

GraphIC: A Graph-Based In-Context Example Retrieval Model for Multi-Step Reasoning

Jiale Fu^{1,2}, Yaqing Wang^{3*}, Simeng Han⁴, Jiaming Fan^{1,2}, Xu Yang^{1,2*}

¹Southeast University

²Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China.

³Beijing Institute of Mathematical Sciences and Applications

⁴Stanford University

jjiale.fu@seu.edu.cn, wangyaqing@bimsa.cn, shan6@stanford.edu, {jiaming.fan, xuyang_palm}@seu.edu.cn

Abstract

In-context learning (ICL) enhances large language models (LLMs) by incorporating demonstration examples, yet its effectiveness heavily depends on the quality of selected examples. Current methods typically use text embeddings to measure semantic similarity, which often introduces bias in multi-step reasoning tasks. This occurs because text embeddings contain irrelevant semantic information and lack deeper reasoning structures. To address this, we propose **GraphIC**, a graph-based retrieval model that leverages reasoning-aware representation and specialized similarity metric for in-context example retrieval. GraphIC first constructs *thought graphs*—directed, node-attributed graphs that explicitly model reasoning steps and their dependencies—for candidate examples and queries. This approach filters out superficial semantics while preserving essential reasoning processes. Next, GraphIC retrieves examples using a novel similarity metric tailored for these graphs, capturing sequential reasoning patterns and asymmetry between examples. Comprehensive evaluations across mathematical reasoning, code generation, and logical reasoning tasks demonstrate that GraphIC outperforms 10 baseline methods. Our results highlight the importance of reasoning-aware retrieval in ICL, offering a robust solution for enhancing LLM performance in multi-step reasoning scenarios.

Code — <https://github.com/jialefu/GraphIC>

1 Introduction

In-context learning (ICL) (Brown et al. 2020; Peng et al. 2024; Jiang et al. 2025) allows large language models (LLMs) to adapt to new tasks by incorporating a few demonstration examples within the input prompt, without updating model parameters. However, studies reveal that ICL performance heavily depends on the quality of selected in-context examples (ICEs) (Zhao et al. 2021; Yang et al. 2023a; Li et al. 2024), motivating extensive research on ICE selection strategies. Current approaches (Liu et al. 2022; Rubin, Herzig, and Berant 2022) typically use text embeddings to measure semantic similarity between queries and candidate

examples, achieving success in semantic-centric tasks like text classification and translation (Agrawal et al. 2023).

However, these text-based methods face significant limitations in multi-step mathematical and logical reasoning tasks. This is because text embedding encodes a substantial amount of shallow semantic information, which is irrelevant to the underlying reasoning processes. This extraneous information introduces bias in the selection of ICEs (An et al. 2023). As shown in Figure 1 (left), when solving a speed calculation problem, text-based methods may retrieve examples about distance/time calculations due to shallow semantic similarity, inducing incorrect reasoning paths (e.g., calculating distance instead of speed). This observation motivates our key insight: Effective ICE selection for reasoning tasks requires representations that explicitly model cognitive processes rather than textual surface forms.

Drawing from cognitive science (Friston 2008) and graph-based reasoning works (Besta et al. 2024; Yao et al. 2023), we propose *thought graphs*—directed node-attributed graphs where nodes represent reasoning steps and edges denote step dependencies (i.e., a child step can only proceed after the parent step is completed). This representation filters irrelevant semantics while preserving essential reasoning patterns. An example of a thought graph is shown in Figure 2 (a).

Furthermore, we propose a similarity metric for thought graphs tailored to multi-step reasoning tasks. Specifically, we introduce a parameter W to model the reasoning pattern of a given process. Given two thought graphs, G_1 and G_2 , we first estimate the reasoning pattern parameter W_1 for G_1 , then assess its applicability to G_2 . This applicability serves as the similarity measure $s(G_1, G_2)$. Our approach naturally captures the sequential structure of reasoning steps and accounts for the inherent asymmetry between examples, which we discuss in detail in Section 3.2.

Building upon these foundations, we introduce GraphIC, a **Graph-based In-Context Example Retrieval Model**. GraphIC achieves reasoning-aware example selection through three key phases: (1) constructing thought graphs for both the query and candidate examples, (2) calculating graph similarity using our proposed metric, and (3) retrieving the top- k most relevant examples. As illustrated in Figure 1 (right), GraphIC effectively identifies examples that

*Corresponding author

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Question:

Marissa is hiking a 12-mile trail. She took 1 hour to walk the first 4 miles, then another hour to walk the next two miles. If she wants her average speed to be 4 miles per hour, what speed (in miles per hour) does she need to walk the remaining distance?

Text-based Retrieval:

Q: Jeannie hikes the 12 miles to Mount Overlook at a pace of 4 miles per hour, and then returns at a pace of 6 miles per hour. How long did her hike take, in hours?

A: ...

Q: Sadie, Ariana and Sarah are running a relay race. Each part of the race is a different length and covers different terrain. It takes Sadie 2 hours to run through the forest at an average speed of 3 miles per hour. Ariana sprints across the open field on her section at 6 miles per hour for half an hour. If Sarah runs along the beach at four miles per hour and their total time for the race is four and half hours, what is the total distance of the race?

A: ...

Output:

12 miles / 4 mph = 3 hours
She has already walked for $1 + 1 + 1 = 3$ hours
She has $3 - 3 = 0$ hours left to walk the remaining distance
She has 0 miles left to walk ❌

Graph-based Retrieval:

Q: Rachel is stuffing envelopes. She has eight hours to complete the task, and there are 1,500 envelopes. In the first hour, Rachel stuffs 135 envelopes. The second hour she stuffs 141 envelopes. How many envelopes will Rachel need to stuff per hour to finish the job?

A: ...

Q: Allie has 9 toys, which are in total worth \$52. If we know that one toy is worth \$12, and all the other toys have the same value, how much does one of the other toys cost?

A: ...

Output:

Marissa has walked $4 + 2 = 6$ miles so far.
She has $12 - 6 = 6$ miles left to walk.
She wants to walk the entire trail in $12 \text{ miles} / 4 \text{ mph} = 3$ hours.
She has already walked for $1 + 1 = 2$ hours.
She has $3 - 2 = 1$ hour left to walk the remaining 6 miles. ✓
She needs to walk $6 \text{ miles} / 1 \text{ hour} = 6 \text{ mph}$.

Figure 1: ICL with different ICE retrieval mechanisms. The left panel shows examples retrieved via BERT embedding (Devlin et al. 2019), while the right panel displays examples retrieved via GraphIC. Semantically related terms are highlighted in blue, and quantities or variables needing resolution are highlighted in green.

align with the reasoning process (such as calculations for the number of envelopes to stuff per hour or unit price), even if they lack semantic similarity to the query (e.g., speed computation). This enables the LLM to solve the problem correctly.

Through comprehensive evaluations across mathematical reasoning, code generation, and logical reasoning tasks, GraphIC demonstrates superior performance over 10 baseline methods including both training-free and training-based approaches. To sum up, our key contributions are:

1. **A representaion.** We introduce a novel graph-based representation, called *thought graph*, to model multi-step reasoning processes. This representation effectively filters out irrelevant shallow semantic information while preserving the essential reasoning steps.
2. **A similarity metric.** We introduce a similarity metric for thought graphs tailored to multi-step reasoning tasks that capture the sequential nature of the steps and the asymmetry between examples.
3. **Empirical validation.** Our experimental results indicate that GraphIC, despite being a training-free model, outperforms both training-free and training-based models across various multi-step reasoning tasks.

2 Related Work

Existing ICE selection techniques can be classified as either training-free or training-based, depending on whether a re-

triever needs to be trained.

Training-free approaches are generally divided into two types: (i) those that use heuristic criteria such as similarity (Liu et al. 2022; Hu et al. 2022), diversity (Cho et al. 2023; Zhang et al. 2022; Levy, Bogin, and Berant 2023; Hongjin et al. 2022; Zhang et al. 2023), complexity (Fu et al. 2022), or combinations of these (Agrawal et al. 2023; Tonglet et al. 2023; Gupta, Gardner, and Singh 2023) to select in-context examples (ICEs); (ii) those that leverage feedback from LLMs, such as probability distributions (Wu et al. 2023; Nguyen and Wong 2023; Li and Qiu 2023; Yang et al. 2023b), perplexity (Gonen et al. 2023), or the model’s generated output (An et al. 2023) to guide the selection process. Training-free approaches eliminate the computational and time costs of model training, but their simpler architecture often leads to lower performance than training-based methods.

Training-based methods are generally divided into two types. The first learns to select individual examples and then extends this to k -shot scenarios (Rubin, Herzig, and Berant 2022; Xiong et al. 2024; Gupta, Rosenbaum, and Elenberg 2024). The second models the selection of a group of examples as a whole (Ye et al. 2023; Wang et al. 2023; Zhang, Feng, and Tan 2022; Scarlatos and Lan 2023; Lu et al. 2022; Xu, Banburski, and Jovic 2024; Yang et al. 2024). Training-based approaches often deliver better performance, but they are computationally expensive and time-consuming.

soning Pattern: Solve Equation (1) for G_1 to obtain W_1 , which encapsulates G_1 's reasoning pattern. (2). **Compute Applicability:** Evaluate how well the reasoning pattern W_1 applies to thought graph G_2 . This applicability, denoted as $\mathcal{A}(W_1; G_2)$, serves as the similarity measure $s(G_1, G_2)$. We now provide a detailed explanation of these two steps:

Firstly, we derive a matrix W that captures directional relationships in G . This is achieved by solving the following optimization problem:

$$W = \arg \max_{\substack{W \in \mathbb{R}^{d \times d}, \\ \|W\|_F = 1}} \mathcal{A}(W; G), \quad \text{where} \quad (1)$$

$$\mathcal{A}(W; G) = \sum_{i=1}^n \left(\underbrace{\sum_{j \in \text{pa}(v_i)} x_j}_{z_i} \right)^\top W x_i.$$

Here, the term $z_i = \sum_{j \in \text{pa}(v_i)} x_j$ calculates the sum of the attributes of v_i 's parent nodes, aggregating precursor information for step v_i . $z_i^\top W$ represents a linear transformation of this information. $z_i^\top W x_i$ measures the inner product similarity between transformed precursor information $W z_i$ and the current step's attribute x_i . By maximizing $\mathcal{A}(W; G)$ under the Frobenius norm constraint ($\|W\|_F = 1$), W learns to predict the attribute of the next step based on precursor information, effectively encoding the directional relationships in G .

For instance, consider a thought graph G with two vertices: v_1 with attribute $x_1 = \text{Emb}(\text{"add"})$ and v_2 with $x_2 = \text{Emb}(\text{"multiply"})$, connected by an edge $v_1 \rightarrow v_2$. In this setup, W learns to map x_1 ("add") to x_2 ("multiply"), indicating that "multiply" typically follows "add" in G . When this learned mapping is applied to another graph G' , a high value of $\mathcal{A}(W; G')$ suggests that G' also contains frequent "add" \rightarrow "multiply" sequences, reflecting similar problem-solving logic. Therefore, $\mathcal{A}(W; G')$ quantifies how well the reasoning pattern of G applies to G' . A higher $\mathcal{A}(W; G')$ indicates an alignment between the two graphs' transitions, indicating that G' follows a similar problem-solving trajectory. This metric, therefore, provides an effective measure of the similarity in reasoning between G and G' , making it suitable for retrieving ICEs that share compatible problem-solving logic.

Proposed Similarity Metric. The similarity metric in GraphIC builds upon the simplified case while introducing two critical enhancements to better model reasoning dynamics and computational efficiency.

Firstly, in Equation (1), the information state z_i for each node aggregates only its direct parent attributes. However, human reasoning often incorporates information from multiple prior steps and occasionally revisits initial premises. To capture this, we propose an iterative aggregation process:

$$Z^{(h+1)} = [(1 - \lambda)\tilde{A} + \lambda\tilde{B} + I]Z^{(h)}, \quad (2)$$

where $Z^{(h)}$ represents node information states at iteration h ; $Z^{(0)}$ is initialized with node attributes X ; I is identity matrix; $\tilde{A} = D_A^{-\frac{1}{2}} A D_A^{-\frac{1}{2}}$, $D_A = \text{diag}(\text{deg}_{\text{out}}(v_1), \dots, \text{deg}_{\text{out}}(v_1))$; $\tilde{B} = D_B^{-\frac{1}{2}} B D_B^{-\frac{1}{2}}$, and

B is traceback matrix enabling backtracking to root nodes, defined as $B_{ij} = 1$ if $\text{deg}_{\text{in}}(v_j) = 0$ and $\text{deg}_{\text{in}}(v_i) > 0$, otherwise $B_{ij} = 0$, with $\text{deg}_{\text{in}}(v_j)$ representing the in-degree of node v_j . A visual example of matrix B is presented in the supplementary material. λ is a balancing hyperparameter ($0 \leq \lambda \leq 1$).

In each iteration, z_i is updated by combining three information sources, corresponding to the matrices A , B , and I . Specifically, $AZ^{(h)}$ aggregates information from direct parent nodes, similar to the simplified case; $BZ^{(h)}$ facilitates backtracking in the reasoning process; and $IZ^{(h)}$ maintains current information.

Secondly, we observe that directly solving the optimization problem in Equation (1) presents challenges related to both uniqueness and computational efficiency. Specifically, the number of vertices n in the thought graph is typically much smaller than the embedding dimension d (i.e., $n \ll d$). For instance, in the GSM8K dataset, thought graphs often contain fewer than 10 vertices, whereas the embedding dimension d can reach up to 768 when using BERT. This large disparity in dimensions leads to a non-unique solution for W . Additionally, storing and computing a $d \times d$ matrix is computationally expensive. To address both the issues of uniqueness and computational efficiency, we constrain W to be rank 1, allowing it to be expressed as $W = \alpha\beta^\top$. As a result, $\mathcal{A}(W; G)$ simplifies to:

$$\mathcal{A}(W; G) = \mathcal{A}(\alpha, \beta; G) = \sum_{i=1}^n z_i^\top \alpha \beta^\top x_i. \quad (3)$$

Both theoretical analysis and empirical evidence, presented in the supplementary material, demonstrate that the rank-1 assumption has minimal impact on solving the optimization problem.

Furthermore, with the rank-1 assumption, we can obtain the closed-form solution of Equation (1), as shown in Equation (4), which greatly reduces the computational complexity of solving the optimization problem. The proof is shown in the supplementary material

$$W = \alpha\beta^\top, \alpha = U[0, :], \beta = V[0, :] \quad (4)$$

where $U, \Sigma, V = \text{SVD}(X^\top Z)$.

Discussion. Here, we compare proposed similarity with a widely used attributed-graph similarity to highlight two key properties of our metric: encoding directional relationships and asymmetry.

A common attributed-graph similarity metric computes the graph's embedding through an iterative formula, and then evaluates the similarity between graphs by calculating the cosine similarity of their embeddings. The iterative formula generally takes the following form:

$$X^{(h+1)} = \tilde{A}X^{(h)}, X^{(0)} = X. \quad (5)$$

Firstly, the embedding-based approach fails to effectively capture the sequential relationships between vertices, which are crucial in multi-step reasoning tasks. In these tasks, the order of operations can significantly alter the reasoning pattern. Therefore, the embedding-based similarity is not well-suited for such tasks. In contrast, our proposed similarity accounts for the transformation of information from previous

vertices to current ones, inherently capturing these sequential relationships.

Secondly, our proposed similarity is asymmetric. Specifically, for two thought graphs G_1 and G_2 , $s(G_1, G_2) \neq s(G_2, G_1)$. This asymmetry reflects real-world scenarios. For instance, as illustrated in the supplementary material, mathematical problem A might be a subproblem of problem B. In such a case, referencing B can help resolve A, but referencing A does not necessarily resolve B.

3.3 Example Retrieval

After introducing thought graphs and the proposed similarity metric, we now describe how GraphIC enhances ICL. During the preparation stage, GraphIC generates thought graphs for all candidate examples and estimates their reasoning patterns, denoted as W_1, \dots, W_N , where N is the number of candidates. Then, given a query, GraphIC creates a thought graph for the query, denoted as G^q , computes the applicability of each reasoning pattern to this thought graph, $\mathcal{A}(W_i, G^q)$, and then selects the top k as ICEs.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluations. We conduct experiments across four multi-step reasoning tasks: mathematical reasoning (GSM8K (Cobbe et al. 2021) and AQUA (Ling et al. 2017)), code generation (MBPP (Austin et al. 2021)), and logical reasoning (ProofWriter (Taffjord, Dalvi, and Clark 2021)). Model performance is measured by answer accuracy for GSM8K, AQUA, and ProofWriter, and by the pass@1 metric (Chen et al. 2021) for MBPP.

Models and Hyper-parameters. We use both an open-source and a closed-source model. For the open-source model, we select Llama-3.1-8B-Instruct (hereafter referred to as Llama-3), one of the most advanced 7B-level models. For the closed-source model, we choose GPT-4o-mini, balancing performance and testing costs. Unless explicitly mentioned otherwise, all evaluations use an 8-shot setting, which is the most common setting in chain-of-thought scenarios. We set the temperature to 1e-5 to minimize randomness, ensuring consistency across generations, and use 3 iterations in Equation (2) with λ values from $\{0, 0.1, 0.2, 0.3\}$ (see the supplementary material for specific details). More experiment details are in the supplementary material.

4.2 Baselines

We compare GraphIC against six training-free retrieval methods spanning random, similarity-based, diversity-based, and complexity-based approaches, including: (1) **Random** randomly selects k unique ICEs from the candidate set; (2) **BM25** (Robertson, Zaragoza et al. 2009) selects the top k examples based on BM25 scoring; (3) **BERT** (Devlin et al. 2019) is a dense retriever using cosine similarity with BERT-base-uncased embeddings; (4) **Complex-CoT** (Fu et al. 2022) selects k examples based on complexity, quantified by newline characters; (5) **Auto-CoT** (Zhang et al. 2022) clusters candidates and selects the closests to each cluster center; and (6) **Skill-kNN** (An

et al. 2023) prompts LLM to generate task-relevant skills for query and candidates, followed by dense retrieval.

We also compare with four training-based methods, which encompass both single-example and combination-example retrieval strategies, including: (1) **EPR** (Rubin, Herzig, and Berant 2022) is trained to retrieve the single most relevant ICE, with top k examples being selected during inference; (2) **CEIL** (Ye et al. 2023) uses determinantal point processes to select ICEs balancing similarity and diversity; (3) **DQ-LoRe** (Xiong et al. 2024) uses dual queries and low-rank approximation re-ranking to identify ICEs; and (4) **GistScore** (Gupta, Rosenbaum, and Elenberg 2024) encodes task-specific information into gist tokens for selecting ICEs.

4.3 Main Results

Table 1 shows the performance comparison results. As a training-free method, GraphIC consistently outperforms both training-free and training-based baselines in most settings.

We find that training-based baselines generally outperform training-free baselines, as they learn explicit patterns from in-context examples. For instance, with Llama-3, training-based methods average over 67% performance, while the top training-free baseline (Complex-CoT) reaches only 66.54%. Among the training-based baselines, DQ-LoRe is the most effective, reaching 68.33%. This method leverages LLM-generated outputs during retrieval, demonstrating that incorporating LLM outputs improves performance on reasoning tasks. Among training-free baselines, Complex-CoT and BM25 perform best. Notably, both use asymmetric retrieval, reinforcing that asymmetric retrieval aligns well with real-world reasoning scenarios.

GraphIC integrates the strengths of existing methods—utilizing LLM outputs for reasoning tasks and asymmetric retrieval—while introducing a reasoning-aware representation and similarity metric. Consequently, it not only surpasses all training-free baselines by (2.57% with GPT-4o-mini and 4.29% with Llama-3), but also outperforms all training-based methods (by 1.18% using GPT-4o-mini and 2.5% using Llama-3), highlighting its effectiveness in reasoning tasks.

A closer analysis highlights GraphIC’s substantial advantages in mathematical and logical reasoning tasks compared to code generation, particularly for complex problem instances. For example, on GSM8K, GraphIC outperforms all baselines by 0.65% and 3.57% with the two LLMs. In the more challenging AQUA dataset, the performance improvements become even more pronounced, reaching 3.47% and 7.64%.

4.4 Ablation Study

We conduct a series of ablation studies to systematically evaluate the contribution of each component in GraphIC, focusing on three main aspects: filtering shallow semantic information, integrating the graph structure, and the proposed similarity metric.

To this end, we develop several variants of the GraphIC model: (1) **Text** relies solely on text embedding, the same as

LLM	Model	GSM8K	AQUA	MBPP	ProofWriter	Avg.	
GPT-4o-mini	Random	92.90 (0.31)	71.58 (0.72)	72.76 (0.74)	64.90 (0.93)	75.54 (0.36)	
	BM25	92.64	70.47	73.4	66.25	75.69	
	BERT	93.02	66.93	74.2	65.25	74.85	
	Complex-CoT	92.49	67.32	74.2	64.25	74.57	
	Auto-CoT	92.72	69.69	73.8	62.25	74.62	
	Skill-kNN	92.34	71.65	72.0	66.00	75.50	
	EPR	93.02	72.04	73.8	68.50	76.84	
	CEIL	92.57	<u>72.44</u>	73.8	<u>69.50</u>	<u>77.08</u>	
	DQ-LoRe	<u>93.32</u>	69.69	<u>74.6</u>	66.50	76.03	
	GistScore	93.25	69.69	<u>72.8</u>	67.00	75.69	
	GraphIC	93.48	73.62	75.2	70.75	78.26	
	Llama-3.1-8B-Instruct	Random	78.86 (0.87)	53.15 (1.85)	57.72 (1.06)	76.10 (2.45)	66.46 (0.84)
		BM25	77.71	46.85	<u>62.0</u>	77.75	66.08
		BERT	74.15	50.39	60.8	73.75	64.77
Complex-CoT		<u>79.30</u>	50.00	58.6	78.25	66.54	
Auto-CoT		72.78	42.91	58.4	78.00	63.02	
Skill-kNN		77.56	50.39	60.8	74.00	65.69	
EPR		75.66	53.94	<u>62.0</u>	79.25	67.71	
CEIL		75.51	51.97	62.4	81.00	67.72	
DQ-LoRe		77.93	<u>54.33</u>	59.8	<u>81.25</u>	<u>68.33</u>	
GistScore		74.60	44.49	60.4	79.50	64.75	
GraphIC		79.98	57.48	61.6	84.25	70.83	

Table 1: Main results on two LLMs and four datasets. For random retrieval, we present the mean and standard deviation derived from five independent experiments.

the BERT approach; (2) **FRR** retrieves examples using text embeddings of FRRs; (3) **Graph** employs Equation (5) to generate graph embeddings and use cosine similarity to retrieve examples; (4) **Graph+B** employs Equation (2) to generate graph embeddings and use cosine similarity to retrieve examples; (5) **Graph+S** excludes the backtracking mechanism from the full GraphIC model during computing Z ; and (6) **Graph+B+S** represents the full GraphIC model, integrating all components.

We conduct experiments using Llama-3 across four datasets, with the results presented in Table 2. First, we observe that employing FRR to compute text embeddings significantly improves performance (FRR versus Text), especially on GSM8K and ProofWriter, with performance increasing from 74.15 to 78.31 and from 73.75 to 82.50, respectively. This improvement is due to FRR’s ability to filter out substantial shallow semantic noise. Second, converting FRR into a graph representation (i.e., a thought graph) further improves performance (Graph versus FRR), as seen on the AQUA dataset, where performance rises from 50.78 to 54.72. This suggests that the graph structure is better suited for capturing the reasoning process than linear text. Third, introducing the reasoning-aware similarity metric also enhances performance (Graph+B+S versus Graph), increasing from 54.72 to 57.48 on AQUA, highlighting the effectiveness of reasoning-aware similarity. Additionally, our experiments show that incorporating the traceback mechanism better simulates the thought process, leading to further perfor-

mance gains (Graph+B versus Graph, and Graph+B+S versus Graph+S).

In summary, the findings emphasize that each component of GraphIC contributes significantly to improving model performance, with the most substantial gains occurring when all components are combined.

Model	GSM8K	AQUA	MBPP	ProofWriter
Text	74.15	50.39	60.8	73.75
FRR	78.31	50.78	60.4	82.50
Graph	78.46	54.72	60.4	83.50
+B	78.92	56.30	61.0	83.75
+S	79.07	49.21	60.4	84.25
+B+S	79.98	57.48	61.6	84.25

Table 2: Ablation Study.

4.5 Analysis

The performance of GraphIC steadily improves as k increases. We analyze how the performance of GraphIC and top training-based/free baselines (DQ-LoRe and Complex-CoT) changes as k increases. Specifically, we vary the number of examples k in the set $\{1, 2, 4, 8\}$ and use Llama-3 as LLM.

As shown in Figure 3, we observe that model performance generally improves with k . Notably, GraphIC consis-

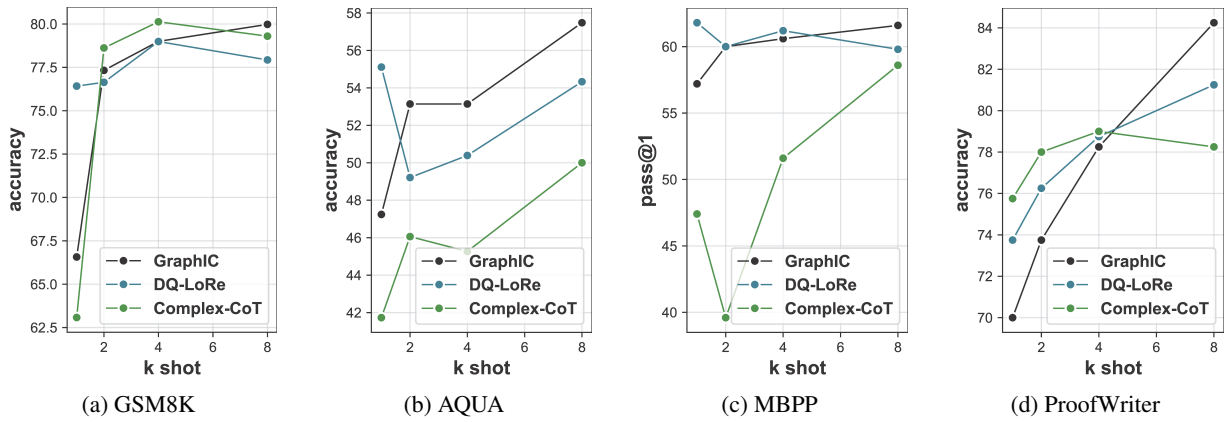


Figure 3: Performance of GraphIC and Top Training-based/Training-free Baselines (DQ-LoRe and Complex-CoT) across 1–8 Shot Settings.

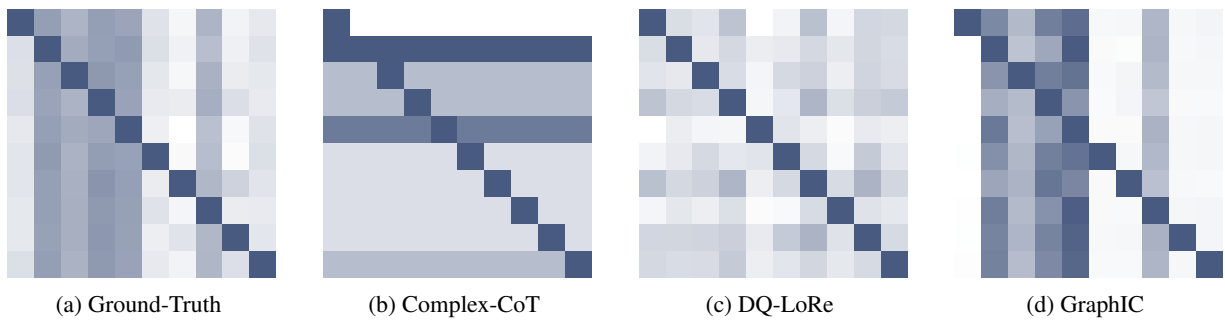


Figure 4: Ground-truth matrix and score matrices of various models. The matrix values have been linearly scaled to the range [0,1], with darker shades representing values closer to 1, and the diagonal elements have been set to 1.

tently performs better as k increases, whereas other methods experience performance degradation. Additionally, training-based and training-free methods show different trends with k increases. Specifically, training-based methods, such as DQ-LoRe, directly optimize the probability of LLM generating correct answers in 1-shot scenarios. As a result, they tend to achieve superior performance in low-shot settings, especially in 1-shot cases. However, as k increases, the performance gains of these methods tend to slow down or even decline. In contrast, training-free methods, like Complex-CoT and GraphIC, typically underperform compared to training-based methods in low-shot settings. However, as the value of k increases, they exhibit more stable improvement.

Asymmetric retrieval aligns more closely with real-world scenarios. As previously discussed, GraphIC is an asymmetric method that reflects real-world scenarios. To further validate this asymmetry, we randomly select 10 examples from GSM8K and compute their “improve matrix” S , which captures the mutual improvement relationships between examples. In this matrix, S_{ij} is defined as the probability of solving the j -th example correctly when using the i -th example as the ICE. The resulting improve matrix (Figure 4 (a)), clearly illustrates the asymmetry in real-world scenarios: while the i -th example may aid in solving the j -th example accurately, the reverse is not necessarily true.

Additionally, we examine the alignment between the score matrices of GraphIC and top training-free/based models with the ground-truth improve matrix. Existing ICE retrieval methods select ICEs by assigning scores to (candidate, query) pairs and comparing them. Similar to the improve matrix, we can compute a score matrix for each method, reflecting the mutual improvement relationships between examples estimated by the method. Naturally, the closer a method’s score matrix aligns with the improve matrix, the better it captures real-world relationships. The score matrices of Complex-CoT, DQ-LoRe, and GraphIC are shown in Figure 4 (b), (c), and (d), respectively. The results clearly demonstrate that GraphIC aligns well with the ground-truth, while the other two models exhibit noticeable deviations.

5 Conclusion

We propose GraphIC, which employs a reasoning-aware representation—*thought graph* and a similarity metric to retrieve ICEs. Our experiments show that GraphIC outperforms both training-free and training-based baselines, highlighting the importance of modeling reasoning structures and providing insights for future research on graph-based solutions for complex problem-solving.

Acknowledgments

This work is supported by the National Science Foundation of China (62576091) and the Fundamental Research Funds for the Central Universities (2242025K30024). Additional support was provided by the Big Data Computing Center of Southeast University and the SEU Kunpeng & Ascend Center of Cultivation. Y. Wang is sponsored by Beijing Nova Program.

References

- Agrawal, S.; Zhou, C.; Lewis, M.; Zettlemoyer, L.; and Ghazvininejad, M. 2023. In-context Examples Selection for Machine Translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, 8857–8873.
- An, S.; Zhou, B.; Lin, Z.; Fu, Q.; Chen, B.; Zheng, N.; Chen, W.; and Lou, J.-G. 2023. Skill-Based Few-Shot Selection for In-Context Learning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.
- Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 17682–17690.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, 1877–1901.
- Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Cho, H.; Kim, H. J.; Kim, J.; Lee, S.-W.; Lee, S.-g.; Yoo, K. M.; and Kim, T. 2023. Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12709–12718.
- Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.
- Friston, K. 2008. Hierarchical models in the brain. *Plos Computational Biology*, 4(11): e1000211.
- Fu, Y.; Peng, H.; Sabharwal, A.; Clark, P.; and Khot, T. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.
- Gonen, H.; Iyer, S.; Blevins, T.; Smith, N. A.; and Zettlemoyer, L. 2023. Demystifying Prompts in Language Models via Perplexity Estimation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Gupta, S.; Gardner, M.; and Singh, S. 2023. Coverage-based Example Selection for In-Context Learning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Gupta, S.; Rosenbaum, C.; and Elenberg, E. R. 2024. GistScore: Learning Better Representations for In-Context Example Selection with Gist Bottlenecks. In *Forty-First International Conference on Machine Learning*.
- Hongjin, S.; Kasai, J.; Wu, C. H.; Shi, W.; Wang, T.; Xin, J.; Zhang, R.; Ostendorf, M.; Zettlemoyer, L.; Smith, N. A.; et al. 2022. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.
- Hu, Y.; Lee, C.-H.; Xie, T.; Yu, T.; Smith, N. A.; and Ostendorf, M. 2022. In-Context Learning for Few-Shot Dialogue State Tracking. In *The 2022 Conference on Empirical Methods in Natural Language Processing*, 2627–2643.
- Jiang, Y.; Fu, J.; Hao, C.; Hu, X.; Peng, Y.; Geng, X.; and Yang, X. 2025. Mimic In-Context Learning for Multimodal Tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 29825–29835.
- Levy, I.; Bogin, B.; and Berant, J. 2023. Diverse Demonstrations Improve In-context Compositional Generalization. In *Findings of the Association for Computational Linguistics: ACL 2023*, 1401–1422.
- Li, L.; Peng, J.; Chen, H.; Gao, C.; and Yang, X. 2024. How to configure good in-context sequence for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26710–26720.
- Li, X.; and Qiu, X. 2023. Finding Support Examples for In-Context Learning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 6219–6235.
- Ling, W.; Yogatama, D.; Dyer, C.; and Blunsom, P. 2017. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. In *Findings of the Association for Computational Linguistics: ACL 2017*, 158–167.
- Liu, J.; Shen, D.; Zhang, Y.; Dolan, W. B.; Carin, L.; and Chen, W. 2022. What Makes Good In-Context Examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 100–114.
- Lu, P.; Qiu, L.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; Rajpurohit, T.; Clark, P.; and Kalyan, A. 2022. Dynamic Prompt Learning via Policy Gradient for Semi-structured Mathematical Reasoning. In *The Eleventh International Conference on Learning Representations*.

- Nguyen, T.; and Wong, E. 2023. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*.
- Peng, Y.; Hu, X.; Peng, J.; Geng, X.; Yang, X.; et al. 2024. Live: Learnable in-context vector for visual question answering. *Advances in Neural Information Processing Systems*, 37: 9773–9800.
- Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.
- Rubin, O.; Herzig, J.; and Berant, J. 2022. Learning To Retrieve Prompts for In-Context Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2655–2671.
- Scarlatos, A.; and Lan, A. 2023. RetICL: Sequential retrieval of in-context examples with reinforcement learning. *arXiv preprint arXiv:2305.14502*.
- Tafford, O.; Dalvi, B.; and Clark, P. 2021. ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3621–3634.
- Tonglet, J.; Reusens, M.; Borchert, P.; and Baesens, B. 2023. SEER: A Knapsack approach to Exemplar Selection for In-Context HybridQA. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 13569–13583.
- Wang, X.; Zhu, W.; Saxon, M.; Steyvers, M.; and Wang, W. Y. 2023. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Advances in Neural Information Processing Systems*, volume 36, 15614–15638.
- Wu, Z.; Wang, Y.; Ye, J.; and Kong, L. 2023. Self-Adaptive In-Context Learning: An Information Compression Perspective for In-Context Example Selection and Ordering. In *Findings of the Association for Computational Linguistics: ACL 2023*.
- Xiong, J.; Li, Z.; Zheng, C.; Guo, Z.; Yin, Y.; Xie, E.; YANG, Z.; Cao, Q.; Wang, H.; Han, X.; Tang, J.; Li, C.; and Liang, X. 2024. DQ-LoRe: Dual Queries with Low Rank Approximation Re-ranking for In-Context Learning. In *The Twelfth International Conference on Learning Representations*.
- Xu, W.; Banburski, A.; and Jojic, N. 2024. Reprompting: Automated Chain-of-Thought Prompt Inference Through Gibbs Sampling. In *Forty-First International Conference on Machine Learning*.
- Yang, X.; Peng, Y.; Ma, H.; Xu, S.; Zhang, C.; Han, Y.; and Zhang, H. 2024. Lever LM: configuring in-context sequence to lever large vision language models. *Advances in Neural Information Processing Systems*, 37: 100341–100368.
- Yang, X.; Wu, Y.; Yang, M.; Chen, H.; and Geng, X. 2023a. Exploring diverse in-context configurations for image captioning. *Advances in Neural Information Processing Systems*, 36: 40924–40943.
- Yang, Z.; Zhang, Y.; Sui, D.; Liu, C.; Zhao, J.; and Liu, K. 2023b. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 11809–11822.
- Ye, J.; Wu, Z.; Feng, J.; Yu, T.; and Kong, L. 2023. Compositional Exemplars for In-context Learning. In *Fortieth International Conference on Machine Learning*.
- Zhang, S.; Xia, X.; Wang, Z.; Chen, L.-H.; Liu, J.; Wu, Q.; and Liu, T. 2023. IDEAL: Influence-driven selective annotations empower in-context learners in large language models. In *The Twelfth International Conference on Learning Representations*.
- Zhang, Y.; Feng, S.; and Tan, C. 2022. Active Example Selection for In-Context Learning. In *The 2022 Conference on Empirical Methods in Natural Language Processing*, 9134–9148.
- Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2022. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International Conference on Learning Representations*.
- Zhao, Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate before use: Improving few-shot performance of language models. In *Thirty-Eighth International Conference on Machine Learning*, 12697–12706.