Rethinking Message Passing as Retrieval for Text-Attributed Graph Learning

Jintang Li, Yuhong Chen, Ruofan Wu, Binli Luo, Jiayi Ji, Hui Li, and Rongrong Ji

Abstract—Graph neural networks (GNNs) are typically conceptualized as message-passing neural networks, yet it remains unclear why neighborhood aggregation reliably outperforms node-wise multilayer perceptrons (MLPs). Despite its empirical success, this paradigm can be computationally expensive and sensitive to imperfect graph structures. In this work, we present a retrieval-augmented view of GNNs: each layer makes predictions by applying an MLP to a node representation together with a permutation-invariant summary of retrieved graph context. Motivated by this perspective, we propose RTA, a simple MLP-based framework that replaces structural message passing with label-aware retrieval and propagation. We provide theoretical insights that (i) connect retrieval-based aggregation to softmax-attention message passing, and (ii) establish the robustness of retrieved-context supervision to mis-retrieved outliers. Experiments on multiple text-attributed graph benchmarks show that RTA matches or even outperforms strong GNN and graph LLM baselines while improving efficiency and robustness across diverse scenarios.

Index Terms—Graph Neural Networks, Retrieval-Augmented Generation, Message Passing Neural Networks

1 Introduction

Graph neural networks (GNNs) have attracted substantial attention due to their strong empirical performance on graph-structured data [1]–[4], with applications ranging from social networks [5] and molecular chemistry [6] to recommendation systems [7]. The dominant paradigm behind modern GNNs is *message passing* [8]: nodes iteratively exchange information along edges and update their representations by aggregating neighbor messages through permutation-invariant operators (e.g., sum or mean). While this mechanism is effective in many settings, it also introduces well-known challenges, including limited scalability, sensitivity to graph structure, degraded performance on heterophilous graphs [9], and representation collapse due to oversmoothing and oversquashing in deep propagation [10]. More fundamentally, although message passing has become the dominant paradigm for GNNs, it remains unclear whether its effectiveness stems from graph-based propagation itself or, more generally, from conditioning node predictions on informative contexts.

In parallel, retrieval-augmented generation (RAG) has emerged as a powerful paradigm in natural language understanding [11]. RAG [12] follows a *retrieve-then-reason* procedure: given a query, it retrieves a small set of relevant contexts and conditions a parametric model (e.g., an LLM) on the retrieved evidence to produce an output. This mechanism reduces reliance on parametric memory and improves robustness by grounding predictions in external information.

- *Jintang Li, Yuhong Chen, Jiayi Ji, Hui Li, Rongrong Ji are with the Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, Xiamen 361005, China (E-mail: edisonlee@xmu.edu.cn, yuhongchen@stu.xmu.edu.cn, jjyxmu@gmail.com, hui@xmu.edu.cn, rrji@xmu.edu.cn).*
- *Ruofan Wu is with the School of Statistics of Fudan University, Shanghai 200433, China (E-mail: wuruofan1989@gmail.com).*
- *Binli Luo is with the School of Computer Science and Engineering of Central South University, Changsha 410000, China (E-mail: lblhandsome@gmail.com).*

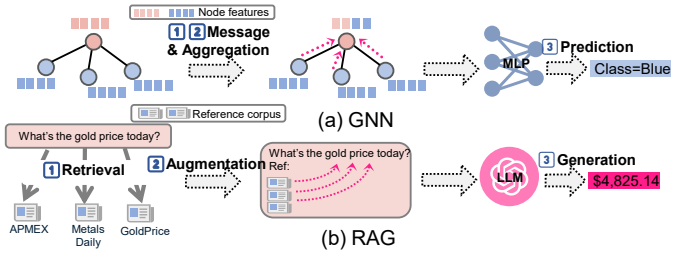

Fig. 1. Technical comparison between GNN and RAG. Both leverage contextual information beyond the raw input.

Despite their different application domains, GNNs and RAG share a common computational pattern: both condition predictions on *contextual information* beyond the raw input (see Figure 1). In a message-passing layer, the context is defined by the node's neighborhood, whereas in RAG it consists of a set of retrieved documents or passages. However, this shared computational structure has not been explicitly characterized from a unified perspective, leaving unclear how context selection, aggregation, and prediction in message passing relate to retrieval-augmented inference.

In this work, we develop a retrieval-augmented view of a broad family of message-passing GNNs:

A message-passing layer can be interpreted as an MLP operating on a node and a retrieved context set.

Concretely, we show that the message passing step of GNNs can be cast as a retrieval and augmentation module, and that different GNN variants correspond to different choices of *retriever* (which contexts are selected) and *set function* (how contexts are aggregated). This perspective clarifies why conditioning on neighbors can generally outperform node-independent models: the model is effectively performing prediction with retrieved evidence, where the evidence is provided by graph-induced or learned retrieval sets.

Building on this connection, we propose **RTA** (**R**etrieve-

Then-Aggregate), a simple retrieval-augmented framework for text-attributed graph representation learning that avoids explicit multi-hop message passing. Crucially, RTA supports label-aware retrieval, where labeled nodes provide necessary supervision signals that guide both retrieval and aggregation without information leakage. RTA proceeds in three stages: (i) a label-aware contextual encoder produces node embeddings by augmenting each node's text with a small set of structural neighbor texts; (ii) a top-k retriever in the structure and embedding space constructs a *retrieval graph* that can connect semantically related nodes beyond local edges; and (iii) an MLP-style update aggregates retrieved neighbors to produce task-specific representations. By replacing recursive message passing on the original graph with localized retrieval-driven interactions, RTA achieves better performance while improving scalability and robustness across diverse settings.

Theoretically, we establish two results that characterize retrieval-based message passing and directly align with our method design. First, we show that top-k retrieval followed by aggregation provides a sparse approximation of softmax-attention message passing, where the approximation error is controlled by the retrieval margin and the score distribution within the retrieved context set. Second, we prove that label-aware retrieved-context supervision reduces the gradient influence of mis-retrieved outliers, providing a theoretical explanation for the robustness of retrieval-based aggregation under noisy contexts. Overall, both results justify retrieval as a principled mechanism for constructing task-relevant context sets and provide theoretical support for label-guided retrieval and aggregation.

The main contributions of our work are as follows:

- **Formulation.** We reveal an intrinsic duality between message-passing aggregation and retrieval-augmented MLPs, offering a unified view of GNN layers as prediction over retrieved contexts.
- **Framework.** Leveraging this connection, we propose RTA, a scalable retrieval-centric alternative to message passing that integrates label-aware contextual encoding, embedding and structural space retrieval, and label-enhanced aggregation.
- **Theoretical results.** We establish theoretical connections between retrieval-based aggregation and message passing, and prove optimization robustness under retrieved-set supervision. We present theoretical guarantees for (i) the equivalence of retrieval-based message passing and (ii) optimization robustness under retrieved-set supervision.
- **Empirical results.** We conduct extensive experiments on multiple text-attributed graph benchmarks, showing that RTA matches or outperforms strong GNN baselines with improved efficiency and robustness.

Overall, our work takes a step toward demystifying message passing by recasting it as retrieval over contextual evidence, and it suggests a practical design space in which graph representation learning can be carried out by retrieval-defined computation graphs rather than fixed-edge diffusion.

2 Related work

Our work connects three lines of research: message-passing graph neural networks, learning over text-attributed graphs with language models, and retrieval-augmented generation. While these areas have been extensively studied in isolation, their underlying computational connection remains underexplored. We review them below and highlight how our work differs by interpreting graph representation learning itself as a retrieval-augmented prediction process.

2.1 Graph neural networks

GNNs are a flexible class of models for learning representations over graph-structured data, which represent a significant leap forward in the ability to model and analyze complex relationships within data. By iteratively aggregating and transforming information from a node's local neighborhood, GNNs learn powerful, context-aware representations that are crucial for making accurate predictions and uncovering hidden patterns within the graphs [1]–[3]. Beyond conventional local message passing, graph Transformers have emerged as an alternative paradigm for capturing long-range dependencies through attention-based interactions. Representative methods include NodeFormer [13], which develops a scalable all-pair attention mechanism for graph representation learning, and DGT [14], which dynamically selects structurally and semantically relevant nodes to enable efficient sparse attention. The research topic of GNNs is vibrant and rapidly evolving, encompassing a wide range of theoretical advancements and algorithmic innovations [4], [15]–[17].

2.2 Learning over text-attributed graphs

In real-world graph tasks, nodes often have textual attributes that convey richer information, resulting in text-attributed graphs (TAGs) [18]. TAG provides a scenario for applying LLMs on graphs that store information in natural language. Combining GNNs with LLMs can leverage the robust textual understanding of LLMs while harnessing GNNs' ability to capture structural relationships. Specifically, there are several lines of research according to the role of LLMs on graphs, including LLM as Predictor [19]–[21], LLM as Aligner [22]–[24] and LLM as Enhancer [18], [25], [26]. While adapting LLMs to graphs has emerged as a research focus, little effort has been dedicated to uncovering the correlations between GNNs and language modeling. A closely related work is AuGLM, which treats retrieval as input augmentation for an LLM classifier, whereas RTA studies retrieval as the context-selection primitive underlying graph message passing.

2.3 Retrieval-augmented generation

Retrieval-augmented generation [12] is an established approach to enhance the generation quality of LLMs by leveraging external knowledge bases. RAG achieves good performance on various tasks such as language modeling [27] and question-answering [28], without additionally fine-tuning LLMs. To improve generation, several studies have been dedicated to enhancing the capability of RAGs with correction [29], verification [30], or critique [31]; see [11]

for a comprehensive survey on RAG techniques. Recently, researchers and practitioners have started to explore the role of graphs in facilitating RAG systems, with the most promising works being GraphRAG [32] and its variants [33]–[35]. GraphRAG-based approaches leverage knowledge graphs as a structured source of context and factual grounding for LLMs. By supplying entity-centric information such as textual descriptions, they provide richer, more organized evidence, which can encourage deeper, relation-aware reasoning by the LLM. Note that GraphRAG-based approaches and our work represent two orthogonal lines of research. GraphRAG-based approaches enhance LLM reasoning by incorporating graph-structured knowledge into RAG. In contrast, RTA advances graph representation learning by leveraging recent advances in RAG.

3 Preliminary

We first introduce the notations and background needed to connect message-passing GNNs with retrieval-augmented generation. The goal of this section is not to review all variants, but to expose the shared computational pattern behind the two paradigms.

3.1 Notations

With graph-structured inputs, we typically assume a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ specifies pairs of nodes in $\mathcal{V}$ that are connected. To unleash the power of LLMs on graphs, we consider a more challenging scenario, i.e., text-attributed graphs, which provide rich semantic information for the representation learning. In the context of text-attributed graphs, we are given a set of textual features $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$, where each t_u is the textual corpus associated with node $u \in \mathcal{V}$. In such scenarios, the text set $\mathcal{T}$ is encoded as the node feature matrix $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

3.2 Message-passing graph neural networks

Given a graph $\mathcal{G}$ and d-dimensional node features $\mathbf{X}$, message-passing GNNs aim to obtain informative node representations capturing both structural properties and connections between node features.

$$\mathbf{x}'_u = \phi\left(\mathbf{x}_u, \bigoplus\left(\{\mathbf{x}_v : v \in \mathcal{N}_u\}\right)\right), \ \mathcal{N}_u = \{v : (u, v) \in \mathcal{E}\}, \tag{1}$$

with the core components including a permutation invariant aggregation function $\bigoplus$ (e.g., mean or sum) and a combine or update function ϕ; $\mathcal{N}_u \subseteq \mathcal{V}$ is a set of nodes adjacent to node u. Specifically, node u aggregates the messages along incoming edges using the aggregation function $\bigoplus$. Then, the update function ϕ updates the representations of the receiver node u based on its current representation ($\mathbf{x}_u$) and the aggregated messages. The update function ϕ allows the node to learn how to integrate its own features with the aggregated context effectively. Typically, ϕ is an MLP network and the design of the function ϕ is what mostly distinguishes one type of GNN from the other. For example, GraphSAGE [3] implements GNN as:

$$\mathbf{x}'_u = \mathrm{MLP}\left(\mathbf{x}_u, \mathrm{MEAN}\left(\{\mathbf{x}_v : v \in \mathcal{N}_u\}\right)\right). \tag{2}$$

In recent years, GNNs have been extended with several advanced variants that incorporate additional components, such as attention mechanisms [2], advanced aggregation functions [16], and residual connections [36].

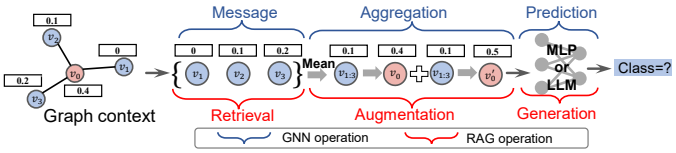

Fig. 2. Illustrative example on how GNNs are decomposed into RAG-like architectures.

3.3 Retrieval-augmented generation

RAG is a natural language querying approach for enhancing off-the-shelf LLMs with external knowledge [12]. In an RAG application, we are given a collection of documents or contexts (e.g. Wikipedia), which provides the grounded knowledge and serves as the external knowledge base. Given a user query $\mathbf{q}$ and a corpus $\mathbf{C}$, a dense embedding based retriever [37] first retrieves top-k contexts $\mathbf{C_k} = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ that are most relevant to the query $\mathbf{q}$. Here $\mathcal{R}_k \subseteq \mathbf{C}$ is a set of retrieved contexts from $\mathbf{C}$. Then, LLM reads the top-k contexts along with the query $\mathbf{q}$ to generate the answer $\mathbf{r}$. Formally, a standard RAG-based system can be formulated as follows:

$$\begin{aligned} \mathbf{r} &= \mathrm{Reason}\left(\mathbf{q}, \mathcal{R}_k\right), \\ \mathcal{R}_k &= \{\mathbf{c}_1, \cdots, \mathbf{c}_K\} = \mathrm{Retrieval}(\mathbf{q} \mid \mathbf{C}). \end{aligned} \tag{3}$$

In practice, advanced RAG may incorporate chunking and reranking operations, which segment the contexts into chunks for improved retrieval and rerank the retrieved results for enhanced generation [38]. In this work, we consider a classical RAG system that comprises only a retriever and a generator, for simplicity.

4 Present Work: RTA

In this section, we first present our perspective on how GNNs relate to the RAG-based framework. Next, we introduce RTA, an RAG-inspired graph learning framework designed to learn representations on text-attributed graphs using a pure RAG scheme. RTA leverages valuable insights from GNNs and label propagation [39], a label-specific variant of message-passing GNNs, to efficiently and effectively capture graph context.

4.1 Connections between GNN and RAG

4.1.1 *Unified view*

GNNs and RAG are important tools for graph structure learning and natural language understanding, respectively. Both approaches leverage similar principles by incorporating *additional context* beyond the raw input to enhance comprehension. In the case of GNNs, the additional context is derived from aggregating information from neighboring nodes and the graph structure itself. For RAGs, context is incorporated by retrieving relevant knowledge from external sources to supplement the input. This shared principle motivates us to align them within a unified formulation.

In this regard, we formulate GNNs and RAG that follow similar rules as below:

$$\mathbf{x}'_u = \underbrace{\underbrace{\phi\Big(\mathbf{x}_u, \underbrace{\bigoplus(\{\mathbf{x}_v : v \in \mathcal{N}_u\})}_{\text{Aggregation}}\Big)}_{\text{Prediction}}}, \underbrace{\mathcal{N}_u = \{v : (u,v) \in \mathcal{E}\}}_{\text{Message}}, \tag{4}$$

$$\mathbf{r} = \underbrace{\underbrace{\phi\Big(\mathbf{q}, \underbrace{\bigoplus(\{\mathbf{c}_j : \mathbf{c}_j \in \mathcal{R}_k\})}_{\text{Augmentation}}\Big)}_{\text{Generation}}}, \underbrace{\mathcal{R}_k(\mathbf{q}) = \text{Top-}k_{\mathbf{c}\in\mathcal{R}}\ \langle \mathbf{q}, \mathbf{c}\rangle}_{\text{Retrieval}}, \tag{5}$$

where the cosine similarity is defined as $\langle x, y\rangle = \frac{x\cdot y}{\|x\|\|y\|}$. Here, we omit the text embedding step for the sake of simplicity. In RAG, $\mathcal{R}_k(\mathbf{q})$ denotes the top-k retrieved corpus from $\mathcal{R}$ that are most relevant to the query $\mathbf{q}$ according to a dense retriever ψ, and $\bigoplus$ is a combination operator over the retrieved contexts, such as concatenation. We overload notations in Eq. (5) for simplicity and omit prompt details.

4.1.2 Label propagation as label-centric RAG

This unified view suggests that the "context" set in Eq. (4)–(5) need not be restricted to feature vectors or text tokens; it can also contain *label evidence* from a labeled set (e.g., human supervision). In particular, classical label propagation (LP) [39] can be interpreted as a label-specific message-passing procedure: each node *collects* (retrieves) label signals from its neighbors and *mixes* (augments) them through a propagation operator. Under the unified view above, LP corresponds to a special case where the retrieval set is the graph neighborhood $\mathcal{N}_u$, the retrieved "contexts" are label message vectors, and the "generation" step reduces to a simple readout on the aggregated label context:

$$\mathbf{p}'_u = \underbrace{\underbrace{\phi\Big(\mathbf{p}_u, \underbrace{\bigoplus\big(\{\mathbf{p}_v : v \in \mathcal{N}_u\}\big)}_{\text{Propagation}}\Big)}_{\text{Inference}}}, \underbrace{\mathcal{N}_u = \{v : (u,v) \in \mathcal{E}\}}_{\text{Message / Retrieval}},$$
$$\mathbf{p}_v = \mathbb{I}[v \in \mathcal{T}]\,\omega(\tilde{y}_v) + \mathbb{I}[v \notin \mathcal{T}]\,\mathbf{0},\ \hat{y}_u = \arg\max_{c\in[C]} [\mathbf{p}'_u]_c, \tag{6}$$

where $\mathcal{T}$ denotes the labeled nodes (training set), $\omega(\tilde{y}_v)$ can be a one-hot label embedding, and unlabeled nodes contribute a padding message (e.g., $\mathbf{0}$). This perspective provides a principled motivation for *label-aware RAG on graphs*: when partial labels are available, they can be treated as additional retrieved context to guide both aggregation and retrieval.

4.1.3 Differences and design implications

The unified formulation in Eqs. (4), (5) and (6) highlights a shared *retrieve-and-aggregate* skeleton, but the three paradigms differ in how the context set is formed and how it is fused. (i) **context selection**: in GNNs/LP, the graph structure fixes the retrieved set $\mathcal{N}_u$, yielding query-dependent and degree-varying neighborhoods; in RAG, a learned retriever returns a fixed-size top-k set $\mathcal{R}_k(\mathbf{q})$, which effectively induces a content-dependent similarity graph with controlled out-degree. (ii) **context fusion**: RAG typically augments inputs via prompt construction (e.g., concatenating retrieved texts), whereas GNNs employ explicit permutation-invariant set operators (mean/sum/max/attention) over vector states; LP further exposes a label-only extreme where the fused context consists purely of propagated supervision signals. These differences suggest a natural hybrid for graphs: learn the context set via embedding and structural space top-k retrieval while retaining efficient set aggregation, and treat available labels as additional retrieved context. We next instantiate this idea by introducing label embeddings into both contextual encoding and retrieved-neighbor aggregation.

4.2 RTA: RAG-based graph learning framework

By drawing parallels between GNN and RAG, we now explore how integrating retrieval-augmented mechanisms into a simple MLP network can efficiently reshape graph representation learning. The overall framework of RTA is shown in Figure 3. Different from conventional message-passing GNNs that rely on static edges, RTA retrieves and aggregates semantically relevant contexts from the embedding and structural space, enabling adaptive graph construction and scalable structure learning.

4.2.1 Label-aware retrieval-augmented node embedding

Existing learning paradigms on TAGs typically require a text encoder, such as BERT [40] or DeBERTa [41], to obtain node feature embeddings from associated text attributes. For example, a language model $\text{LLM}(\cdot)$ processes the text attribute t_u to generate $\mathbf{x}_u = \text{LLM}(t_u)$. However, this operation treats each node in isolation and ignores the rich contextual information residing in neighboring textual attributes, leading to less informative embeddings.

Inspired by the retrieve-then-reason paradigm in RAG, we enrich the textual representation of each node by augmenting it with contextual text retrieved from its structural neighbors. Since the graph itself provides a localized retrieval source, we can bypass explicit document retrieval and instead treat the neighbor texts as intrinsic contexts:

$$\mathbf{x}_u = \text{Enc}_\theta\left(t_u, \bigoplus\left(\{t_v : v \in \mathcal{N}_u^{(k_1)}\}\right)\right) \in \mathbb{R}^d, \tag{7}$$

where $\mathcal{N}_u^{(k_1)}$ denotes the k_1 sampled neighbors of node u, and $\bigoplus$ represents a permutation-invariant combination such as prompt concatenation for the neighbor contexts. Generally, Enc_θ is implemented as an LLM backbone for semantic textual embedding. To further enhance retrieval quality, we incorporate partial label signals into both the embedding and aggregation stages. When some node labels are available, we may inject them as additional context, which act as supervision-aware anchors that guide the retrieval process toward semantically and label-consistent neighborhoods:

$$\mathbf{x}_u = \text{Enc}_\theta\Big(t_u, \bigoplus\left(\{(t_v, y_v) : v \in \mathcal{N}_u^{(k_1)}\}\right)\Big), \tag{8}$$

where y_v denotes the label of node v. The introduction of label information is motivated by the label propagation [39] algorithm to bring supervision to the message-passing (retrieval) step. This retrieval-augmented encoding stage allows the LLM to generate context-aware node embeddings that better reflect both semantic and structural correlations.

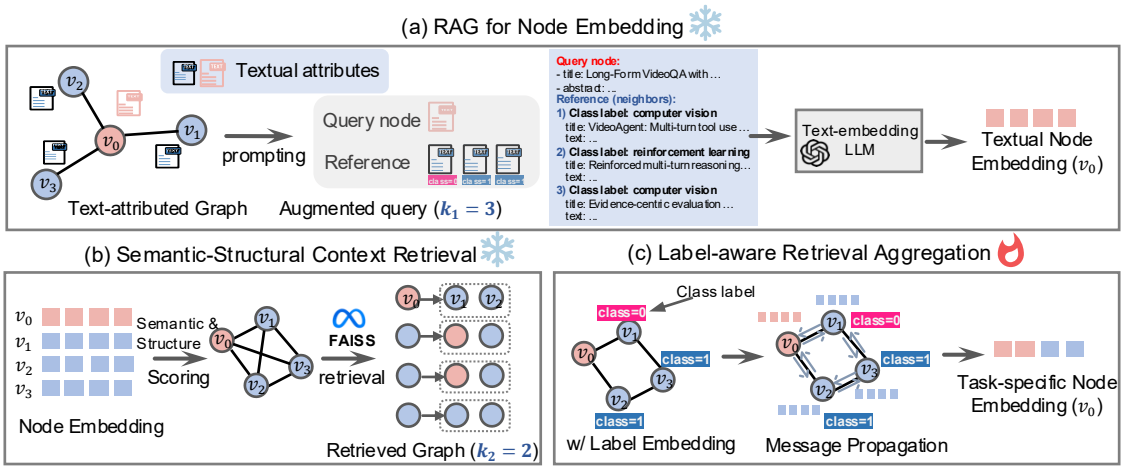

Fig. 3. Overall framework of RTA. RTA recasts graph representation learning as a retrieval-augmented pipeline that avoids explicit multi-hop message passing over the original graph. Specifically, **(a)** for a query node v_0, we augment its text with a small set of structural neighbors (size k_1) and corresponding labels as the resulting prompt to obtain the contextual node embedding $\mathbf{x}_{v_0}$. Then, **(b)** based on the learned embeddings and the original graph structure, we construct a retrieval-defined context graph by jointly considering semantic similarity and structural proximity. Specifically, semantic retrieval is performed in the embedding space (e.g., via FAISS), while structural retrieval exploits graph diffusion information. Finally, **(c)** we aggregate features and available label embeddings from retrieved contexts and apply an MLP-style update to produce task-specific node representations for prediction.

Empirically, such contextual augmentation yields richer semantic alignment across neighboring nodes and provides a stronger basis for subsequent structure learning.

4.2.2 Retrieval-augmented aggregation

Once the contextual embeddings $\{\mathbf{x}_u\}_{u\in V}$ are obtained, RTA performs graph structure learning by constructing a retrieval-defined context graph. Rather than propagating messages along pre-defined edges, we rebuild the graph topology through a retrieval process that identifies task-relevant nodes from both semantic and structural perspectives. For each node u, we retrieve a set of relevant contexts according to a retrieval score function:

$$\mathcal{R}_u^{(k_2)} = \text{Top-}k_{2v\neq u} S(u, v), \tag{9}$$

where $S(u, v)$ denotes the relevance score between nodes u and v. Specifically, we consider two complementary retrieval signals. The semantic retrieval score is calculated based on the contextual embeddings:

$$S_{\text{sem}}(u, v) = \frac{\mathbf{x}_u^\top \mathbf{x}_v}{\|\mathbf{x}_u\|_2 \|\mathbf{x}_v\|_2}, \tag{10}$$

which identifies nodes with similar semantic meanings and can be efficiently implemented using approximate nearest neighbor search, such as FAISS [42]. In addition to semantic similarity, we introduce a structural retrieval score to preserve the topological information of the original graph. Specifically, we employ personalized PageRank (PPR) [43] to measure the diffusion-based proximity between nodes. For a query node u, its PPR vector $\boldsymbol{\pi}_u$ is defined as:

$$\boldsymbol{\pi}_u = \alpha \mathbf{e}_u + (1 - \alpha)\mathbf{P}^\top \boldsymbol{\pi}_u, \tag{11}$$

where $\mathbf{P}$ is the transition matrix of the original graph, $\mathbf{e}_u$ is the one-hot personalization vector centered at node u, and α is the teleport probability. The structural relevance between nodes u and v is then defined as:

$$S_{\text{str}}(u, v) = \pi_u(v), \tag{12}$$

where $\pi_u(v)$ denotes the probability mass assigned to node v in the personalized PageRank vector of node u. Since semantic similarity and PPR scores may lie on different numerical scales, we apply query-wise min-max normalization to make the two retrieval signals comparable:

$$\widetilde{S}_r(u, v) = \frac{S_r(u, v) - \min_{j\neq u} S_r(u, j)}{\max_{j\neq u} S_r(u, j) - \min_{j\neq u} S_r(u, j) + \epsilon}, r \in \{\text{sem}, \text{str}\}, \tag{13}$$

where ϵ is a small constant for numerical stability. The final retrieval score combines the normalized semantic and structural evidence:

$$S(u, v) = \lambda \widetilde{S}_{\text{sem}}(u, v) + (1 - \lambda)\widetilde{S}_{\text{str}}(u, v), \tag{14}$$

where $\lambda \in [0, 1]$ controls the balance between semantic and structural retrieval. Since the fused score is used only for top-k ranking, we set $\lambda = 0.5$ by default to equally balance the two retrieval signals without dataset-specific tuning. Optionally, we further symmetrize the retrieved edges by mutual-kNN to ensure a stable, undirected retrieval graph. This process adaptively redefines connectivity according to both semantic relevance and structural proximity, resulting in a retrieval-defined graph that better reflects task-relevant relations. As illustrated in Figure 3(b), the retrieval-aggregation process enables the model to perform structure learning in a data-driven and scalable manner.

4.2.3 Label-aware retrieval aggregation

With the dynamically constructed similarity graph, message passing can be performed in a simplified yet expressive form. Instead of iterating over fixed edges, each node aggregates information from its retrieved contexts:

$$\mathbf{x}'_u = \text{MLP}\left(\mathbf{x}_u, \bigoplus\left(\{\mathbf{x}_v : v \in \mathcal{R}_u^{(k_2)}\}\right)\right), \tag{15}$$

This design can be viewed as an MLP operating over a retrieval-defined neighborhood. It eliminates the need for explicit graph convolutions and avoids common issues such

as oversmoothing and computational overhead caused by multi-hop message passing. Moreover, the retrieval mechanism allows efficient graph updates through approximate nearest neighbor (ANN) search tools such as FAISS [42], ensuring scalability to large text-attributed graphs. We embed labels via $\omega : [C] \to \mathbb{R}^d$ (one-hot or learned embedding). Define $\mathbf{z}_v = \omega(y_v)$ if $v \in \mathcal{T}$ and $\mathbf{z}_v = \mathbf{0}$ otherwise. We aggregate retrieved neighbors and apply an MLP update:

$$\mathbf{x}'_u = \mathrm{MLP}\left(\mathbf{x}_u, \bigoplus\left(\{\mathbf{x}_v + \mathbf{z}_v : v \in \mathcal{R}_u^{(k_2)}\}\right)\right), \tag{16}$$

Unlike transductive label propagation that directly diffuses labels along graph edges, RTA incorporates label embeddings as auxiliary retrieved context and does not directly overwrite unlabeled predictions. Labeled nodes influence which contexts are retrieved, but do not directly overwrite unlabeled representations.

4.2.4 Relation to prior work

While prior efforts such as GraphRAG [32] and G-Retriever [33] enhance RAG systems using graph structures, they treat graphs as auxiliary knowledge for retrieval in language models. In contrast, RTA reverses this direction: it treats retrieval as the fundamental operation for graph learning itself. By shifting from message passing on fixed edges to retrieval-based dynamic connectivity, RTA transforms graph learning into a retrieval-augmented reasoning process. This perspective unifies semantic retrieval and structural aggregation, resulting in a scalable, adaptable, and label-aware framework that generalizes beyond traditional GNN design.

4.3 Algorithm

To facilitate a better understanding of the functionality of RTA, we provide PyTorch-style pseudocode for its implementation in Algorithm 1.

Algorithm 1 PyTorch-style pseudocode for our model.

▷ Inputs: text attributes t, optional labels y, graph g
▷ L-I: label-aware contextual embedding
$C = \mathrm{Sample}(g, k_1)$ ▷ $C[u]$: structural contexts for node u
$X = \mathrm{LLM}(t, C, y)$ ▷ $X = \{x_u\}_{u \in V}$: contextual node embeddings
$Z = w(y)$ ▷ set $Z[u] = 0$ if y_u is unavailable
▷ Semantic-structural context retrieval
$S_{\mathrm{sem}} = \mathrm{Norm}(\mathrm{CosSim}(X))$ ▷ semantic relevance
$S_{\mathrm{str}} = \mathrm{Norm}(\mathrm{PPR}(g))$ ▷ structural relevance
$S = \lambda S_{\mathrm{sem}} + (1-\lambda) S_{\mathrm{str}}$ ▷ retrieval score fusion
$N = \mathrm{TopK}(S, k_2)$ ▷ $N[u] = \mathcal{R}_u$: retrieved contexts for node u
▷ L-II: label-enhanced retrieval aggregation
$M = (X + Z)[N].\mathrm{mean}(\mathrm{dim} = 1)$ ▷ aggregate retrieved contexts
$H = X + M$ ▷ update node representations
out $= \mathrm{MLP}(H)$
return out

5 Theoretical Analysis

In this section, we are motivated to bridge the gap between RAG and GNNs. We first develop a theoretical connection between message-passing GNNs and RAG by viewing both as instances of prediction over retrieved context sets. This perspective clarifies why message passing improves over node-independent models. Then, we demonstrate the robustness and optimization stability of retrieval-induced graph learning. Due to space limitations, we present simplified statements here and defer the full theorems and proofs to the supplementary material.

Theorem 1 (Informal). *Let $\gamma_u > 0$ be the top-k margin of retrieval scores with query node u, i.e., the gap between the k-th and the $k+1$-th largest score. Further let a GNN oracle be an arbitrary softmax-attention message passing at temperature τ induced by retrieval scores. Then under the regularity conditions as detailed in the supplementary material, there exists a RAG-style construction that approximates the oracle message at node u with ℓ_∞ error $O(\exp(-\gamma_u/\tau))$.*

Theorem 2. *Let $\mathcal{R}_u$ be a retrieved set for some query node u, with prediction distribution $p(\mathcal{R}_u) = (p_1, \ldots, p_C)$ induced by retrieved-context aggregation. Consider an outlier node $o \in \mathcal{R}_u$ whose individual distribution is $p_o = (p'_1, \ldots, p'_C)$ and define $m' = \arg\max_{c \in [C]} p'_c$. Let y_u be the ground-truth label of node u, treated as fixed during differentiation. If $y_u \neq m'$ and $p'_{m'} \geq p_{m'}$, then*

$$0 \leq \frac{\partial \mathcal{L}(\mathcal{R}_u)}{\partial \ell'_{m'}} \leq \frac{\partial \mathcal{L}(o)}{\partial \ell'_{m'}},$$

where $\mathcal{L}(\mathcal{R}_u) = \mathrm{CE}(p(\mathcal{R}_u), y_u)$ and $\mathcal{L}(o) = \frac{1}{|\mathcal{R}_u|}\mathrm{CE}(p_o, y_u)$, and $\ell'_{m'}$ denotes the outlier logit for class m'.

Remark 1. *If the attention scores are constant on the context set, softmax attention reduces to uniform averaging, which matches the mean aggregator used in models such as GCN and GraphSAGE. Sum aggregation differs only by a degree-dependent scaling and can be viewed as the same update up to normalization. Therefore, the message-level conclusion remains informative for a broad class of message-passing GNNs, even when the final layer implementation does not explicitly use attention.*

Remark 2. *Theorem 2 shows that when retrieval introduces an outlier node o whose most confident class m' conflicts with the ground-truth label y_u and satisfies $p'_{m'} \geq p_{m'}$, the retrieved-context objective $\mathcal{L}(\mathcal{R}_u)$ is more tolerant compared to the individual supervision $\mathcal{L}(o)$, as evidenced by a smaller penalty imposed by the gradient.*

6 Experiments

In this section, we present empirical evaluation results of RTA on real-world graph benchmarks. The source code, including the datasets and all necessary scripts for reproducing the results, is available at https://github.com/EdisonLeeeee/RTA.

6.1 Experimental settings

6.1.1 Datasets

We evaluate RTA on a diverse collection of text-attributed graphs covering both homophilic and heterophilic settings. Specifically, we use six homophilic datasets from the

CSTAG benchmark [44], including Books-Children (Children), Books-History (History), Ele-Computers (Computers), Ele-Photo (Photo), Sports-Fitness (Fitness), and Ogbn-Arxiv-TA (Arxiv). We further include four heterophilic webpage graphs, namely Cornell, Texas, Wisconsin, and Washington [45]. In all datasets, each node is associated with textual attributes and a category label, and the task is node classification. For Arxiv, we use the official public split provided by the OGB benchmark [46]. For all remaining datasets, following prior work [24], [44], we adopt a random 60%/20%/20% split for train/validation/test. Dataset statistics are summarized in Table 1.

We briefly describe each dataset group below.

- **Ogbn-Arxiv-TA (Arxiv)** [44] is derived from the citation dataset Ogbn-Arxiv [46]. The task is to predict paper categories, formulated as a 40-class classification problem. The textual attributes of each paper node are extracted from its title and abstract.
- **Books-Children and Books-History** [44] are extracted from the Amazon-Books dataset. Books-Children contains items under the second-level category "Children", while Books-History contains items under the second-level category "History". Nodes represent books, and edges indicate frequent co-purchase or co-viewing relations. The labels correspond to third-level book categories, and the title and description of each book are used as textual attributes.
- **Ele-Computers and Ele-Photo** [44] are extracted from the Amazon-Electronics dataset. Ele-Computers contains products under the second-level category "Computers", while Ele-Photo contains products under the second-level category "Photo". Nodes represent electronics-related products, and edges indicate frequent co-purchase or co-viewing relations.
- **Sports-Fitness** [44] is extracted from the Amazon-Sports dataset and consists of items under the second-level category "Fitness". Nodes represent fitness-related products, and edges indicate frequent co-purchase or co-viewing relations.
- **Cornell, Texas, Wisconsin, and Washington** [45] are university web-page graphs. Nodes represent web pages, edges represent hyperlinks, and textual features are derived from page content. Each node is classified into one of several page types, such as student, faculty, or department.

TABLE 1
Datasets statistics. Both homophilic and heterophilic graphs are included in our experiments.

Dataset	#Nodes	#Edges	#Classes	Type
Children	76,875	1,554,578	24	Homophilic
History	41,551	358,574	12	Homophilic
Computers	87,229	721,081	10	Homophilic
Photo	48,362	500,928	12	Homophilic
Fitness	173,055	1,773,500	13	Homophilic
Arxiv	169,343	1,166,243	40	Homophilic
Cornell	191	292	5	Heterophilic
Texas	187	310	5	Heterophilic
Wisconsin	265	510	5	Heterophilic
Washington	229	394	5	Heterophilic

6.1.2 Baselines

We compare RTA with a broad range of baselines covering graph neural networks, self-supervised graph learning methods, LLM-based predictors, and graph-oriented LLM methods. Specifically, the baselines fall into four categories. ► *Standard GNNs*: GCN [1], GAT [2], GraphSAGE [3], and DGT [14]; ► *Graph contrastive learning (GCL) methods*: DGI [47], BGRL [48], MaskGAE [4], and GraphMAE [49]; ► *Vanilla LLMs*: QWEN2.5-7B-INSTRUCT, LLAMA-3.1-8B-INSTRUCT, QWEN3-8B, GPT-4O, and DEEPSEEK-V4; ► *Graph LLMs*: GLEM [23], TAPE [18], LlaGA [50], OFA [26], GOFA [51], LLM-BP [24], AuGLM [52], and DENSE [21].

In adversarial robustness experiments, we additionally compare with robust GNNs, including RGCN [53], MedianGCN [16], SimPGCN [54], and ProGNN [55]. We briefly describe these baselines below.

Standard GNNs.

- **GCN** [1]: A canonical graph convolutional model that mixes node features with a normalized adjacency operator.
- **GAT** [2]: Replaces fixed neighbor weights with learned attention coefficients, so different neighbors contribute unequally to the update.
- **GraphSAGE** [3]: An inductive GNN that samples neighbors and aggregates them with simple set functions, such as mean aggregation, followed by an MLP.
- **DGT** [14]: A sparse graph Transformer that dynamically selects structurally and semantically relevant nodes and applies sparse attention with learnable Katz positional encodings for efficient large-scale graph representation learning.

Graph contrastive learning methods.

- **DGI** [47]: A self-supervised method that trains an encoder by maximizing mutual information between local patch embeddings and a global graph summary, using corrupted graphs as negatives. It learns representations without labels and then fits a downstream classifier.
- **BGRL** [48]: A negative-free bootstrap approach with an online network predicting the target network's representations under graph augmentations. It stabilizes training via a stop-gradient target, similar in spirit to BYOL on graphs.
- **MaskGAE** [4]: A masked graph autoencoding framework that hides parts of the graph structure and trains an encoder–decoder to reconstruct them. The masking objective encourages representations that capture both semantics and topology.
- **GraphMAE** [49]: A masked autoencoder that reconstructs masked node attributes from partially observed graphs via an encoder–decoder pipeline. It is commonly used as self-supervised pretraining before supervised evaluation.

Vanilla LLMs. We include QWEN2.5-7B-INSTRUCT, LLAMA-3.1-8B-INSTRUCT, QWEN3-8B, GPT-4O, and

DEEPSEEK-V4 as zero-shot or prompt-based LLM baselines without task-specific fine-tuning. These baselines evaluate whether language models alone can perform node classification on text-attributed graphs without explicit graph modeling.

Graph LLMs.

- **GLEM** [23]: Couples a text encoder with a graph model and iteratively leverages text and structure signals, typically via pseudo-labeling or co-training style updates. It is a representative hybrid that explicitly fuses language and graph inductive biases.
- **TAPE** [18]: Uses an LLM to produce task-aware textual signals, such as explanations or enriched descriptions, that are then distilled or encoded for learning on text-attributed graphs.
- **LlaGA** [50]: A graph-aware LLM framework that injects graph context into LLM-based prediction, for example by linearizing local neighborhoods into text, and adapts the model for graph tasks.
- **OFA** [26]: A general graph foundation-style baseline that targets broad task coverage with a unified backbone and lightweight task adaptation.
- **GOFA** [51]: A stronger generalization-oriented extension in the same spirit as OFA, emphasizing robustness and transfer across datasets and tasks.
- **LLM-BP** [24]: Treats LLM predictions as local evidence and refines them with a belief-propagation-style graph inference procedure.
- **AuGLM** [52]: An LLM-based node classifier that enriches input with topological and semantic retrieval and employs a lightweight GNN for candidate label pruning.
- **DENSE** [21]: A zero-shot LLM-on-graph baseline that reduces cost and noise by grouping or bundling nodes and contexts before querying the LLM.

Robust GNN baselines. The following baselines are used specifically in adversarial robustness experiments.

- **RGCN** [53]: A robustness-oriented GCN variant that reduces sensitivity to noisy or adversarial edges by explicitly modeling uncertainty or unreliable neighborhoods.
- **MedianGCN** [16]: Replaces mean aggregation with a coordinate-wise median operator to suppress outlier neighbor messages.
- **SimPGCN** [54]: Encourages similarity-preserving representations and typically reweights edges based on feature similarity, reducing the impact of adversarially injected edges.
- **ProGNN** [55]: Jointly learns a cleaned graph structure together with GNN parameters under structural constraints.

We next describe how RTA generates robustness-aware node embeddings, which serve as the basis for subsequent retrieval and aggregation.

6.1.3 LLM-based Embedding Generation

For LLM-based embedding generation, we use a robustness-aware prompt that asks the model to distinguish consistent neighbor evidence from potentially misleading or adversarial neighbor information. The prompt encourages the generated embedding to preserve the target node's core semantics while downweighting inconsistent neighbor signals. The full prompt is shown in Table 2.

TABLE 2
Prompt used to generate robustness-aware text embeddings.

Task:
You are analyzing a target node in a graph where each node has associated text and a class label. You are also provided with its neighbors' texts and labels. However, due to adversarial perturbations, some neighbors may intentionally provide misleading information.

Please follow these steps:
1. Read the target node's text.
2. Check the consistency of each neighbor's text and label with the target node's content.
3. For consistent neighbors, softly incorporate their information; for conflicting ones, discard or downweight them.
4. Generate a numerical embedding that reflects the target node's core semantics and is robust against inconsistent or adversarial neighbors. The embedding should capture the semantic meaning of the text and be useful for similarity comparison, clustering, and retrieval tasks.

Input Text:
{input_text}

Reference Labels:
{reference_labels}

6.1.4 Hyperparameters

We use a unified hyperparameter setting across datasets for fair comparison. Specifically, hyperparameters are tuned on OGBN-ARXIV-TA and then kept fixed for all other datasets. All trainable models are optimized with Adam [56] for at most 500 epochs, using a learning rate of 0.001 and early stopping with a patience of 10 epochs. For RTA and neural baselines, we use a hidden dimension of 128 and adopt a two-layer architecture unless otherwise specified.

For the retrieval components in RTA, we retrieve $k_1 = 3$ labeled examples for label-aware text embedding and $k_2 = 5$ nodes for retrieval-augmented message aggregation; we also set the PPR teleport probability $\alpha = 0.5$ and the semantic-structural fusion weight $\lambda = 0.5$ throughout all experiments. Following prior work [21], [24], we use `bge-en-icl` [57] as the default text embedding model for dense retrieval.

6.1.5 Implementation and Evaluation

We implement RTA and all trainable baselines using PyTorch [58] and PyTorch Geometric [59]. Accuracy is used as the evaluation metric. For MLP, GNN-based and GraphLLM methods, we report the average accuracy and standard deviation over ten runs. For LLM-based baselines, due to computational cost, we report the average results over three runs to account for the variability introduced by few-shot sampling.

To avoid label leakage, RTA uses only training labels as label-aware guidance. For validation and test nodes, as well as nodes without labels, we replace the label input with a padding token so that no ground-truth label information

TABLE 3
Node classification results (%) on text-attributed graphs. The best and second-best performances are highlighted as **boldfaced** and underlined, respectively.

	Children	History	Photo	Computers	Fitness	Arxiv	Cornell	Texas	Wisc.	Wash.
MLP	$51.2_{\pm 0.2}$	$82.0_{\pm 0.9}$	$60.7_{\pm 0.9}$	$65.3_{\pm 0.4}$	$87.2_{\pm 0.3}$	$46.5_{\pm 0.8}$	$49.5_{\pm 0.7}$	$62.0_{\pm 1.8}$	$46.5_{\pm 1.5}$	$66.0_{\pm 0.9}$
GCN	$56.3_{\pm 0.3}$	$84.4_{\pm 0.3}$	$84.0_{\pm 0.9}$	$87.6_{\pm 0.4}$	$92.1_{\pm 0.7}$	$72.5_{\pm 0.6}$	$52.6_{\pm 0.4}$	$64.8_{\pm 1.5}$	$49.0_{\pm 0.7}$	$69.5_{\pm 0.5}$
GAT	$56.4_{\pm 0.8}$	$84.0_{\pm 0.6}$	$85.3_{\pm 0.5}$	$88.6_{\pm 0.2}$	$92.5_{\pm 0.5}$	$72.7_{\pm 0.8}$	$53.2_{\pm 0.6}$	$65.9_{\pm 1.4}$	$52.3_{\pm 0.4}$	$73.1_{\pm 0.7}$
GraphSAGE	$\underline{58.5_{\pm 0.1}}$	$86.1_{\pm 0.8}$	$82.9_{\pm 0.7}$	$87.9_{\pm 0.8}$	$92.8_{\pm 0.5}$	$72.9_{\pm 0.8}$	$54.8_{\pm 0.5}$	$64.2_{\pm 1.3}$	$50.1_{\pm 1.0}$	$71.4_{\pm 0.6}$
DGT	$57.4_{\pm 0.7}$	$86.2_{\pm 0.3}$	$85.1_{\pm 0.5}$	$87.3_{\pm 0.3}$	$92.0_{\pm 0.5}$	$74.6_{\pm 0.4}$	$54.5_{\pm 0.3}$	$67.3_{\pm 1.0}$	$51.4_{\pm 1.2}$	$71.5_{\pm 0.5}$
DGI	$54.9_{\pm 0.5}$	$84.9_{\pm 0.3}$	$82.7_{\pm 0.6}$	$86.9_{\pm 0.4}$	$92.9_{\pm 0.7}$	$72.1_{\pm 0.5}$	$14.7_{\pm 0.9}$	$11.2_{\pm 1.4}$	$12.1_{\pm 0.7}$	$21.0_{\pm 1.2}$
BGRL	$55.9_{\pm 0.4}$	$85.9_{\pm 0.6}$	$83.7_{\pm 0.5}$	$87.9_{\pm 0.2}$	$93.9_{\pm 0.4}$	$73.3_{\pm 0.7}$	$16.5_{\pm 1.0}$	$12.8_{\pm 1.7}$	$13.6_{\pm 1.4}$	$22.3_{\pm 1.1}$
MaskGAE	$57.1_{\pm 0.8}$	$86.1_{\pm 0.9}$	$83.8_{\pm 0.4}$	$\underline{89.1_{\pm 0.3}}$	$93.2_{\pm 0.6}$	$73.4_{\pm 0.5}$	$24.9_{\pm 1.1}$	$19.3_{\pm 2.0}$	$24.0_{\pm 1.6}$	$26.1_{\pm 1.4}$
GraphMAE	$57.0_{\pm 0.6}$	$\underline{86.2_{\pm 0.5}}$	$83.7_{\pm 0.7}$	$87.9_{\pm 0.4}$	$\underline{93.4_{\pm 0.5}}$	$73.3_{\pm 0.6}$	$23.0_{\pm 0.7}$	$17.7_{\pm 0.4}$	$23.0_{\pm 1.7}$	$24.9_{\pm 0.9}$
LLAMA3.1-8B	20.9	36.0	47.8	53.9	43.2	38.1	40.2	58.5	50.0	44.3
QWEN2.5-7B	23.0	19.2	41.0	52.4	56.2	34.3	38.1	56.3	48.7	42.0
QWEN3-8B	26.6	29.4	45.6	55.3	58.4	38.5	42.3	59.6	52.3	47.5
DEEPSEEK-V4	30.5	61.4	55.1	56.3	48.8	45.9	47.0	63.4	53.4	49.6
GPT-4O	30.8	53.3	55.6	60.3	66.4	46.8	45.5	63.1	56.6	48.9
GLEM	$52.1_{\pm 0.5}$	$84.0_{\pm 0.8}$	$79.8_{\pm 0.3}$	$80.2_{\pm 0.6}$	$88.9_{\pm 0.2}$	$74.7_{\pm 1.3}$	$57.8_{\pm 1.0}$	$70.6_{\pm 0.8}$	$63.2_{\pm 0.6}$	$68.9_{\pm 0.5}$
TAPE	$53.4_{\pm 0.7}$	$83.8_{\pm 0.6}$	$86.1_{\pm 0.5}$	$84.5_{\pm 0.2}$	$85.7_{\pm 0.7}$	$\underline{75.2_{\pm 0.3}}$	$78.6_{\pm 0.6}$	$84.2_{\pm 0.9}$	$80.3_{\pm 1.1}$	$74.9_{\pm 1.2}$
LLaGA	$53.2_{\pm 0.4}$	$84.9_{\pm 0.3}$	$\underline{87.1_{\pm 0.7}}$	$88.0_{\pm 0.4}$	$92.0_{\pm 0.3}$	$\underline{75.2_{\pm 0.9}}$	$82.6_{\pm 0.8}$	$85.5_{\pm 1.6}$	$85.1_{\pm 0.9}$	$70.5_{\pm 0.7}$
OFA	$50.1_{\pm 0.5}$	$78.3_{\pm 0.3}$	$82.9_{\pm 0.5}$	$81.5_{\pm 0.6}$	$87.2_{\pm 0.6}$	$69.8_{\pm 0.5}$	$75.8_{\pm 0.5}$	$81.8_{\pm 0.4}$	$74.8_{\pm 0.3}$	$76.0_{\pm 0.6}$
GOFA	$52.2_{\pm 0.3}$	$76.3_{\pm 0.4}$	$82.4_{\pm 0.6}$	$80.9_{\pm 0.2}$	$85.7_{\pm 0.9}$	$69.0_{\pm 0.9}$	$39.5_{\pm 0.6}$	$38.4_{\pm 1.0}$	$32.5_{\pm 1.5}$	$31.0_{\pm 1.1}$
LLM-BP	$54.8_{\pm 0.2}$	$79.9_{\pm 0.7}$	$84.1_{\pm 0.3}$	$84.1_{\pm 0.6}$	$81.2_{\pm 0.5}$	$67.8_{\pm 0.8}$	$83.3_{\pm 0.2}$	$81.7_{\pm 1.4}$	$77.8_{\pm 1.3}$	$73.1_{\pm 1.5}$
AuGLM	$55.2_{\pm 0.6}$	$81.4_{\pm 0.2}$	$83.5_{\pm 0.7}$	$82.7_{\pm 0.8}$	$80.9_{\pm 0.6}$	$69.4_{\pm 0.4}$	$81.5_{\pm 0.7}$	$80.1_{\pm 0.8}$	$79.4_{\pm 1.2}$	$78.5_{\pm 0.8}$
DENSE	$51.4_{\pm 0.7}$	$77.4_{\pm 0.3}$	$82.5_{\pm 0.8}$	$86.5_{\pm 0.6}$	$84.1_{\pm 0.4}$	$70.5_{\pm 0.9}$	$\underline{84.8_{\pm 1.1}}$	$\underline{92.5_{\pm 1.5}}$	$\underline{87.2_{\pm 0.9}}$	$\underline{81.7_{\pm 1.7}}$
RTA (ours)	$\mathbf{61.9}_{\pm 0.7}$	$\mathbf{87.0}_{\pm 0.5}$	$\mathbf{88.7}_{\pm 0.4}$	$\mathbf{91.2}_{\pm 0.3}$	$\mathbf{94.2}_{\pm 0.5}$	$\mathbf{75.4}_{\pm 0.9}$	$\mathbf{89.9}_{\pm 0.2}$	$\mathbf{97.4}_{\pm 0.4}$	$\mathbf{96.3}_{\pm 0.3}$	$\mathbf{82.6}_{\pm 0.2}$

is injected beyond the training set. All datasets used in the experiments are publicly available. Unless otherwise specified, all experiments are conducted on an NVIDIA RTX 4090 GPU with 24 GB memory. We use the Transformers [60] library to implement the LLM models.

6.2 Main results

The node classification results on both homophilic and heterophilic datasets are presented in Table 3. Several key observations emerge from the results.

RAG is a strong alternative to message passing: Traditional GNNs and GCL methods rely on explicit message passing and structural encoding to capture graph relationships. However, our results show that RTA, which replaces conventional message passing with retrieval-based augmentation, consistently outperforms these approaches. Although we do not explicitly use recursive message passing in the original graph structure, RTA surpasses standard GNNs and GCL methods by a large margin, demonstrating the advantage of leveraging textual information in conjunction with graph structures.

Simple MLP with RAG outperforms graph LLMs: RTA is a simple MLP network augmented with RAG mechanisms, making it more parameter-efficient compared to graph LLMs. Compared to graph LLMs such as TAPE and LLaGA, RTA achieves significant improvements, particularly on datasets with richer textual features and heterophilic settings. This result highlights the effectiveness of retrieval-based augmentation in enhancing model performance without the need for specialized graph-specific pretraining.

Graph-aware adaptation is crucial for LLMs to perform well on graph tasks: We observe that vanilla LLMs perform poorly without fine-tuning, emphasizing the necessity of graph-aware adaptations for learning on text-attributed graphs. While LLMs excel at unstructured tasks, their inability to inherently model graph topology leads to subpar performance when applied directly to node classification tasks. The retrieval-based augmentation in RTA addresses this limitation by dynamically providing relevant contextual information, enabling LLMs to reason over graph data more effectively.

6.3 Robustness against adversarial attacks

We evaluate adversarial robustness on four benchmark graphs (Children, History, Photo, and Computers) under the structure-perturbation attack `SGA` [61], which degrades node classification by adversarially modifying edges at test time. Figure 4 reports accuracy under `SGA` (clean results omitted).

RTA consistently achieves the best performance across all datasets. Under `SGA` attack, RTA attains $56.7/79.3/80.3/84.5$ accuracy on Children/History/Photo/Computers, outperforming the strongest baseline by clear margins. In contrast, standard GNNs (e.g., GCN, GAT) collapse under attack, and robust defenses (e.g., MedianGCN) improve stability but remain substantially below RTA. We attribute this robustness to retrieval-based message passing: instead of relying on a single, potentially poisoned local neighborhood, RTA retrieves task-relevant context that is less coupled to any specific adjacency and provides redundant evidence from multiple supports, leading to more stable predictions under edge perturbations.

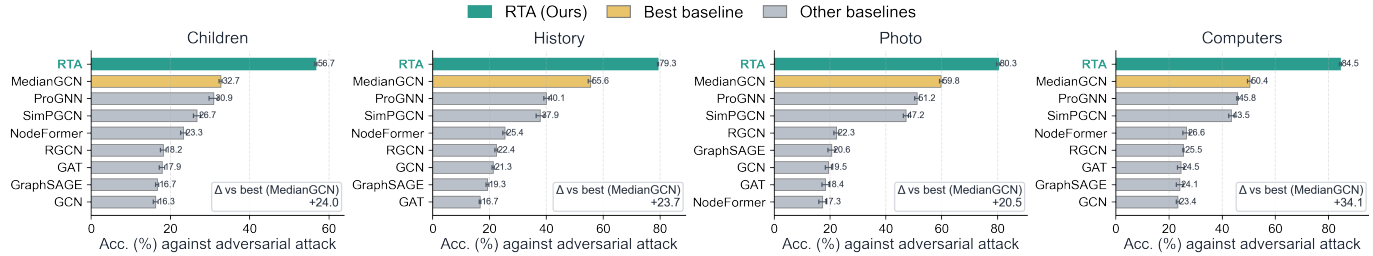

Fig. 4. Performance comparison against graph adversarial attack SGA [61].

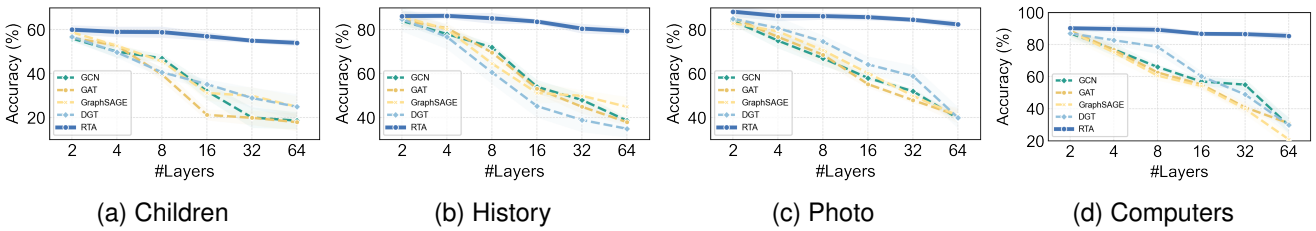

Fig. 5. Node classification performance with different network depth. Here we compare only with representative GNNs.

6.4 Robustness against oversmoothing

Although message passing enables GNNs to capture local graph structures, standard GNNs are known to suffer from the oversmoothing problem. Oversmoothing is a common phenomenon in GNNs, where increasing network depth leads to a deterioration in performance due to node representations becoming indistinguishable. To further investigate whether RTA can mitigate the performance degradation of GNN methods in deep architectures, we conduct experiments on the Children and History datasets with varying numbers of layers/depths, $L \in \{2, 4, 8, 16, 32\}$. The results of different GNN methods are shown in Figure 5.

As observed in Figure 5, standard GNNs experience severe oversmoothing as network depth increases. The learned representations in these methods collapse at greater depths, leading to significantly degraded performance beyond four layers. In contrast, RTA demonstrates remarkable robustness against oversmoothing, scaling effectively to deeper architectures without substantial performance loss. Notably, RTA achieves state-of-the-art performance across all depth levels. As the number of layers increases, RTA outperforms all baselines by increasingly larger margins. This highlights the effectiveness of retrieval-based augmentation in preserving representation quality and mitigating the limitations of traditional message-passing-based GNNs when scaling to deeper networks.

6.5 Efficiency comparison

We provide a comparison of training time between RTA and standard GNNs, including GCN, GAT, GraphSAGE, and DGT on the Fitness dataset - the largest dataset our experiments. The results in Figure 6 highlight that RTA achieves significantly higher computational efficiency while maintaining strong performance. Unlike traditional GNNs that require iterative message passing across multiple layers, RTA efficiently retrieves and incorporates contextual information through retrieval-augmented mechanisms, reducing

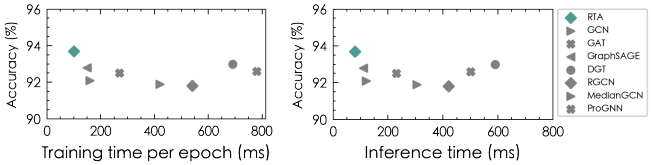

Fig. 6. Comparison of training and inference time on Fitness dataset across different GNNs.

redundant computations. Additionally, RTA avoids costly graph convolutions and neighbor aggregation steps, leading to faster convergence and lower training time. The efficiency of RTA is particularly advantageous for large-scale graph datasets, where standard GNNs often suffer from scalability issues due to high memory consumption and increased computational complexity. These results demonstrate that RTA provides a practical and scalable alternative to existing GNN models, offering a balance between computational efficiency and predictive accuracy.

6.6 Ablation study

In this subsection, we conduct ablation studies on eight representative benchmarks, including homophilic graph datasets Children, History, Photo, and Computers, and heterophilic graph datasets Cornell, Texas, Wisconsin, and Washington, to investigate the key factors that influence the performance of RTA.

6.6.1 *Impact of retrieved samples.*

We first conduct an ablation study on the number of retrieved text samples (k_1) and the number of neighbors used for message aggregation (k_2). The results are shown in Figure 7. Both k_1 and k_2 are important for the downstream performance of RTA. When $k_1 = 0$ or $k_2 = 0$, the method degenerates into a plain text-embedding baseline or a simple MLP without retrieved context or neighborhood

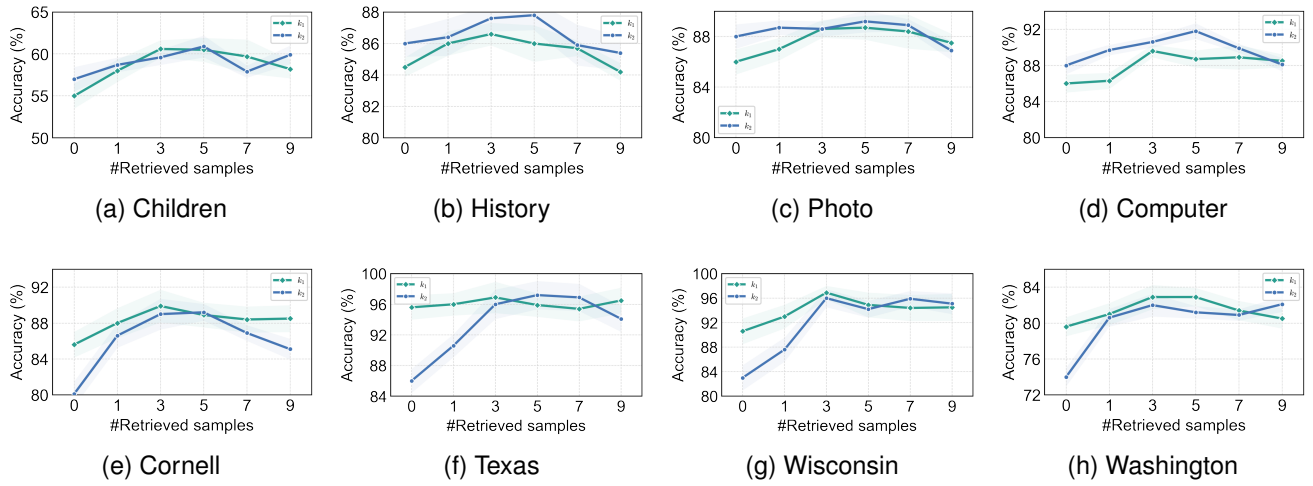

Fig. 7. Ablation results on the number of retrieved samples in text embedding (k_1) and message aggregation (k_2).

TABLE 4
Comparisons between RTA and its variants. L-I: label-aware contextual embedding; L-II: label-aware aggregation.

	Children	History	Photo	Computers	Cornell	Texas	Wisc.	Wash.
RTA \ (L-I)	$56.6_{\pm 0.3}$	$85.4_{\pm 0.9}$	$86.2_{\pm 0.8}$	$86.9_{\pm 0.6}$	$85.4_{\pm 0.4}$	$95.1_{\pm 0.7}$	$90.2_{\pm 0.6}$	$80.7_{\pm 0.9}$
RTA \ (L-II)	$58.3_{\pm 0.6}$	$86.6_{\pm 0.7}$	$85.4_{\pm 0.5}$	$88.8_{\pm 0.3}$	$79.4_{\pm 1.1}$	$86.1_{\pm 1.5}$	$83.3_{\pm 1.6}$	$74.6_{\pm 1.4}$
RTA \ (L-I&II)	$55.9_{\pm 0.7}$	$84.7_{\pm 0.4}$	$83.2_{\pm 0.3}$	$85.4_{\pm 0.4}$	$76.6_{\pm 1.4}$	$68.6_{\pm 0.9}$	$80.6_{\pm 0.8}$	$69.6_{\pm 1.2}$
RTA \ (semantic retrieval)	$57.8_{\pm 0.7}$	$84.9_{\pm 0.8}$	$84.6_{\pm 0.7}$	$87.1_{\pm 0.6}$	$78.8_{\pm 1.3}$	$84.7_{\pm 1.6}$	$82.1_{\pm 1.5}$	$73.8_{\pm 1.4}$
RTA \ (structural retrieval)	$59.9_{\pm 0.6}$	$85.9_{\pm 0.6}$	$87.1_{\pm 0.6}$	$89.8_{\pm 0.5}$	$85.6_{\pm 0.9}$	$93.2_{\pm 1.0}$	$91.5_{\pm 1.1}$	$79.8_{\pm 1.0}$
RTA	$\mathbf{61.9}_{\pm 0.7}$	$\mathbf{87.0}_{\pm 0.5}$	$\mathbf{88.7}_{\pm 0.4}$	$\mathbf{91.2}_{\pm 0.3}$	$\mathbf{89.9}_{\pm 0.2}$	$\mathbf{97.4}_{\pm 0.4}$	$\mathbf{96.3}_{\pm 0.3}$	$\mathbf{82.6}_{\pm 0.2}$

TABLE 5
Retriever backbone ablation for RTA on eight node classification benchmarks.

Retriever backbone	Children	History	Photo	Computers	Cornell	Texas	Wisc.	Wash.
`bert-base-uncased`	$60.1_{\pm 0.4}$	$85.7_{\pm 0.8}$	$87.6_{\pm 0.6}$	$90.3_{\pm 0.2}$	$81.4_{\pm 1.6}$	$80.4_{\pm 0.6}$	$77.8_{\pm 2.0}$	$68.6_{\pm 1.6}$
`all-mpnet-base-v2`	$60.6_{\pm 0.3}$	$86.2_{\pm 0.3}$	$88.2_{\pm 0.3}$	$90.8_{\pm 0.3}$	$82.6_{\pm 1.2}$	$84.9_{\pm 0.8}$	$79.1_{\pm 1.4}$	$69.6_{\pm 1.5}$
`bge-en-icl`	$\mathbf{61.9}_{\pm 0.7}$	$\mathbf{87.0}_{\pm 0.5}$	$\mathbf{88.7}_{\pm 0.4}$	$\mathbf{91.2}_{\pm 0.3}$	$\mathbf{89.9}_{\pm 0.2}$	$\mathbf{97.4}_{\pm 0.4}$	$\mathbf{96.3}_{\pm 0.3}$	$\mathbf{82.6}_{\pm 0.2}$

aggregation. Notably, relatively small values of k_1 and k_2 already yield strong performance, suggesting that a modest amount of high-relevance context is sufficient and helps avoid overwhelming the model. In contrast, increasing k_1 and k_2 beyond a certain point consistently hurts accuracy, likely because additional retrieved samples or neighbors become increasingly irrelevant or redundant and introduce noise. This degradation is more pronounced on heterophilic datasets, where indiscriminate neighborhood aggregation is more likely to mix semantically mismatched information across classes. Overall, these results reveal a clear trade-off in which appropriately chosen k_1 and k_2 improve performance by injecting useful contextual signals.

6.6.2 *Ablation on label-aware and retrieval components.*

RTA couples retrieval with lightweight aggregation and incorporates label information into both the contextual embedding stage (L-I) and the aggregation stage (L-II). We evaluate variants by removing L-I, L-II, or both components, with results reported in Table 4. Removing either component consistently degrades performance, confirming the benefit of label supervision at different stages of RTA. In particular, removing L-I weakens the label-informed contextual representations used for subsequent retrieval, whereas removing L-II leads to more pronounced degradation on heterophilic graphs, where retrieved contexts are more likely to contain class-inconsistent information and label-aware aggregation helps suppress such noise. Removing both components results in the largest overall performance drop, indicating that label guidance in contextual representation learning and retrieval aggregation provides complementary benefits. We further examine the contribution of the two retrieval signals by individually removing semantic or structural retrieval. Both variants perform consistently worse than the full model, demonstrating that semantic and structural evidence provide complementary information for context construction. Semantic retrieval generally contributes more substantially, particularly on heterophilic graphs, where graph topology alone may provide less reliable task-relevant context. Nevertheless, removing structural retrieval also leads to clear performance degradation, showing that PPR-based structural proximity captures useful high-order topological dependencies that are not fully reflected by embedding similarity. Overall, jointly exploiting label-aware modeling together with semantic and structural retrieval enables RTA to construct more informative and

task-relevant context sets.

6.6.3 Ablation on the retrieval embedding model.

In our main experiments, we follow prior work [21], [24] and adopt `bge-en-icl` [57] as the default retrieval embedding model used to encode texts for dense retrieval. To study the effect of embedding quality, we keep the retrieval pipeline fixed and only swap the text encoder among `bert-base-uncased` [40], `all-mpnet-base-v2` [62], and `bge-en-icl`. As shown in Table 5, `bge-en-icl` achieves the best performance across all eight benchmarks. The improvements are relatively modest on homophilic datasets, where neighborhood signals are already informative and retrieved texts mainly provide complementary evidence. In contrast, on heterophilic datasets, `bge-en-icl` yields substantially larger gains, which suggests that stronger semantic embeddings are crucial when informative neighborhoods are harder to recover and naive aggregation is more likely to introduce mismatched context. From a practical perspective, `all-mpnet-base-v2` offers a favorable accuracy–efficiency trade-off, while `bge-en-icl` is preferred when the additional compute is justified by the performance gains in challenging, especially heterophilic settings.

7 Conclusion

In this work, we establish a close connection between message-passing GNNs and retrieval-augmented generation. We show that RAG-inspired designs can serve as a simplified yet effective alternative to conventional GNN architectures for text-attributed graphs. By casting graph representation learning as a retrieval-augmented problem, RTA queries and aggregates task-relevant context from text-attributed nodes, which reduces the reliance on explicit message passing. We further draw inspiration from label propagation and use training labels as guidance signals that shape both retrieval embeddings and neighborhood aggregation. Theoretically, we formalize the relationship between RAG-based aggregation and standard message-passing GNNs, and we provide robustness guarantees for the optimization of label-aware retrieval. Empirically, RTA achieves state-of-the-art results on a range of text-attributed graph benchmarks compared with strong GNN, LLM, and graph-LLM baselines, while demonstrating improved scalability and robustness under adversarial perturbations and mitigating oversmoothing.

Acknowledgments

This work was supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0122701), the Young Scientists Fund (C) of the National Natural Science Foundation of China (No. 62602538), and the National Science Fund for Distinguished Young Scholars (No. 62525605).

References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR (Poster)*. OpenReview.net, 2017.

[2] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR (Poster)*. OpenReview.net, 2018.

[3] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[4] J. Li, R. Wu, W. Sun, L. Chen, S. Tian, L. Zhu, C. Meng, Z. Zheng, and W. Wang, "What's behind the mask: Understanding masked graph modeling for graph autoencoders," in *KDD*. ACM, 2023, pp. 1268–1279.

[5] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *WWW*. ACM, 2019, pp. 417–426.

[6] R. Sun, H. Dai, and A. W. Yu, "Does GNN pretraining help molecular representation?" in *NeurIPS*, 2022.

[7] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*. ACM, 2020, pp. 639–648.

[8] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1263–1272.

[9] J. Li, Z. Wei, Y. Zhu, R. Wu, H. Zhang, L. Chen, and Z. Zheng, "Heterophily-aware representation learning on heterogeneous graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 9, pp. 7852–7866, 2025.

[10] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," in *ICLR*. OpenReview.net, 2021.

[11] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T. Chua, and Q. Li, "A survey on RAG meeting llms: Towards retrieval-augmented large language models," in *KDD*. ACM, 2024, pp. 6491–6501.

[12] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *NeurIPS*, 2020.

[13] Q. Wu, W. Zhao, Z. Li, D. P. Wipf, and J. Yan, "Nodeformer: A scalable graph structure learning transformer for node classification," in *NeurIPS*, 2022.

[14] J. Park, S. Yun, H. Park, J. Kang, J. Jeong, K. Kim, J. Ha, and H. J. Kim, "Deformable graph transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 7, pp. 5385–5396, 2025.

[15] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. K. Prasanna, "Graphsaint: Graph sampling based inductive learning method," in *ICLR*. OpenReview.net, 2020.

[16] L. Chen, J. Li, Q. Peng, Y. Liu, Z. Zheng, and C. Yang, "Understanding structural vulnerability in graph convolutional networks," in *IJCAI*. ijcai.org, 2021, pp. 2249–2255.

[17] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*. OpenReview.net, 2019.

[18] X. He, X. Bresson, T. Laurent, A. Perold, Y. LeCun, and B. Hooi, "Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning," in *ICLR*. OpenReview.net, 2024.

[19] J. Tang, Y. Yang, W. Wei, L. Shi, L. Su, S. Cheng, D. Yin, and C. Huang, "Graphgpt: Graph instruction tuning for large language models," in *SIGIR*. ACM, 2024, pp. 491–500.

[20] Z. Chen, H. Mao, H. Li, W. Jin, H. Wen, X. Wei, S. Wang, D. Yin, W. Fan, H. Liu, and J. Tang, "Exploring the potential of large language models (llms)in learning on graphs," *SIGKDD Explor.*, vol. 25, no. 2, pp. 42–61, 2023.

[21] Y. Zhao, Q. Zhang, X. Luo, W. Zhang, Z. Xiao, W. Ju, P. S. Yu, and M. Zhang, "Dynamic bundling with large language models for zero-shot inference on text-attributed graphs," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=1nSynwHvu2

[22] Y. Shi, A. Zhang, E. Zhang, Z. Liu, and X. Wang, "Relm: Leveraging language models for enhanced chemical reaction prediction," in *EMNLP (Findings)*. Association for Computational Linguistics, 2023, pp. 5506–5520.

[23] J. Zhao, M. Qu, C. Li, H. Yan, Q. Liu, R. Li, X. Xie, and J. Tang, "Learning on large-scale text-attributed graphs via variational inference," in *ICLR*. OpenReview.net, 2023.

[24] H. P. Wang, S. Liu, R. Wei, and P. Li, "Generalization principles for inference over text-attributed graphs with

large language models," in *ICML*, 2025. [Online]. Available: https://openreview.net/forum?id=dfOqiHuklY
[25] E. Chien, W. Chang, C. Hsieh, H. Yu, J. Zhang, O. Milenkovic, and I. S. Dhillon, "Node feature extraction by self-supervised multi-scale neighborhood prediction," in *ICLR*. OpenReview.net, 2022.
[26] H. Liu, J. Feng, L. Kong, N. Liang, D. Tao, Y. Chen, and M. Zhang, "One for all: Towards training one graph model for all classification tasks," in *ICLR*. OpenReview.net, 2024.
[27] Z. Wang, Z. Wang, L. T. Le, H. S. Zheng, S. Mishra, V. Perot, Y. Zhang, A. Mattapalli, A. Taly, J. Shang, C. Lee, and T. Pfister, "Speculative RAG: enhancing retrieval augmented generation through drafting," *CoRR*, vol. abs/2407.08223, 2024.
[28] G. Izacard, P. S. H. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, "Atlas: Few-shot learning with retrieval augmented language models," *J. Mach. Learn. Res.*, vol. 24, pp. 251:1–251:43, 2023.
[29] S. Yan, J. Gu, Y. Zhu, and Z. Ling, "Corrective retrieval augmented generation," *CoRR*, vol. abs/2401.15884, 2024.
[30] X. Li, C. Zhu, L. Li, Z. Yin, T. Sun, and X. Qiu, "Llatrieval: Llm-verified retrieval for verifiable generation," in *NAACL-HLT*. Association for Computational Linguistics, 2024, pp. 5453–5471.
[31] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-rag: Learning to retrieve, generate, and critique through self-reflection," in *ICLR*. OpenReview.net, 2024.
[32] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson, "From local to global: A graph RAG approach to query-focused summarization," *CoRR*, vol. abs/2404.16130, 2024.
[33] X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," in *NeurIPS*, 2024.
[34] B. J. Gutierrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su, "Hipporag: Neurobiologically inspired long-term memory for large language models," in *NeurIPS*, 2024.
[35] B. J. Gutiérrez, Y. Shu, W. Qi, S. Zhou, and Y. Su, "From RAG to memory: Non-parametric continual learning for large language models," in *ICML*. OpenReview.net, 2025.
[36] G. Li, M. Müller, B. Ghanem, and V. Koltun, "Training graph neural networks with 1000 layers," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 6437–6449.
[37] V. Karpukhin, B. Oguz, S. Min, P. S. H. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense passage retrieval for open-domain question answering," in *EMNLP (1)*. Association for Computational Linguistics, 2020, pp. 6769–6781.
[38] J. Jin, Y. Zhu, X. Yang, C. Zhang, and Z. Dou, "Flashrag: A modular toolkit for efficient retrieval-augmented generation research," *CoRR*, vol. abs/2405.13576, 2024.
[39] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NeurIPS*. MIT Press, 2003, pp. 321–328.
[40] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*. Association for Computational Linguistics, 2019, pp. 4171–4186.
[41] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: decoding-enhanced bert with disentangled attention," in *ICLR*. OpenReview.net, 2021.
[42] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.
[43] G. Jeh and J. Widom, "Scaling personalized web search," in *WWW*. ACM, 2003, pp. 271–279.
[44] H. Yan, C. Li, R. Long, C. Yan, J. Zhao, W. Zhuang, J. Yin, P. Zhang, W. Han, H. Sun, W. Deng, Q. Zhang, L. Sun, X. Xie, and S. Wang, "A comprehensive study on text-attributed graphs: Benchmarking and rethinking," in *NeurIPS*, 2023.
[45] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, "Learning to extract symbolic knowledge from the world wide web," *AAAI/IAAI*, vol. 3, no. 3.6, p. 2, 1998.
[46] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *arXiv preprint arXiv:2005.00687*, 2020.
[47] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax." *ICLR (Poster)*, vol. 2, no. 3, p. 4, 2019.
[48] S. Thakoor, C. Tallec, M. G. Azar, R. Munos, P. Veličković, and M. Valko, "Bootstrapped representation learning on graphs," in *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.
[49] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "Graphmae: Self-supervised masked graph autoencoders," in *KDD*. ACM, 2022, pp. 594–604.
[50] R. Chen, T. Zhao, A. K. Jaiswal, N. Shah, and Z. Wang, "Llaga: Large language and graph assistant," in *ICML*. OpenReview.net, 2024.
[51] L. Kong, J. Feng, H. Liu, C. Huang, J. Huang, Y. Chen, and M. Zhang, "GOFA: A generative one-for-all model for joint graph language modeling," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=mIjblC9hfm
[52] Z. Xu, K. Hassani, S. Zhang, H. Zeng, M. Yasunaga, L. Wang, D. Fu, N. Yao, B. Long, and H. Tong, "How to make lms strong node classifiers?" in *EACL (Findings)*, ser. Findings of ACL. Association for Computational Linguistics, 2026, pp. 252–274.
[53] D. Zhu, Z. Zhang, P. Cui, and W. Zhu, "Robust graph convolutional networks against adversarial attacks," in *KDD*, 2019, pp. 1399–1407.
[54] W. Jin, T. Derr, Y. Wang, Y. Ma, Z. Liu, and J. Tang, "Node similarity preserving graph convolutional networks," in *WSDM*, 2021, pp. 148–156.
[55] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," in *KDD*. ACM, 2020, pp. 66–74.
[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.
[57] C. Li, M. Qin, S. Xiao, J. Chen, K. Luo, Y. Shao, D. Lian, and Z. Liu, "Making text embedders few-shot learners," 2024. [Online]. Available: https://arxiv.org/abs/2409.15700
[58] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8024–8035.
[59] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
[60] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *CoRR*, vol. abs/1910.03771, 2019.
[61] J. Li, T. Xie, L. Chen, F. Xie, X. He, and Z. Zheng, "Adversarial attack on large scale graph," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 82–95, 2023.
[62] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: http://arxiv.org/abs/1908.10084

APPENDIX

This section provides theoretical support for the retrieval-augmented view developed in Section 4. Rather than treating message passing as a graph-specific primitive, we analyze it as *prediction over a retrieved context set*. Concretely, for a query node u we denote by $\mathcal{R}_u^{(k)}$ its retrieved set (constructed by score-based Top-k retrieval), and we view the downstream computation as a permutation-invariant summarization of $\mathcal{R}_u^{(k)}$ followed by a parametric predictor.

We further consider the label-aware setting where a subset of nodes (e.g., the training set) can provide supervision as retrieved *label evidence*. Throughout, let $\mathcal{T} \subseteq \mathcal{V}$ denote the set of labeled nodes: for each $v \in \mathcal{T}$ we observe a (possibly noisy) label $\tilde{y}_v \in [C]$, while nodes outside $\mathcal{T}$ contribute only a fixed padding embedding (taken as $\mathbf{0}$ for simplicity). Under this notation, we establish two

complementary results that align directly with the design of RTA: (i) an *approximation* showing that Top-k retrieval with mean aggregation provides a sparse approximation to softmax-attention message passing under a margin condition; and (ii) an *optimization robustness* guarantee showing that retrieved-context supervision attenuates the gradient impact of mis-retrieved outliers. Together, these statements clarify when retrieval-defined context sets can replace traditional message passing on fixed edges, and why injecting label evidence into retrieval and aggregation can improve stability without sacrificing robustness to residual retrieval noise.

Assumption 1 (Bounded retrieved features)**.** *Define normalized embeddings $\bar{\mathbf{x}}_u = \mathbf{x}_u/\|\mathbf{x}_u\|_2$, such that $\|\bar{\mathbf{x}}_u\|_2 = 1$. Let $\omega : [C] \to \mathbb{R}^d$ be the label embedding map and define the label embedding used in aggregation as*

$$\mathbf{z}_v \;=\; \omega(\tilde{y}_v) \;\; \textit{if } v \in \mathcal{T}, \qquad \mathbf{z}_v = \mathbf{0} \;\; \textit{otherwise}.$$

Let the label-augmented neighbor feature be

$$\mathbf{r}_v \;:=\; \mathbf{x}_v + \mathbf{z}_v \in \mathbb{R}^d.$$

Assume coordinate-wise boundedness: for all v and coordinates j,

$$|(\mathbf{r}_v)_j| \le B.$$

Assumption 2 (Top-k retrieval margin)**.** *Fix a node u. Let $s(u, v)$ be the retrieval score used in Eq. (9) to construct $\mathcal{R}_u^{(k)}$. In RTA, $s(u, v) = S(u, v)$ denotes the generalized retrieval score that combines semantic and structural relevance. Let $v_{(1)}, v_{(2)}, \dots$ be nodes sorted by $s(u, \cdot)$ in descending order and define the top-k margin*

$$\gamma_u \;=\; s\big(u, v_{(k)}\big) - s\big(u, v_{(k+1)}\big).$$

Assume $\gamma_u > 0$ for the nodes of interest.

Theorem 3 (Formal statement of Theorem 1)**.** *Fix a node u and let $s(u, v)$ be the retrieval score used in Eq. (9). Let $\mathcal{R}_u^{(k)} = \{v_{(1)}, \dots, v_{(k)}\}$ be the Top-k set under $s(u, \cdot)$. For any temperature $\tau > 0$, define softmax attention weights over all candidates $v \neq u$ by*

$$a_{uv}(\tau) \;:=\; \frac{\exp\big(s(u, v)/\tau\big)}{\sum_{w \neq u} \exp\big(s(u, w)/\tau\big)}.$$

Define the attention message and the Top-k mean message as

$$\mathbf{m}_u^{\mathrm{att}}(\tau) \;:=\; \sum_{v \neq u} a_{uv}(\tau)\, \mathbf{r}_v, \qquad \mathbf{m}_u^{\mathrm{mean}} \;:=\; \frac{1}{k} \sum_{v \in \mathcal{R}_u^{(k)}} \mathbf{r}_v,$$

where $\mathbf{r}_v = \mathbf{x}_v + \mathbf{z}_v$ as in Assumption 1. Let $\pi_u(\tau) := \sum_{v \notin \mathcal{R}_u^{(k)}} a_{uv}(\tau)$ be the attention mass outside the retrieved set, and let

$$\delta_u \;:=\; s\big(u, v_{(1)}\big) - s\big(u, v_{(k)}\big) \;\ge 0$$

be the score spread within the retrieved set. Under Assumption 1, the following bound holds:

$$\big\|\mathbf{m}_u^{\mathrm{att}}(\tau) - \mathbf{m}_u^{\mathrm{mean}}\big\|_\infty \;\le\; 2B\, \pi_u(\tau) \;+\; B\big(\exp(\delta_u/\tau) - 1\big).$$

Moreover, under Assumption 2,

$$\pi_u(\tau) \;\le\; (N - 1 - k) \exp\big(-\gamma_u/\tau\big),$$

where $N = |\mathcal{V}|$.

Proof. Let $S = \mathcal{R}_u^{(k)}$. Define the renormalized weights on S by

$$\tilde{a}_{uv}(\tau) = \frac{a_{uv}(\tau)}{\sum_{w \in S} a_{uw}(\tau)} = \frac{a_{uv}(\tau)}{1 - \pi_u(\tau)} \qquad (v \in S),$$

and the truncated attention message $\tilde{\mathbf{m}}_u(\tau) = \sum_{v \in S} \tilde{a}_{uv}(\tau)\, \mathbf{r}_v$.

First, bound the leakage error:

$$\mathbf{m}_u^{\mathrm{att}}(\tau) - \tilde{\mathbf{m}}_u(\tau) = \sum_{v \notin S} a_{uv}(\tau)\, \mathbf{r}_v + \sum_{v \in S} \Big(a_{uv}(\tau) - \tilde{a}_{uv}(\tau)\Big)\mathbf{r}_v.$$

Since $\sum_{v \in S} |a_{uv}(\tau) - \tilde{a}_{uv}(\tau)| = \pi_u(\tau)$ and $\|\mathbf{r}_v\|_\infty \le B$ by Assumption 1, we obtain

$$\|\mathbf{m}_u^{\mathrm{att}}(\tau) - \tilde{\mathbf{m}}_u(\tau)\|_\infty \le B\pi_u(\tau) + B\pi_u(\tau) = 2B\pi_u(\tau).$$

Second, bound the within-S weighting error. By triangle inequality and $\|\mathbf{r}_v\|_\infty \le B$,

$$\|\tilde{\mathbf{m}}_u(\tau) - \mathbf{m}_u^{\mathrm{mean}}\|_\infty = \Big\| \sum_{v \in S} \big(\tilde{a}_{uv}(\tau) - \tfrac{1}{k}\big)\mathbf{r}_v \Big\|_\infty \le B \sum_{v \in S} \Big|\tilde{a}_{uv}(\tau) - \tfrac{1}{k}\Big|.$$

By definition of δ_u, for all $v \in S$ we have $s(u, v) \in [s(u, v_{(k)}),\ s(u, v_{(k)}) + \delta_u]$. Thus $\exp(s(u, v)/\tau)$ varies within a multiplicative factor $\exp(\delta_u/\tau)$ over $v \in S$, which implies

$$\frac{1}{k \exp(\delta_u/\tau)} \;\le\; \tilde{a}_{uv}(\tau) \;\le\; \frac{\exp(\delta_u/\tau)}{k} \qquad \forall v \in S.$$

Hence, $\big|\tilde{a}_{uv}(\tau) - \frac{1}{k}\big| \le \frac{\exp(\delta_u/\tau) - 1}{k}$, and summing over $v \in S$ yields

$$\sum_{v \in S} \Big|\tilde{a}_{uv}(\tau) - \tfrac{1}{k}\Big| \le \exp(\delta_u/\tau) - 1.$$

Therefore,

$$\|\tilde{\mathbf{m}}_u(\tau) - \mathbf{m}_u^{\mathrm{mean}}\|_\infty \le B\big(\exp(\delta_u/\tau) - 1\big).$$

Combining the two bounds gives

$$\|\mathbf{m}_u^{\mathrm{att}}(\tau) - \mathbf{m}_u^{\mathrm{mean}}\|_\infty \le 2B\pi_u(\tau) + B\big(\exp(\delta_u/\tau) - 1\big).$$

Finally, bound $\pi_u(\tau)$ under Assumption 2. For any $v \notin S$, $s(u, v) \le s(u, v_{(k+1)}) \le s(u, v_{(k)}) - \gamma_u$. Therefore,

$$\sum_{v \notin S} \exp(s(u, v)/\tau) \le (N - 1 - k) \exp\big((s(u, v_{(k)}) - \gamma_u)/\tau\big),$$

while

$$\sum_{v \in S} \exp(s(u, v)/\tau) \ge \exp\big(s(u, v_{(k)})/\tau\big).$$

Thus,

$$\pi_u(\tau) = \frac{\sum_{v \notin S} \exp(s(u, v)/\tau)}{\sum_{w \neq u} \exp(s(u, w)/\tau)} \le \frac{\sum_{v \notin S} \exp(s(u, v)/\tau)}{\sum_{v \in S} \exp(s(u, v)/\tau)} \le (N - 1 - k) \exp(-\gamma_u/\tau).$$

□

Remark 3. *Theorem 3 justifies Top-k retrieval as a principled sparsification of attention-based message passing. The theorem is stated for softmax-attention message passing because it provides a convenient envelope for many classical GNN layers. If the attention scores are constant on the context set, softmax attention reduces to uniform averaging, which matches the mean aggregator used in models such as GCN and GraphSAGE. Sum aggregation*

differs only by a degree-dependent scaling and can be viewed as the same update up to normalization. Therefore, the message-level conclusion remains informative for a broad class of message-passing GNNs, even when the final layer implementation does not explicitly use attention.

Remark 4. *Theorem 3 separates two sources of approximation error between $m_u^{\mathrm{att}}(\tau)$ and the Top-k mean message m_u^{mean}. The first term $2B\,\pi_u(\tau)$ captures* retrieval leakage*: even if the Top-k set is fixed, attention still assigns some probability mass to $v \notin \mathcal{R}_u^{(k)}$, which vanishes when the margin γ_u is large (stable retrieval). The second term $B(\exp(\delta_u/\tau)-1)$ captures* within-set reweighting*: attention is generally non-uniform on $\mathcal{R}_u^{(k)}$, and it approaches uniform (hence mean aggregation) when the score spread $\delta_u = s(u, v_{(1)}) - s(u, v_{(k)})$ is small, i.e., the retrieved items are similarly relevant. This decomposition clarifies why a retrieval-based graph learner can replace attention-style message passing: good retrieval quality corresponds to simultaneously (i) suppressing leakage via a clear margin and (ii) keeping the retrieved set "coherent" so that a simple permutation-invariant mean operator is an accurate surrogate for attention.*

Theorem 4 (Restatement of Theorem 2)**.** *Let $\mathcal{R}_u$ be a retrieved set for some query node u, with prediction distribution $p(\mathcal{R}_u) = (p_1, \ldots, p_C)$ induced by retrieved-context aggregation. Consider an outlier node $o \in \mathcal{R}_u$ whose individual distribution is $p_o = (p'_1, \ldots, p'_C)$ and define $m' = \arg\max_{c\in[C]} p'_c$. Let y_u be the ground-truth label of node u, treated as fixed during differentiation. If $y_u \neq m'$ and $p'_{m'} \geq p_{m'}$, then*

$$0 \leq \frac{\partial \mathcal{L}(\mathcal{R}_u)}{\partial \ell'_{m'}} \leq \frac{\partial \mathcal{L}(o)}{\partial \ell'_{m'}},$$

where $\mathcal{L}(\mathcal{R}_u) = \mathrm{CE}(p(\mathcal{R}_u), y_u)$ and $\mathcal{L}(o) = \frac{1}{|\mathcal{R}_u|}\mathrm{CE}(p_o, y_u)$, and $\ell'_{m'}$ denotes the outlier logit for class m'.

Proof. Since y_u is the ground-truth label of the query node, it is fixed with respect to model parameters during differentiation.

Let $\mathbf{e} \in \{0,1\}^C$ be the one-hot vector of y_u (i.e., $e_{y_u} = 1$). For each node $i \in \mathcal{R}_u$, let $\boldsymbol{\ell}_i = (\ell_{i,1}, \ldots, \ell_{i,C})$ be its logits. Define the set-level logits by mean pooling:

$$\ell_c = \frac{1}{|\mathcal{R}_u|} \sum_{i\in\mathcal{R}_u} \ell_{i,c}, \qquad c = 1, \ldots, C,$$

and set $p(\mathcal{R}_u) = p = \mathrm{softmax}(\boldsymbol{\ell})$. For the outlier node o, denote its logits by $\boldsymbol{\ell}_o = \boldsymbol{\ell}' = (\ell'_1, \ldots, \ell'_C)$ and its distribution by $p_o = p' = \mathrm{softmax}(\boldsymbol{\ell}')$.

Using $\frac{\partial \mathcal{L}(\mathcal{R}_u)}{\partial \ell_c} = p_c - e_c$ and $\frac{\partial \ell_c}{\partial \ell'_c} = \frac{1}{|\mathcal{R}_u|}$ for the outlier node, the chain rule yields

$$\frac{\partial \mathcal{L}(\mathcal{R}_u)}{\partial \ell'_c} = \frac{1}{|\mathcal{R}_u|}(p_c - e_c).$$

Likewise, since $\nabla_{\boldsymbol{\ell}'} \mathrm{CE}(\mathrm{softmax}(\boldsymbol{\ell}'), y_u) = p' - \mathbf{e}$, we obtain

$$\frac{\partial \mathcal{L}(o)}{\partial \ell'_c} = \frac{1}{|\mathcal{R}_u|}(p'_c - e_c).$$

Evaluating at $c = m'$ and using $y_u \neq m'$ (hence $e_{m'} = 0$), we have

$$\frac{\partial \mathcal{L}(\mathcal{R}_u)}{\partial \ell'_{m'}} = \frac{1}{|\mathcal{R}_u|} p_{m'} \geq 0, \qquad \frac{\partial \mathcal{L}(o)}{\partial \ell'_{m'}} = \frac{1}{|\mathcal{R}_u|} p'_{m'}.$$

Together with $p'_{m'} \geq p_{m'}$, this implies

$$0 \leq \frac{\partial \mathcal{L}(\mathcal{R}_u)}{\partial \ell'_{m'}} \leq \frac{\partial \mathcal{L}(o)}{\partial \ell'_{m'}}.$$

□

Remark 5. *Theorem 4 shows that when retrieval introduces an outlier node o whose most confident class m' conflicts with the ground-truth label y_u and satisfies $p'_{m'} \geq p_{m'}$, the retrieved-context objective $\mathcal{L}(\mathcal{R}_u)$ is more tolerant compared to the individual supervision $\mathcal{L}(o)$, as evidenced by a smaller penalty imposed by the gradient.*

Remark 6. *Theorem 4 further indicates that even under imperfect retrieval, outlier neighbors whose predictions conflict with the ground-truth label have their optimization impact attenuated under retrieved-context supervision. Consequently, injecting label information can improve semantic coherence and retrieval precision, while the retrieved-context objective maintains robustness to the remaining retrieval noise. Together, these effects help explain why combining retrieval with label-aware aggregation can simultaneously enhance optimization stability and downstream performance in graph representation learning.*