GRAPHREACT: REASONING AND ACTING FOR MULTI-STEP GRAPH INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning-acting frameworks enhance large language models (LLMs) by interleaving reasoning with actions for dynamic information acquisition. However, extending this paradigm to graph learning remains underexplored. Graph data is inherently structured, with information distributed across nodes and edges and encoded through both topology and latent representations. As a result, effective reasoning over graphs requires not only retrieving informative evidence from the graph, but also progressively refining the accumulated context during multi-step inference. In this work, we propose GRAPHREACT, a graph reasoning-acting framework that enables step-by-step inference over graph-structured data. Specifically, we design a graph-based action space with two complementary retrieval actions: topological retrieval, which captures local structural dependencies, and semantic retrieval, which accesses non-local but relevant evidence in the representation space. These actions dynamically expand the reasoning context. To further support multi-step reasoning, we introduce another type of action, context refinement, which distills and reorganizes accumulated information into a compact representation. By interleaving reasoning with both retrieval and refinement actions, our framework enables a progressive transition from context expansion to compression. Extensive experiments on six benchmark datasets demonstrate that GRAPHREACT consistently outperforms state-of-the-art methods, validating the effectiveness of reasoning-acting for graph learning. Code is available at `https://anonymous.4open.science/r/GraphReAct-B416`.

1 INTRODUCTION

Recent advances in large language models have demonstrated strong reasoning capabilities through Chain-of-Thought (CoT) prompting (Wei et al., 2022; Feng et al., 2023; Zhang et al., 2023), which decomposes complex problems into intermediate reasoning steps. Building upon this paradigm, the reasoning-acting framework (Yao et al., 2022; Shen et al., 2025; Fu et al., 2025) further enhances reasoning by interleaving it with actions, allowing models to interact with external environments and acquire additional information during inference, as shown in Fig. 1(a). This reasoning-acting synergy has proven particularly effective in tasks such as question answering and decision making, where actions (e.g., search) help reduce hallucination and improve factual accuracy. These successes suggest that integrating reasoning with information acquisition is a powerful paradigm for solving complex problems.

Despite its effectiveness in natural language processing, extending the reasoning-acting framework to graph learning remains largely underexplored. Graph-structured data (Xia et al., 2021; Cook & Holder, 2006) is inherently different from textual environments: information is distributed across nodes and edges, and meaningful signals are often encoded implicitly through topology and latent representations (Kipf & Welling, 2017; Veličković et al., 2018a). Existing graph learning methods, including end-to-end graph encoders (Xu et al., 2019; Hamilton et al., 2017), finetuning (You et al., 2020; Veličković et al., 2018b), and graph prompting (Liu et al., 2023; Yu et al., 2023), typically rely on a fixed receptive field and perform single-pass inference, which may limit their ability to resolve ambiguity under sparse or incomplete information. While recent work has incorporated CoT-style reasoning into graph-related tasks (Yu et al., 2025), it lacks explicit operations for acquiring additional graph evidence during reasoning, as shown in Fig. 1(b). A key obstacle is that actions developed for natural language domains, such as web search (Yao et al., 2022), are not designed to exploit

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

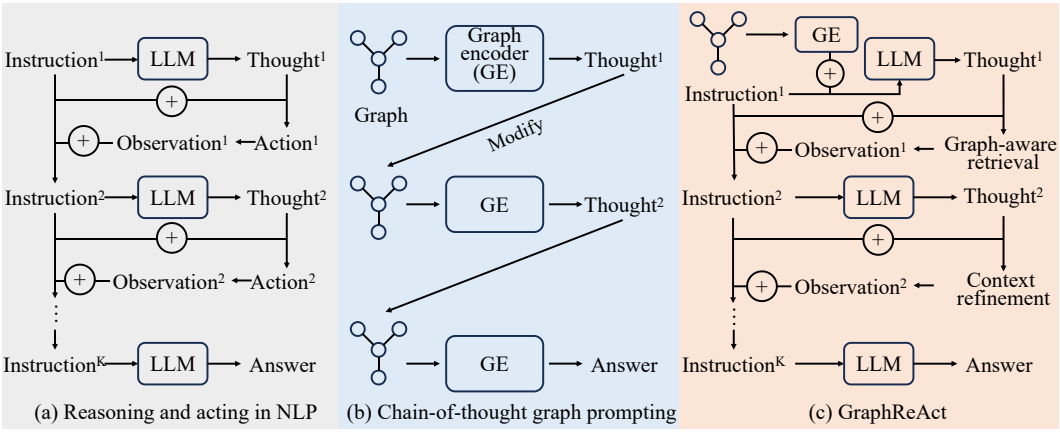


Figure 1: Comparison of reasoning paradigms. (a) Reasoning and acting in NLP. (b) CoT-based graph prompting. (c) Our proposed graph reasoning-acting (GRAPHREACT) framework.

information encoded in graph structure and representations. This raises a fundamental question: *how can we enable reasoning-acting synergy in graph learning through appropriate action design?* This problem is non-trivial due to two key challenges.

First, in natural language domains, actions typically correspond to interactions with external knowledge sources such as Wikipedia (Yao et al., 2022; Fu et al., 2025). In contrast, for graph learning tasks, while external information remains accessible, it is often misaligned with the task objective and may introduce irrelevant or noisy signals. Instead, the most informative evidence is inherently structured within the graph itself, encoded through both topology and latent representations. Therefore, directly transferring text-based action designs to graph domains is suboptimal. This raises a key challenge: *how to design graph-aware actions that can effectively retrieve informative evidence from both topological structure and semantic similarity?* In this work, we design two complementary graph-based retrieval actions for structured information acquisition, as shown in Fig. 1(c): (1) *topological retrieval*, which gathers textual information from structurally neighboring nodes to capture local dependencies, and (2) *semantic retrieval*, which identifies nodes with similar embeddings to provide non-local, semantically relevant evidence. By jointly leveraging these two actions, our framework incorporates both local and non-local graph evidence to expand the reasoning context beyond fixed receptive fields.

Second, *how to support multi-step reasoning with actions in graph domains?* In standard reasoning-acting frameworks, actions are typically used for information acquisition, where the model queries external environments to gather additional evidence (Yao et al., 2022). However, graph reasoning exhibits a different requirement: while early-stage reasoning benefits from expanding the context through information retrieval, later stages require consolidating the accumulated information to reduce redundancy and noise. This introduces an inherent tension between context expansion and compression, which is not explicitly addressed in existing reasoning-acting frameworks. To address this challenge, we introduce another type of action, termed *context refinement*, which distills, summarizes, and reorganizes the retrieved information into a more compact and informative context. Rather than allowing the model to freely select actions at each step, we organize retrieval and refinement in a predefined order: graph-based retrieval first expands the available evidence, while subsequent refinement progressively consolidates it. This structured process enables multi-step reasoning over a progressively constructed context without repeatedly retrieving additional graph information.

In summary, our contributions are fourfold. (1) We propose GRAPHREACT, a novel reasoning-acting framework for graph learning that enables step-by-step inference via structured interaction with graph-structured data. To the best of our knowledge, this is the first work that systematically extends the reasoning-acting paradigm to graph domains. (2) We propose a graph-based retrieval action space with topology-based and semantic retrieval, enabling structured and complementary information acquisition from both local neighborhoods and global semantic space. (3) We design context refinement action that balances context expansion and compression, enabling progressive

refinement of graph-aware information during inference. (4) We conduct extensive experiments on six datasets, demonstrating that GRAPHREACT consistently outperforms state-of-the-art methods and validating the effectiveness of reasoning-acting in graph learning.

2 RELATED WORK

LLMs for graph learning. Recent studies integrate LLMs into graph learning in three main ways. Some use LLMs as semantic feature extractors to enrich node representations and mitigate domain gaps (Wang et al., 2024b; He et al., 2025; Liu et al., 2024; Yuan et al., 2025). Others employ LLMs as predictors to leverage their reasoning capabilities (Tang et al., 2024a;b; Chen et al., 2024). There are also efforts to align graph representations with LLM embedding spaces to enable graph–language interaction (Tang et al., 2024a; Chen et al., 2024). Despite these advances, most methods rely on static graph–language interfaces with single-pass inference, lacking explicit multi-step reasoning and structured evidence acquisition. In contrast, our method enables iterative evidence integration through graph-aware operations, supporting multi-step reasoning over graph data.

Reasoning and acting in LLMs. Chain-of-Thought (CoT) prompting improves reasoning by decomposing problems into intermediate steps (Wei et al., 2022; Zhou et al., 2023; Wang et al., 2023; Yao et al., 2023). Building on this, reasoning-acting frameworks interleave reasoning with external actions to acquire additional information (Yao et al., 2022; Fu et al., 2025). However, directly applying such approaches to graph learning is non-trivial, as graph information is inherently structured and distributed across topology and latent semantic space. Existing methods neither exploit these structured inductive biases nor consider how graph-specific actions should be organized across different stages of multi-step reasoning. Our work instead adopts a structured reasoning-acting process tailored to graph data, organizing graph-based retrieval and progressive context refinement in a predefined sequence.

3 PRELIMINARIES

Graph encoder. A graph is defined as $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, respectively, and $\mathbf{X}$ is the node feature matrix, whose i-th row x_i corresponds to the feature vector of node $v_i \in \mathcal{V}$. We denote a collection of graphs as $\mathcal{G}$. A mainstream technique for graph representation learning is GNNs, which recursively update node representations through message passing. Let $\mathbf{H}^l \in \mathbb{R}^{|\mathcal{V}| \times d}$ denote the embedding matrix at the l-th layer, where the i-th row $\mathbf{h}_i^l$ represents the embedding of node v_i. The layer-wise update is defined as

$$\mathbf{h}_v^l = \texttt{MP}(\mathbf{h}_v^{l-1}, \{\mathbf{h}_u^{l-1} : u \in \mathcal{N}_v\}; \theta^l), \tag{1}$$

where $\mathcal{N}_v$ denotes the neighbors of v, $\texttt{MP}$ denotes the message-passing function, and θ^l represents the learnable parameters of the l-th layer. The initial embedding is given by $\mathbf{h}_v^0 = x_v$, and after L layers, the final node representations are denoted as

$$\mathbf{H} = \texttt{GE}(\mathbf{X}, G; \Theta), \tag{2}$$

where $\Theta = \{\theta^1, \ldots, \theta^L\}$.

Graph–LLM interface. Recent approaches introduce a graph-to-language interface that integrates graph encoders with LLMs, enabling LLMs to perform prediction over graph-structured data. To bridge the gap between graph representations and the LLM token space, existing methods typically first conduct pre-training based on multimodal contrastive learning (Wang et al., 2024a; Chen et al., 2024). Given a graph G and its associated text, the graph encoder $\texttt{GE}$ is trained to produce representations that are aligned with the embedding space of the LLM. Specifically, the pre-training objective encourages graph representations to align with corresponding textual representations, thereby embedding graph features into the LLM semantic space. This alignment enables graph embeddings to be interpreted as pseudo-tokens that are compatible with LLM inputs. After pre-training, the graph encoder is frozen. Given an input graph, it first produces representations $\mathbf{H}_{\mathcal{V}} = \texttt{GE}(\mathbf{X}, G; \Theta)$, and then maps them into the token embedding space of the LLM via a projection function:

$$\mathbf{H}^{\text{tok}} = \texttt{Proj}(\mathbf{H}; \phi), \tag{3}$$

where ϕ is the learnable parameter. The graph embedding tokens $\mathbf{H}^{\text{tok}}$ share the same embedding dimension as LLM tokens and are incorporated into an instruction template for the LLM.

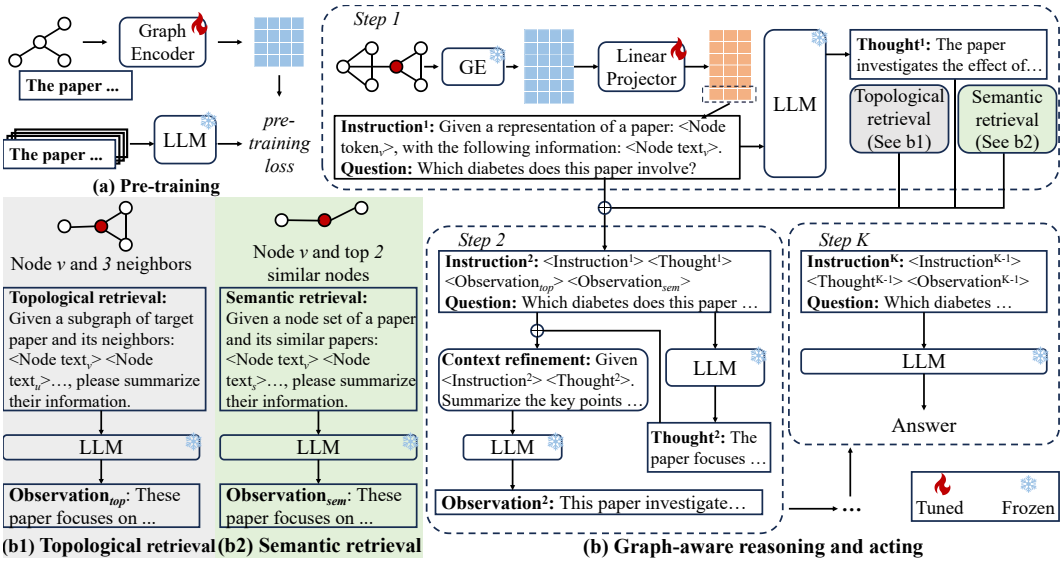


Figure 2: Overall framework of GRAPHREACT.

4 PROPOSED APPROACH

In this section, we introduce GRAPHREACT. We first provide a high-level overview of the framework and then describe its core components in detail.

4.1 OVERALL FRAMEWORK

We illustrate the overall framework of GRAPHREACT in Fig. 2, which consists of two phases: (1) pre-training, and (2) graph-based reasoning and acting. First, we pre-train a graph encoder to produce representations aligned with the embedding space of a large language model, as shown in Fig. 2(a). Details of the pre-training phase are provided in Sect. 3. Second, we propose a graph-based reasoning and acting framework to guide the LLM to perform multi-step inference via iterative context updating, as illustrated in Fig. 2(b). Specifically, at each step, the LLM generates an intermediate thought based on the current context, which is used to guide a predefined action for updating the context. We consider two types of actions. The first type is graph-based retrieval, which expands the context by incorporating information from topologically related nodes and semantically similar nodes, as shown in Fig. 2(b1,b2). The second type is context refinement, which compresses and reorganizes the accumulated context to reduce redundancy and enhance relevance. Through this progressive context updating process, the model constructs a high-quality context for final prediction.

4.2 GRAPH-BASED RETRIEVAL

Graph-based retrieval is designed to acquire informative evidence from graph-structured data, where useful signals reside in both graph topology and latent representation space. To this end, we design a unified set of action primitives that captures two complementary sources of information: structural locality and semantic similarity.

Topological retrieval. We first introduce a topology-based action that extracts structurally grounded evidence from the graph. In graphs, informative signals are often distributed along connectivity patterns, making local neighborhoods a primary source of context for reasoning. To capture such structural dependencies, we retrieve nodes that are topologically related to the target node v via a breadth-first traversal (Bundy & Wallen, 1984), collecting the first N visited neighbors. Unlike random neighbor sampling, this traversal expands the receptive field in a structured manner, ensuring a consistent retrieval budget across nodes while adaptively incorporating higher-order neighborhoods when necessary.

We then transform the retrieved structural evidence into a compact representation suitable for reasoning. Specifically, we construct a topology-based instruction by integrating the target node text with the textual information of the retrieved neighbors, and use the frozen LLM to generate a topology-based summary. This summary, denoted as s^{top}, serves as a structured abstraction of the local neighborhood, encoding aggregated structural signals into a form that can be directly consumed by subsequent reasoning steps.

Semantic retrieval. While topological retrieval captures local structural dependencies, it may fail to access semantically relevant but structurally distant evidence. To address this limitation, we introduce a complementary semantic retrieval action that operates in the representation space. Specifically, we identify nodes that are semantically similar to the target node v based on cosine similarity (Xia et al., 2015) between node embeddings, and select the top-M most similar nodes. This process enables the model to access globally relevant evidence that is not constrained by graph connectivity, thereby extending the reasoning context beyond local neighborhoods.

Similar to topological retrieval, we convert the retrieved semantic evidence into a reasoning-compatible representation. We construct a semantic-based instruction by integrating the target node text with the textual information of the retrieved semantically similar nodes, and feed it into the frozen LLM to produce a semantic-based summary. This summary, denoted as s^{sem}, provides a global semantic abstraction of the target node, complementing the locality of topology-based context and enabling reasoning over both structural and semantic dimensions. TThis dual-retrieval design provides a structured mechanism for accessing both local and global evidence, forming the foundation for graph-aware reasoning in subsequent steps.

Comparison with heterophilic graph learning. Existing methods for modeling non-local dependencies, particularly in heterophilic graphs (Ma et al., 2022; Luan et al., 2022), typically extend message passing with multi-hop aggregation or similarity-based propagation. However, these approaches rely on fixed aggregation schemes learned during training, and rely on fixed aggregation schemes and lack instance-specific evidence selection at inference time. Moreover, they operate in a single-modality setting and are not designed for zero-shot generalization across datasets. In contrast, our approach formulates graph information access as an explicit action in the reasoning process. Instead of static propagation, our approach performs explicit evidence acquisition via topological and semantic retrieval, and converts them into natural language summaries for subsequent reasoning. This enables flexible and instance-specific evidence integration, which is particularly suitable for zero-shot graph learning.

Context construction. Graph-based retrieval are applied after the initial reasoning step to construct an evidence-augmented context for subsequent reasoning. We first perform an initial inference using the graph–LLM interface:

$$\text{Thought}^1 = \texttt{LLM}(\mathcal{I}, \mathcal{Q}), \tag{4}$$

where $\mathcal{I}$ denotes the initial instruction formed by combining the target node text x^{text} and its node token embedding $\mathbf{h}_V^{\text{tok}}$, and $\mathcal{Q}$ denotes a task-level instruction specifying the prediction objective. $\mathcal{Q}$ is shared across all nodes and remains fixed during inference. We then retrieve structured evidence from the graph via graph-based retrieval:

$$\text{Observation}^1 = \texttt{Act}_{\text{retrieve}}(G, v) = \{s^{\text{top}},\ s^{\text{sem}}\}, \tag{5}$$

where s^{top} and s^{sem} denote the topological and semantic summaries, respectively. $\texttt{Act}_{\text{retrieve}}(\cdot)$ denotes a composite operation that includes topological and semantic retrieval and LLM-based abstraction into textual summaries. Finally, we construct the initial context by integrating the initial thought with the retrieved evidence:

$$\mathcal{C}^1 = \texttt{Init}(\text{Thought}^1,\ \text{Observation}^1), \tag{6}$$

where $\texttt{Init}(\cdot)$ denotes a prompt construction operation that organizes the thought and retrieved summaries into a structured instruction that serves as the input context for the LLM. $\mathcal{C}^1$ corresponds to a structured textual instruction that encodes accumulated evidence and reasoning states fed into the LLM. This initial context serves as the starting point for subsequent multi-step reasoning, where the context is progressively refined through iterative updates. Notably, these retrieval operations are applied in a predefined manner rather than being dynamically selected by the model, reflecting the structured nature of graph data. We provide detailed instruction templates in Appendix H.

4.3 CONTEXT REFINEMENT

Context refinement is designed to support multi-step reasoning by progressively updating the reasoning context under the guidance of intermediate thoughts. Unlike graph-based retrieval, which acquires external evidence from the graph, refinement is formulated as another type of action within the same action space, operating on the accumulated context and reasoning signals to enable structured consolidation and reorganization of information across steps.

At each reasoning step, the model first generates a new thought based on the current context. This thought is then used to guide a predefined refinement operation that updates the context. To unify different forms of context updates, we define a generalized refinement operator that produces an observation:

$$\text{Observation}^{k+1} = \texttt{Act}_{\text{refine}}(\mathcal{C}^k,\ \text{Thought}^{k+1}), \tag{7}$$

where $\texttt{Act}_{\text{refine}}(\cdot)$ denotes a refinement action that transforms the accumulated context into a more compact and informative representation. The updated context is then constructed by integrating the previous context, the newly generated thought, and the resulting observation:

$$\mathcal{C}^{k+1} = \texttt{Update}(\mathcal{C}^k,\ \text{Thought}^{k+1},\ \text{Observation}^{k+1}), \tag{8}$$

where $\texttt{Update}(\cdot)$ incorporates the observation into the existing context while preserving relevant historical information. Specifically, the refinement action is implemented via instruction-guided generation, where the LLM performs a reasoning-conditioned transformation over $\mathcal{C}^k$ and Thought^{k+1}, conducting information distillation and structural reorganization. The resulting observation serves as a condensed abstraction of the accumulated context, analogous to the evidence summaries obtained during graph-based retrieval.

To summarize the overall reasoning process, at each step k, the LLM operates on a structured input consisting of the node-specific instruction, current context, and task query:

$$\text{Thought}^{k+1} = \texttt{LLM}(\mathcal{I},\ \mathcal{C}^k,\ \mathcal{Q}). \tag{9}$$

The generated thought is then used to guide the corresponding refinement operation as defined in Eq. (7) and Eq. (8). At the final step, the LLM directly produces the prediction:

$$\hat{y} = \texttt{LLM}(\mathcal{I},\ \mathcal{C}^K,\ \mathcal{Q}). \tag{10}$$

4.4 ADAPTATION AND INFERENCE

We consider a node classification task with a labeled training set $\mathcal{D} = \{(v_1, y_1), (v_2, y_2), \dots\}$, where $v_i \in \mathcal{V}$ denotes a target node and $y_i \in Y$ is its class label. Given a target node v, our framework performs multi-step reasoning over the constructed context and produces a predictive distribution over labels via the LLM. Let $p(y \mid v)$ denote the probability of predicting label y from the LLM output conditioned on the final instruction. The training objective is defined as the negative log-likelihood:

$$\mathcal{L}_{\text{down}}(\mathcal{D}; \phi) = -\sum_{(v_i, y_i) \in \mathcal{D}} \log p(y_i \mid v_i; \phi), \tag{11}$$

where the probability is derived from the LLM output via label verbalization or answer matching. During training, both the graph encoder and the LLM are kept frozen. The only trainable component is the projection function $\texttt{Proj}(\cdot; \phi)$, which maps graph representations into the LLM token embedding space. By optimizing ϕ, the model learns to better align graph-derived representations with the LLM input space for downstream prediction.

For cross-dataset inference, the learned projection function can be directly applied to unseen graphs without further training. Specifically, given a new graph and a target instance, we construct the corresponding instruction using the projected graph representations, and perform the same multi-step reasoning process to obtain predictions. This enables GRAPHREACT to be applied to unseen graphs in a zero-shot manner, leveraging the learned graph–LLM alignment without requiring additional task-specific training.

5 EXPERIMENTS

In this section, we conduct experiments to evaluate GRAPHREACT and analyze the empirical results.

Table 1: Accuracy of zero-shot node classification.

Model	Citation		E-commerce			
	Cora	Pubmed	Children	History	Photo	Sports
MLP	0.021±0.006	0.323±0.027	0.029±0.037	0.080±0.041	0.110±0.070	0.042±0.021
GCN	0.017±0.004	0.288±0.092	0.030±0.018	0.063±0.042	0.103±0.047	0.042±0.025
GRAPHSAGE	0.014±0.007	0.316±0.058	0.008±0.007	0.195±0.206	0.056±0.055	0.051±0.015
GAT	0.016±0.004	0.343±0.064	0.086±0.084	0.172±0.098	0.050±0.027	0.142±0.138
NODEFORMER	0.016±0.007	0.308±0.093	0.048±0.028	0.168±0.127	0.073±0.015	0.165±0.057
DIFFORMER	0.029±0.014	0.361±0.071	0.129±0.030	0.275±0.171	0.321±0.055	0.306±0.131
DGI	0.026±0.009	0.329±0.103	0.082±0.035	0.218±0.168	0.224±0.127	0.049±0.017
GKD	0.042±0.008	0.399±0.033	0.202±0.064	0.339±0.138	0.166±0.086	0.208±0.077
GLNN	0.031±0.006	0.390±0.011	0.187±0.012	0.283±0.021	0.403±0.019	0.317±0.048
VICUNA-7B-V1.5	0.156±0.001	0.719±0.010	0.270±0.001	0.363±0.001	0.378±0.004	0.370±0.001
VICUNA-7B-SPT	0.168±0.018	0.768±0.036	0.227±0.015	0.281±0.088	0.350±0.061	0.230±0.018
OFA	0.130±0.019	0.314±0.059	0.064±0.086	0.052±0.049	0.340±0.026	0.101±0.071
GRAPHGPT-STD	0.126	0.701	—	—	—	—
GRAPHGPT-COT	0.181	0.521	—	—	—	—
LLAGA	0.168±0.032	0.793±0.036	0.199±0.007	0.146±0.067	0.276±0.069	0.352±0.033
TEA-GLM	<u>0.202</u>±0.014	**0.848**±0.010	<u>0.271</u>±0.010	<u>0.528</u>±0.058	<u>0.497</u>±0.027	<u>0.404</u>±0.010
GRAPHREACT	**0.273**±0.002	<u>0.819</u>±0.001	**0.294**±0.001	**0.645**±0.001	**0.523**±0.003	**0.483**±0.001

The best method is bolded and the runner-up is underlined.

5.1 EXPERIMENTAL SETUP

Datasets. We evaluate GRAPHREACT on eight text-attributed graph datasets from two domains: citation networks and e-commerce networks. For citation networks, we use Arxiv Hu et al. (2020), PubMed He et al. (2024), and an expanded version of Cora Wen & Fang (2023), where nodes represent papers and edges denote citation relationships. For e-commerce networks, we use Computer, Photo, Children, History, and Sports from the TAG benchmark Yan et al. (2023), where nodes represent products and edges indicate frequent co-viewing or co-purchasing relationships. Detailed statistics of all datasets are provided in Table 4, and further descriptions are provided in Appendix B.

Baselines. We compare GRAPHREACT with representative methods across five categories. (1) *Non-graph model*: MLP (Taud & Mas, 2017) serves as a structure-agnostic baseline that relies solely on node features without exploiting graph topology. (2) *Supervised graph methods*: GCN (Kipf & Welling, 2017), GRAPHSAGE (Hamilton et al., 2017), and GAT (Veličković et al., 2018a) perform supervised learning via message passing over graph structures, while NODEFORMER (Wu et al., 2022) and DIFFORMER (Wu et al., 2023) extend transformer architectures to graphs to capture long-range dependencies. (3) *Self-supervised graph methods*: DGI (Veličković et al., 2018b) learns node representations via contrastive objectives on unlabeled graphs, followed by a classifier for downstream prediction. (4) *Graph knowledge distillation*: GKD (Yang et al., 2022) transfers structural knowledge from a teacher GNN trained on a full graph to a student model operating under restricted structural access, while GLNN (Zhang et al., 2022) distills graph-aware representations into an MLP-like architecture to reduce reliance on graph connectivity during inference. (5) *Large language models*: VICUNA-7B-V1.5 (Chiang et al., 2023) and VICUNA-7B-SPT evaluate the capability of LLMs for graph tasks via textual prompting without explicit graph modeling. (6) *Graph–LLM methods*: OFA (Liu et al., 2024), GRAPHGPT (Tang et al., 2024a), LLAGA (Chen et al., 2024), and TEA-GLM (Wang et al., 2024a) integrate graph representations with LLMs to enable zero-shot or few-shot inference on graph tasks. Detailed descriptions of all baselines are provided in Appendix C, with implementation details in Appendix D.

Evaluation setting. We follow the cross-dataset zero-shot evaluation protocol of TEA-GLM (Wang et al., 2024a). Specifically, Arxiv and Computer are used as source datasets to conduct pre-training and downstream adaptation, and all methods are evaluated on unseen target datasets without further adaptation. For the citation domain, PubMed and Cora are used as target datasets; for the e-commerce domain, we evaluate on Children, History, Photo, and Sports. We adapt the same data splits as TEA-GLM, i.e., 90,941 nodes for training on Arxiv and 62,748 nodes for training on Computer, with 1,000 nodes used for evaluation on the other datasets. For all baseline methods, we directly report

Table 2: Ablation study of key componentss, reporting accuracy.

Methods	TR	SR	CF	Cora	Children	History	Sports
VARIANT 1	×	×	×	0.248±0.001	0.270±0.003	0.568±0.002	0.404±0.008
VARIANT 2	✓	×	×	0.274±0.003	0.290±0.002	0.603±0.006	0.445±0.001
VARIANT 3	×	✓	×	0.256±0.002	0.283±0.003	0.599±0.001	0.418±0.002
VARIANT 4	✓	✓	×	**0.285**±0.001	0.293±0.004	0.631±0.001	0.421±0.001
VARIANT 5	×	×	✓	0.255±0.006	0.272±0.003	0.578±0.005	0.410±0.003
GRAPHREACT	✓	✓	✓	0.273±0.002	**0.294**±0.001	**0.645**±0.001	**0.483**±0.001

TR: topological retrieval; SR: semantic retrieval; CF: context refinement.

the results reported in TEA-GLM. We report accuracy and Macro-F1 for node classification. Each experiment is conducted with five random seeds, and we report the mean and standard deviation.

5.2 PERFORMANCE EVALUATION

We make the following observations. First, GRAPHREACT consistently outperforms or remains highly competitive with all baselines across datasets, demonstrating the effectiveness of incorporating graph-aware reasoning and acting into zero-shot graph learning. Compared with conventional graph learning methods, the performance gains indicate that dynamically acquiring and refining graph-aware evidence provides informative and transferable context for cross-dataset prediction. Second, while LLM-based and graph–LLM methods, such as LLaGA and TEA-GLM, incorporate textual or graph representations into LLM inference, they typically rely on a fixed and static context, limiting their ability to adaptively exploit available evidence. In contrast, GRAPHREACT introduces a dynamic reasoning process that iteratively expands and refines the context through graph-aware retrieval and context refinement actions. This allows the model to progressively integrate topological and semantic evidence while filtering out irrelevant information, leading to more informative and task-relevant representations for prediction. On PubMed, GRAPHREACT achieves performance comparable to TEA-GLM. One possible reason is that PubMed contains only three coarse-grained classes, where semantically or topologically similar nodes (e.g., papers on related types of diabetes) may introduce ambiguous signals during context expansion, limiting the advantage of reasoning-based refinement.

Ablation study. We analyze the contribution of each component in Table 2. First, comparing Variant 1 and Variants 2–4 shows that both topological retrieval and semantic retrieval consistently improve performance over the no-retrieval baseline, demonstrating the effectiveness of graph-aware evidence acquisition. Among them, topological retrieval brings slightly larger gains than SR, indicating the importance of structural signals. Second, combining topological and semantic retrieval (Variant 4) further improves performance, suggesting that the two types of retrieval provide complementary information from structural and semantic perspectives. Third, context refinement alone (Variant 5) yields limited improvement, as it operates without additional external evidence. However, when combined with both topological and semantic retrieval, the full model achieves the best performance across all datasets, highlighting that refinement plays a crucial role in consolidating retrieved information and enhancing reasoning quality.

5.3 COMPARISON WITH TEXTUAL SEARCH ACTION

We further analyze whether the standard text-based **Search** action is useful for graph reasoning. Unlike our graph-aware actions, **Search** retrieves external evidence from a Wikipedia-based [1] knowledge source. We compare two variants in Table 3: SEARCH, which relies solely on external retrieval, and GRAPHREACT+SEARCH, which augments GRAPHREACT with this action. Specifically, given the current context, we first prompt the LLM to generate a query entity, then retrieve up to the top-6 relevant Wikipedia pages based on textual similarity, and finally concatenate the retrieved passages into the instruction as additional context for the next reasoning step.

As shown in Table 3, SEARCH alone performs substantially worse than graph-based methods, indicating that external textual evidence is insufficient for node classification in graph domains. Moreover, incorporating textual search into GRAPHREACT does not lead to consistent improvements,

[1] https://dumps.wikimedia.org/backup-index.html

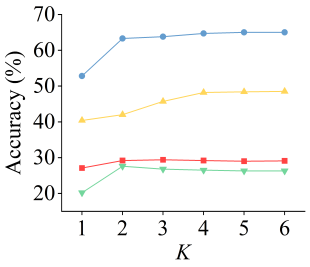


Figure 3: Impact of inference steps in GRAPHREACT.

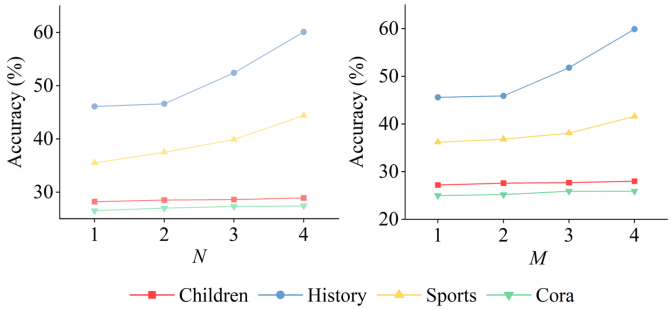


Figure 4: Impact of topological and semantic retrieval size.

and even slightly degrades performance compared to the full model. This suggests that external information may introduce noise or misaligned signals that interfere with graph-aware reasoning. We further analyze the effect of the number of retrieved entities in the search process in Appendix G. These results highlight that, unlike in NLP tasks, effective evidence for graph reasoning is primarily encoded within the graph itself, and directly applying text-based search actions is neither necessary nor beneficial.

5.4 PARAMETER SENSITIVITY ANALYSIS

We study the sensitivity of three key hyperparameters: the number of inference steps,the size of topological retrieved and semantic retrieval.

Effect of inference steps. We analyze the impact of the number of reasoning steps K in GRAPHREACT, as shown in Fig. 3. When $K = 1$, the model reduces to single-step inference without graph-aware retrieval or context refinement, which is equivalent to TEA-GLM. When $K = 2$, graph-aware retrieval is introduced in the first step, while context refinement is not yet involved. When $K \geq 3$, the full framework is enabled. We observe that increasing K from 1 to 2 yields significant improvement, highlighting the effectiveness of graph-aware retrieval. Further increasing K brings additional gains, indicating the benefit of iterative context refinement, although the improvement gradually saturates. In our experiments, we set $K = 4$.

Table 3: Analysis of the textual search action.

Methods	Cora	Children	History	Sports
SEARCH	0.253	0.225	0.535	0.427
GRAPHREACT+SEARCH	0.269	0.282	0.635	0.479
GRAPHREACT	**0.273**	**0.294**	**0.645**	**0.483**

Effect of retrieval size. We analyze the impact of the number of retrieved nodes, including the number of topological neighbors N and semantically similar nodes M, as shown in Fig. 4. We observe that increasing both N and M leads to consistent performance improvements, indicating the benefit of incorporating richer structural and semantic evidence. The gain from enlarging N is generally more pronounced, suggesting that structural context serves as the primary signal, while semantic retrieval provides complementary information. The improvement gradually saturates as N and M increase, implying that most useful information can be captured with a small number of retrieved nodes. Due to the input length limitation of the LLM, we set the maximum number of N and M to 4 in our experiments. In our experiment, we set $N = M = 4$.

6 CONCLUSION

In this paper, we proposed GRAPHREACT, a structured reasoning-acting framework for graph learning. We introduced graph-based retrieval that integrates topological and semantic evidence, along with a context refinement mechanism for multi-step reasoning. By combining retrieval and refinement within a step-by-step process, GRAPHREACT constructs informative contexts for prediction. Extensive experiments demonstrate strong performance under zero-shot settings, highlighting the effectiveness of structured reasoning with graph-aware evidence acquisition for graph inference.

AI USE STATEMENT

We used AI tools solely to polish the language and improve the readability of the manuscript. These tools were not used to develop research ideas, design or implement methods, conduct experiments, or analyze and interpret results. The authors take full responsibility for the final content of this paper.

REFERENCES

Alan Bundy and Lincoln Wallen. Breadth-first search. In *Catalogue of artificial intelligence tools*, pp. 13–13. 1984.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. *ICML*, 2024.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Diane J Cook and Lawrence B Holder. *Mining graph data*. John Wiley & Sons, 2006.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. volume 36, pp. 70757–70798, 2023.

Dayuan Fu, Jianzhao Huang, Siyuan Lu, Guanting Dong, Yejie Wang, Keqing He, and Weiran Xu. Preact: Prediction enhances agent's planning ability. In *ACL*, pp. 1–16, 2025.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *ICLR*, 2024.

Yufei He, Yuan Sui, Xiaoxin He, Yue Liu, Yifei Sun, and Bryan Hooi. Unigraph2: Learning a unified embedding space to bind multimodal graphs. In *WWW 2025*, pp. 1759–1770, 2025.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *ICLR*, 2024.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. GraphPrompt: Unifying pre-training and downstream tasks for graph neural networks. In *WWW*, pp. 417–428, 2023.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. *NeurIPS*, pp. 1362–1375, 2022.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? In *ICLR*, 2022.

Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory W Wornell, Subhro Das, David Daniel Cox, and Chuang Gan. Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search. In *ICML*, 2025.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *SIGIR*, pp. 491–500, 2024a.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. Higpt: Heterogeneous graph language model. In *KDD*, pp. 2842–2853, 2024b.

Hind Taud and Jean-Franccois Mas. Multilayer perceptron (mlp). In *Geomatic approaches for modeling land change scenarios*, pp. 451–455. 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018a.

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *ICLR*, 2018b.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *NeurIPS*, 37:5950–5973, 2024a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ICLR*, 2023.

Zehong Wang, Zheyuan Zhang, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. Gft: Graph foundation model with transferable tree vocabulary. *NeurIPS*, 37:107403–107443, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, volume 35, pp. 24824–24837, 2022.

Zhihao Wen and Yuan Fang. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *SIGIR*, 2023.

Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. *NeurIPS*, 35:27387–27401, 2022.

Qitian Wu, Chenxiao Yang, Wentao Zhao, Yixuan He, David Wipf, and Junchi Yan. DIFFormer: Scalable (graph) transformers induced by energy constrained diffusion. In *International Conference on Learning Representations*, 2023.

Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021.

Peipei Xia, Li Zhang, and Fanzhang Li. Learning similarity with cosine similarity ensemble. *Information sciences*, 307:39–52, 2015.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *NeurIPS*, 36:17238–17264, 2023.

Chenxiao Yang, Qitian Wu, and Junchi Yan. Geometric knowledge distillation: Topology compression for graph neural networks. *NeurIPS*, 35:29761–29775, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *ICLR*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *NeurIPS*, 36:11809–11822, 2023.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, volume 33, pp. 5812–5823, 2020.

Xingtong Yu, Zhenghao Liu, Yuan Fang, Zemin Liu, Sihong Chen, and Xinming Zhang. Generalized graph prompt: Toward a unification of pre-training and downstream tasks on graphs. *IEEE TKDE*, 36(11):6237– 6250, 2023.

Xingtong Yu, Chang Zhou, Zhongwei Kuai, Xinming Zhang, and Yuan Fang. GCoT: Chain-of-thought prompt learning for graphs. *arXiv preprint arXiv:2502.08092*, 2025.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S Yu. Graver: Generative graph vocabularies for robust graph foundation models fine-tuning. *NeurIPS*, 2025.

Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old mlps new tricks via distillation. In *ICLR*, 2022.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *ICLR*, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *ICLR*, 2023.

A ALOGRITHM

We summarize the overall procedure of GRAPHREACT in Algorithm 1. Lines 2–4 initialize the node representation by encoding the target node and projecting it into the LLM embedding space, followed by constructing the initial instruction. In lines 6–8, the model performs the first reasoning step and invokes graph-based retrieval to obtain both topological and semantic summaries, which are used to build the initial context. Then, lines 10–13 implement multi-step reasoning via context refinement, where at each step the LLM generates a new thought based on the current context, and the context is updated through a refinement action that distills and reorganizes the accumulated information. Finally, in lines 14–15, the model produces the prediction based on the refined context after K steps.

B FURTHER DESCRIPTIONS OF DATASETS

We summarize all datasets in Table 4 and provide further comprehensive descriptions of these datasets.

- **Arxiv** Hu et al. (2020) is a large-scale citation network built from Computer Science papers on the arXiv preprint server. The graph contains 169,343 paper nodes and 1,166,243 citation edges, with labels covering 40 arXiv CS sub-categories.
- **PubMed** He et al. (2024) consists of 19,717 diabetes-related publications connected by 44,338 citation links. Each node is labeled as one of three categories: experimentally induced diabetes, type 1 diabetes, or type 2 diabetes.
- **Cora** Wen & Fang (2023), formally known as the Cora Research Paper Classification Dataset, is an expanded citation graph for research paper classification. It includes 25,120 papers and 91,140 citation edges, where the 70 node labels correspond to fine-grained research categories.
- **Computer** Yan et al. (2023) is an e-commerce graph from the TAG benchmark, extracted from computer-related products in Amazon-Electronics. It has 87,229 product nodes, 721,081 co-viewing or co-purchasing edges, and 10 third-level product categories.
- **Photo** Yan et al. (2023) is constructed from photo-related products in Amazon-Electronics. The graph contains 48,362 products and 500,928 behavioral links, where an edge indicates that two products are frequently co-viewed or co-purchased. The prediction labels are 12 third-level product categories.
- **Children** Yan et al. (2023) is an Amazon-Books graph whose nodes are children's book products. It comprises 76,875 nodes and 1,554,578 co-viewing or co-purchasing edges, with 24 labels defined by third-level book categories.
- **History** Yan et al. (2023) is another Amazon-Books graph, focusing on history-related books. It contains 41,551 book nodes, 358,574 product-relation edges, and 12 third-level category labels.
- **Sports** Yan et al. (2023) is an Amazon-Sports graph built from fitness-related products. It is the largest e-commerce target dataset in our experiments, with 173,055 product nodes, 1,773,500 co-viewing or co-purchasing edges, and 13 fine-grained product categories.

C FURTHER DESCRIPTIONS OF BASELINES

In this section, we provide additional details about the baselines used in our experiments.

(1) Non-graph model.

- **MLP** (Taud & Mas, 2017): A multilayer perceptron that predicts node labels based only on node features. It does not explicitly model graph topology, and therefore serves as a structure-agnostic baseline for evaluating whether graph structural information is necessary.

(2) Supervised graph methods.

Algorithm 1 GRAPH-AWARE REASONING AND ACTING

Input: Graph $G = (\mathcal{V}, \mathcal{E})$, target node v, node text x^{text}, projection $\texttt{Proj}(\cdot;\phi)$, frozen LLM, steps K
Output: Prediction $\hat{y}$

Initialization:
$\mathbf{h}_v \leftarrow \texttt{GraphEncoder}(G, v)$
$\mathbf{h}_v^{\text{tok}} \leftarrow \texttt{Proj}(\mathbf{h}_v;\phi)$
Construct instruction $\mathcal{I}$ from $(x^{\text{text}}, \mathbf{h}_v^{\text{tok}})$
Graph-based retrieval:
$\text{Thought}^1 \leftarrow \texttt{LLM}(\mathcal{I}, \mathcal{Q})$
$\{s^{\text{top}}, s^{\text{sem}}\} \leftarrow \texttt{Act}_{\text{retrieve}}(G, v)$
$\mathcal{C}^1 \leftarrow \texttt{Init}(\text{Thought}^1,\ \{s^{\text{top}}, s^{\text{sem}}\})$
Context refinement:
for $k = 2$ to K **do**
    $\text{Thought}^{k+1} \leftarrow \texttt{LLM}(\mathcal{I},\ \mathcal{C}^k,\ \mathcal{Q})$
    $\text{Observation}^{k+1} \leftarrow \texttt{Act}_{\text{refine}}(\mathcal{C}^k,\ \text{Thought}^{k+1})$
    $\mathcal{C}^{k+1} \leftarrow \texttt{Update}(\mathcal{C}^k,\ \text{Thought}^{k+1},\ \text{Observation}^{k+1})$
Final prediction:
$\hat{y} \leftarrow \texttt{LLM}(\mathcal{I},\ \mathcal{C}^K,\ \mathcal{Q})$
return $\hat{y}$

- **GCN** (Kipf & Welling, 2017): A representative graph neural network that aggregates and transforms information from local neighborhoods through graph convolution. It captures structural information by recursively propagating node features over the graph.
- **GraphSAGE** (Hamilton et al., 2017): An inductive graph representation learning method that samples and aggregates neighborhood information. It learns node representations by combining target node features with sampled neighbor representations, making it suitable for large-scale graphs.
- **GAT** (Veličković et al., 2018a): A graph attention network that introduces attention weights into neighborhood aggregation. Instead of treating all neighbors equally, GAT assigns different importance scores to neighboring nodes, allowing the model to selectively aggregate more informative structural evidence.
- **NodeFormer** (Wu et al., 2022): A scalable graph Transformer for node classification. It models node interactions through an efficient attention mechanism and is designed to handle large-scale graph structure learning.
- **DIFFormer** (Wu et al., 2023): A scalable graph Transformer induced by energy-constrained diffusion. It captures long-range dependencies on graphs through a diffusion-based formulation, improving the ability to model global structural information.

(3) Self-supervised graph methods.

- **DGI** (Veličković et al., 2018b): A self-supervised graph representation learning method based on mutual information maximization. It learns node embeddings by contrasting local node representations with a global graph summary, without relying on task-specific labels during representation learning.

(4) Graph knowledge distillation.

- **GKD** (Yang et al., 2022): A graph knowledge distillation framework that transfers structural knowledge from a teacher GNN trained on complete graph information to a student model. It aims to preserve useful topological knowledge while reducing the dependency on full graph access.
- **GLNN** (Zhang et al., 2022): A graph-less neural network framework that distills graph-aware knowledge from GNNs into an MLP-like architecture. By transferring graph-aware predictions into a structure-free student model, GLNN reduces the reliance on graph connectivity during inference.

(5) Large language models.

Table 4: Summary of datasets.

Domain	Dataset	#Nodes	#Edges	#Classes
Citation	Arxiv	169,343	1,166,243	40
	Pubmed	19,717	44,338	3
	Cora	25,120	91,140	70
E-commerce	Ele-Computer	87,229	721,081	10
	Ele-Photo	48,362	500,928	12
	Book-Children	76,875	1,554,578	24
	Book-History	41,551	358,574	12
	Sports-Fitness	173,055	1,773,500	13

- **Vicuna-7B-v1.5** (Chiang et al., 2023): An instruction-tuned open-source large language model. In our experiments, Vicuna directly performs prediction based on textual node information, without explicitly using graph structure.
- **Vicuna-7B-SPT**: A soft prompt tuning variant of Vicuna-7B-v1.5. It introduces learnable soft prompts to adapt the LLM to graph-related prediction tasks while still relying mainly on textual information.

(6) Graph–LLM methods.

- **OFA** (Liu et al., 2024): A unified graph learning framework designed to handle different graph classification tasks with a shared model. It improves transferability across tasks by formulating graph learning problems under a unified task interface.
- **GraphGPT** (Tang et al., 2024a): A graph instruction tuning framework that aligns graph representations with large language models. We report two variants following the TEA-GLM setting: **GraphGPT-std**, which uses the standard graph instruction tuning pipeline, and **GraphGPT-cot**, which further incorporates Chain-of-Thought-style instruction data generated by LLMs.
- **LLaGA** (Chen et al., 2024): A graph–language assistant that connects graph structural information with LLMs for graph-related tasks. It uses graph-aware representations to support LLM-based prediction and reasoning.
- **TEA-GLM** (Wang et al., 2024a): A token embedding-aligned graph language model that aligns GNN representations with the token embedding space of a frozen LLM. It maps graph representations into a fixed number of graph token embeddings and inserts them into instruction templates for zero-shot graph learning. Since GRAPHREACT is built upon the TEA-GLM graph–LLM interface, TEA-GLM is used as the most direct backbone baseline for evaluating the effectiveness of our graph-aware reasoning-acting mechanism.

D IMPLEMENTATION DETAILS

Environment. All experiments are conducted on Ubuntu 22.04 with a 25 vCPU Intel(R) Xeon(R) Platinum 8481C and an vGPU-48GB with 48GB memory.

Optimizer. For all experiment, Adam is used as the optimizer.

Details of baselines. For fair and consistent comparison, we directly adopt the reported results of all non-TEA-GLM baseline methods from TEA-GLM Wang et al. (2024a), including MLP, GCN, GRAPHSAGE, GAT, DGI, GKD, GLNN, NODEFORMER, DIFFormer, VICUNA-7B-V1.5, VICUNA-7B-SPT, GRAPHGPT, LLAGA, OFA and TEA-GLM. These baselines are evaluated under the same dataset setting and evaluation metrics as TEA-GLM.

For TEA-GLM, we follow its recommended setting: raw node texts are encoded by a pretrained BERT model, GraphSAGE is used as the graph encoder, Vicuna-7B-v1.5 is used as the frozen LLM backbone, and the number of GNN layers is set to 2. The graph encoder is pretrained on the source

dataset, and the linear projector maps graph representations into graph token embeddings for zero-shot inference on unseen target datasets.

Implementation details of GRAPHREACT. We adopt a 3-layer GraphSAGE encoder as the graph backbone to obtain node representations, which are then mapped into the LLM embedding space via a linear projection layer. For graph-aware acting, we retrieve a fixed number of nodes from both structural and semantic perspectives. Specifically, we select $N = 4$ neighbors for topological retrieval using breadth-first traversal, and $M = 4$ nodes for semantic retrieval based on embedding similarity. For multi-step reasoning, we set the number of inference steps to $K = 4$. The prompt design used in our experiments is provided in Sect. H.

E SUPERVISED PERFORMANCE

Table 5: Accuracy and Macro-F1 on training datasets.

Model	Arxiv		Computer	
	Acc	F1	Acc	F1
MLP	0.546±0.004	0.295±0.007	0.420±0.006	0.267±0.005
GCN	0.545±0.005	0.317±0.006	0.424±0.012	0.386±0.014
GRAPHSAGE	0.556±0.006	0.315±0.008	0.534±0.037	0.347±0.036
GAT	0.561±0.003	0.339±0.005	0.609±0.035	<u>0.598</u>±0.039
NODEFORMER	0.544±0.016	0.297±0.029	0.434±0.012	0.288±0.012
DIFFORMER	0.616±0.025	0.356±0.024	0.629±0.012	0.467±0.022
DGI	0.342±0.024	0.336±0.011	0.594±0.004	0.452±0.008
GKD	0.393±0.085	0.164±0.029	0.351±0.031	0.155±0.016
GLNN	0.602±0.004	0.362±0.008	0.393±0.005	0.243±0.007
VICUNA-7B-V1.5	0.347±0.000	0.164±0.001	0.372±0.010	0.304±0.002
OFA	0.682±0.006	0.495±0.006	**0.753**±0.004	**0.687**±0.006
GRAPHGPT-STD	0.626	0.262	-	-
GRAPHGPT-COT	0.576	0.228	-	-
LLAGA	**0.749**±0.001	<u>0.575</u>±0.003	0.642±0.004	0.562±0.001
TEA-GLM	0.655±0.001	0.445±0.002	0.578±0.002	0.496±0.010
GRAPHREACT	<u>0.738</u>±0.011	**0.645**±0.012	<u>0.687</u>±0.007	0.513±0.015

We further evaluate GRAPHREACT under a supervised setting, where models are trained on the Arxiv and Computer datasets and evaluated on held-out test splits. The results are reported in Table 5. We make the following observations. First, GRAPHREACT achieves competitive performance compared with existing baselines, while not consistently outperforming state-of-the-art graph–LLM methods in this setting. This is expected, as supervised methods can directly optimize task-specific objectives with labeled data, whereas GRAPHREACT is primarily designed for reasoning-based inference rather than end-to-end supervised training. Second, GRAPHREACT demonstrates relatively stronger performance in terms of Macro-F1, particularly on Arxiv, indicating improved class-level balance. This suggests that the reasoning-acting mechanism, which dynamically aggregates and refines evidence, can help mitigate bias toward dominant classes even in supervised scenarios. Noote that the supervised setting is not the primary focus of this work. Instead, GRAPHREACT is designed to address the more challenging zero-shot graph learning scenario, where labeled data is unavailable and effective reasoning over structured context becomes critical. As demonstrated in Sect. 5.2, GRAPHREACT achieves significant improvements under zero-shot settings, highlighting the advantage of reasoning-acting synergy for cross-dataset generalization.

F MACRO-F1 RESULTS

We further report Macro-F1 scores for zero-shot node classification in Table 6. Overall, GRAPHREACT achieves competitive or superior performance on most datasets, particularly in the e-commerce

Table 6: Macro-F1 score of node classification.

Model	Citation		E-commerce			
	Cora	Pubmed	Children	History	Photo	Sports
MLP	0.009±0.004	0.246±0.042	0.007±0.007	0.023±0.008	0.041±0.023	0.019±0.005
GCN	0.007±0.001	0.187±0.021	0.006±0.004	0.024±0.013	0.034±0.007	0.017±0.009
GRAPHSAGE	0.007±0.003	0.257±0.084	0.005±0.003	0.029±0.024	0.020±0.011	0.021±0.004
GAT	0.006±0.001	0.259±0.065	0.063±0.067	0.159±0.117	0.036±0.035	0.091±0.090
NODEFORMER	0.008±0.003	0.232±0.089	0.019±0.008	0.046±0.031	0.055±0.006	0.049±0.009
DIFFORMER	0.007±0.002	0.187±0.007	0.002±0.002	0.050±0.019	0.069±0.010	0.045±0.007
DGI	0.004±0.002	0.213±0.127	0.012±0.004	0.038±0.015	0.045±0.015	0.018±0.005
GKD	0.004±0.001	0.247±0.039	0.028±0.003	0.060±0.008	0.049±0.015	0.050±0.008
GLNN	0.006±0.001	0.221±0.033	0.021±0.003	0.064±0.007	0.057±0.002	0.052±0.003
VICUNA-7B-V1.5	0.109±0.002	0.629±0.024	**0.279**±0.002	0.349±0.003	0.383±0.001	0.410±0.002
OFA	0.091±0.013	0.287±0.059	0.017±0.010	0.026±0.007	0.103±0.007	0.043±0.021
GRAPHGPT-STD	0.082	0.649	—	—	—	—
GRAPHGPT-COT	**0.127**	0.482	—	—	—	—
LLAGA	0.108±0.014	0.778±0.056	0.163±0.029	0.144±0.025	0.362±0.039	0.446±0.035
TEA-GLM	0.107±0.012	**0.846**±0.011	0.209±0.010	0.336±0.021	0.404±0.017	0.396±0.007
GRAPHREACT	0.116±0.016	0.759±0.017	0.216±0.008	**0.351**±0.022	**0.409**±0.011	**0.464**±0.009

The best method is bolded and the runner-up is underlined.

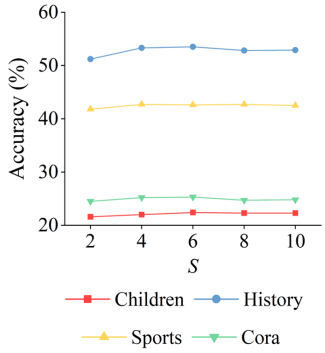


Figure 5: Numbers of Wiki entities for Search.

domain, where it consistently outperforms existing graph–LLM methods. Compared with accuracy, the improvements in Macro-F1 are more pronounced, indicating that GRAPHREACT provides better class-level balance under zero-shot settings. This can be attributed to the reasoning-acting mechanism, which dynamically aggregates and refines evidence from both structural and semantic sources, thereby reducing bias toward dominant classes. On citation datasets, GRAPHREACT remains competitive with strong baselines such as TEA-GLM, showing that the proposed framework maintains robust performance across different domains.

G EFFECT OF THE NUMBER OF RETRIEVED ENTITIES IN TEXTUAL SEARCH

We further analyze the impact of the number of retrieved Wikipedia entities S in the text-based Search action, as shown in Fig. 5. We observe that increasing S brings only marginal improvements across all datasets. The performance slightly improves when increasing S from 2 to 4, indicating that a small amount of external evidence can provide limited complementary information. However, further increasing S does not lead to consistent gains and quickly reaches saturation, with performance remaining largely stable thereafter. This suggests that simply retrieving more external textual entities does not substantially enhance graph reasoning, and may introduce redundant or noisy information. The results further confirm that external search plays a limited role compared to graph-aware evidence.

H PROMPT DESIGN

Table 7: Prompt design for node classification on children.

Type	Prompts
Task Instruction	Given a representation of a book: <Token 1>, with the following information: Description: {description} Title: {title} Question: Which category does this book belong to? Please directly give the most likely answer from the following categories: {categories}
Thought Generation	USER: The task is to classify a children's book by its main topic and content type. Your goal is to identify what kind of story or subject the book presents to young readers (for example, adventure, humor, fantasy, education, or everyday life). Based on the book information above, generate one concise sentence as your current thought to help solve the task. ASSISTANT:
Topological retrieval	USER: You are given a subgraph consisting of a target node and its neighboring nodes from a children's book e-commerce graph. Each neighbor is described by a title and a description. Your task is to infer the most likely category of the subgraph based ONLY on the textual information of the neighbors, and generate a one-sentence summary in the following fixed format: "This subgraph xxx, so the category might be xxx." IMPORTANT RULES (must be followed strictly): 1. Text-based inference: Analyze the titles and descriptions of all neighbors as a whole. Identify the dominant themes, genres, or target age group, such as fantasy, animals, educational concepts, activities, or everyday life experiences. Give more weight to recurring elements shared across neighbors, like magical creatures, vehicle types, specific emotions, or learning topics. 2. Noise filtering: Ignore non-informative promotional content, including but not limited to: bestseller status, author biography details (e.g., "has written over one hundred books"), series accolades, review quotes, and generic publisher blurbs. Focus only on information that reflects the book's core story, subject matter, characters, or intended educational/entertainment purpose for children. 3. Category constraint: The final category MUST be chosen from the following list and cannot be invented or modified: {categories} 4. Output constraint: Output exactly ONE sentence. Use the fixed format exactly as specified. Do not add explanations, bullet points, or extra text. The phrase "This subgraph xxx" should concisely describe the overall thematic focus of the children's books in the subgraph. Here are correct examples: - "This subgraph features stories about fairies and magical adventures, so the category might be Fairy Tales, Folk Tales & Myths." - "This subgraph contains books about trucks, trains, and airplanes, so the category might be Cars, Trains & Things That Go." - "This subgraph explores concepts like feelings, friendship, and school life, so the category might be Growing Up & Facts of Life." Here is the neighbor information: {nodes} Please give your reply:

Type	Prompts
Semantic retrieval	USER: You are given a center node and some similar nodes, which are based on similar semantic node embeddings, from a children's book e-commerce graph. Each node is described by a title and a description. Your task is to infer the most likely category of the node set based ONLY on the textual information, and generate a one-sentence summary in the following fixed format: "This node set xxx, so the category might be xxx." IMPORTANT RULES (must be followed strictly): 1. Text-based inference: Analyze the titles and descriptions of all nodes as a whole. Identify the dominant themes, genres, or target age group, such as fantasy, animals, educational concepts, activities, or everyday life experiences. Give more weight to recurring elements shared across the node set, like magical creatures, vehicle types, specific emotions, or learning topics. 2. Noise filtering: Ignore non-informative promotional content, including but not limited to: bestseller status, author biography details (e.g., "has written over one hundred books"), series accolades, review quotes, and generic publisher blurbs. Focus only on information that reflects the book's core story, subject matter, characters, or intended educational/entertainment purpose for children. 3. Category constraint: The final category MUST be chosen from the following list and cannot be invented or modified: {categories} 4. Output constraint: Output exactly ONE sentence. Use the fixed format exactly as specified. Do not add explanations, bullet points, or extra text. The phrase "This node set xxx" should concisely describe the overall thematic focus of the children's books in the set. Here are correct examples: - "This node set features stories about fairies and magical adventures, so the category might be Fairy Tales, Folk Tales & Myths." - "This node set contains books about trucks, trains, and airplanes, so the category might be Cars, Trains & Things That Go." - "This node set explores concepts like feelings, friendship, and school life, so the category might be Growing Up & Facts of Life." Here is the node information: {nodes} Please give your reply:
Context refinement	USER: Please summarize the key points from our discussion so far in one sentence. Focus on the most important information that will help answer the original question. ASSISTANT:
Textual Search	USER: To better understand the children's book above, select ONE specific story theme, character type, literary genre, or educational concept that is central to the book and can be searched directly on Wikipedia. Avoid overly generic terms like 'story' or 'children'. Output only a short phrase (maximum 5 words). Do NOT explain. ASSISTANT:
Full prompt example	A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers step by step to the user's questions. USER: Given a representation of a book: <Token 1> <Token 2> <Token 3> <Token 4> <Token 5>, with the following information: Description: Daisy Meadows has written over one hundred books for children. Her RAINBOW MAGIC series is a New York Times bestseller! Title: Amber: The Orange Fairy (Rainbow Magic: The Rainbow Fairies, No. 2). Thought: The book is a fantasy story featuring a fairy character (Amber the Orange Fairy) as part of a magical rainbow-themed series aimed at young children.

Type	Prompts
	In addition to the information above, the book is part of an e-commerce graph, where neighboring nodes represent children's books that are often related or similar. As a result, adjacent books may share the same or closely related categories. The following summary describes the content and category tendencies of the neighboring books, and can be used as contextual evidence to help answer the question. Neighbor summary: This subgraph features books written by Daisy Meadows, with a focus on the RAINBOW MAGIC series, which is a New York Times bestseller. The category might be Literature & Fiction or Children's Books. Here is another summary generated from nodes that are similar to the center node. Node summary: This node set features humorous stories and books about everyday life experiences, so the category might be Humor. Thought: The book might belong in the Children's Books category with a focus on fantasy and humorous stories. Summary: The book "Amber: The Orange Fairy" by Daisy Meadows is a part of the RAINBOW MAGIC series, which is a New York Times bestseller. It falls under the category of Children's Books with a focus on fantasy and humorous stories. Question: Which category does this book belong to? Please directly give the most likely answer from the following categories: "Literature & Fiction", "Animals", "Growing Up & Facts of Life", "Humor", "Cars, Trains & Things That Go", "Fairy Tales, Folk Tales & Myths", "Activities, Crafts & Games", "Science Fiction & Fantasy", "Classics", "Mysteries & Detectives", "Action & Adventure", "Geography & Cultures", "Education & Reference", "Arts, Music & Photography", "Holidays & Celebrations", "Science, Nature & How It Works", "Early Learning", "Biographies", "History", "Children's Cookbooks", "Religions", "Sports & Outdoors", "Comics & Graphic Novels", "Computers & Technology". ASSISTANT:

In this section, we illustrate the prompt design used in our framework in Table 7, taking the Children dataset as a representative example. For each type of prompt, we describe its role in the reasoning-acting process and provide a concrete instance. For other datasets, the prompt structure remains the same, while dataset-specific descriptions are adjusted accordingly.

I LIMITATIONS

Despite its effectiveness, GRAPHREACT has several limitations. First, the framework introduces additional computational overhead due to multi-step LLM inference. Second, its performance may depend on the availability and quality of node-associated text. Finally, the current design is constrained by the input length of the LLM, which limits the number of reasoning steps and retrieved evidence. We leave improving efficiency and scalability for future work.

J BROADER IMPACT

This work introduces a graph-aware reasoning framework that integrates large language models with structured data, which may benefit a wide range of applications involving relational reasoning, such as recommendation systems, knowledge discovery, and scientific data analysis. By enabling zero-shot generalization across graphs, our approach may reduce the need for extensive labeled data and improve the accessibility of graph learning techniques in real-world scenarios. Potential risks are limited but may arise if the method is applied to sensitive domains, where incorrect predictions or biases inherited from pre-trained language models could propagate through the reasoning process. However, our framework does not introduce new data sources or amplify such biases beyond those already present in existing models. Overall, we believe this work has a positive impact by advancing

the integration of structured and unstructured knowledge for more flexible and generalizable machine learning systems.