can be particularly harmful in multi-hop QA, as errors introduced at early stages of reasoning tend to propagate and mislead downstream inference. Existing approaches (Shi et al., 2025c; Lee and Lee, 2025) typically adopt a generate-then-score paradigm, where intermediate sentences are generated once and evaluated afterward. In this setting, symbolic reasoning components, when used, often serve as passive scoring modules without the ability to influence the generation process itself, limiting their effectiveness in preventing hallucinations.

To address this limitation, we propose **ITERLOGIC**, an iterative natural logic verification framework for multi-hop question answering. Instead of treating verification as a post hoc scoring step, ITERLOGIC tightly integrates natural logic into the generation loop. Given a pair of retrieved facts, a neural generator composes candidate intermediate sentences, while a natural logic verifier evaluates their logical consistency with the source evidence. Candidates that receive low verification scores are rejected and regenerated, enabling the system to iteratively refine intermediate sentences

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 43407–43419
October 24-29, 2026 ©2026 Association for Computational Linguistics

and progressively eliminate hallucinated compositions. This lightweight interaction allows symbolic verification to actively guide generation, without requiring additional supervision.

We evaluate ITERLOGIC on three representative multi-hop QA benchmarks: QASC (Khot et al., 2020), WorldTree (Jansen et al., 2018), and WikiHop (Welbl et al., 2018). Experimental results show that ITERLOGIC consistently improves answer accuracy compared to strong baselines, demonstrating that iterative natural logic verification leads to more reliable intermediate reasoning and more accurate final predictions. Further analysis indicates that the proposed framework effectively reduces logically inconsistent intermediate sentences, highlighting the importance of integrating symbolic verification into the generation process for multi-hop reasoning tasks.

Our key contributions can be summarized as follows:

- We reformulate knowledge composition as a constrained generation problem, where neural generation is iteratively guided and filtered by symbolic natural logic verification.
- We propose **ITERLOGIC**, an iterative natural logic verification framework that upgrades natural logic from a passive evaluator to an active signal guiding intermediate sentence generation.
- We demonstrate the effectiveness of ITERLOGIC on multiple multi-hop QA benchmarks, showing improved accuracy and more reliable reasoning. Our method yields comparable performance across benchmarks.

2 Preliminaries

2.1 Problem Setup

Given a question Q and a set of candidate answers $\{o_1, \ldots, o_M\}$, the goal is to select the correct answer by composing information from multiple pieces of external evidence.

For each candidate answer o_i, a retrieval module identifies a set of relevant facts from an external knowledge source. Following prior work (Khot et al., 2020; Shi et al., 2025c), we focus on fact-pair composition, where pairs of retrieved facts (f_1, f_2) are combined to support reasoning in a principled manner. The central challenge addressed in this work is to generate intermediate sentences that validly compose such fact pairs while avoiding hallucinated content that is not logically supported by the evidence.

Relation: Name	Example	Set Theoretic Definition
$\equiv$: Equivalence	couch $\equiv$ sofa	$x = y$
$\sqsubseteq$: Forward Entailment	tulip $\sqsubseteq$ flower	$x \subset y$
$\sqsupseteq$: Reverse Entailment	flower $\sqsupseteq$ tulip	$x \supset y$
$\wedge$: Negation	true $\wedge$ false	$x \cap y = \varnothing \wedge x \cup y = U$
$\Downarrow$: Alternation	sunflower $\Downarrow$ tulip	$x \cap y = \varnothing \wedge x \cup y \neq U$
$\smile$: Cover	mammal $\smile$ nonhuman	$x \cap y \neq \varnothing \wedge x \cup y = U$
$\#$: Independence	banana $\#$ ocean	All other cases

Table 1: A set $\mathcal{L}$ of seven logical relations proposed by MacCartney and Manning (MacCartney and Manning, 2009).

2.2 Natural Logic

Natural Logic (Nairn et al., 2006; MacCartney and Manning, 2009; Angeli et al., 2016; Sanchez, 2016; Shi et al., 2021, 2025b) investigates how logical inference can be carried out directly within natural language itself, rather than through an external symbolic formalism. Unlike traditional approaches based on first-order logic, this framework treats syntactic structures as primary carriers of semantic information and performs reasoning without explicit translation into abstract logical expressions. The underlying intuition can be traced back to Aristotle's syllogistic tradition (Rose, 1965), which already emphasized inference patterns grounded in linguistic form.

A central contribution of natural logic is the relational calculus proposed by MacCartney and Manning (2009). This calculus defines a set of seven semantic relations, $\mathcal{L} = \{\equiv, \sqsubseteq, \sqsupseteq, \wedge, \Downarrow, \smile, \#\}$, which serve as the basic units for reasoning. As reported in Table 1, these relations encode equivalence, entailment, contradiction, and various forms of semantic independence between expressions, thereby providing a rich vocabulary for modeling meaning comparisons in text. Together with this relational system, monotonicity theory (MacCartney and Manning, 2009; Icard III and Moss, 2014) plays a crucial role by describing how semantic relations interact with linguistic environments. Depending on whether a context is upward or downward monotone, an existing relation may either be preserved or systematically transformed, as exemplified in Table 2.

In practice, reasoning under the natural logic framework follows a compositional procedure (MacCartney and Manning, 2009; Angeli and Manning, 2014; Feng et al., 2020; Shi et al., 2025a). Sentence pairs are first decomposed into aligned

r	$\equiv$	$\sqsubseteq$	$\sqsupseteq$	$\smile$	$\downharpoonleft\upharpoonright$	$\curlywedge$	$\#$
$\delta(r)$	$\equiv$	$\sqsupseteq$	$\sqsubseteq$	$\downharpoonleft\upharpoonright$	$\smile$	$\curlywedge$	$\#$

Table 2: The projection function δ projects an input relation r into a different relation under downward monotonicity.

textual segments that correspond to each other semantically. Pairwise relations are then assigned to these segments, after which contextual monotonicity is applied to propagate local inferences through the sentence structure. By composing the resulting relations across all aligned components, the model ultimately determines the global inference relation holding between the two sentences.

3 Methods

We present ITERLOGIC, an iterative natural logic verification framework for multi-hop question answering. ITERLOGIC is designed to reduce hallucinated intermediate reasoning steps by tightly integrating symbolic verification into the generation process. Instead of generating intermediate sentences once and scoring them afterward, ITERLOGIC enables a verifier to actively control generation through iterative rejection and regeneration. The overall architecture of ITERLOGIC is depicted in Figure 1.

3.1 Knowledge Selection

Answering a compositional question requires integrating information from multiple pieces of external knowledge. To this end, we retrieve pairs of supporting facts from an external knowledge base $\mathcal{K}$. For each candidate answer o_i, we transform the question–answer pair into a declarative hypothesis S_i, following standard multiple-choice QA formulations. The hypothesis S_i serves as the query to initiate a two-stage information retrieval procedure. This process follows the general strategy introduced by Khot et al. (2020) and is implemented with ElasticSearch (Gormley and Tong, 2015). In the first stage, an initial fact f_1 is retrieved from $\mathcal{K}$ based on its relevance to S_i. In the second stage, a complementary fact f_2 is retrieved conditioned on f_1.

3.2 Intermediate Sentence Generation

Given a fact pair (f_1, f_2), the generator produces a candidate intermediate sentence:

$$s^{(t)} = \mathcal{G}(f_1, f_2), \tag{1}$$

where t denotes the iteration index. The generator $\mathcal{G}$ is implemented using a pretrained large language model. To encourage diversity across iterations, we vary decoding randomness when regeneration is triggered.

3.3 Natural Logic Verifier

The natural logic verifier in ITERLOGIC is responsible for assessing whether a generated intermediate sentence is logically consistent with a given source fact. This verification process is grounded in span-level natural logic reasoning and proceeds through three stages: span chunking and alignment, logical relation prediction, and deterministic inference via an automaton.

Span Chunking and Alignment. Given a generated sentence and a source fact, we first segment both texts into a sequence of spans using the chunker (Akbik et al., 2019). To ensure semantic completeness, spans that do not contain content words are merged with adjacent spans. Following chunking, we align spans between the generated sentence and the source fact. We formulate span alignment as an optimal bipartite matching problem and solve it using the Hungarian algorithm (Crouse, 2016). This alignment produces a set of span pairs $\{(s_1, \tilde{s}_1), \ldots, (s_n, \tilde{s}_n)\}$, where each pair corresponds to semantically related text segments.

Logical Relation Prediction. For each aligned span pair $(s_n, \tilde{s}_n)$, the verifier predicts a probabilistic distribution over a predefined set of natural logic relations $\mathcal{L}$. We employ a DeBERTa-based classifier fine-tuned for natural logic inference (Shi et al., 2021), which outputs:

$$p_n^r = \mathrm{softmax}(\mathrm{DeBERTa}(s_n, \tilde{s}_n)), \tag{2}$$

where p_n^r denotes the probability assigned to each logical relation for the n-th span pair. Because the validity of natural logic relations depends on contextual monotonicity, we further project the predicted relations according to their monotonicity environments. To this end, we fine-tune a monotonicity classifier following the method of Rozanova et al. (2021). This projection maps each predicted relation to its contextually valid counterpart, yielding updated probabilities $p_n^{r'}$.

Deterministic Inference. The sequence of projected logical relations is then evaluated using a

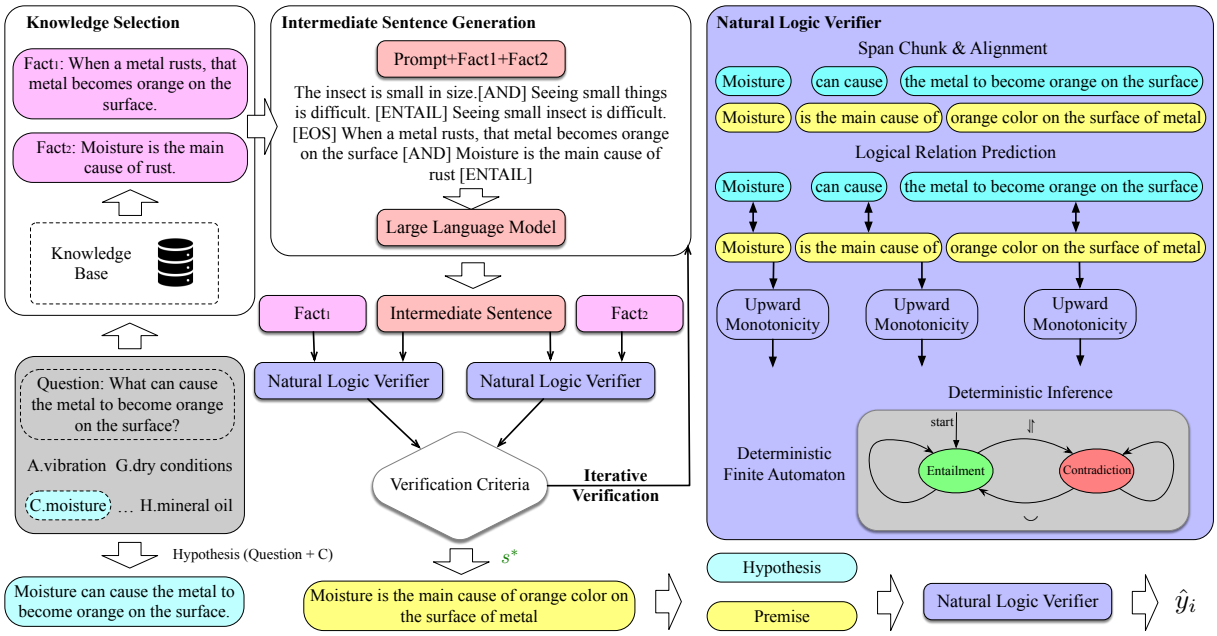


Figure 1: Overview of the proposed **ITERLOGIC** framework.

deterministic finite automaton (DFA), which aggregates local span-level relations into a global inference decision. The overall verification score is computed as the product of the probabilities of the projected relations:

$$p = \prod_{n} p_n^{r'}. \tag{3}$$

While the original NaturalLI framework (Angeli and Manning, 2014) employs a multi-state DFA to distinguish entailment, contradiction, and unknown outcomes, ITERLOGIC adopts a simplified automaton with two terminal states: entailment and contradiction. This binary design streamlines verification and provides a sharper signal for controlling iterative regeneration, making it particularly suitable for intermediate sentence validation and answer selection.

3.4 Iterative Verification and Regeneration

To assess the validity of a composed sentence $s^{(t)}$, the verifier computes natural logic consistency scores between $s^{(t)}$ and each source fact:

$$\begin{aligned} p_1 &= V(s^{(t)} \mid f_1), \\ p_2 &= V(s^{(t)} \mid f_2), \end{aligned} \tag{4}$$

where each probability reflects the likelihood that $s^{(t)}$ is logically supported by the corresponding fact under natural logic constraints.

We aggregate the two probabilities into a single composition score using a log-sum formulation:

$$\text{Score}(s^{(t)}) = \log(p_1) + \log(p_2). \tag{5}$$

Compared to averaging-based alternatives, this formulation amplifies weak logical support and provides a more discriminative signal for iterative rejection. Given the composition score, ITERLOGIC applies a simple yet effective rejection criterion. If $\text{Score}(s^{(t)})$ falls below a predefined threshold, the candidate sentence is rejected and a new sentence is generated. The iterative process is defined as:

$$s^* = \arg\max_{t \leqslant T} \text{Score}(s^{(t)}), \tag{6}$$

where T denotes the maximum number of generation attempts. In our experiments, the threshold is fixed to 0.5, and the maximum number of generation attempts is set to 5. This rejection-based interaction allows ITERLOGIC to progressively filter out hallucinated compositions without requiring explicit error annotations. Crucially, regeneration is formulated as sampling from the language model, rather than as a diagnostic-feedback loop. This design keeps the framework lightweight, requires neither model retraining nor verifier-specific prompt engineering, and avoids propagating potentially noisy span-level rationales into the generation process. Instead, the verifier constrains the output by selecting candidates with stronger logical support.

3.5 Hypothesis Entailment Scoring

After obtaining a logically consistent intermediate sentence through iterative natural logic verification,

ITERLOGIC evaluates how well the validated sentence supports a candidate answer. For each option o_i, we evaluate how well the validated sentence supports the corresponding hypothesis S_i.

Given the selected intermediate sentence s^* associated with option o_i, we compute a hypothesis-level entailment score that measures whether s^* entails S_i. This step is performed using the natural logic verifier, which outputs a probability distribution over entailment labels. We denote the entailment probability as:

$$\hat{y}_i = V(S_i \mid s^*). \tag{7}$$

We pair the question with each candidate choice to construct M input sequences, where the sequence corresponding to the correct answer is treated as a positive instance, and all remaining sequences are treated as negative instances. For each input sequence, the model predicts the probabilities of the positive and negative classes, denoted as $\hat{y}_i^+$ and $\hat{y}_i^-$, respectively. Based on these predictions, we define the training objective using a binary cross-entropy loss over all candidates:

$$\begin{aligned}\mathcal{L} &= -\sum_{i=1}^{M} \left(o_i^+ \log \hat{y}_i^+ + o_i^- \log \hat{y}_i^-\right) \\ &= -\log \hat{y}_{\text{gold}}^+ - \sum_{i=1,\, i\neq \text{gold}}^{M} \log \hat{y}_i^-,\end{aligned} \tag{8}$$

where gold denotes the index of the correct answer. The label indicators o_i^+ and o_i^- are defined as follows: if the i-th candidate is the correct answer, then $o_i^+ = 1$ and $o_i^- = 0$; otherwise, $o_i^+ = 0$ and $o_i^- = 1$.

4 Experimental Settings

4.1 Datasets

We evaluate our approach on three benchmark datasets that require retrieving and synthesizing external facts for multiple-choice question answering: QASC (Khot et al., 2020), WorldTree (Jansen et al., 2018), and WikiHop (Welbl et al., 2018). Detailed descriptions of these datasets are provided in Appendix A.

4.2 Baselines

Large Language Models. We evaluate two categories of large language models: proprietary API-based models and open-weight models. For the former, we use GPT-3.5 and GPT-4 via the OpenAI API (OpenAI, 2022; Achiam et al., 2023), specifically `gpt-3.5-turbo-1106` and `gpt-4-1106-preview`. For the latter, we consider representative open-weight LLMs, including LLAMA 2 (Touvron et al., 2023), LLAMA 3 (AI@Meta, 2024), Vicuna (Chiang et al., 2023), ChatGLM (Du et al., 2022), Baichuan (Yang et al., 2023), Mistral (Jiang et al., 2023), Gemma (Mesnard et al., 2024), Qwen (Bai et al., 2023), and Flan-T5 (Chung et al., 2024). To ensure fair comparison, we use chat-oriented variants with parameter sizes comparable to our model.

Fine-tuned Models. We additionally compare against fine-tuned baselines trained on the target datasets, including BERT$_{large}$ (Devlin et al., 2019), RoBERTa$_{large}$ (Liu et al., 2019), and DeBERTa$_{large}$ (He et al., 2023). We further include TEAM (Ghosal et al., 2022), a strong baseline that formulates MCQA as a binary classification problem using `DeBERTa-large`. In addition, we compare with several neural-symbolic reasoning approaches. NeuNLI (Shi et al., 2021) integrates natural logic into neural inference, while DILR (Wang and Pan, 2022) performs first-order logical reasoning over query-relevant facts. We also consider FINAL (Shi et al., 2025a), which combines first-order logic with natural logic for question answering. Finally, we include Dual-NeSy (Shi et al., 2025c), a dual-system framework that employs fast, heuristic reasoning for knowledge composition and structured, multi-step symbolic reasoning for deliberate logical verification. Details of the evaluation metrics and implementation settings are provided in Appendix A.

5 Experimental Results and Analysis

5.1 Main Results

Table 3 reports the performance of ITERLOGIC on QASC, WorldTree, and WikiHop. Overall, ITERLOGIC achieves strong and consistent results across all three benchmarks, outperforming prior neuro-symbolic and neural baselines.

On QASC, ITERLOGIC attains an accuracy of 90.43%, establishing the best performance among all compared models. This result demonstrates the effectiveness of iterative natural logic verification in two-hop compositional question answering. Compared to previous neuro-symbolic approaches such as FINAL and Dual-NeSy, ITERLOGIC further improves accuracy by enforcing logical consistency during generation rather than relying on one-shot verification. On WorldTree, ITERLOGIC

Model	QASC Accuracy (%)	WorldTree Accuracy (%)			WikiHop Accuracy (%)				
		Easy	Challenge	Overall	Publisher	Developer	Country	Record_label	Average
Open-weight LLMs 6B/7B/8B									
Baichuan2-Chat-7B (Yang et al., 2023)	45.43	72.18	54.12	66.56	38.89	34.48	62.89	22.97	39.81
LLAMA 2-CHAT-7B (Touvron et al., 2023)	33.26	64.38	45.88	58.62	37.04	24.14	41.75	42.76	36.43
Vicuna-7B (Chiang et al., 2023)	48.37	75.20	58.76	70.09	51.85	48.28	71.13	20.49	47.94
ChatGLM2-6B (Du et al., 2022)	41.52	76.37	56.70	70.25	42.59	37.93	72.68	40.64	48.46
ChatGLM3-6B (Du et al., 2022)	56.20	81.72	68.04	77.47	72.22	58.62	73.71	59.72	66.07
Mistral-7B-Instruct-v0.2 (Jiang et al., 2023)	60.76	82.42	74.23	79.87	64.81	51.72	72.68	64.66	63.47
Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)	58.91	82.77	76.03	80.67	64.81	51.72	77.32	67.84	65.43
Gemma-7B-it (Mesnard et al., 2024)	64.13	85.33	72.16	81.23	61.11	51.72	69.59	41.70	56.03
Gemma-1.1-7B-it (Mesnard et al., 2024)	66.20	90.45	81.70	87.73	62.96	48.28	71.65	38.16	55.26
Qwen-7B-Chat (Bai et al., 2023)	66.85	87.19	67.53	81.07	51.85	44.83	68.04	42.05	51.69
Qwen1.5-7B-Chat (Bai et al., 2023)	72.61	91.85	78.35	87.65	66.67	51.72	75.77	65.72	64.97
LLAMA 3-8B-INSTRUCT (AI@Meta, 2024)	71.30	91.50	80.41	88.05	62.96	48.28	76.29	55.83	60.84
Open-weight LLMs 11B/13B									
Baichuan2-Chat-13B (Yang et al., 2023)	56.63	82.07	68.04	77.71	44.44	34.48	50.00	33.92	40.71
LLAMA 2-CHAT-13B (Touvron et al., 2023)	41.52	65.89	47.42	60.14	55.56	51.72	64.43	62.19	58.48
Vicuna-13B (Chiang et al., 2023)	58.80	82.89	69.85	78.83	62.96	41.38	73.71	44.52	55.64
FLAN-T5-11B (Chung et al., 2024)	78.91	88.94	79.64	86.05	59.26	48.28	72.16	70.67	62.59
LLMs API									
GPT-3.5 (OpenAI, 2022)	77.72	94.76	89.43	93.10	64.81	65.52	75.77	71.02	69.28
GPT-4 (Achiam et al., 2023)	86.52	95.34	91.24	94.07	75.93	68.97	79.38	72.44	74.29
Finetuned Models									
BERT$_{large}$ (Devlin et al., 2019)	73.15	52.62	31.96	46.19	87.04	75.86	78.87	80.57	80.59
RoBERTa$_{large}$ (Liu et al., 2019)	73.26	57.04	35.05	50.20	88.89	75.86	81.96	81.63	82.09
DeBERTa$_{large}$ (He et al., 2023)	85.65	81.61	65.72	76.66	90.74	79.31	82.47	83.39	83.98
TEAM-DeBERTa (Ghosal et al., 2022)	89.35	88.82	78.87	85.73	88.89	79.31	82.99	83.04	83.56
DILR-DeBERTa (Wang and Pan, 2022)	47.66	59.14	47.68	55.57	88.89	82.76	84.54	83.75	84.99
NeuNLI-DeBERTa (Shi et al., 2021)	88.15	89.87	79.12	86.53	90.74	82.76	84.02	83.04	85.14
FINAL (Shi et al., 2025a)	90.33	**91.97**	81.70	88.77	**94.44**	86.21	85.05	85.51	87.80
Dual-NeSy (Shi et al., 2025c)	89.02	91.73	83.25	89.09	92.59	**89.66**	85.57	84.81	88.16
ITERLOGIC (Ours)	**90.43**	**91.97**	**83.76**	**89.41**	**94.44**	**89.66**	**86.08**	**85.87**	**89.01**

Table 3: Performance results on the QASC, WorldTree and WikiHop datasets. The best results achieved by open source models are marked in **bold**. The best results across all models are marked in underline.

achieves an overall accuracy of 89.41%, with particularly strong performance on the challenge subset (83.76%). These results indicate that iterative verification is particularly effective in more challenging multi-hop settings, as evidenced by the improvement on the WorldTree challenge subset. On WikiHop, ITERLOGIC consistently outperforms both open-weight and finetuned baselines across all four subtasks. It achieves the best average accuracy of 89.01%. This suggests that the proposed framework generalizes well in multi-hop reasoning scenarios.

These results show that ITERLOGIC provides a robust and general solution for multi-hop question answering. By integrating natural logic verification into an iterative generation loop, ITERLOGIC yields more reliable intermediate reasoning and consistently improves answer accuracy across diverse multi-hop QA benchmarks.

5.2 Ablation Study

To validate the effectiveness of the natural logic verifier, we conduct an ablation study on the WorldTree dataset by replacing it with an alternative neural NLI scorer while keeping all other components unchanged.

As shown in Table 4, the ablation results demonstrate the importance of both the iterative rejection–regeneration mechanism and the natural logic verifier. Removing iteration, which reduces the framework to one-shot generation followed by verification, decreases overall accuracy from 89.41% to 89.09%. This result shows that regeneration is not merely redundant resampling; rather, it gives the verifier multiple opportunities to filter out unsupported intermediate sentences and select a more logically consistent candidate.

Model	WorldTree Accuracy (%)		
	Easy	Challenge	Overall
ITERLOGIC (Ours)	91.97	83.76	89.41
w/o iteration	91.37	83.25	89.09
w/o verifier	91.62	79.90	87.97

Table 4: Ablation study on the WorldTree dataset.

Replacing the natural logic verifier with a standard NLI scorer further reduces the overall accuracy to 87.97%, with a more substantial 3.86-point drop on the Challenge subset. This degradation highlights the importance of natural logic verification for assessing logical consistency during intermediate composition. Unlike standard NLI scoring, the natural logic verifier produces explicit logical consistency scores that directly guide rejection and regeneration, providing a structured control signal

Model	QASC Accuracy (%)	WorldTree Accuracy (%)		
		Easy	Challenge	Overall
DeepSeek-R1-Distill-LLaMA-8B (Guo et al., 2025)	71.52	85.68	69.59	80.67
DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025)	68.70	87.66	73.45	83.24
Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)	75.43	88.01	77.58	84.76
LLaMA 3-8B-Instruct (AI@Meta, 2024)	72.93	90.22	79.90	87.01
Qwen-7B-Chat (Bai et al., 2023)	77.07	91.15	78.09	87.09
Gemma-1.1-7B-it (Mesnard et al., 2024)	77.61	91.15	79.12	87.41
LLaMA 3.1-8B-Instruct (AI@Meta, 2024)	78.15	**92.08**	79.12	88.05
IterLogic (Ours)	**90.43**	91.97	**83.76**	**89.41**

Table 5: Accuracy (%) on the QASC and WorldTree dataset using LoRA fine-tuned large language models.

Model	QASC Accuracy (%)	WikiHop Accuracy (%)				
		Publisher	Developer	Country	Record_label	Average
DeepSeek-R1 (Guo et al., 2025)	85.33	77.78	82.76	82.47	84.81	81.96
o1-preview (OpenAI, 2024)	87.28	85.19	79.31	81.96	**86.22**	83.17
IterLogic (Ours)	**90.43**	**94.44**	**89.66**	**86.08**	85.87	**89.01**

Table 6: Accuracy (%) of the o1-preview and DeepSeek-R1 on the QASC and WikiHop dataset.

for filtering out fluent but unsupported candidates before they influence final answer selection.

5.3 Performance Comparison between IterLogic and LoRA-Finetuned Models

To further assess the effectiveness of the proposed framework, we compare IterLogic with large language models fine-tuned using LoRA (Low-Rank Adaptation) (Hu et al., 2022). LoRA introduces low-rank decompositions into the model weight matrices and updates only the newly added parameters while freezing the original weights, substantially reducing the computational costs of fine-tuning. Under this setting, we evaluate IterLogic and LoRA-finetuned models on the same datasets using identical evaluation metrics.

Table 5 reports the accuracy of LoRA-finetuned large language models on the QASC and WorldTree test sets, including Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Gemma-1.1-7B-it (Mesnard et al., 2024), Qwen-7B-Chat (Bai et al., 2023), LLaMA 3-8B-Instruct (AI@Meta, 2024), LLaMA 3.1-8B-Instruct (AI@Meta, 2024), and DeepSeek-R1-Distill (Guo et al., 2025). Note that we do not include LoRA-based experiments on the WikiHop dataset. This is because, under the zero-shot evaluation setting, the required supporting knowledge is already provided in the input prompts, making additional parameter-efficient fine-tuning unnecessary.

The results show that on the QASC test set, IterLogic achieves an accuracy of 90.43%, substantially outperforming all LoRA-finetuned baselines. For example, Mistral-7B-Instruct attains only 75.43% accuracy, and models from the DeepSeek-R1 series also underperform relative to our approach. On the WorldTree test set, IterLogic consistently delivers strong performance across the easy subset, the challenge subset, and the full dataset. In particular, it achieves an accuracy of 83.76% on the challenge subset, demonstrating its superior ability to compose external knowledge and perform logical verification in complex reasoning scenarios, where other models suffer from performance degradation. Overall, while LoRA fine-tuning generally improves the accuracy of large language models, IterLogic achieves higher performance without relying on parameter-efficient fine-tuning. By leveraging a complementary neuro-symbolic architecture, IterLogic demonstrates greater stability and robustness in multi-step reasoning tasks, surpassing LoRA-finetuned large language models.

5.4 Performance Comparison between IterLogic and Reasoning Models

Table 6 presents a comparative evaluation of IterLogic against two reasoning models, o1-preview (OpenAI, 2024) and DeepSeek-R1 (Guo et al., 2025), on the QASC test set and the WikiHop validation set.

On the QASC dataset, IterLogic achieves an

accuracy of 90.43%, surpassing the strongest reasoning baseline, o1-preview (87.28%), by 3.15 percentage points. This margin indicates that ITERLOGIC is more effective at distinguishing correct answers from closely related distractors in commonsense reasoning tasks where misleading options are prevalent. In contrast, the performance of DeepSeek-R1 on QASC may be attributed to the adversarial noise intentionally introduced in this dataset, which can lead to over-reasoning during inference. For example, for the question "What are used for observing materials up close?", DeepSeek-R1 incorrectly selects "light" as the answer and justifies it by claiming that "Light is critical as it illuminates the specimen, enhancing visibility." Although this explanation appears semantically plausible, it fails to establish a correct correspondence with the requirement of "observing materials at close range," resulting in over-reasoning rather than valid evidential support.

On the WikiHop dataset, ITERLOGIC demonstrates consistently strong performance across four subtasks, namely publisher, developer, country, and record label, achieving an average accuracy of 89.01%. This result exceeds the best average accuracy among the two reasoning models, 83.17% achieved by o1-preview, by 5.84 percentage points. These findings indicate that the proposed framework exhibits strong adaptability and robustness in multi-domain reasoning scenarios.

5.5 Case Study

Natural logic verification is particularly useful when a generated intermediate sentence is semantically relevant but not fully supported by the external fact. For example, given the fact "Plants grow faster when exposed to sunlight", the generated sentence "Plants grow faster because sunlight directly increases plant mass" shares closely related concepts, such as *sunlight* and *plant growth*. A holistic semantic scorer may therefore assign it a high score. However, the generated sentence introduces an unsupported causal mechanism. The source fact only states that sunlight exposure is associated with faster plant growth, but does not entail that sunlight directly increases plant mass. By comparing aligned spans and their logical relations, the natural logic verifier can identify this unsupported addition and reject the candidate. This case illustrates how span-level logical evaluation enables the verifier to identify unsupported intermediate reasoning steps that holistic semantic similarity may overlook.

6 Related Work

Neuro-Symbolic Reasoning. Neuro-symbolic approaches (Qi et al., 2025; Wang et al., 2025a; He et al., 2025; Liang et al., 2026) aim to combine the flexibility of neural models with the rigor of symbolic reasoning. Natural logic, in particular, enables inference directly over linguistic surface forms without translating text into formal logical representations, and has been successfully applied to interpretable question answering and textual inference (Angeli and Manning, 2014; Angeli et al., 2016; Shi et al., 2021). Prior work (Shi et al., 2025c) integrates natural logic as a scoring mechanism, but treats it as a passive component that does not influence generation. In contrast, ITERLOGIC tightly integrates natural logic verification into the generation loop, allowing symbolic reasoning to actively guide the rejection and regeneration of intermediate candidates.

Iterative Refinement in Reasoning Systems. Iterative refinement has recently emerged as an effective strategy for improving reasoning reliability in language models (Klein and Nabi, 2021; Quan et al., 2024; Wang and Yu, 2025; Xiao et al., 2025; Lin et al., 2025; Huang et al., 2025b), including self-refinement and verification-guided generation. Unlike approaches that rely on model self-consistency or heuristic feedback (Wang et al., 2023; Aggarwal et al., 2023; Chen et al., 2024a; Jiang et al., 2025), ITERLOGIC employs an explicit symbolic verifier grounded in natural logic, providing a principled and lightweight control signal in the generation process. This design enables more reliable intermediate reasoning without additional supervision or task-specific fine-tuning.

7 Conclusions

We proposed ITERLOGIC, an iterative natural logic verification framework that integrates symbolic verification directly into the generation process for multi-hop question answering. By enabling natural logic to reject logically inconsistent intermediate sentences and regenerate improved alternatives, ITERLOGIC effectively reduces hallucinated reasoning steps. Experiments on three datasets show consistent improvements over strong neural and neuro-symbolic baselines, particularly in challenging multi-hop settings. These results highlight the importance of iterative verification for reliable multi-step reasoning.

Limitations

The effectiveness of ITERLOGIC depends on the quality of retrieved evidence, and failures in retrieval may limit the benefits of iterative verification. In addition, the natural logic verifier primarily captures lexical and entailment-level consistency, and may not fully handle implicit knowledge beyond the given evidence. Finally, the iterative rejection and regeneration process introduces additional computational overhead compared to single-pass generation.

Acknowledgments

We thank the anonymous reviewers for their constructive comments, and gratefully acknowledge the support of the Fundamental Research Funds for the Central Universities (Grant No.DUT25RC(3)153), the Doctoral Research Start-up Fund Project of Liaoning Province (Grant No.2026-BS-0022) and the Natural Science Foundation of China (Grant No.62676066).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396.

AI@Meta. 2024. Llama 3 model card.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 54–59.

Gabor Angeli, Neha Nayak Kennard, and Christopher D Manning. 2016. Combining natural logic and shallow reasoning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 442–452.

Gabor Angeli and Christopher D Manning. 2014. Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of the conference on empirical methods in natural language processing*, pages 534–545.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Angelica Chen, Jason Phang, Alicia Parrish, Vishakh Padmakumar, Chen Zhao, Samuel R. Bowman, and Kyunghyun Cho. 2024a. Two failures of self-consistency in the multi-step reasoning of llms. *Transactions on Machine Learning Research*.

Guoxin Chen, Kexin Tang, Chao Yang, Fuying Ye, Yu Qiao, and Yiming Qian. 2024b. SEER: facilitating structured reasoning and explanation via reinforcement learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, August 11-16*, pages 5901–5921.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

David F Crouse. 2016. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Virtual Event / Punta Cana, Dominican Republic, 7-11 November*, pages 7358–7370.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the conference of the North American chapter of the association for computational linguistics*, pages 4171–4186.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th annual meeting of the association for computational linguistics*, pages 320–335.

Yue Fan, Hu Zhang, Ru Li, Yujie Wang, Hongye Tan, and Jiye Liang. 2024. Frva: Fact-retrieval and verification augmented entailment tree generation for explainable question answering. In *Findings of the Association for Computational Linguistics: ACL, Bangkok, Thailand and virtual meeting, August 11-16*, pages 9111–9128.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Yufei Feng, Zi'ou Zheng, Quan Liu, Michael Greenspan, and Xiaodan Zhu. 2020. Exploring end-to-end differentiable natural logic modeling. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1172–1185.

Deepanway Ghosal, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2022. Two is better than many? binary classification as an effective approach to multi-choice question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, December 7-11*, pages 10158–10166.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The Definitive Guide*, 1st edition. O'Reilly Media, Inc.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Nature*, 645(8081):633–638.

Feijuan He, Han Lai, Jiaqi Liu, Bo Wang, Haoran Chen, Haohan Liu, and Chenxi Zhang. 2025. Solving mathematical problems using large language models: A survey. *Data Intelligence*, 7(4):907–946.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, Kigali, Rwanda, May 1-5*.

Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. 2022. Metgen: A module-based entailment tree generation framework for answer explanation. In *Findings of the Association for Computational Linguistics: NAACL, Seattle, WA, United States, July 10-15*, pages 1887–1905.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The International Conference on Learning Representations, Virtual Event, April 25-29*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Zhisheng Huang, Xudong Jia, Tao Chen, and Zhongwei Zhang. 2025b. A general scalable approach for knowledge selection based on iterative hybrid encoding and re-ranking. *Data Intelligence*, 7(1):124–142.

Thomas F Icard III and Lawrence S Moss. 2014. Recent progress on monotonicity. *Linguistic Issues in Language Technology*, 9:167–194.

Saidul Islam, Hanae Elmekki, Ahmed Elsebai, Jamal Bentahar, Nagat Drawel, Gaith Rjoub, and Witold Pedrycz. 2024. A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems with Applications*, 241:122666.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, May 7-12*.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 137–150.

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*.

Chunyang Jiang, Chi-Min Chan, Wei Xue, Qifeng Liu, and Yike Guo. 2025. Importance weighting can help large language models self-improve. In *Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, February 25 - March 4,*, pages 24257–24265.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, February 7-12*, pages 8082–8090.

Tassilo Klein and Moin Nabi. 2021. Towards zero-shot commonsense reasoning with self-supervised refinement of language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Virtual Event / Punta Cana, Dominican Republic, 7-11 November*, pages 8737–8743.

Jihyung Lee and Gary Geunbae Lee. 2025. Structured reasoning and answer verification: Enhancing question answering system accuracy and explainability. *Knowledge-Based Systems*, 311:113091.

Feng Liang, Weixin Zeng, Runhao Zhao, and Xiang Zhao. 2026. Nestr: A neuro-symbolic abductive framework for temporal reasoning in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence, Singapore, January 20-27*, pages 31907–31915.

Shaoqing Lin, Chong Teng, Fei Li, Donghong Ji, Lizhen Qu, and Zhuang Li. 2025. DiscoSG: Towards

discourse-level text scene graph parsing through iterative graph refinement. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7837–7862.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, New Orleans, LA, USA, May 6-9*.

Bill MacCartney and Christopher D Manning. 2009. An extended model of natural logic. In *Proceedings of the eight international conference on computational semantics*, pages 140–156.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of the fifth international workshop on inference in computational semantics*.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2024. Introducing openai o1-preview.

Shrey Pandit, Jiawei Xu, Junyuan Hong, Zhangyang Wang, Tianlong Chen, Kaidi Xu, and Ying Ding. 2025. MedHallu: A comprehensive benchmark for detecting medical hallucinations in large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Suzhou, China, November 4-9*, pages 2858–2873.

Chengwen Qi, Ren Ma, Bowen Li, He Du, Binyuan Hui, Jinwang Wu, Yuanjun Laili, and Conghui He. 2025. Large language models meet symbolic provers for logical reasoning evaluation. In *The International Conference on Learning Representations, Singapore, April 24-28*.

Xin Quan, Marco Valentino, Louise Dennis, and André Freitas. 2024. Enhancing ethical explanations of large language models through iterative symbolic refinement. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, St. Julian's, Malta, March 17-22*, pages 1–22.

Joshua Robinson and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations, Kigali, Rwanda, May 1-5*.

Lynn E Rose. 1965. Aristotle's syllogistic and the fourth figure. *Mind*, 74(295):382–389.

Julia Rozanova, Deborah Ferreira, Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Supporting context monotonicity abstractions in neural nli models. In *Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning*, pages 41–50.

Victor Sanchez. 2016. *Studies on natural logic and categorial grammar*. Ph.D. thesis, University of Amsterdam.

Jihao Shi, Xiao Ding, Li Du, Ting Liu, and Bing Qin. 2021. Neural natural logic inference for interpretable question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Virtual Event / Punta Cana, Dominican Republic, 7-11 November*, pages 3673–3684.

Jihao Shi, Xiao Ding, Siu Cheung Hui, Yuxiong Yan, Hengwei Zhao, Ting Liu, and Bing Qin. 2025a. Final: Combining first-order logic with natural logic for question answering. *IEEE Transactions on Knowledge and Data Engineering*, 37(6):3103–3117.

Jihao Shi, Xiao Ding, Kai Xiong, Hengwei Zhao, Bing Qin, and Ting Liu. 2025b. Natural logic at the core: Dynamic rewards for entailment tree generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17372–17382.

Jihao Shi, Xiao Ding, Hengwei Zhao, Ting Liu, and Bing Qin. 2025c. Bridging neural and symbolic reasoning: A dual-system framework for interpretable question answering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL, Online Event, August 1-6*, pages 3621–3634.

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

Shuai Wang and Yinan Yu. 2025. iquest: An iterative question-guided framework for knowledge base question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, Vienna, Austria, July 27 - August 1*, pages 15616–15628.

Wenguan Wang, Yi Yang, and Fei Wu. 2025a. Towards data-and knowledge-driven ai: A survey on neuro-symbolic computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):878–899.

Wenya Wang and Sinno Jialin Pan. 2022. Deep inductive logic reasoning for multi-hop reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, May 22-27, 2022*, pages 4999–5009.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The International Conference on Learning Representations, Kigali, Rwanda, May 1-5*.

Yangfan Wang, Jie Liu, Chen Tang, Lian Yan, and Jingchi Jiang. 2025b. KCS: Diversify multi-hop question generation with knowledge composition sampling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Suzhou, China, November 4-9*, pages 23173–23185.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Yisheng Xiao, Pei Guo, Zechen Sun, Juntao Li, Kai Song, and Min Zhang. 2025. An empirical study of iterative refinements for non-autoregressive translation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, Vienna, Austria, July 27 - August 1*, pages 29851–29865.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Li Yuan, Yi Cai, Haopeng Ren, and Jiexin Wang. 2024. A logical pattern memory pre-trained model for entailment tree generation. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation, Torino, Italy, 20-25 May*, pages 759–772.

Hongxiang Zhang, Hao Chen, Muhao Chen, and Tianyi Zhang. 2025. Active layer-contrastive decoding reduces hallucination in large language model generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Suzhou, China, November 4-9*, pages 3028–3046.

Longyin Zhang, Bowei Zou, and Aiti Aw. 2024. Empowering tree-structured entailment reasoning: Rhetorical perception and llm-driven interpretability. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation, Torino, Italy, 20-25 May*, pages 5783–5793.

A Appendix: Experimental Settings

A.1 Datasets

QASC (Khot et al., 2020) is a multiple-choice science question answering dataset designed to assess commonsense and scientific reasoning. It contains 9,980 questions, each with one correct answer and seven distractors, including four adversarial choices. Answering QASC questions requires retrieving and composing justification sentences from a large knowledge base containing 17.2 million facts.

WorldTree (Jansen et al., 2018) is also a multiple-choice science question answering dataset. The dataset consists of 2,459 questions, each with four candidate answers, only one of which is correct, and is partitioned into Easy and Challenge subsets. Solving WorldTree questions relies on a knowledge base of approximately 10,000 commonsense and scientific facts for constructing explanatory reasoning chains.

WikiHop (Welbl et al., 2018) is a multi-hop reading comprehension dataset that requires aggregating evidence from multiple documents. Each question is formulated as a query of the form $p(e, X)$, where e denotes a given entity, X is the target entity to be predicted from a candidate set C, and p represents the query relation. We evaluate WikiHop across four domains: $publisher$, $developer$, $country$, and $record_label$, each with separate training and development splits. Since the test set is not publicly available, we report results on the development set. Dataset statistics are summarized in Table 7 and Table 8.

Train-Dev-Test Split	QASC	WorldTree		
		Easy	Challenge	Overall
Train (# of questions)	8,134	708	278	986
Dev (# of questions)	926	158	68	226
Test (# of questions)	920	859	388	1,247

Table 7: Statistics of the QASC and WorldTree datasets.

Train-Dev Split	WikiHop			
	Publisher	Developer	Country	Record_label
Train (# of questions)	509	267	742	2,305
Dev (# of questions)	54	29	194	283

Table 8: Statistics of the WikiHop dataset.

A.2 Baselines

While LLMs are commonly evaluated using *cloze prompting*, this strategy may underperform in

multiple-choice question answering (MCQA) settings (Robinson and Wingate, 2023). Following prior work on *multiple-choice prompting* (Robinson and Wingate, 2023), we adopt the prompt templates shown in Table 9. A unified prompt is used for QASC and WorldTree. For WikiHop, where performance was initially poor, we introduce a modified prompt that explicitly incorporates the provided knowledge and separates the query from candidate answers, reflecting the dataset's triplet structure.

Prompt Template for QASC and WorldTree

Choose the best answer to the following multiple-choice question delimited by triple backticks. And give the reason for your choice. Provide them in JSON format with the following keys: Answer, Reason.
```[question with multiple choices]```

Prompt Template for Wikihop

Assuming you know the following knowledge.
Knowledge:```[knowledge]```.
Choose the best answer to the following query from the candidates. And give the reason for your choice. Provide them in JSON format with the following keys: Answer, Reason.
Query: ```[query]```
Candidates: ```[candidates]```

One-shot Prompt Template for Intermediate Sentence Generation

The insect is small in size. [AND] Seeing small things is difficult. [ENTAIL] Seeing a small insect is difficult. [EOS]
```fact1 [AND] fact2 [ENTAIL]```

Table 9: Prompts used for different datasets and intermediate sentence generation.

A.3 Evaluation Metrics and Implementation Details

Model performance is evaluated using accuracy across all datasets, following standard practice in prior work (Shi et al., 2025a). For ITERLOGIC, the intermediate-sentence generator $\mathcal{G}$ is implemented with LLAMA 3-8B-INSTRUCT (AI@Meta, 2024). During regeneration, we sample with a temperature of 0.7 and set `max_new_tokens` to 50, which provides moderate generation diversity while keeping intermediate sentences concise. We employed the pre-trained `DeBERTa-large` model (He et al., 2023), which has a hidden dimension of 1024. The model and training pipeline were implemented using the Transformers library (Wolf et al., 2020). Fine-tuning was performed with the AdamW optimizer (Loshchilov and Hutter, 2019), using an initial learning rate of 1×10^{-6}. Training was carried out for 50 epochs on RTX PRO 6000 GPUs with 96 GB of memory. For both QASC and WorldTree, we selected the checkpoint achieving the best performance on the development set and report results on the test set.

B Appendix: Iteration Budget

Regarding sensitivity to the iteration budget, the maximum iteration count in ITERLOGIC acts as an upper bound on regeneration attempts rather than as a directly optimized parameter. In practice, the process can terminate once a candidate satisfies the verifier criterion, so the budget mainly controls how many additional opportunities the generator has to recover from an initially weak composition.