

Unveiling the Vulnerability of Graph-LLMs: An Interpretable Multi-Dimensional Adversarial Attack on TAGs

Bowen Fan
Beijing Institute of Technology
China
fan1085165825@gmail.com

Zhilin Guo
Shandong University
China
frank04180@outlook.com

Xunkai Li
Beijing Institute of Technology
China
cs.xunkai.li@gmail.com

Yihan Zhou
Beijing Institute of Technology
China
z2005416z@163.com

Bing Zhou
Shenzhen Institute of Technology
China
bingzhou6@gmail.com

Zhenjun Li
Shenzhen Institute of Technology
China
15323940@qq.com

Rong-Hua Li
Beijing Institute of Technology
China
lironghuabit@126.com

Guoren Wang
Beijing Institute of Technology
China
wanggrbit@gmail.com

Abstract

Graph Neural Networks (GNNs) have become a pivotal framework for modeling graph-structured data, enabling a wide range of applications from social network analysis to molecular chemistry. By integrating large language models (LLMs), text-attributed graphs (TAGs) enhance node representations with rich textual semantics, significantly boosting the expressive power of graph-based learning. However, this sophisticated synergy introduces critical vulnerabilities, as Graph-LLMs are susceptible to adversarial attacks on both their structural topology and textual attributes. Although specialized attack methods have been designed for each of these aspects, no work has yet unified them into a comprehensive approach. In this work, we propose the Interpretable Multi-Dimensional Graph Attack (IMDGA), a novel human-centric adversarial attack framework designed to orchestrate multi-level perturbations across both graph structure and textual features. IMDGA utilizes three tightly integrated modules to craft attacks that balance interpretability and impact, enabling a deeper understanding of Graph-LLM vulnerabilities. Through rigorous theoretical analysis and comprehensive empirical evaluations on diverse datasets and architectures, IMDGA demonstrates superior interpretability, attack effectiveness, stealthiness, and robustness compared to existing methods. By exposing critical weaknesses in TAG representation learning, this work uncovers a previously underexplored semantic dimension of vulnerability in Graph-LLMs, offering valuable insights for improving their resilience. Our code and resources are publicly available at <https://anonymous.4open.science/r/IMDGA-7289>.

CCS Concepts

• **Computing methodologies** → **Semi-supervised learning settings**; **Neural networks**.

Keywords

Text-attributed graphs; Graph-LLMs; Adversarial attacks

1 Introduction

In the contemporary landscape of data science, graphs are indispensable data structures for representing intricate and interactive entities [61]. Within domains such as citation networks [20, 47, 58] and social media platforms [28, 29, 59], where nodes are characteristically imbued with copious semantic content, text-attributed graphs (TAGs) distinguish themselves from traditional graphs by offering a more semantically enriched structural paradigm [7, 9, 18]. Coinciding with this, Graph Neural Networks (GNNs) [12, 34, 52] have rapidly developed into a powerful tool for modeling TAGs, effectively capturing the intricate interactions and profound semantic connections within graphs. This ability to understand both the structure and semantics of TAGs has led to outstanding performance in various downstream tasks, including the biological networks [13, 42, 70] and recommendation systems [2, 19, 51].

With the ascendance of large language models (LLMs), encoding models such as Sentence-BERT [44] and RoBERTa [30] have been adapted to the graph domain, giving rise to the new innovative paradigm known as Graph-LLMs [8, 22, 31, 41]. This approach transcends shallow textual encoding methods (e.g., skip-gram [36] and BoW [17]), thereby endowing node features in TAGs with more profound semantic information. Concurrently, this non-decoupled paradigm dismantles the conventional separation of text feature processing and model architecture designing, significantly enhancing the capability of representation learning on TAGs [5, 69].

However, recent investigations have underscored the inherent vulnerability of GNNs to adversarial examples, which are meticulously crafted by introducing subtle perturbations to the original input data [15, 25, 53]. In the conventional GNN settings, attacks typically involve alterations to the graph structure or node features. Notably, Graph Modification Attacks (GMAs) [11, 54, 62, 72] and Graph Injection Attacks (GIAs) [6, 57, 68, 71], as emblematic instances, pose formidable challenges to the robustness of GNNs. In the context of TAGs, the incorporation of additional textual information into Graph-LLMs introduces new security concerns, suggesting that attack methodologies from the field of Natural

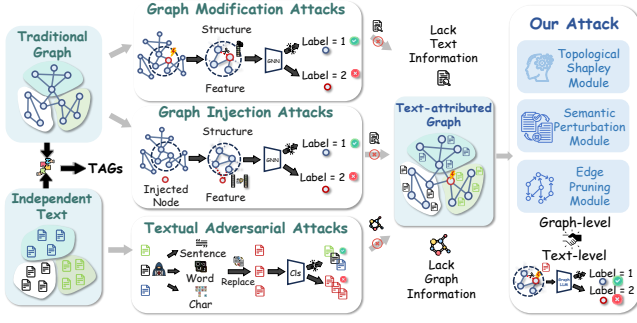


Figure 1: Illustration of different adversarial attack spaces

Language Processing (NLP) may have a significant impact on the representation learning of existing TAGs [23, 27, 56].

While prior adversarial attack methods have explored diverse perspectives (as illustrated in Figure 1), they clearly manifest the following shortcomings in scenarios integrating LLMs with graphs:

- (1) *Incompatibility with real-world constraints*: A large portion of conventional adversarial attacks rely on either unrestricted access to the target model’s internal gradients or the deployment of poisoning data during the model training phase, both of which are often strictly unattainable in practical environments [3, 4, 49].
- (2) *Limited scope in attack paradigms*: Although GMAs and GIAs are meticulously designed attack frameworks that compromise GNNs by leveraging graph-structure knowledge, they fail to address vulnerabilities originating at the raw textual level [26]. Conversely, while textual attacks from the NLP domain can significantly mislead models in classification, the textual features on TAGs are inherently interconnected through the graph’s message-passing mechanism [60, 63], making a naive direct transfer of these methods unviable.
- (3) *Insufficient interpretability in graph attacks*: Prevailing strategies primarily focus on perturbing node features and modifying the graph topology. However, these modifications often lack a human-centric perspective and suffer from limited interpretability, hindering a deeper understanding of Graph-LLMs’ vulnerabilities and constraining the pursuit of more resilient defense approaches [21, 38].

To address the aforementioned limitations, we propose an Interpretable Multi-Dimensional Graph Attack (IMDGA) framework in this study, with careful consideration of the algorithm’s effectiveness, stealth, and interpretability. To ensure the algorithm aligns more closely with practical scenarios, we adopt a black-box setting [37, 40], wherein the attack leverages only limited information (e.g., model outputs) without any access to the internal parameters of the target model. Within the framework, IMDGA unfolds through three progressively connected stages to achieve effective attacks on TAGs. In the **warm-up stage**, IMDGA introduces the word-level *Topological SHAP Module* to precisely quantify the contributions of salient tokens in graph information propagation [32, 45, 50]. This approach illuminates the semantic weight of each word from a human-centric perspective, allowing us to pinpoint pivotal words that exert varying degrees of influence on node classification predictions. Subsequently, during the **manipulation stage**, the framework incorporates the *Semantic Perturbation Module*, which leverages the contextual understanding of the mask model to generate a diverse pool of semantically plausible substitutes for the previously identified pivotal words [14]. A subsequent phase then

employs a graph-aware scoring function to meticulously evaluate these candidates, quantifying their potential to disrupt predictions based on their topological and semantic relationships within the graph. This refined, two-stage process identifies and applies the optimal substitute, thereby introducing highly targeted perturbations while scrupulously preserving the attack’s stealthiness. To transcend the inherent limitations of existing graph text attacks and to achieve seamless integration with edge-level attack algorithms, we introduce a novel Influence-based *Edge Pruning Module* in the **refinement stage** to identify the most vulnerable edges, which are most susceptible to text-based perturbations [1, 39]. The strategic design of this module serves a dual function: it alleviates the computational bottleneck inherent in the Shapley strategy [33] and refines the precision of the attack. By selectively targeting a minimal yet highly representative subset of samples, it significantly curtails computational overhead, while simultaneously identifying edges that exert a substantial and positive reinforcing influence on target node’s classification confidence. This integrated approach, which strategically leverages the aforementioned three modules, not only advances the stealth and effectiveness of the attack but also provides an unprecedented level of interpretability, offering valuable insights into the most vulnerable components of Graph-LLMs.

The main contributions of our work are summarized as follows:

- ① **A Novel Perspective on TAGs Security**: We present a pioneering and crucial perspective on the security vulnerabilities of TAGs, which arise from the non-decoupled nature of text encoding and GNN message passing. By synthesizing knowledge from both advanced NLP adversarial attacks and graph attack fields, we introduce a new paradigm for security analysis and attacks on TAGs.
- ② **A Unified Multi-Dimensional Attack Framework**: We propose IMDGA, a holistic framework that effectively integrates three novel modules to perform interpretable attacks on both text and edge attributes. This methodology includes a human-centric approach for targeted text substitution and, critically, an edge-level attack strategy that transcends the inherent limitations of text-only perturbations, thereby further optimizing the attack’s potency.
- ③ **Proven Effectiveness and Stealthiness**: The IMDGA method demonstrates exceptional attack success rates under stringent black-box conditions, showcasing robust performance across multiple datasets. The stealthiness of our attack is rigorously validated through empirical results and theoretical analysis, ensuring that the perturbations remain imperceptible while being highly effective.

2 Background and Preliminary

In this section, we will briefly introduce the key concepts and definitions to better explain the fundamentals of our proposed method.

Text-Attributed Graphs. Typically, a text-attributed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{T})$, where \mathcal{V} denotes the set of nodes with $|\mathcal{V}| = n$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges, which can also be represented by the adjacency matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$. Any node $v_i \in \mathcal{V}$ has an associated text description $\mathcal{T}_i \in \mathcal{T}$. The feature matrix $\mathcal{X} \in \mathbb{R}^{n \times d}$, derived by encoding the texts \mathcal{T} , contains the feature vectors for all nodes, and d represents the dimension of node feature. Specifically, we define the label set as $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$, where each label y_i is uniquely associated with a node $v_i \in \mathcal{V}$.

LLM-based Graph Learning. In this section, we elucidate the

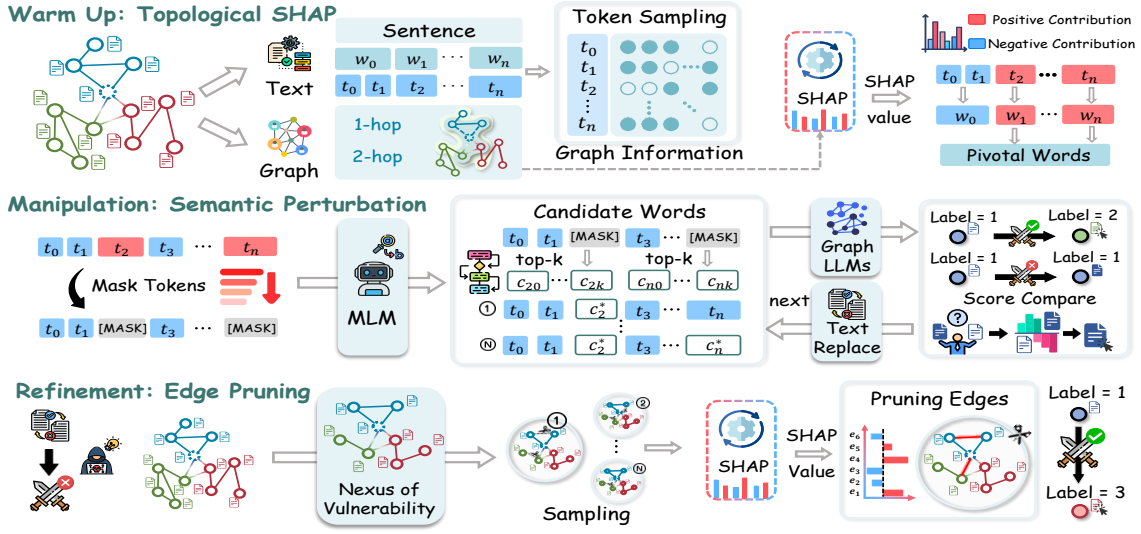


Figure 2: An overview of IMDGA framework, illustrating the key components and methodologies.

workflow of graph learning on TAGs when LLMs serve as enhancers. Broadly, such Graph-LLMs comprise three pivotal components: initialization, aggregation, and updating, which seamlessly integrate textual information with graph-structured data to enable efficient representation learning. For a given node v_i and its associated textual attribute t_i , the transformer-based model generally encodes the raw text into a semantically rich embedding vector to initialize the node representation, expressed as follows:

$$h_{v_i} = x_i, \quad x_i = \psi_\theta(t_i) \in \mathbb{R}^d, \quad \forall v_i \in \mathcal{V}, \quad (1)$$

where ψ_θ denotes the text encoder with parameters θ . To enrich the final representation of node v_i , update operation $f_{Up}^{(k-1)}$ and aggregation function AGGR is employed to consolidate node representations derived from the preceding iteration:

$$h_{v_i}^{(k)} = f_{Up}^{(k-1)}(h_{v_i}^{(k-1)}, \text{AGGR}(\{h_u^{(k-1)}, u \in \mathcal{N}(v_i)\})). \quad (2)$$

Embedded from raw text and then iteratively refined, these node representations capture both rich textual semantics and graph structure information, making them suitable for various downstream tasks such as node classification and link prediction.

Adversarial Attack. The vulnerability of TAGs stems from two principal attack vectors: graph and text adversarial attacks.

Graph adversarial attacks are designed to deceive a GNN \mathcal{F}_θ by subtly altering the original graph \mathcal{G} , causing the model to produce incorrect predictions. This objective can be formally expressed as:

$$\max \mathcal{L}(\mathcal{F}_\theta(\mathcal{G}')), \quad \text{s.t. } \|\mathcal{G}' - \mathcal{G}\| \leq \Delta, \quad (3)$$

where \mathcal{G}' denotes the perturbed graph and Δ denotes the perturbation budgets, containing Δ_A and Δ_X . In GMAs, modifications are restricted to making subtle perturbations on the existing adjacency matrix \mathcal{A} and feature matrix \mathcal{X} . Specifically, these perturbations must satisfy the constraint $\|\mathcal{A}' - \mathcal{A}\|_0 \leq \Delta_A$ and $\|\mathcal{X}' - \mathcal{X}\|_\infty \leq \Delta_X$. For GIAs, the graph structure is expanded by introducing malicious nodes \mathcal{V}_{atk} with corresponding adversarial features \mathcal{X}_{atk} , resulting in $\mathcal{X}' = \begin{pmatrix} \mathcal{X} \\ \mathcal{X}_{\text{atk}} \end{pmatrix}$ and $\mathcal{A}' = \begin{pmatrix} \mathcal{A} & \mathcal{A}_{\text{atk}} \\ \mathcal{A}_{\text{atk}}^T & \mathcal{O}_{\text{atk}} \end{pmatrix}$. Similarly, the number

of attacked nodes $|\mathcal{V}_{\text{atk}}|$, the degree of nodes d_v , and the feature matrix \mathcal{X}_{atk} are bounded to ensure stealthiness.

Text adversarial attacks are commonly employed in tasks such as text classification, aiming to craft adversarial samples for the textual attributes \mathcal{T}^* that significantly diminish the confidence of classifier Φ in accurate predictions, thereby inducing misclassification:

$$\mathcal{T}^* = \arg \min_{\mathcal{T}'} \{-D(\mathcal{T}, \mathcal{T}')\}, \quad \text{s.t. } \Phi(\mathcal{T}') \neq y, \quad (4)$$

where D denotes the semantic similarity. Drawing inspiration from the above adversarial attacks, we propose a novel perspective to combine their advantages, investigating the vulnerabilities of TAGs. **Shapley Value.** SHAP (SHapley Additive exPlanation) is a game-theoretic framework that employs the concept of Shapley values [48, 50] to quantify the contribution of individual players (e.g., feature) in cooperative scenarios. Intuitively, the more pivotal a player is to the prediction, the higher its corresponding Shapley value will be. In general, the Shapley value of a player is computed as the weighted average of all possible marginal contributions that the player provides across different coalitions S :

$$\phi(i) = \sum_{S \subseteq \{1, \dots, n\} \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [f(S \cup \{i\}) - f(S)], \quad (5)$$

where n denotes the total number of players and $f(S \cup \{i\}) - f(S)$ quantifies the marginal contribution of player i to the coalition S . From an explainable perspective, the raw text, features, or edges within TAGs can be seen as collaborative ‘players’ that jointly explain model predictions. This leads to the concept of Topological SHAP, which makes the attack process more interpretable.

3 Methodology

In this section, we present a comprehensive overview of our innovative method, which systematically probes the vulnerabilities of Graph-LLMs with a focus on interpretability. IMDGA introduces a pioneering paradigm for adversarial attacks on representation

learning in TAGs under black-box conditions, achieving precise and efficient targeted manipulations. It strategically accounts for both textual attributes and graph structural dynamics in its attack design, addressing three important challenges through synergistic modules. Specifically, IMDGA first introduces the Topological SHAP to tackle **Challenge ①**: How to pinpoint pivotal words in the raw text that critically influence model predictions from a graph-centric perspective (Section 3.1)? Subsequently, it employs the Semantic Perturbation to address **Challenge ②**: How to execute stealthy, semantically coherent perturbations to these pivotal words (Section 3.2)? Finally, IMDGA proposes the Edge Pruning to achieve more advanced adversarial effects, resolving **Challenge ③**: How to precisely disrupt key message-passing pathways in a human-intuitive manner (Section 3.3)? For clarity, we illustrate the overall framework in Figure 2 and provide pseudocode in Appendix H.

3.1 Topological SHAP (Warm Up)

Words serve as the fundamental building blocks of sentences, embodying independent semantic units that often encapsulate core intent, such as subjects or sentiments. While numerous NLP techniques have pursued word-level textual attacks, the texts in TAGs differ markedly from those in prior studies as they are intrinsically tied to nodes, interconnected via edges, rendering them interdependent rather than isolated passages. Consequently, attacking a single node’s text in isolation without accounting for its connected neighbors fails to fully exploit the graph’s textual attributes. To address this, we introduce the Topological SHAP Module in the warm-up stage, which maps the SHAP framework onto graph structures to quantify word importance from a topological viewpoint:

$$\phi(i) = \sum_{S \subseteq \mathcal{W} \setminus \{W_i\}} \frac{|S|!(|\mathcal{W}| - |S| - 1)!}{|\mathcal{W}|!} [f(\mathcal{T}_S) - f(\mathcal{T}_{S \cup \{W_i\}})], \quad (6)$$

where for a node v , $\mathcal{W} = \{W_i \mid W_i \in \text{Tokenize}(\mathcal{T}_v), i = 1, \dots, m\}$ denotes the subset of words derived from the node’s text \mathcal{T}_v , with W_i representing the i -th word. The masked text \mathcal{T}_S is defined as:

$$\mathcal{T}_S = \{W_i \cdot \mathbb{I}_{\{W_i \notin S\}} + [\text{MASK}] \cdot \mathbb{I}_{\{W_i \in S\}} \mid W_i \in \mathcal{W}\}. \quad (7)$$

Notably, while \mathcal{T}_S is defined as a set, it retains the original text sequence order when used as input. In addition, the coalitional function $f(\mathcal{T}_S)$ is defined as the aggregation of the GNN’s predictive scores over v and its neighborhood $\mathcal{N}(v)$, which serves as the basis for computing the marginal contribution:

$$f(\mathcal{T}_S) = \sum_{u \in \{v\} \cup \mathcal{N}(v)} \mathcal{F}_\theta(\mathcal{G}, \psi_\theta(\mathcal{T}_S), u), \quad \forall v \in \mathcal{V}. \quad (8)$$

In contrast to traditional SHAP, our approach employs a mirrored operation, bypassing the conventional paradigm of incrementally adding features. By masking words, we reformulate the marginal contribution from the additive transition (S to $S \cup \{i\}$) to a subtractive shift ($S \cup \{W_i\}$ to S). The Shapley value $\phi(i)$ quantifies a word’s impact on the classification of both the node and its neighborhood, providing a precise measure of its significance in Graph-LLM predictions. Recognizing that $\phi(i)$ encompasses contributions across all classes, we refine the final score to focus on the sum of its Shapley values for the true labels of relevant nodes $\phi^{yu}(i)$:

$$\xi(i) = \sum_{u \in \{v\} \cup \mathcal{N}(v)} \phi^{yu}(i), \quad \forall v \in \mathcal{V}. \quad (9)$$

Once the importance score $\xi(i)$ of each word within a sentence has been obtained, we further refine the selection to identify the most influential tokens. Accordingly, we define the pivotal word set \mathcal{P} as the top- k words whose $\xi(i)$ values exceed a threshold τ :

$$\mathcal{P} = \{W_i \mid \xi(i) > \tau, i \in \mathcal{I}_k\}. \quad (10)$$

The pivotal word set \mathcal{P} thus encapsulates the most semantically and topologically critical words that govern the prediction behavior of both the target node and its neighbors. By distilling the text into this compact subset of decisive words, we establish a principled foundation for manipulation stage, where carefully designed perturbations can be applied in a targeted and stealthy manner.

3.2 Semantic Perturbation (Manipulation)

Inspired by the limitations of conventional textual attacks, which fail to leverage inter-node message passing to amplify their impact through graph structures, we introduce the adversarial Semantic Substitution Module. This module generates a diverse pool of semantically plausible substitutes for the pivotal words in \mathcal{P} , meticulously balancing maximal disruption of model predictions with minimal surface-level detectability. In addition, it ingeniously transforms a Masked Language Model (MLM) into a context-aware "semantic proxy," transcending the limitations of traditional static substitutions. Crucially, since each encoder corresponds to its own specialized MLM, the resulting candidate words are highly aligned with the model’s internal semantic representation. Drawing from MLM’s pre-training objective, which maximizes the product of conditional probabilities for masked tokens given surrounding context:

$$\prod_{i=1}^m P(W_i \mid W_1, \dots, W_{i-1}, W_{i+1}, \dots, W_m). \quad (11)$$

Building upon this principle, our framework exploits such contextual dependency modeling to synthesize semantically coherent and contextually plausible candidate substitutions. Taking node v as an example, we define its candidate word set \mathcal{C} as the union of top- k replacements for each pivotal word $W_i \in \mathcal{P}$, where each pivotal word generates multiple semantically proximate candidates to ensure diversity and contextual adaptability. For each candidate $r \in \mathcal{C}$, we generate the perturbed text $\mathcal{T}' = (\mathcal{T} \setminus \{W_i\}) \cup \{r\}$. Subsequently, we compute GNN’s predictive probability distribution $p_u(\mathcal{G}, \mathcal{T}')$ for nodes $u \in \{v\} \cup \mathcal{N}(v)$. The confidence gap is defined as:

$$\delta_u(r) = p_u^{(1)}(\mathcal{G}, \mathcal{T}') - p_u^{(2)}(\mathcal{G}, \mathcal{T}'), \quad (12)$$

where $p_u^{(1)}$ and $p_u^{(2)}$ denote the largest and the second-largest predicted probabilities, respectively, quantifying the model’s decision certainty for node u post-replacement. Evidently, a smaller confidence gap indicates a more ambiguous decision boundary for the model, thereby rendering it more vulnerable to adversarial attacks. Therefore, aggregating these gaps across the neighborhood yields the base score for candidate word r :

$$\Delta(r) = \sum_{u \in \{v\} \cup \mathcal{N}(v)} \delta_u(r). \quad (13)$$

To incorporate adaptive characteristics, we introduce the label-flip indicator function $\mathbb{I}_{\text{flip}}(r)$, which is 1 if the replacement induces a

prediction label flip for node v or its neighborhood, and 0 otherwise. The final replacement score is defined as:

$$\sigma(r) = \Delta(r) \cdot (1 + \alpha \cdot \mathbb{I}_{\text{flip}}(r)), \quad (14)$$

where $\alpha > 0$ is a hyperparameter that dynamically amplifies the confidence gap weight in label-flip scenarios, prioritizing high-impact replacements. The aggregated score $\Delta(r)$ ensures a comprehensive neighborhood evaluation, while the adaptive factor $(1 + \alpha \cdot \mathbb{I}_{\text{flip}}(r))$ elegantly highlights boundary perturbations, balancing stability and sensitivity analysis. This innovative mechanism implicitly reinforces the graph interactions through neighborhood aggregation, endows the attack with greater strategic depth and robustness.

3.3 Edge Pruning (Refinement)

Although textual attacks effectively disrupt node representations by perturbing high-contribution words in node texts, their efficacy is limited in scenarios where graph topology strongly dominates GNN predictions. To address this, we propose an interpretable Edge Pruning Module as a strategic extension of textual perturbations, activated only when textual disruptions fail to induce target label flips. The core of this module lies in our innovative concept, nexus of vulnerability, which represents a curated subset of nodes highly intertwined with the target node v and inherently susceptible to attacks. We identify this nexus through a robust multifaceted scoring mechanism that elegantly fuses three complementary dimensions: predictive disparity, feature influence, and vertex centrality. Drawing from the confidence gap $\delta_u(r)$ in Eq. (12), we similarly define predictive disparity $\delta(u)$ for node u , quantifying decision boundary ambiguity. Critically, feature influence is derived from message propagation dynamics, measured via the L1 norm of the expected Jacobian for node v 's impact on u after k layers:

$$I(u, v, k) = \left\| \mathbb{E} \left[(\partial X_u^{(k)}) / (\partial X_v^{(0)}) \right] \right\|_1, \quad (15)$$

where $X_u^{(k)}$ denotes node u 's feature after the k -th layer, and $X_v^{(0)}$ is node v 's initial feature embedding. Normalized, it yields:

$$I_u(v, k) = \frac{I(u, v, k)}{\sum_{w \in V} I(u, w, k)}, \quad (16)$$

reflecting node v 's relative contribution to node u 's representation. This allows us to quantify the feature influence from target node v to other nodes. Typically, nodes with lower degrees, owing to their reduced structural redundancy, are more prone to amplified perturbation effects. To capture this property, we incorporate $\frac{1}{\deg(u)}$ as a surrogate indicator of structural centrality and attack susceptibility, thereby forming the final term in the computation of the vulnerability score. We consolidate the above dimensions into a unified vulnerability scoring function, formally defined as:

$$\text{Score}(u) = \alpha_1 \cdot (1 - \delta(u)) + \alpha_2 \cdot I_u + \alpha_3 \cdot \left(\frac{1}{\deg(u)} \right), \quad (17)$$

Nodes with the highest scores form the nexus of vulnerability $\mathcal{G}_n(v)$, constraining the attack domain to a high-impact subgraph. On this basis, we delineate critical paths linking nodes in $\mathcal{G}_n(v)$ to v , and employ an efficient Shapley value approximation inspired by GNNShap, cast as a least-squares solution:

$$\hat{\phi} = (M^T U M)^{-1} M^T U \hat{y}, \quad (18)$$

where $M \in \mathbb{R}^{k \times n}$ is the mask matrix encoding k subgraph samples across n edges, U is a diagonal weight matrix reflecting sample importance and \hat{y} approximates nexus predictions under masked configurations. Finally, we prune the top- k edges with the highest attributions, further eroding model confidence in node's original label. This interpretable pruning mechanism not only complements textual attacks but also uncovers topological vulnerabilities, forging a multi-dimensional adversarial strategy that advances the frontier of explainable TAGs attacks.

3.4 Time Complexity Analysis

To substantiate the scalability of IMDGA, we provide a theoretical analysis highlighting how interpretability is achieved without sacrificing computational efficiency. First, the exponential cost of exact Shapley computation is alleviated through partition-based sampling: for a node with m tokens, the coalition space is reduced to $s \ll 2^m$. Denoting by t_g the inference time of the underlying Graph-LLM, the complexity of the Topological SHAP Module becomes $O(s \cdot t_g)$. For token substitution, the Semantic Perturbation bounds computation by restricting to $|\mathcal{P}|$ pivotal tokens, each producing k_c semantically coherent candidates; since the masked language model requires only one forward pass, the cost is $O(k_c |\mathcal{P}| \cdot t_g + t_m)$, where t_m is the MLM inference time. Finally, the Edge Pruning Module narrows the sampling space to a subgraph $\mathcal{G}_n(v)$ around the target node and further samples $k \ll 2^{|\mathcal{N}(v)|}$ coalitions within this restricted domain, which not only limits the computational scope but also improves the precision of Shapley-based attribution, yielding a complexity of $O(k \cdot t_g)$. Integrating these components, the overall per-node complexity converges to $O((s + q + k_c |\mathcal{P}|) \cdot t_g + t_m)$, demonstrating that the complexity of IMDGA increases slowly with the scale of the dataset, being primarily constrained by the underlying Graph-LLMs and the chosen sampling hyperparameters.

4 Experiments

In this section, we conduct a comprehensive evaluation of the proposed IMDGA framework to demonstrate its effectiveness, interpretability, stealthiness, and robustness on TAGs. To rigorously assess these attributes, we structure our experiments around following research questions: **Q1**: To what extent does IMDGA method surpass traditional approaches in executing effective attacks on TAGs? **Q2**: Can our method optimally reconcile the trade-off between attack stealthiness and effectiveness? **Q3**: Does IMDGA maintain robust attack performance when facing different Graph-LLM architectures? **Q4**: What distinct roles do the individual modules play in bolstering the overall performance of the IMDGA framework? **Q5**: How significantly do critical hyperparameters influence the robustness and adaptability of our attack strategy on TAGs?

4.1 Experimental Setup

Datasets. We evaluate our proposed method on several widely used benchmark datasets. Specifically, we adopt Cora, Citeseer, and PubMed [64], which are among the most frequently used citation datasets. Beyond these commonly used datasets, we further incorporate the large-scale ogbn-arxiv [20] dataset to assess the scalability and practicality of our approach in more challenging, real-world scenarios. Detailed dataset statistics are provided in Appendix A.

Table 1: Comprehensive Comparison of Adversarial Attack Effectiveness Across TAG Datasets.

Dataset	Methods	SBERT		BERT		RoBERTa		DeBERTa		DistilBERT	
		ACC ↓	ASR ↑	ACC ↓	ASR ↑	ACC ↓	ASR ↑	ACC ↓	ASR ↑	ACC ↓	ASR ↑
Cora	Clean	82.14±1.71	-	80.61±1.85	-	77.48±0.91	-	76.19±1.16	-	81.31±0.06	-
	HLBB	76.97±0.98	23.33±0.53	76.42±0.86	32.41±1.94	73.70±1.05	24.54±0.36	74.80±1.06	9.72±1.07	78.22±1.15	23.15±0.78
	TextHoaxer	78.45±1.35	16.68±0.61	76.42±1.92	32.41±0.08	73.70±1.17	24.54±0.55	74.80±1.04	9.72±1.88	78.22±0.33	23.15±1.49
	SemAttack	81.13±0.29	4.44±0.15	79.42±0.91	6.05±1.14	76.65±0.58	5.09±0.13	75.36±0.36	6.48±1.07	80.30±1.96	6.48±0.43
	FGSM	71.02 ±1.63	43.70±0.49	70.37±1.97	57.21±0.11	64.61±1.42	63.43±0.78	69.17±0.25	45.37±1.51	70.47±0.89	61.11±1.06
	PA-F	63.68 ±1.22	70.74±0.76	71.94±1.91	46.05±0.37	64.28±0.85	61.11±1.45	67.60±0.04	48.61±1.19	75.50±1.58	37.96±0.52
	IMDGA	61.51±1.73	95.19±0.02	64.33±1.48	94.42±0.87	61.98±0.55	94.44±1.94	61.24±0.33	95.83±1.21	65.25±0.69	93.06±1.08
Citeseer	Clean	71.58±1.16	-	72.54±0.50	-	71.48±0.82	-	70.26±0.91	-	72.85±0.37	-
	HLBB	64.14±1.05	43.08±0.67	67.28±1.93	40.16±0.18	69.16±1.34	18.50±0.42	66.42±0.27	26.38±1.58	69.48±1.77	29.13±0.96
	TextHoaxer	66.65±1.41	27.36±0.98	69.75±0.25	23.23±1.80	69.83±0.07	14.57±1.63	68.73±0.56	11.02±1.97	70.46±0.39	20.47±1.11
	SemAttack	70.26±1.25	7.23±0.78	71.32±1.91	8.66±0.33	71.17±0.62	3.15±1.06	68.07±1.54	15.35±0.09	71.95±1.77	9.06±0.48
	FGSM	59.83 ±1.04	58.81±0.61	64.26±1.98	51.97±0.15	61.91±1.42	61.02±0.73	63.52±0.36	43.31±1.55	64.73±0.89	55.51±1.22
	PA-F	68.77±1.33	14.15±0.69	69.95±1.92	19.29±0.18	62.18±1.04	58.27±0.77	61.75±0.25	55.51±1.55	69.16±0.48	27.95±1.87
	IMDGA	53.82±1.04	92.77±0.61	58.65±1.98	94.49±0.15	57.94±1.42	93.31±0.73	56.89±0.36	93.31±1.55	59.40±0.89	94.10±1.22
Pubmed	Clean	82.31±1.45	-	82.31±0.78	-	83.71±0.29	-	82.62±1.94	-	82.04±0.05	-
	HLBB	81.12±1.73	27.67±0.02	81.25±1.48	30.00±0.87	82.95±0.55	25.80±1.94	82.03±0.33	18.60±1.21	81.29±0.69	23.80±1.08
	TextHoaxer	81.15±1.19	28.67±0.58	81.27±1.92	28.20±0.03	82.88±1.35	25.00±0.81	82.03±0.46	16.80±1.64	81.33±0.99	22.00±1.87
	SemAttack	81.93±1.04	16.00±0.61	82.10±1.98	5.40±0.15	83.54±1.42	5.40±0.73	82.43±0.36	17.20±1.55	81.97±0.89	4.00±0.22
	FGSM	80.27±1.19	64.33±0.58	78.65 ±1.92	69.00±0.03	80.67±1.35	72.40±0.81	81.67±0.46	35.20±1.64	78.59±0.99	57.40±1.87
	PA-F	80.20±1.04	13.29±0.61	80.76±1.98	29.40±0.15	80.65±1.42	51.00±0.73	81.70±0.36	32.60±1.55	80.67±0.89	17.80±1.22
	IMDGA	77.62±1.19	85.88±0.58	77.84±1.92	86.40±0.03	80.17±1.35	84.40±0.81	79.35±0.46	77.40±1.64	77.88±0.99	86.00±1.87

Compared Baselines. To ensure a fair and comprehensive comparison of adversarial performance on TAGs, we select baselines from both the text adversarial attack domain and the graph adversarial attack domain (see Appendix B). From the former category, we select HLBB [37], TextHoaxer [65], and SemAttack [56], adapting them to TAGs to evaluate their transferability from text-only tasks to TAGs settings. From the latter, we adopt FGSM [15] and PA-F [35], two representative approaches in graph adversarial attack, which enable a direct evaluation against graph-specific perturbations.

Evaluation. Since our attack primarily targets the node classification task, we evaluate effectiveness from both a global perspective and a local perspective. From the global view, we measure the overall classification ACC on the test set, which reflects how the attack influences the general predictive capability of the victim model. From the local view, we adopt the ASR on originally correctly classified nodes, providing a more fine-grained measure of how effectively the attack disrupts predictions at the target node. For more detailed experimental settings, refer to the Appendix C.

4.2 Performance Comparison (Q1)

Our experimental results consistently demonstrate the superior effectiveness and robustness of the proposed method across both local and global perspectives, substantially outperforming existing baselines and exposing the vulnerability of Graph-LLMs to adversarial manipulations. Through comprehensive data analysis, we derive the following key observations: (1) **Comprehensive SOTA performance:** At the local level, our strategy consistently surpasses all competing methods, producing the strongest disruptive effect on target nodes across every dataset and Graph-LLM

variant. On smaller benchmarks such as Cora and Citeseer, the ASR exceeds 90%, showing that nearly all targeted nodes can be reliably compromised. Even on larger-scale datasets like ogbn-arxiv, where adversarial robustness is typically higher, our method still demonstrates clear superiority over existing baselines. From the global perspective, although the degree of degradation is influenced by factors such as node selection strategy and perturbation scale, under a unified random selection protocol our approach still achieves significantly stronger global disruption than any baseline. (2) **Overcoming prior limitations:** Traditional NLP adversarial attacks, despite their success in text classification, exhibit severely diminished transferability in the context of TAGs. Even in its best case, HLBB achieves only 40.16% ASR on Citeseer (with BERT encoding), while in other scenarios certain methods yield ASR values dropping below 10%. Meanwhile, graph-specific adversarial baselines such as FGSM and PA-F perform better under the same similarity constraints, but remain limited to embedding-level manipulations without semantic interpretability. By contrast, our method not only achieves stronger quantitative performance but also introduces meaningful and explainable perturbations, thereby addressing the twofold shortcomings of insufficient transferability in text-based attacks and lack of interpretability in graph-based ones.

4.3 Stealthiness Evaluation (Q2)

In this section, we evaluate stealthiness from two perspectives aligned with IMDGA’s modules. For the textual perspective, we select BERT cosine similarity (Sim), GPT-2 perplexity (PPL) [43], and human ratings (see Appendix F) as our main metrics. From the structural perspective, we report both degree distributions and

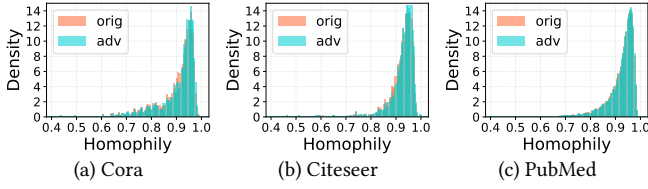


Figure 3: Feature-cosine homophily (before and after attack).

Table 2: Textual stealthiness.

Method	Cora			Citeseer		
	PPL	Sim	Human	PPL	Sim	Human
HLBB	638.35	0.917	3.254	574.46	0.934	3.391
TextHoaxer	618.55	0.914	3.290	580.38	0.955	3.326
SemAttack	498.46	0.954	3.486	209.99	0.958	3.493
IMDGA	196.74	0.955	3.565	186.18	0.960	3.529

the feature-cosine homophily. Specifically, the feature-cosine homophily of each node u is formally defined as:

$$h_u = \cos(r_u, x_u), \quad r_u = \sum_{j \in N(u)} \frac{1}{\sqrt{d_u d_j}} x_j, \quad (19)$$

where $N(u)$ denotes the immediate neighbor set of u , d_u indicates its structural degree, and x_u characterizes its node feature vector. In particular, r_u represents the normalized aggregation of neighbor features. Structural stealthiness is measured by Δh_u and Δd_u , where smaller deviations imply higher stealthiness.

Under the same perturbation ratio, IMDGA consistently achieves higher Sim and lower PPL than text-only baselines, as clearly shown in Table 2, which is consistent with human ratings and further indicates better semantic preservation and fluency.

The *Semantic Substitution Module* explains these improvements. Eq. (11) employs an MLM to propose context-aware candidates, while Eqs. (12)–(14) evaluate replacements via neighborhood confidence gaps with an adaptive label-flip factor. If a fraction ρ of tokens is replaced with candidates under an MLM similarity threshold γ , then with ℓ_2 -normalized embeddings we obtain:

$$\|h' - h\| \leq \rho \sqrt{2(1 - \gamma)} \Rightarrow \cos(h', h) \geq 1 - \rho^2(1 - \gamma). \quad (20)$$

This inequality establishes a lower bound on similarity with small ρ and large γ . Moreover, replacements with sufficiently high relative likelihood incur only a limited increase in negative log-likelihood (NLL), hence a mild rise in PPL. Together with the adaptive scoring in Eq. (14), module favors substitutions that are both impactful and linguistically plausible, yielding effective yet stealthy perturbations.

The homophily distributions after the *Edge Pruning Module* almost coincide with the originals, with minor tail deviations, as shown in Figure 3. This arises from pruning high-attribution edges with a small budget (top- k in Eq. (18)) and degree-normalized aggregation in Eq. (19). Deleting an edge (u, v) updates $r'_u = r_u - \frac{x_v}{\sqrt{d_u d_v}}$, and for unit-normalized features a first-order bound gives:

$$|\cos(r'_u, x_u) - \cos(r_u, x_u)| \lesssim \frac{1}{\sqrt{d_u d_v}} \cdot \frac{\|x_v\|}{\|r_u\|}, \quad (21)$$

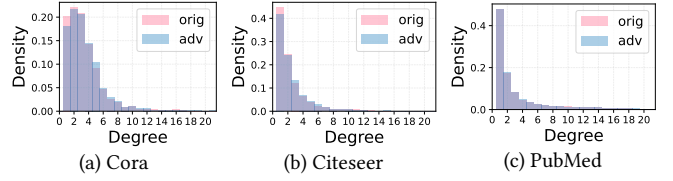


Figure 4: Degree distributions (before and after attack).

so the change in h_u is constrained by degree normalization and the limited number of pruned edges.

The degree distributions, as clearly shown in Figure 4, preserve the head and body of the original curves with only negligible tail differences, thereby indicating that the global topology is statistically indistinguishable from the original. Let $|\Delta E|$ be the number of pruned edges. Since each pruned edge decreases the degrees of its two endpoints by one, we have:

$$\frac{1}{|V|} \sum_u |d'_u - d_u| = \frac{2|\Delta E|}{|V|}, \quad (22)$$

and node-wise caps together with the locality of the nexus $G_n(v)$ (Eq. (17)) keep $|\Delta E|/|V|$ small, explaining the near overlap.

Table 3: Generalizable Robustness and Effectiveness of IMDGA Across Multiple Backbones (ASR).

Encoder	Backbone	Cora	Citeseer	PubMed
SBERT	GCN	92.19 \pm 0.84	94.87 \pm 0.42	87.22 \pm 1.02
	GAT	83.58 \pm 0.67	93.48 \pm 1.35	90.72 \pm 0.64
	SAGE	95.21 \pm 1.89	92.74 \pm 0.58	88.79 \pm 1.05
	RGCN	89.04 \pm 0.35	91.19 \pm 1.81	84.66 \pm 0.92
	Guard	93.92 \pm 1.76	98.12 \pm 0.36	90.12 \pm 1.58
	GLEM	88.73 \pm 0.54	94.55 \pm 0.43	90.11 \pm 0.47
	GIANT	87.19 \pm 0.98	93.67 \pm 1.24	84.39 \pm 1.26
BERT	GCN	92.37 \pm 1.42	95.63 \pm 0.72	85.42 \pm 1.37
	GAT	90.12 \pm 0.74	93.26 \pm 1.18	91.73 \pm 0.72
	SAGE	89.77 \pm 1.63	92.41 \pm 0.66	89.04 \pm 0.92
	RGCN	87.03 \pm 0.42	91.86 \pm 1.64	85.13 \pm 1.08
	Guard	86.47 \pm 1.59	97.89 \pm 0.41	91.28 \pm 1.33
	GLEM	89.21 \pm 0.61	94.32 \pm 0.49	90.54 \pm 0.53
	GIANT	87.68 \pm 1.05	93.24 \pm 1.36	84.92 \pm 1.12

4.4 Attack Robustness Assessment (Q3)

Having demonstrated the effectiveness and stealthiness of IMDGA in previous sections, we further investigate its generalization and robustness through extensive experiments. To verify that our attack is not limited to a specific GNN architecture, we evaluate IMDGA on widely-used models such as GCN [24], GAT [55], and GraphSAGE [16], as well as on more robust architectures including RGCN [46] and GNNGuard [66]. For Graph-LLMs, although Q1 compared multiple encoder models, the attack performance on more advanced Graph-LLMs remains unexplored. Therefore, we conduct comprehensive and rigorous experiments using GLEM [67] and GIANT [10] as representative Graph-LLMs.

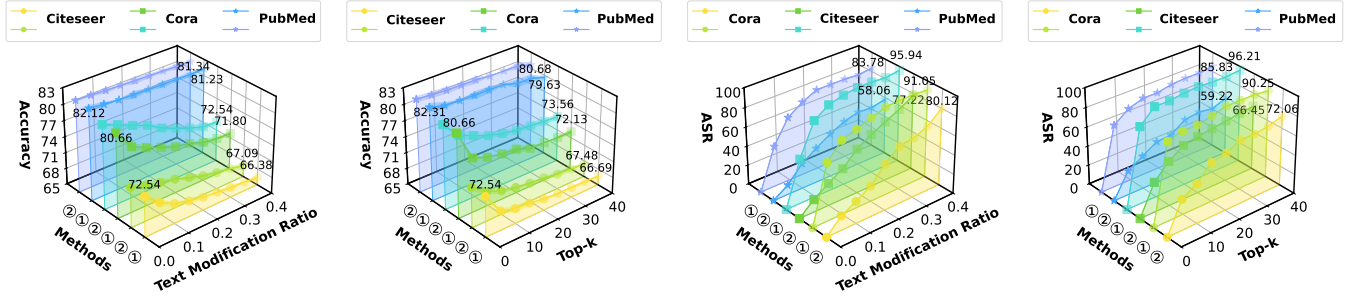


Figure 5: Sensitivity of IMDGA (① represents the full IMDGA and ② denotes text-only perturbations).

Table 4: Ablation study (ASR).

Model	Cora	Citeseer	PubMed
IMDGA (SBERT)	91.78±0.72	94.86±0.79	85.47±0.41
w/o Topology SHAP	29.54±0.42	32.89±0.29	23.76±0.58
w/o Semantic Perturb	36.97±0.53	41.21±0.19	32.84±0.44
w/o Edge Pruning	66.12±0.35	62.47±0.33	59.21±0.55
IMDGA (BERT)	92.06±0.61	95.11±0.87	85.16±0.35
w/o Topology SHAP	29.13±0.38	33.18±0.21	23.41±0.63
w/o Semantic Perturb	37.32±0.46	40.93±0.16	33.21±0.38
w/o Edge Pruning	65.87±0.30	62.23±0.26	58.83±0.62

The results presented in Table 3 indicate that all tested models exhibit relatively weak resistance to IMDGA, with only minor differences in ASR across architectures. Notably, even the robust Guard model achieves an ASR of 98.12% on Citeseer, suggesting that while these backbones leverage node similarity to enhance robustness, they remain vulnerable to attacks exploiting textual information. Similarly, GLEM and GIANT, which employ fine-tuned encoders to incorporate richer node representations, do not demonstrate improved resistance against our proposed adversarial attack. These findings collectively highlight that IMDGA consistently compromises diverse GNNs and Graph-LLMs, underscoring the persistent vulnerabilities present across modern graph-based models.

4.5 Ablation Study (Q4)

To evaluate the contribution of each component in our adversarial framework, we conduct ablation experiments and report the results in Table 4. For the Topological SHAP Module, we replace the identified pivotal words with randomly selected words for comparison. For the Semantic Perturbation Module, while it is infeasible to completely remove the effect of the MLM, we substitute the graph-aware scoring function originally designed to integrate both structural and textual information with a simpler function that only considers the prediction probability drop of a single node. For the Edge Pruning Module, we directly eliminate it, leaving only text-based perturbations in the algorithm. It is worth noting that, to preserve stealthiness, we did not apply the Edge Pruning strategy when ablating the first two modules.

From the analysis of the results, several conclusions can be drawn: (1) Text modification modules are critical to attack success. When

perturbations are restricted to naive random word substitutions or applied without incorporating graph structural information, the ASR sharply drops from around 92% to below 40%. Such simplified attacks are insufficient to substantially compromise the robustness of Graph-LLMs. By contrast, precisely identifying pivotal words in message passing and applying carefully designed graph-aware substitutions leads to a stronger adversarial influence on TAGs.

(2) Word-level perturbations alone face inherent limitations. Even when pivotal words are correctly identified and semantically reasonable substitutions are introduced, the ASR remains capped at around 60%. Without the complementary structural guidance provided by edge-level manipulations, attacks struggle to overcome the inherent robustness of representation learning in TAGs.

4.6 Hyperparameter Analysis (Q5)

To address Q5, we conducted extensive experiments focusing on two pivotal hyperparameters: the text modification ratio β and the top- k selection parameter. These two factors largely govern the strength of adversarial perturbations and thus serve as the most representative indicators for sensitivity analysis. In contrast, other hyperparameters play a more secondary role and are discussed in detail, along with their corresponding search spaces, in Appendix D. For experimental efficiency and comparability, we randomly selected 100 nodes as target instances in each trial, ensuring stable and reliable evaluation across different settings.

The results, summarized in Figure 5, reveal a consistent and interpretable trend: as β and top- k increase, both ACC and ASR undergo a rapid initial change, followed by a clear attenuation in marginal gains. This observation suggests that moderate relaxation of these constraints brings substantial improvements in attack effectiveness at first, as models quickly become more vulnerable under stronger perturbations. However, further increases lead to diminishing returns, with only marginal benefits despite greater perturbation. More critically, loosening these hyperparameters significantly raises the risk of exposure, as excessive modifications are more detectable, while also incurring higher computational and temporal overhead. Consequently, there exists an inherent trade-off among effectiveness, stealth, and efficiency when choosing hyperparameters. Practically, this trade-off implies that optimal settings for β and top- k should be chosen according to the deployment scenario—prioritizing higher stealth for stealth-sensitive applications and higher strength where maximal disruption is required.

5 Conclusion

In this work, we introduced IMDGA to investigate the heightened vulnerabilities of TAGs induced by the integration of textual features into Graph-LLMs. By jointly leveraging Topological SHAP, Semantic Perturbation, and Edge Pruning, IMDGA orchestrates multi-layered adversarial manipulations that expose weaknesses

in both textual and structural dimensions. Through extensive evaluations, our method consistently outperforms conventional NLP and graph adversarial attack baselines in terms of interpretability, effectiveness, and stealthiness, achieving high ASR across diverse datasets. These findings not only uncover fundamental fragilities in TAG representation learning but also underscore the urgent need for systematic defenses to guide the development of more resilient Graph-LLMs against increasingly sophisticated adversarial threats.

References

- [1] Selahattin Akkas and Ariful Azad. 2024. Gnnshap: Scalable and accurate gnn explanation using shapley values. In *Proceedings of the ACM Web Conference 2024*. 827–838.
- [2] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *International Conference on Learning Representations, ICLR*.
- [3] Anirban Chakraborty, Manar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069* (2018).
- [4] Liang Chen, Jintang Li, Jiaying Peng, Tao Xie, Zengxu Cao, Kun Xu, Xiangnan He, Zibin Zheng, and Bingzhe Wu. 2020. A survey of adversarial learning on graphs. *arXiv preprint arXiv:2003.05730* (2020).
- [5] Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. 2024. Llag: Large language and graph assistant. *arXiv preprint arXiv:2402.08170* (2024).
- [6] Yongqiang Chen, Han Yang, Yonggang Zhang, Kaili Ma, Tongliang Liu, Bo Han, and James Cheng. 2022. Understanding and improving graph injection attack by promoting unnoticeability. *arXiv preprint arXiv:2202.08057* (2022).
- [7] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter* 25, 2 (2024), 42–61.
- [8] Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. 2023. Label-free node classification on graphs with large language models (llms). *arXiv preprint arXiv:2310.04668* (2023).
- [9] Eli Chien, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, Jiong Zhang, Olga Milenkovic, and Inderjit S Dhillon. 2021. Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064* (2021).
- [10] Eli Chien, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, Jiong Zhang, Olga Milenkovic, and Inderjit S Dhillon. 2022. Node Feature Extraction by Self-Supervised Multi-scale Neighborhood Prediction. In *International Conference on Learning Representations*.
- [11] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial attack on graph structured data. In *International conference on machine learning*. PMLR, 1115–1124.
- [12] Michal Defferrard, X. Bresson, and P. Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *Advances in Neural Information Processing Systems, NeurIPS* (2016).
- [13] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein Interface Prediction using Graph Convolutional Networks. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.
- [14] Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970* (2020).
- [15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
- [16] Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems, NeurIPS* (2017).
- [17] Zellig S Harris. 1954. Distributional structure. *Word* 10, 2-3 (1954), 146–162.
- [18] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2023. Harnessing explanations: Llm-to-llm interpreter for enhanced text-attributed graph representation learning. *arXiv preprint arXiv:2305.19523* (2023).
- [19] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 639–648.
- [20] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems, NeurIPS* (2020).
- [21] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review* 37 (2020), 100270.
- [22] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2024. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering* (2024).
- [23] Di Jin, Zhijiang Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 8018–8025.
- [24] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations, ICLR*.
- [25] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.
- [26] Runlin Lei, Yuwei Hu, Yuchen Ren, and Zhewei Wei. 2024. Intruding with words: Towards understanding graph injection attacks at the text level. *Advances in Neural Information Processing Systems* 37 (2024), 49214–49251.
- [27] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984* (2020).
- [28] Quan Li, Xiaoting Li, Lingwei Chen, and Dinghao Wu. 2022. Distilling knowledge on text graph for social media attribute inference. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024–2028.
- [29] Xuefeng Li, Yang Xin, Chensu Zhao, Yixian Yang, and Yuling Chen. 2020. Graph Convolutional Networks for Privacy Metrics in Online Social Networks. *Applied Sciences* 10, 4 (2020).
- [30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [31] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*. 417–428.
- [32] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).
- [33] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. Curran Associates, Inc. <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>
- [34] Yuankai Luo, Lei Shi, and Xiao-Ming Wu. 2024. Classic GNNs are Strong Baselines: Reassessing GNNs for Node Classification. *arXiv preprint arXiv:2406.08993* (2024).
- [35] Jiaqi Ma, Shuangrui Ding, and Qiaozhu Mei. 2020. Towards more practical adversarial attacks on graph neural networks. *Advances in neural information processing systems* 33 (2020), 4756–4766.
- [36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).
- [37] Jiaming Mu, Binghui Wang, Qi Li, Kun Sun, Mingwei Xu, and Zhuotao Liu. 2021. A hard label black-box adversarial attack against graph neural networks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 108–125.
- [38] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences* 116, 44 (2019), 22071–22080.
- [39] Maximilian Muschalik, Fabian Fumagalli, Paolo Frazzetto, Janine Strother, Luca Hermes, Alessandro Sperduti, Eyke Hüllermeier, and Barbara Hammer. 2025. Exact Computation of Any-Order Shapley Interactions for Graph Neural Networks. *arXiv preprint arXiv:2501.16944* (2025).
- [40] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.
- [41] Yijian Qin, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2023. Disentangled representation learning with large language models for text-attributed graphs. *arXiv preprint arXiv:2310.18152* (2023).
- [42] Zongshuai Qu, Tao Yao, Xinghui Liu, and Gang Wang. 2023. A Graph Convolutional Network Based on Univariate Neurodegeneration Biomarker for Alzheimer’s Disease Diagnosis. *IEEE Journal of Translational Engineering in Health and Medicine* (2023).
- [43] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
- [44] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410
- [45] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why should i trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.
- [46] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, 593–607.
- [47] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

- [48] Lloyd S Shapley et al. 1953. A value for n-person games. (1953).
- [49] Yash Sharma. 2018. Gradient-based Adversarial Attacks to Deep Neural Networks in Limited Access Settings. *THE COOPER UNION ALBERT NERKEN SCHOOL OF ENGINEERING* (2018).
- [50] Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems* 41, 3 (2014), 647–665.
- [51] Daohan Su, Bowen Fan, Zhi Zhang, Haoyan Fu, and Zhida Qin. 2024. DCL: Diversified Graph Recommendation With Contrastive Learning. *IEEE Transactions on Computational Social Systems* (2024).
- [52] Henan Sun, Xunkai Li, Zhengyu Wu, Daohan Su, Rong-Hua Li, and Guoren Wang. 2023. Breaking the Entanglement of Homophily and Heterophily in Semi-supervised Node Classification. *arXiv preprint arXiv:2312.04111* (2023).
- [53] Lichao Sun, Yingdong Dou, Carl Yang, Kai Zhang, Ji Wang, Philip S Yu, Lifang He, and Bo Li. 2022. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 7693–7711.
- [54] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. 2020. Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach. In *Proceedings of the Web Conference 2020*. 673–683.
- [55] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations, ICLR*.
- [56] Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. 2022. SemAttack: Natural textual attacks via different semantic spaces. *arXiv preprint arXiv:2205.01287* (2022).
- [57] Jihong Wang, Minnan Luo, Fnu Suyu, Jundong Li, Zijiang Yang, and Qinghua Zheng. 2020. Scalable attack on graph data by injecting vicious nodes. *Data Mining and Knowledge Discovery* 34, 5 (2020), 1363–1389.
- [58] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (2020), 396–413.
- [59] Zhouxia Wang, Tianshui Chen, Jimmy Ren, Weihao Yu, Hui Cheng, and Liang Lin. 2018. Deep Reasoning with Knowledge Graph for Social Relationship Understanding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 1021–1028.
- [60] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2021), 4–24. doi:10.1109/TNNLS.2020.2978386
- [61] Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. 2021. Graph Learning: A Survey. *IEEE Transactions on Artificial Intelligence* 2, 2 (April 2021), 109–127. doi:10.1109/taai.2021.3076021
- [62] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. 2019. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214* (2019).
- [63] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? *International Conference on Learning Representations, ICLR*.
- [64] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *International Conference on Machine Learning, ICML*.
- [65] Muchao Ye, Chenglin Miao, Ting Wang, and Fenglong Ma. 2022. Texthoaxer: Budgeted hard-label adversarial attacks on text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 3877–3884.
- [66] Xiang Zhang and Marinka Zitnik. 2020. GnnGuard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems* 33 (2020), 9263–9275.
- [67] Jianan Zhao, Meng Qu, Chaozhao Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2023. Learning on Large-scale Text-attributed Graphs via Variational Inference. In *The Eleventh International Conference on Learning Representations*.
- [68] Qinkai Zheng, Xu Zou, Yuxiao Dong, Yukuo Cen, Da Yin, Jiarong Xu, Yang Yang, and Jie Tang. 2021. Graph robustness benchmark: Benchmarking the adversarial robustness of graph machine learning. *arXiv preprint arXiv:2111.04314* (2021).
- [69] Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. 2024. Efficient tuning and inference for large language models on textual graphs. *arXiv preprint arXiv:2401.15569* (2024).
- [70] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, 13 (06 2018), 1457–1466.
- [71] Xu Zou, Qinkai Zheng, Yuxiao Dong, Xinyu Guan, Evgeny Kharlamov, Jialiang Lu, and Jie Tang. 2021. Tdgia: Effective injection attacks on graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2461–2471.
- [72] Daniel Zügner and Stephan Günnemann. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations, ICLR*.

A Datasets

Experiments are carried out on several widely used, representative text-attributed graph datasets (Table 5). In these datasets, each node is associated with a short textual description (e.g., paper title or abstract, product description), and edges encode citations. Node features are obtained by encoding node texts with our encoding model. We adopt stratified splits of 10%/10%/80% (train/val/test) for Cora, Citeseer, and PubMed, and 20%/20%/60% for ogbn-arxiv.

Table 5: Dataset statistics.

Dataset	#Nodes	#Edges	#Classes	Avg. Degree
Cora	2,708	10,556	7	3.90
Citeseer	3,186	8,450	6	2.65
PubMed	19,717	88,648	3	4.50
ogbn-arxiv	169,343	2,315,598	40	6.89

B Attack Methods

Two families of baselines are considered on text-attributed graphs: text-side attacks that edit node texts and feature-side attacks that perturb continuous text-derived features. For fair and stealthy comparison, all methods attack the same pre-specified nodes under an untargeted setting with matched cosine-similarity thresholds.

HLBB [37] is a hard-label, decision-based black-box attack that observes only the predicted label and uses a population-based word-substitution search to push the example across the decision boundary while preserving semantic similarity.

TextHoaxer [65] is a budget-aware hard-label attack that casts discrete word substitution as a continuous optimization in embedding space. It iteratively refines a single candidate with a loss combining semantic similarity, pairwise perturbation, and sparsity to reduce queries while maintaining fluency.

SemAttack [56] is a semantics-preserving attack that defines perturbations across multiple semantic spaces (typos, WordNet synonyms, and contextualized BERT neighborhoods) and optimizes within these spaces while controlling the magnitude of changes.

PA-F [35] is a black-box baseline that perturbs node features while keeping the graph fixed. The original method includes RWCS-based node selection with a GC-RWCS variant. In our setting, we disable node selection and attack a fixed node set, applying only feature perturbation. Perturbations are further bounded by a cosine-similarity threshold for fairness.

FGSM [15] is a one-step gradient-sign attack in the continuous feature space. Given features x , it computes $\eta = \epsilon \text{sign}(\nabla_x J)$ and projects $x' = \Pi_C(x + \eta)$ onto a feasible set that enforces either an ℓ_∞ or an ℓ_2 bound together with a cosine-similarity threshold. The graph topology and raw tokens remain fixed.

C Experimental Settings

In this section, we present a unified description of the experimental setups for each problem to ensure reproducibility and consistency. To guarantee the reliability of the results, each experiment is repeated five times independently, and both the mean and standard deviation are reported. For Q1, we select 10% of nodes from Cora and Citeseer as target nodes, while for the larger datasets PubMed

and ogbn-arxiv, we sample 500 nodes. Regarding the key parameters, the text perturbation ratio and the candidate set size are fixed at 30% and top-30, respectively. It is worth noting that since PA-F and FGSM perturb the original features, we constrain them using the same similarity measure as IMDGA. For the remaining experiments, to improve efficiency, the number of target nodes is fixed at 100, while all other hyperparameters remain unchanged. Unless otherwise specified, we choose a 2-layer GCN as the backbone.

D Hyperparameter Settings

Table 6 details the hyperparameters and search ranges used in all reported experiments, with notation following the main text.

Table 6: Search Space for IMDGA.

Hyperparameter	Description	Search Space
β	Text Modification Ratio	$\{0, 0.05, \dots, 0.4\}$
α	Label Flip Weight	$\{0, 1, \dots, 5\}$
top- k_1	Candidate Word Number	$\{0, 5, \dots, 40\}$
top- k_2	Edge Pruning Number	$\{0, 2, 4, 6\}$
$\alpha_1, \alpha_2, \alpha_3$	Scoring Function Weight	Grid Search

E Experiment

Table 7 reports additional ASR results on ogbn-arxiv under identical budgets and cosine-similarity constraints. **OOT**: out of time (12-h wall-clock limit exceeded).

Table 7: ASR comparison on ogbn-arxiv.

Dataset	Methods	SBERT	BERT	RoBERTa	DeBERTa	DistilBERT
Arxiv	HLBB	35.12 \pm 0.32	21.76 \pm 0.84	18.14 \pm 1.57	39.76 \pm 0.49	37.16 \pm 1.22
	TextHoaxer	36.41 \pm 1.15	22.24 \pm 0.82	22.39 \pm 0.43	40.03 \pm 0.11	38.83 \pm 1.56
	SemAttack	OOT	OOT	OOT	OOT	OOT
	FGSM	30.44 \pm 0.26	35.09 \pm 0.88	18.81 \pm 1.07	31.87 \pm 0.70	30.55 \pm 1.92
	PA-F	4.37 \pm 0.42	21.13 \pm 1.16	15.27 \pm 0.51	4.56 \pm 0.67	5.62 \pm 0.45
	IMDGA	47.82\pm1.03	45.56\pm0.22	47.32\pm0.48	44.73\pm0.11	48.12\pm0.45

F Human Evaluation

We recruit three trained student annotators to evaluate text-side adversarial examples after a short warm-up. For each case, sentences are shown in random, blind order, and each annotator rates them independently without discussion. Scores are averaged per sentence across annotators and then combined into per-method means across all cases for reporting.

Below are the annotation instructions and the concise guideline for language-quality ratings used herein.

Please rate overall language quality on a 1–5 scale (coherence, fluency, grammar), considering clarity and readability.

- 5 – Natural, coherent; no errors; fully fluent.
- 4 – Minor issues; easy to read; few typos.
- 3 – Clear meaning; some roughness; occasional errors.
- 2 – Awkward; hinders understanding; frequent errors.
- 1 – Incoherent; severely ungrammatical; unreadable.

G Adversarial Examples

Table 8: Examples of successful IMDGA attacks on Cora: original tokens in bold and adversarial substitutions in red.

Type	Text
Orig.	Prior information and generalized questions: This paper ... uses a Bayesian decision theoretic framework , contrasting parallel and inverse decision problems, ... a subsequent risk minimization ...
Adv.	Prior knowledge and simplified questions: This paper ... uses a Bayesian decision theoretic system , contrasting parallel and opposite decision problems, ... a subsequent cost minimization ...
Orig.	Several computer algorithms for discovering patterns in groups of protein sequences ... and these algorithms are sometimes prone to producing models that are incorrect because two or ...
Adv.	Several computer algorithms for discovering patterns in sets of protein sequences ... and these methods are sometimes vulnerable to producing models that are inaccurate because two or ...

H Pseudo-code

For completeness and clarity, the full procedures of the three IMDGA modules are provided in Algorithms 1–3 below.

Algorithm 1: Topological SHAP Module

Input : Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{T})$, target node $v \in \mathcal{V}$, victim Graph-LLM $\mathcal{F}_\theta(\cdot)$, text encoder $\psi_\theta(\cdot)$, tokenizer $\text{Tokenize}(\cdot)$, mask operation $\text{Mask}(\cdot)$, coalition sampler $\mathcal{S}(\cdot)$

Output : Pivotal word set \mathcal{P}

```

1 foreach  $W_i \in \mathcal{W}$  do
2   foreach  $S \in \mathcal{S}(\mathcal{W} \setminus \{W_i\})$  do
3      $\mathcal{T}_{S \cup \{W_i\}} \leftarrow \text{Mask}(S \cup \{W_i\})$ ,  $\mathcal{T}_S \leftarrow \text{Mask}(S)$ 
4      $z_S \leftarrow \psi_\theta(\mathcal{T}_S)$ ,  $z_{S \cup \{W_i\}} \leftarrow \psi_\theta(\mathcal{T}_{S \cup \{W_i\}})$ 
5      $f(\mathcal{T}_S) \leftarrow \sum_{u \in \{v\} \cup \mathcal{N}(v)} \mathcal{F}_\theta(\mathcal{G}, z_S, u)$ 
6      $f(\mathcal{T}_{S \cup \{W_i\}}) \leftarrow \sum_{u \in \{v\} \cup \mathcal{N}(v)} \mathcal{F}_\theta(\mathcal{G}, z_{S \cup \{W_i\}}, u)$ 
7      $w_S \leftarrow \frac{|S|! (m - |S| - 1)!}{m!}$ 
8      $\phi(W_i) \leftarrow \phi(W_i) + w_S \cdot (f_S - f_{S \cup \{W_i\}})$ 
    /* Compute SHAP contribution for word  $W_i$  */
9 foreach  $W_i \in \mathcal{W}$  do
10    $\xi(W_i) \leftarrow \sum_{u \in \{v\} \cup \mathcal{N}(v)} \phi^{y_u}(W_i)$ 
    /* Aggregate SHAP values over target node  $v$  and its neighbors */
11  $\mathcal{I}_k \leftarrow$  indices of top- $k$  tokens by  $\xi(\cdot)$ 
    /* Select top- $k$  pivotal tokens */
12  $\mathcal{P} \leftarrow \{W_i \mid \xi(W_i) > \tau, i \in \mathcal{I}_k\}$ 
13 return  $\mathcal{P}$ 

```

Algorithm 2: Semantic Perturbation Module

Input : Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{T})$, target node $v \in \mathcal{V}$, Graph-LLM $\mathcal{F}_\theta(\cdot)$, text encoder $\psi_\theta(\cdot)$, Masked Language Model (MLM), pivotal word set \mathcal{P}

Output : Perturbed text \mathcal{T}'_v for node v

```

1  $C \leftarrow \{\}$ ,  $\text{count} \leftarrow 0$ 
2 foreach  $W_i \in \mathcal{P}$  do
3    $C[i] \leftarrow$  top- $k$  candidates from  $\text{MLM}(W_i, \mathcal{T}_v)$ 
    /* Generate top- $k$  replacement candidates for pivotal word  $W_i$  using MLM */
4 for  $i \leftarrow 0$  to  $|\mathcal{P}|$  do
5   foreach  $r \in C[i]$  do
6      $\mathcal{T}' \leftarrow (\mathcal{T}_v \setminus \{W_i\}) \cup \{r\}$ 
7      $p_u(\mathcal{G}, \mathcal{T}') \leftarrow \mathcal{F}_\theta(\mathcal{G}, \psi_\theta(\mathcal{T}'), u)$ 
8      $\delta_u(r) \leftarrow p_u^{(1)}(\mathcal{G}, \mathcal{T}') - p_u^{(2)}(\mathcal{G}, \mathcal{T}')$ 
9      $\Delta(r) \leftarrow \sum_{u \in \{v\} \cup \mathcal{N}(v)} \delta_u(r)$ 
10    if  $\mathcal{F}_\theta(\mathcal{G}, \psi_\theta(\mathcal{T}'), v)$  flips label then
11       $\mathbb{I}_{\text{flip}}(r) \leftarrow 1$ 
12    else
13       $\mathbb{I}_{\text{flip}}(r) \leftarrow 0$ 
14       $\sigma(r) \leftarrow \Delta(r) \cdot (1 + \alpha \cdot \mathbb{I}_{\text{flip}}(r))$ 
    /* Compute score  $\sigma(r)$  to select optimal replacement  $r^*$  */
15     $r^* \leftarrow \arg \max_{r \in C} \sigma(r)$ 
16     $\mathcal{T}'_v \leftarrow (\mathcal{T}_v \setminus \{W_i\}) \cup \{r^*\}$ 
17     $\text{count} \leftarrow \text{count} + 1$  if  $\text{count} > \beta \cdot |\mathcal{T}|$  then return  $\mathcal{T}'_v$ 
    /* Stop if modification exceeds ratio */
18 return  $\mathcal{T}'_v$ 

```

Algorithm 3: Edge Pruning Module

Input : Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{T}')$, target node $v \in \mathcal{V}$, victim Graph-LLM $\mathcal{F}_\theta(\cdot)$, text encoder $\psi_\theta(\cdot)$, number of layers k , mask matrix $M \in \mathbb{R}^{k \times n}$, weight matrix U

Output : Pruned graph \mathcal{G}'

Parameter: weights $\alpha_1, \alpha_2, \alpha_3$

```

1  $\mathcal{G}_n(v) \leftarrow \emptyset$  foreach  $u \in \mathcal{V}$  do
2    $\delta(u) \leftarrow p_u^{(1)}(\mathcal{G}, \mathcal{T}'_v) - p_u^{(2)}(\mathcal{G}, \mathcal{T}'_v)$ 
3    $I(u, v, k) \leftarrow \left\| \mathbb{E} \left[ (\partial \mathcal{X}_u^{(k)}) / (\partial \mathcal{X}_v^{(0)}) \right] \right\|_1$ 
4    $I_u(v, k) \leftarrow \frac{I(u, v, k)}{\sum_{w \in \mathcal{V}} I(u, w, k)}$ 
5    $\text{Score}(u) \leftarrow \alpha_1 \cdot (1 - \delta(u)) + \alpha_2 \cdot I_u(v, k) + \alpha_3 \cdot \left( \frac{1}{\deg(u)} \right)$ 
    /* Calculate the weighted score of predictive disparity, feature, and degree. */
6  $\mathcal{G}_n(v) \leftarrow$  top- $k$  nodes by  $\text{Score}(u)$ 
7  $\hat{\phi} \leftarrow (M^\top U M)^{-1} M^\top U \hat{y}$ 
8 foreach edge  $e \in \mathcal{E}$  in  $\mathcal{G}_n(v)$  do
9   Assign attribution  $\phi_e$  from  $\hat{\phi}$ 
10  $\mathcal{E}' \leftarrow \mathcal{E} \setminus$  top- $k$  edges by  $\phi_e$ 
11  $\mathcal{G}' \leftarrow (\mathcal{V}, \mathcal{E}', \mathcal{X}, \mathcal{T})$ 
12 return  $\mathcal{G}'$ 

```