# Exploratory Inference Chain: Exploratorily Chaining Multi-hop Inferences with Large Language Models for Question-Answering

1st Shosuke Haji
*Meiji University*
Kanagawa, Japan
hajisho@cs.meiji.ac.jp

2nd Keiichi Suekane
*Meiji University*
Kanagawa, Japan
kaiichi_s@cs.meiji.ac.jp

3rd Hirofumi Sano
*Meiji University*
Kanagawa, Japan
hiro-sano@cs.meiji.ac.jp

4th Tomohiro Takagi
*Meiji University*
Kanagawa, Japan
takagi@cs.meiji.ac.jp

2023 IEEE 17th International Conference on Semantic Computing (ICSC) | 978-1-6654-8263-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSC56153.2023.00036

***Abstract*—Successful few-shot question-answering with large language models (LLMs) has been reported for a variety of tasks. In the usual approach, an answer is generated by a single call to an LLM, but it has been pointed out that the performance of multi-hop inference by LLMs is not sufficient. Thus, an LLM is unable to perform the complex processing necessary to get an answer, which leads to poor performance. Moreover, the inference process is opaque. Against this, approaches that call an LLM multiple times have been proposed, but many of these approaches can only be used for a limited number of effective tasks, and LLMs essentially require complex processing.**

**To address these problems, we propose the Exploratory Inference Chain (EIC) framework that combines the implicit processing of LLMs with explicit inference chains, and this is based on the dual process theory of human cognitive processes. The EIC framework first generates the information needed to answer a multi-hop question as keywords and then performs 1-hop inference for each keyword. If the inference is not sufficient, additional inferences are performed. This process is repeated, and when sufficient inferences are obtained, they are aggregated, and the final answer is generated. This makes the information per inference by LLM simplified, and logical inference is achieved through an explicit inference chain.**

**We conducted experiments on two multi-hop QA datasets and confirmed through a quantitative evaluation that our EIC framework performed better than existing approaches. Moreover, a qualitative evaluation confirmed that our approach can effectively perform inference so as to get closer to the answer in question-answering tasks that require knowledge. In addition, compared with existing approaches, the EIC framework improves the interpretability of the output.**

***Index Terms*—Multi-hop Inference, Large Language Models, Neuro-Symbolic, Logical Reasoning**

## I. Introduction

Large language models (LLMs) train on huge amounts of data on the Web, so they have a lot of knowledge and common sense, which allows them to read, infer, and joke. By using these capabilities, LLMs have achieved impressive language generation in few-shot learning for a variety of tasks [1]–[3]. However, in general, when humans answer a question, they do not often use a single piece of knowledge to get to the answer but rather use a number of pieces of knowledge and build up inferences from them to get to the final answer. How much and which knowledge is needed to obtain an answer, how inferences need to be made, and how they are built up from that knowledge is dynamic, depending on the difficulty of the answer. Moreover, these are not known from the question but are decided through a process of exploration and reasoning to lead to the answer. In the usual approach, the LLM is required to perform this process in a single call, which leads to poor performance in multi-hop inference. For this reason, logical reasoning with LLMs has been studied. [4], [5] encourage logical multi-hop reasoning by explicitly generating not only the answer but also the logical reasoning to the question. However, these approaches require multi-hop inference directly from the question, and the processing complexity required by the LLM is no different than a single call to the model.

When a human answers a multi-hop question that requires multiple pieces of information, the human repeatedly refers to the necessary information to logically get to the answer and infers it. The dual process theory of human cognitive processes suggests that our brains have implicit, intuitive processes (System 1) and explicit, logical processes (System 2) [6]–[11]. While the ability of LLMs improves the accuracy of a single call to a model, corresponding to System 1, the explicit inference process, as in System 2, has not been studied enough. Moreover, how to combine both to make such inferences is a critical challenge in modeling human-like inference. Therefore, we study a neuro-symbolic approach to question-answering that combines LLMs and this symbolic approach.

There are various types of neuro-symbolic approaches [12]. In [13]–[17], multiple modules specialized for a single process are prepared and chained together to generate answers to complex questions by inputting the output of one module into the next module. [18]–[22] construct dynamic graphs using language models. [23], [24] build modules using only LLMs to realize step-by-step reasoning.

The neuro-symbolic approach with LLMs that we draw most inspiration from is Selection Inference (SI) [23]. Selection Inference uses a selection module to select

978-1-6654-8263-9/23/$31.00 ©2023 IEEE
DOI 10.1109/ICSC56153.2023.00036



Authorized licensed use limited to: Northcentral University. Downloaded on May 06,2023 at 00:06:52 UTC from IEEE Xplore. Restrictions apply.

sentences from a given context and an inference module to perform 1-hop inference from the selected sentences. These are realized with LLM's few-shot learning. Repeating these two modules alternately achieves multi-hop inference. This approach decomposes the inference process into sub-processes and handles them with modules, thereby reducing the complexity of a single LLM call while performing multi-hop inference. However, this method needs to set the number of iterations of selection and inference as a hyperparameter. As mentioned above, it is impossible for a human to determine the number of inferences in a search from a question before the search begins, and this would leave the underlying issue unresolved as a general-purpose problem solver. Moreover, the tasks are limited to simple logical reasoning and do not fully use the knowledge and processing abilities of LLMs.

We addressed a neuro-symbolic approach to multi-hop inference that requires knowledge, which better uses the ability of LLMs. Therefore, we considered breadth and depth in reasoning for multi-hop questions. The breadth is the number of pieces of information required directly from the question. The two questions "When was film X released?" and "Which file came first, X or Y?" have different breadths. The former requires only one inference about "film X", while the latter requires two parallel inferences about "X" and "Y".

In addition, the depth varies with the obtained information. For the question "When is the birthday of the director of film X?", if direct information is available, the answer can be obtained by 1-hop inference. However, when only "the director of film X is Z" is available, it is necessary to refer to the "director Z" information and perform 2-hop inference. This means that the number of required inference hops will vary depending on the results of each inference. Our Exploratory Inference Chain (EIC) framework dynamically determines the breadth and depth depending on the question-related information generated by the LLMs. This allows LLMs to get to the answer and its reasoning without having to make complex inferences in a single call. Moreover, unlike existing methods, the proposed method refers to and infers only the necessary paragraphs, so it can effectively advance inferences in distractive tasks that include paragraphs that are not directly related to the question.

In this paper, our contributions are the following:

- We propose the Exploratory Inference Chain (EIC) framework that combines the ability of LLMs with a symbolic approach. The proposed framework exploratorily chains multi-hop inferences.
- The proposed method can efficiently get to an answer by referring to only the necessary information for reasoning in distractive contextual QA. We show that our approach can lead to an answer in a faster time and with better accuracy than the existing symbolic approach with LLMs.
- The proposed framework iteratively and dynamically advance simple inferences for multi-hop questions, which greatly improves interpretability. As a result, even for wrong answers, the reason for the mistake is clear.

## II. Related Work

### A. Large Language Models

LLMs have been successful in a variety of tasks. In transformer-based language models, the size and performance of LLMs continues to increase on the basis of scaling law [25], which states that the larger the model size, the better the model performance will be.

However, because of the large size of LLMs, training or fine-tuning them is expensive in terms of time and computing resources. Moreover, fine-tuning LLMs with a small amount of data may result in the loss of knowledge and processing ability gained by pre-training. For this reason, there are many studies on improving the performance of LLMs with fixed parameters.

[26]–[32] adjust the input prompts or sample multiple generated texts. Meanwhile, [19], [20], [33], [34] use LLMs to score answers.

### B. LLMs to Logical Reasoning

While LLMs are good at 1-hop logical reasoning, they struggle with multi-hop reasoning. For this reason, methods of logical reasoning with LLMs have been studied [4], [5], [23], [24], [35]–[37].

Chain of Thought (COT) has demonstrated that explicitly generating intermediate inference steps improves accuracy during complex inference in LLMs [4], [5]. However, it has been reported that this property arises only in LLMs with more than 100B parameters, and moreover, LLMs are still essentially required to make multi-hop inferences at the same time.

Selection Inference (SI) [23] is a symbolic approach that calls LLMs multiple times. However, the number of iterations is fixed and must be set in advance, so the task must have a fixed number of required inference hops. Against that, Faithful Reasoning [24] fixed the problem of SI being unable to judge when to stop the iteration in the number of steps and prepared an LLM, "Halter", that judges whether reasoning is enough. Halter can determine the stop iteration itself, but it needs to be fine-tuned for each task, and the number of tasks that work with Halter is limited.

While these studies have improved accuracy for logical reasoning with LLMs, the given context is often simple.

### C. Decomposing Tasks

An approach that breaks complex questions into simple subtasks is also being considered.

[13]–[15] program the necessary processing steps from question to answer and follow these steps, achieving high accuracy. [16], [17] have agents that decompose questions and sub-models that solve simple tasks. The agent decomposes a question, determines the next necessary process, and passes it to the appropriate sub-model for the specific process. These approaches achieve more accurate processing by preparing modules that solve subtasks. Decomposing tasks into subtasks is a standard approach that has been considered for a long time, but it requires designing modules for each sub-task, and



Authorized licensed use limited to: Northcentral University. Downloaded on May 06,2023 at 00:06:52 UTC from IEEE Xplore. Restrictions apply.

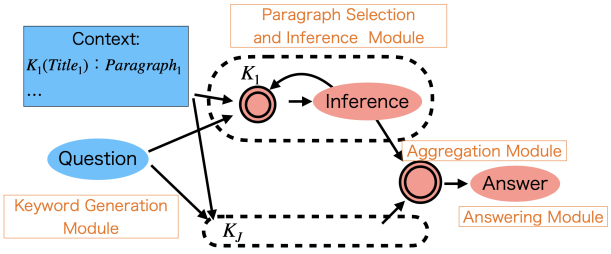



Fig. 1: Overview of the EIC framework.

it is not easy to decompose complex questions. Therefore, it is only studied for tasks that require prepared modules or decomposed data.

The effectiveness of decomposing questions in multi-hop inference in LLMs has also been shown. [38]–[42] achieve better accuracy than a single call to a model by decomposing a multi-hop question and repeatedly answering the 1-hop questions. However, decomposing questions is difficult and requires either manual or finely designed task-specific models. That is, it is necessary to prepare specialized mechanisms for each task, suggesting a major challenge for the realization of general-purpose artificial intelligence. Compared with these studies, our approach dynamically advances inference, decomposing multi-hop inference without specializing in a particular task.

## III. The Exploratory Inference Chain framework

In this section, we detail our proposed framework for more logical and accurate reasoning by decomposing the process required to get to the answer from a question into four steps and exploratorily chaining inferences. An overview of the proposed method is shown in Fig. 1.

(1) Generate the necessary information from the question text as keywords. This determines the breadth of the exploratory inference chain.

(2) Make paragraph references and inferences. Determining whether or not to refer to further paragraphs depending on the inference, the exploratory inference chain is made with the necessary depth for the answer.

(3) Aggregate exploratory inference chains. The inferences in (2) are still multi-step information because they are inferred about one paragraph at a time. For each keyword, these pieces of information are aggregated into one, and the model generates a direct reason for the answer.

(4) Generate a final answer on the basis of the reasons generated in (3).

As mentioned above, step-by-step inference makes LLM's one-time inference simple, and dynamically chaining inferences realizes logical inference.

### A. Definition

For each question $q$ in a dataset, an answer $a$ and the context $C$ related to the question are associated. At this time, the context $C$ has multiple paragraphs: $C = \{p_1, ..., p_n\}$. Each paragraph has a paragraph title $t$ and its content document $d$: $p_i = \{t_i, d_i\}$. This context is under a distract setting and contains paragraphs that are not directly related to the answer. Moreover, even within document $d$, which contains the answer, information is contained that is not related to the answer. Therefore, the model needs to select the necessary paragraphs from the given context and extract the appropriate information from them to infer and answer the questions.

Question $q$ is a multi-hop task requiring multiple paragraph references to answer.

### B. Keyword Generation Module

Depending on the question, the number of pieces of information needed vary, and parallel inferences are sometimes required. Hence, according to the following form, the LLM generates the information directly needed to answer the question $q$ as keywords and then infers for each keyword:

```
Please select the keywords needed
for one-hop inference from Context.
# n-shot keywords prompt
# First example
Context: <Paragraph Titles>
Question: <Question>
Keywords: <K'_1>,<K'_2>,...
...
# Problem to answer
Context: <t_1>,<t_2>,...
Question: <Question q>
Keywords:
```

Here, because the paragraphs given for each question are limited, their titles are also inputted, and a string is generated so that only those titles directly related to the question are selected from among them. We denote the generated $J$ keywords as $K_1, ..., K_j, ..., K_J$.

Note that inputting the paragraph title at the same time as the question depends on the method for selecting the next paragraph, and the purpose of this step is to extract important keywords to lead to the answer from the question.

### C. Paragraph Selection and Inference Module

In this module, each obtained keyword is processed independently. Now, let us consider the keyword $K_j$. Moreover, the inferences advanced from $K_j$ are added to the empty list $Inference_{K_j}$.

**Paragraph Selection Step.** This step references the necessary paragraphs for $K_j$. For simplicity, we determine the necessity of a reference by matching the keyword with the title string $K_j = t_i$. If there is a matched paragraph title, refer to the paragraph $p_i = \{t_i, d_i\}$ and proceed to the next step. This makes it possible to refer to only paragraphs that are directly related to the question from multiple distract contexts.

Note that we use string matches for simplicity, but it is possible to extend this method such as on the basis of the similarity of keywords to the distributed representation of



Authorized licensed use limited to: Northcentral University. Downloaded on May 06,2023 at 00:06:52 UTC from IEEE Xplore. Restrictions apply.

documents.

**Inference Step.** This step infers information about the paragraphs referenced by the Select Paragraph step. A question and the referenced paragraph are input to the LLM, and it extracts the information directly related to the question in the paragraph according to the following form:

```
Please state what can be directly
inferred from the context.
# n-shot inference prompt
# First example
Context: <Inferences about K'_j>
<Paragraph>
Question: <Question>
Reason: <K'_j>::<Relation>::<Object>
...
# Problem to answer
Context: <Inferences about K_j>
<Paragraph d_i>
Question: <Question q>
Reason: <K_j>::
```

The generated inference is added to $Inference_{K_j}$. Since only one paragraph is referenced per inference, complex inferences over multiple paragraphs can be avoided.

When generating an inference, the keyword $K_j$ is included in the input prompt, and the generated sentence is induced to follow the following form, separated by "::".

$$< K_j >::< Relation >::< Object >$$

This prevents the LLM from generating a sentence that ignores the input content, called hallucinating, and also makes symbolic processing easier.

**Halting Step.** A single inference may not provide the information necessary to answer a question and may require additional references to other paragraphs. Thus, it is necessary to determine whether additional paragraph references are needed on the basis of the results of the inference. For simplicity, we focus on the $\langle Object \rangle$ of a generated inference. If title $t_{i'}$ is included in $\langle Object \rangle$, go back to the Paragraph Selection step and infer information about the paragraph. This dynamically determines the inference depth.

However, if the matched title has been already referenced in $Inference_{K_j}$, it is not referenced a second time.

### D. Aggregation Module

For each keyword, when additional references are no longer needed, the inferences chained from the keyword are aggregated in the following form:

```
Please state what can be directly
inferred from the context.
# n-shot multi-hop inference prompt
# First example
Context: <Inference about K'_j>
Question: <Question>
Reason: <Multi-hop Inference about K'_j>
...
# Problem to answer
Context: <Inference about K_j>
Question: <Question q>
Reason:
```

Because each inference in $Inference_{K_j}$ is a simple inference about a question from a single paragraph, aggregating these inferences generates a multi-hop inference across paragraphs.

### E. Answering Module

Finally, we concatenate inferences about all keywords and generate answers according to the following form:

```
# n-shot answer prompt
# First example
Context: <Multi-hop Inference about K'_1>
<Multi-hop Inference about K'_2>...
Question: <Question>
Answer: <Answer>
...
# Problem to answer
Context: <Multi-hop Inference about K_1>
<Multi-hop Inference about K_2>...
Question: <Question q>
Answer:
```

## IV. Experiment

In this section, we show the effectiveness of the Exploratory Inference Chain (EIC) framework in both quantitative and qualitative evaluations.

### A. Experimental Settings

To confirm the effectiveness of the EIC framework, we conducted experiments on two datasets, the HotpotQA dev set in the distractor setting [43] and 2WikiMultihopQA dev set [44], which require multi-hop inference. Both datasets have questions and answers, together with multiple paragraphs extracted from Wikipedia articles. These include paragraphs that not only contain sentences that support answers but also those that are not needed for answers. Therefore, it is necessary to go directly to the answer by extracting only the information necessary for the answer from the given paragraphs.

We used 1000 questions sampled randomly from each dataset.

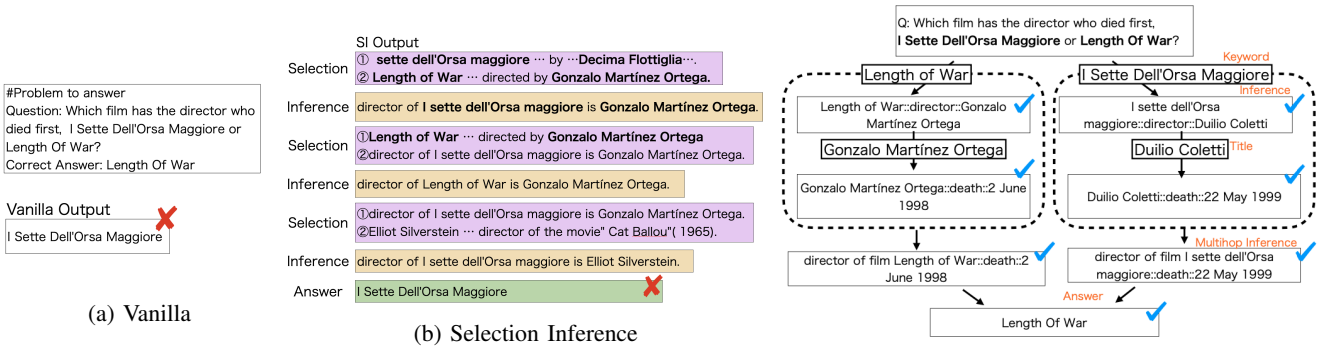
All models in the paper used GPT-j-6B [45]. Moreover, 5-shot prompts were sampled from the training sets. To ensure that accuracy was not affected by the differences in input prompts, all methods used for each dataset used the same questions for prompts.

Baseline.

- No_context: Only questions are input to the model, and the model generates the answer. The model answers are based only on the knowledge gained by pre-training.



Authorized licensed use limited to: Northcentral University. Downloaded on May 06,2023 at 00:06:52 UTC from IEEE Xplore. Restrictions apply.




(a) Vanilla

(b) Selection Inference

(c) Exploratory Inference Chain

Fig. 2: Difference of output by methods.

- Vanilla: A question and context are input to the model, and the model generates an answer. If the sequence length of the context exceeds the maximum length of the model, characters are removed from the beginning of the 5-shot prompt until the input length is acceptable.

LLMs for Logical Reasoning.

- Chain of Thought (COT): Questions and context are input to the model, and the model generates the reasoning and answer. [4]
- Selection Inference (SI): The model selects two sentences from the context and makes a 1-hop inference from those two sentences. [23] This process of selection and 1-hop inference is treated as 1 step. In SI, the number of steps is a hyperparameter, so it is not possible to stop at the appropriate number of steps. We conducted all steps from 1 to 6 and used the number of steps with the best score in the evaluation data.
- Selection Inference-Once (SI-Once): In the SI experiment, there were many cases where the same sentence was repeatedly selected and inference was not advanced. Thus, for this SI, paragraphs selected once are not selected again.

Proposed Method.

- Exploratory Inference Chain (EIC): Our proposed method. Unlike the methods described above, only paragraphs matching the title with the generated text can be referenced.
- Exploratory Inference Chain (EIC)-Reasoning: Like Chain of Thought, after aggregating inferences for each keyword, the model generates the reasoning and answer.

Gold Support.

- Input Support: The two datasets have, in addition to answers, which sentences are the supporting evidence in the context as labels. Here, questions and their support sentences are input to the model, and the model generates answers.

TABLE I: Accuracy in exact string match (lower case).

| Method | 2WikiMultihopQA | HotpotQA |
|---|---|---|
| No_Context | 0.260 | 0.113 |
| Vanilla | 0.214 | 0.129 |
| COT | 0.16 | 0.121 |
| SI | 0.234 | 0.156 |
| SI-Once | 0.227 | 0.185 |
| EIC | **0.334** | **0.213** |
| EIC-reasoning | 0.301 | 0.186 |
| Input Support | 0.430 | 0.438 |

### *B. Quantitative Evaluation*

We measured the accuracy by exact string match (lower case) between the generated answer and the ground truth answer. The results of the quantitative evaluation are shown in Table I.

Our EIC had the best accuracy on the two datasets.

For 2WikiMultihopQA, the accuracy of Vanilla was lower than that of No_context. That may be due to the input to Vanilla containig a lot of information that did not lead to an answer, and this became noise. Thus, it indicates that even if the context is closely related to the question, the accuracy will get worse without selecting the appropriate text.

The accuracy of COT was lower than that of Vanilla, which is consistent with previous works showing that this only occurs at a scale of more than 100B model parameters. That is, the 6B parameters GPT used in this experiment is insufficient in its ability to perform multi-hop inference in a single call.

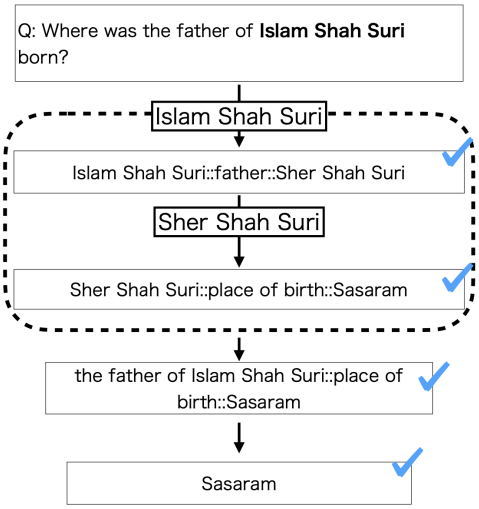
In comparison, Selection Inference and our EIC showed better accuracy than the methods above. That confirms that the neuro-symbolic approach in LLMs is effective, regardless of the scale of the model.

As shown in the experimental settings, even though Selection Inference had a more favorable setting compared with the EIC, the EIC achieved better accuracy. This quantitatively confirms the effectiveness of our EIC.

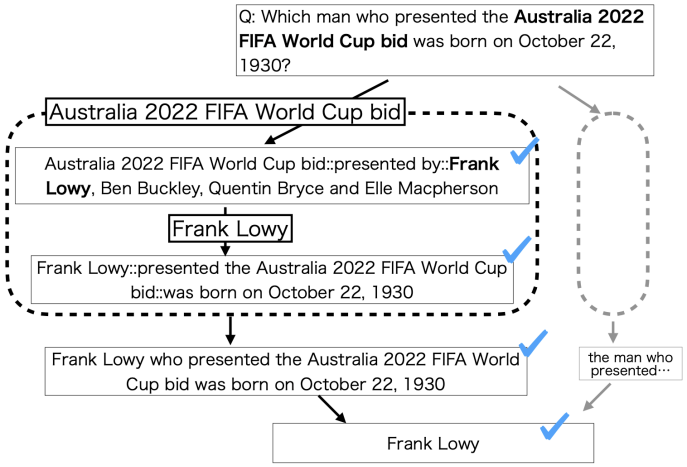
Finally, the accuracy scores for Input Support were 0.430 and 0.438, respectively, which are lower than those of the larger LLMs in previous work. As shown in the following qualitative evaluation, the model used in the experiment fails even the simplest inferences. Since LLM performs better with



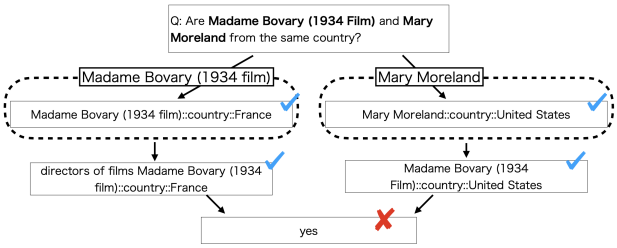
Authorized licensed use limited to: Northcentral University. Downloaded on May 06,2023 at 00:06:52 UTC from IEEE Xplore. Restrictions apply.



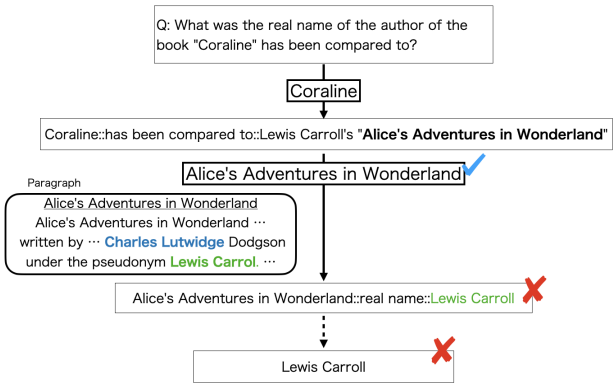

(a) Only one keyword.




(b) Answer is based on the appropriate inference path from multiple keywords.




(c) Error in comparing two names of country.




(d) Error in reading paragraph.

Fig. 3: The examples of exploratory inference chain.

larger models, it is possible that the EIC will improve accuracy even further if larger models are used.

### C. Qualitative Evaluation

In this section, we confirm that our EIC is able to advance inference dynamically by verifying the output of each method. Moreover, we confirm the interpretability of the EIC framework.

The differences in the output of the three methods are shown in Fig. 2. These were the outputs of the three models for the same question.

Vanilla got the answer wrong and the reason for the output was opaque.

Selection Inference symbolically repeats selection and inference, so it is clear how the inference proceeded. However, there are two problems in this example. First, two different pieces of information were mixed up. The given question required referring to the directors of two films, and during the first selection, sentences about these two directors were selected. As a result, the model incorrectly inferred a correspondence between the film and its director.

Second, similar or same sentences were selected repeatedly. In this example, after inferring information about the director of the film, the model moreover needed to refer to the death anniversary of the director. However, because Selection Inference selects sentences that are related to the question, it selected only sentences about the movie contained in the question text, so inference was not advanced. As a result, the model failed to make inferences that lead directly to an answer.

In comparison, for the EIC, it was confirmed that the correct paragraph was referred to for each hop, and the necessary information was obtained and inferred.

First, the inference was processed for the two movies independently as keywords to the question, so that the information from the two films was not mixed up like in the case of Selection Inference. Moreover, by determining whether to refer to other paragraphs additionally on the basis of the results of inference, the proposed method was able to refer to and infer paragraphs as necessary and sufficient. Compared with SI, which selects sentences on the basis of likelihood, the EIC improved the interpretability of the inference process and the reason for referring to a paragraph.



Authorized licensed use limited to: Northcentral University. Downloaded on May 06,2023 at 00:06:52 UTC from IEEE Xplore. Restrictions apply.

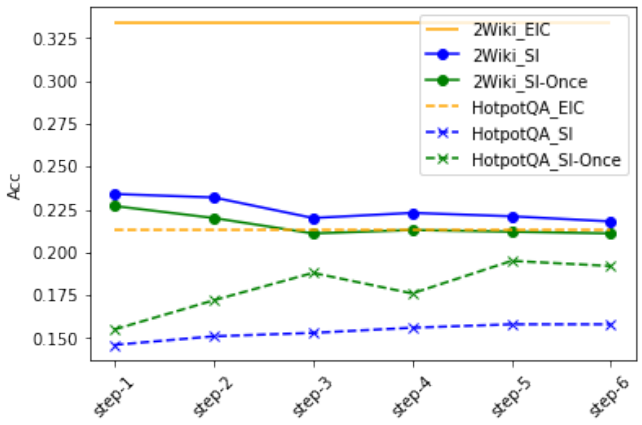



Fig. 4: Accuracy of SI for each specified number of steps and that of the EIC.

Next, Figs. 2c and 3 show that the EIC exploratorily chaining inferences and is able to dynamically control the range of materials collected before getting to the final answer. It can be seen that the number of inferences varied appropriately for each question. In Fig. 2c, the question branches into keywords about the two films, and each refers to its paragraph. Fig. 3a refers to and makes inferences about only one person from the question. In Fig. 3b, multiple paragraphs were referenced, but the appropriate path was used to answer the question.

Figs. 3c and 3d show mistaken examples. In Fig. 3c, the system was able to correctly infer and even aggregate information. However, the model got it wrong in comparing the last two countries. In Fig. 3d, the final answer was wrong due to the error of taking out the pen name of the author of "Alice in Wonderland" when the real name should have been taken out. Thus, the EIC framework was found in the quantitative evaluation to have not only improved, but it also clarified the reasons for mistakes due to the improved interpretability. At the same time, the results reveal that LLMs get even simple inferences wrong.

### *D. Comparison with SI*

To confirm the problems of the existing method and the efficiency of the EIC framework, we compared the proposed method with Selection Inference. Both SI and the EIC framework are neuro-symbolic approaches that do not fine-tune LLMs.

The accuracy of SI at each specified step and of our EIC is shown in Fig. 4.

First, in 2WikiMultihopQA (solid line), the more the number of steps for SI increased, the more the accuracy decreased. This quantitatively indicates that SI did not advance the reasoning to solve the problem. In SI, two sentences are selected on the basis of the question and context at the selection step, but these two sentences are not selected such as to get closer to the answer. Moreover, emulating previous works, at the inference step, to prevent hallucination, a question is not inputted, and an inference is made from those two sentences [23], [24]. That may induce inferences that do not lead to an answer. This suggests that SI does not work well in tasks that require multiple paragraph references.

In HotpotQA (dashed line), SI tends to increase in accuracy as the number of specified steps increases. Here, to ensure that the number of SI steps in the HotpotQA is enough, the number of support sentences for the QA sampled from HotpotQA is shown in Table II. Because two sentences are selected and inferred in one step, (number of supports - 1) step is the required number of steps, which is satisfied at 6 steps. However, Fig. 4 shows that the maximum accuracy of SI was step-5, which does not reach the accuracy of the EIC, thus indicating that the EIC is better at advancing inference to lead to an answer than SI.

In the EIC, inference in one paragraph is made by only one call of the model. However, HotpotQA was built on information from two paragraphs, regardless of the number of supporting sentences, and questions with more than two supports require multi-hop reasoning for one paragraph. This may be the reason why the difference in accuracy between the proposed method and SI is smaller for HotpotQA than for 2WikiMultihopQA. Therefore, using SI for inference in each paragraph could further improve accuracy.

Finally, Table III shows the average inference time per question for each dataset.

Compared with SI, the EIC significantly reduced the inference time. This is because it additionally refers to only the necessary paragraphs after inference, while SI refers to all paragraphs at the time of selection. In addition, the inference time shows that the EIC can efficiently proceed with inference.

From the above, it can be concluded that the EIC framework is an effective and efficient approach for advancing inference.

TABLE II: Number of supporting sentences in HotpotQA

| Support sentence Num | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| Count | 683 | 239 | 67 | 9 | 1 | 1 |

TABLE III: Inference time per question (sec)

| Method | 2WikiMultihopQA | HotpotQA |
|---|---|---|
| SI | 35.14 | 55.19 |
| EIC | 8.09 | 14.42 |

## V. Conclusion

In this paper, we proposed the Exploratory Inference Chain (EIC) framework that uses LLMs to dynamically proceed with inference and make reference to necessary knowledge. For multi-hop QA, by dividing the inference process into modules and exploratorily chaining inferences, LLMs can lead to an answer through iterations of simple inference. A quantitative evaluation showed the effectiveness of the neuro-symbolic approach using LLMs. Moreover, a qualitative evaluation confirmed that the interpretability is improved. Because our



Authorized licensed use limited to: Northcentral University. Downloaded on May 06,2023 at 00:06:52 UTC from IEEE Xplore. Restrictions apply.

EIC framework does not adjust the parameters of the LLMs, it could be applied to other tasks by defining appropriate inference patterns.

However, the improved interpretability reveals that LLMs can err even on simple inferences. Our future work is to solve these problems and make LLMs perform more accurate logical reasoning.

## REFERENCES


[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[3] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Pretrained transformers as universal computation engines," *arXiv preprint arXiv:2103.05247*, 2021.

[4] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *arXiv preprint arXiv:2201.11903*, 2022.

[5] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *arXiv preprint arXiv:2205.11916*, 2022.

[6] J. S. B. Evans, "Heuristic and analytic processes in reasoning," *British Journal of Psychology*, vol. 75, no. 4, pp. 451–468, 1984.

[7] J. S. B. Evans, "In two minds: dual-process accounts of reasoning," *Trends in cognitive sciences*, vol. 7, no. 10, pp. 454–459, 2003.

[8] J. St Evans *et al.*, "Dual-processing accounts of reasoning, judgment, and social cognition," *Annual Review of Psychology*, vol. 59, no. 1, pp. 255–278, 2008.

[9] S. A. Sloman, "The empirical case for two systems of reasoning.," *Psychological bulletin*, vol. 119, no. 1, p. 3, 1996.

[10] D. Kahneman, *Thinking, fast and slow*. Macmillan, 2011.

[11] K. E. Stanovich and R. F. West, "Individual differences in reasoning: Implications for the rationality debate?," *Behavioral and brain sciences*, vol. 23, no. 5, pp. 645–665, 2000.

[12] A. d. Garcez and L. C. Lamb, "Neurosymbolic ai: the 3rd wave," *arXiv preprint arXiv:2012.05876*, 2020.

[13] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," *Advances in neural information processing systems*, vol. 31, 2018.

[14] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," *arXiv preprint arXiv:1904.12584*, 2019.

[15] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 39–48, 2016.

[16] T. Khot, K. Richardson, D. Khashabi, and A. Sabharwal, "Learning to solve complex tasks by talking to agents," *arXiv preprint arXiv:2110.08542*, 2021.

[17] T. Khot, D. Khashabi, K. Richardson, P. Clark, and A. Sabharwal, "Text modular networks: Learning to decompose tasks in the language of existing models," *arXiv preprint arXiv:2009.00751*, 2020.

[18] A. Bosselut, R. Le Bras, and Y. Choi, "Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering.," in *AAAI*, pp. 4923–4931, 2021.

[19] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," *arXiv preprint arXiv:1911.00172*, 2019.

[20] U. Alon, F. Xu, J. He, S. Sengupta, D. Roth, and G. Neubig, "Neuro-symbolic language modeling with automaton-augmented retrieval," in *International Conference on Machine Learning*, pp. 468–485, PMLR, 2022.

[21] J. Jung, L. Qin, S. Welleck, F. Brahman, C. Bhagavatula, R. L. Bras, and Y. Choi, "Maieutic prompting: Logically consistent reasoning with recursive explanations," *arXiv preprint arXiv:2205.11822*, 2022.

[22] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, "Cognitive graph for multi-hop reading comprehension at scale," *arXiv preprint arXiv:1905.05460*, 2019.

[23] A. Creswell, M. Shanahan, and I. Higgins, "Selection-inference: Exploiting large language models for interpretable logical reasoning," *arXiv preprint arXiv:2205.09712*, 2022.

[24] A. Creswell and M. Shanahan, "Faithful reasoning using large language models," *arXiv preprint arXiv:2208.14271*, 2022.

[25] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[26] O. Rubin, J. Herzig, and J. Berant, "Learning to retrieve prompts for in-context learning," *arXiv preprint arXiv:2112.08633*, 2021.

[27] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for gpt-3?," *arXiv preprint arXiv:2101.06804*, 2021.

[28] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, and E. Chi, "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[29] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, and D. Zhou, "Rationale-augmented ensembles in language models," *arXiv preprint arXiv:2207.00747*, 2022.

[30] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[31] E. Zelikman, Y. Wu, and N. D. Goodman, "Star: Bootstrapping reasoning with reasoning," *arXiv preprint arXiv:2203.14465*, 2022.

[32] Y. Li, Z. Lin, S. Zhang, Q. Fu, B. Chen, J.-G. Lou, and W. Chen, "On the advance of making language models better reasoners," *arXiv preprint arXiv:2206.02336*, 2022.

[33] A. Lazaridou, E. Gribovskaya, W. Stokowiec, and N. Grigorev, "Internet-augmented language models through few-shot prompting for open-domain question answering," *arXiv preprint arXiv:2203.05115*, 2022.

[34] W. Saunders, C. Yeh, J. Wu, S. Bills, L. Ouyang, J. Ward, and J. Leike, "Self-critiquing models for assisting human evaluators," *arXiv preprint arXiv:2206.05802*, 2022.

[35] O. Tafjord, B. D. Mishra, and P. Clark, "Proofwriter: Generating implications, proofs, and abductive statements over natural language," *arXiv preprint arXiv:2012.13048*, 2020.

[36] B. Dalvi, P. Jansen, O. Tafjord, Z. Xie, H. Smith, L. Pipatanangkura, and P. Clark, "Explaining answers with entailment trees," *arXiv preprint arXiv:2104.08661*, 2021.

[37] D. Dohan, W. Xu, A. Lewkowycz, J. Austin, D. Bieber, R. G. Lopes, Y. Wu, H. Michalewski, R. A. Saurous, J. Sohl-dickstein, *et al.*, "Language model cascades," *arXiv preprint arXiv:2207.10342*, 2022.

[38] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 346–361, 2021.

[39] P. Patel, S. Mishra, M. Parmar, and C. Baral, "Is a question decomposition unit all we need?," *arXiv preprint arXiv:2205.12538*, 2022.

[40] R. Fu, H. Wang, X. Zhang, J. Zhou, and Y. Yan, "Decomposing complex questions makes multi-hop qa easier and more interpretable," *arXiv preprint arXiv:2110.13472*, 2021.

[41] Z. Liang, T. Khot, S. Bethard, M. Surdeanu, and A. Sabharwal, "Better retrieval may not lead to better question answering," *arXiv preprint arXiv:2205.03685*, 2022.

[42] S. Wang, Z. Wei, Z. Fan, Q. Zhang, and X. Huang, "Locate then ask: Interpretable stepwise reasoning for multi-hop question answering," *arXiv preprint arXiv:2208.10297*, 2022.

[43] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," *arXiv preprint arXiv:1809.09600*, 2018.

[44] X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 6609–6625, International Committee on Computational Linguistics, Dec. 2020.

[45] B. Wang and A. Komatsuzaki, "GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model." https://github.com/kingoflolz/mesh-transformer-jax, May 2021.




Authorized licensed use limited to: Northcentral University. Downloaded on May 06,2023 at 00:06:52 UTC from IEEE Xplore. Restrictions apply.