

GRAPHTEXTACK: A Realistic Black-Box Node Injection Attack on LLM-Enhanced GNNs

Jiaji Ma^{1*}, Puja Trivedi¹, Danai Koutra¹

¹University of Michigan, Ann Arbor
 jjaji@umich.edu, pujat@umich.edu, dkoutra@umich.edu

Abstract

Text-attributed graphs (TAGs), which combine structural and textual node information, are ubiquitous across many domains. Recent work integrates Large Language Models (LLMs) with Graph Neural Networks (GNNs) to jointly model semantics and structure, resulting in more general and expressive models that achieve state-of-the-art performance on TAG benchmarks. However, this integration introduces dual vulnerabilities: GNNs are sensitive to structural perturbations, while LLM-derived features are vulnerable to prompt injection and adversarial phrasing. While existing adversarial attacks largely perturb structure or text independently, we find that uni-modal attacks cause only modest degradation in LLM-enhanced GNNs. Moreover, many existing attacks assume unrealistic capabilities, such as white-box access or direct modification of graph data. To address these gaps, we propose GRAPHTEXTACK, the first black-box, multi-modal node injection attack for LLM-enhanced GNNs. GRAPHTEXTACK injects nodes with carefully crafted structure and semantics to degrade model performance, operating under a realistic threat model without relying on model internals or surrogate models. To navigate the combinatorial, non-differentiable search space of connectivity and feature assignments, GRAPHTEXTACK introduces a novel evolutionary optimization framework with a multi-objective fitness function that balances local prediction disruption and global graph influence. Extensive experiments on five datasets and two state-of-the-art LLM-enhanced GNN models show that GRAPHTEXTACK significantly outperforms 12 strong baselines.

1 Introduction

Text-attributed graphs (TAGs), where nodes are associated with natural language text, are common in many real-world applications. Examples include citation networks, where each node represents a paper described by its abstract, and product co-purchase graphs, where products are annotated with textual descriptions. TAGs combine rich semantic information with relational structure, making them a powerful representation for a range of learning tasks. Early graph learning methods relied on shallow node features, often ignoring the rich semantic and contextual information in textual attributes. Recent advances leverage Large Language

Models (LLMs) to extract expressive node representations from raw text, which are then integrated with graph structure via Graph Neural Networks (GNNs). This combination enables models to capture complex relational and semantic patterns, leading to improved performance on downstream tasks involving TAGs (Duan et al. 2023; Tang et al. 2024; Fatemi, Halcrow, and Perozzi 2024).

Compared to directly applying LLMs to graph learning tasks (i.e., the LLM-as-predictor paradigm), combining LLM-derived embeddings with graph structure aggregation (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Wu et al. 2020), has achieved state-of-the-art performance on node classification benchmarks (Chien et al. 2022; Duan et al. 2023; He et al. 2024; Liu et al. 2024). These LLM-enhanced GNN approaches offer a powerful and general framework for learning on TAGs, effectively integrating language modeling with structural reasoning.

However, this integration also introduces new vulnerabilities that are underexplored. GNNs are known to be susceptible to adversarial perturbations in both structure and node features (Zügner, Akbarnejad, and Günnemann 2018; Ma et al. 2021), with structural modifications often being particularly effective (Zhu et al. 2022). At the same time, LLMs are vulnerable to prompt injection and adversarial phrasing (Li et al. 2025; Wei, Haghtalab, and Steinhardt 2023). As a result, LLM-enhanced GNNs inherit vulnerabilities from both modalities (structural and semantic) making them sensitive to even small perturbations in either input space.

Yet, because these vulnerabilities are distributed across two interdependent modalities, they are difficult to fully exploit through uni-modal attacks that perturb structure or text independently. To motivate the need for stronger attack strategies, we briefly evaluate state-of-the-art uni-modal attacks, including structural perturbations (Geisler et al. 2021) and textual attacks (Li et al. 2020; Ebrahimi et al. 2018; Eger et al. 2019). Our observations show that uni-modal attacks cause only *modest* degradation, highlighting the importance of developing effective multi-modal adversarial approaches.

Existing adversarial attacks on LLM-enhanced GNNs, however, largely remain *uni-modal*, perturbing either structure or textual features independently (Guo et al. 2024). Moreover, many existing attacks assume unrealistic capabilities, such as direct modification of existing nodes, edges, or textual content, or full white-box access to model internals.

*Corresponding Author

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

In practical deployment settings, however, existing graph data is often immutable, and deployed models expose only prediction outputs without revealing internal architectures or training dynamics. These constraints make such strong attack assumptions infeasible for real-world adversaries.

This motivates a more realistic threat model: *node injection attacks*, where adversaries introduce new nodes into the graph without modifying the existing graph elements. This better reflects common real-world scenarios where adversaries can create new user profiles, products, or documents, but cannot arbitrarily alter existing records. Such attacks are especially relevant in domains like e-commerce (where fake products or reviews may be injected to manipulate recommendations) and citation networks, where low-quality or fraudulent publications can be added to influence scholarly metrics. In these settings, attackers can realistically create new entities with realistic structure and textual content, which are added to the data and can influence the model, making node injection a practical adversarial attack. **Our Approach.** To address existing limitations, we propose GRAPHTEXTACK, the first black-box, multi-modal poisoning attack that injects nodes, designed specifically for LLM-enhanced GNNs. It jointly perturbs both structure and semantics without relying on model internals or surrogate models. To efficiently explore the combinatorial search space of injections, we adopt an evolutionary optimization framework (Eiben and Smith 2003). This approach is well-suited for our setting because the search space includes both structural and semantic changes, which makes standard optimization difficult. Moreover, gradient-based methods are not applicable: LLM-enhanced GNNs rely on LLM encoders that produce high-dimensional, non-differentiable embeddings. Surrogate models are also ineffective, as they cannot accurately approximate the LLM-based representations.

In contrast to standard evolutionary attacks that operate solely on structural perturbations under surrogate-assisted settings (Li et al. 2023; Fang et al. 2024), GRAPHTEXTACK systematically adapts the evolutionary optimization framework to the multi-modal nature of the node injection problem, and jointly optimizes both structure and semantics. Our approach introduces a joint candidate encoding, multi-modal crossover and mutation operations, and a novel multi-objective fitness function that balances structural and semantic attack objectives, enabling more effective and realistic attacks under practical threat models.

We summarize our main contributions as follows:

- **Realistic, Multi-modal Attack Model.** We introduce GRAPHTEXTACK, a realistic, black-box, multi-modal node injection attack that jointly optimizes structure and semantic features to degrade LLM-enhanced GNNs.
- **Theoretical Analysis.** We provide a search space and complexity analysis supporting the efficiency and scalability of our approach, offering theoretical justification.
- **Empirical Study.** We evaluate GRAPHTEXTACK across 5 diverse node classification benchmarks and 2 representative LLM-enhanced GNN target models, demonstrating its superior effectiveness and efficiency compared to 12 state-of-the-art baselines.

Method	Black-box	Multi-modal	Node Injection
BertAttack	✓	x (text)	x
HotFlip	✓	x (text)	x
VIPER	✓	x (text)	x
PRBCD	x (white-box)	x (edges only)	x
Nettack	x (surrog.)	x (graph)	x
PSO	✓	x (graph)	✓
TDGIA	✓	x (graph)	✓
AFGSM	x (surrog.)	x (graph)	✓
$G^2 A2C$	✓	x (graph)	✓
GANI	x (surrog.)	x (graph)	✓
WTGIA	x (surrog.)	✓	✓
Ours	✓	✓ (graph + text)	✓

Table 1: Qualitative comparison with prior attacks.

2 Related Work

Our work builds on several lines of research on graph adversarial attacks, node injection strategies, and robustness of LLM-enhanced GNNs. To contextualize our contributions, we now review the most relevant prior work and highlight how GRAPHTEXTACK advances the state of the art in the context of realistic, multi-modal, and black-box node injection attacks. A qualitative comparison of prior attack methods and our approach is provided in Table 1.

Graph Adversarial Attacks. Structural modifications. Adversarial attacks on graphs have traditionally focused on perturbing node features or graph structure to degrade model performance (Ma et al. 2021; Jin et al. 2020). Among these, structural attacks have received greater attention and generally prove more effective, due to decreasing homophily, amplification effects through message passing, and the creation of inconsistent neighborhoods (Zhu et al. 2022, 2020). Attack strategies range from greedy approximations (Zügner, Akbarnejad, and Günnemann 2018), to meta-learning (Zügner et al. 2020), and evolutionary search methods (Li et al. 2023). However, these approaches typically assume direct access to and control over the existing graph structure, an assumption often unrealistic in practice, where attackers must operate without the ability to modify existing nodes or edges.

Node injection attacks. To overcome this limitation, node injection attacks introduce new nodes connected to the existing graph without modifying existing data. AFGSM (Wang et al. 2020) applies an approximate fast gradient sign method to generate targeted perturbations; G-NIA (Tao et al. 2021) proposes a parametric poisoning strategy that preserves learned structural patterns; TDGIA (Zou et al. 2021) is a targeted evasion attack based on topological defective edge selection mechanisms; and GANI (Fang et al. 2024) uses a genetic algorithm to optimize node injections via a surrogate model. Unlike these methods, GRAPHTEXTACK operates fully black-box and jointly optimizes both structural connectivity and textual feature assignments, without relying on surrogate models or white-box assumptions.

Attacks on LLM-Enhanced GNNs. The robustness of LLM-enhanced GNNs remains underexplored. Existing ad-

versarial strategies for LLM-enhanced GNNs largely remain uni-modal, targeting either structural or textual perturbations independently. Structural perturbations exploit vulnerabilities in message passing, while textual perturbations specifically affect the LLM component and the resulting semantic embeddings. Previous work (Guo et al. 2024) study adversarial attacks on LLM-as-enhancer and LLM-as-predictor pipelines but primarily consider white-box *modification* attacks, i.e., uni-modal attacks where the edges of the original graph are perturbed. On the other hand, WTGIA (Lei et al. 2024) introduces a text-only node injection attack, without considering joint structural and semantic vulnerabilities.

Natural language adversarial attacks such as BERT-Attack (Li et al. 2020), HotFlip (Ebrahimi et al. 2018), and VIPER (Eger et al. 2019) were originally developed to degrade LLM performance through minimal text perturbations (Sclar et al. 2024; Mizrahi et al. 2024), and can be adapted to attack the text modality in LLM-enhanced GNNs.

Unlike prior work on uni-modal attacks, GRAPHTEXTACK introduces the first black-box, global node injection attack that jointly optimizes structural and semantic modalities, demonstrating superior effectiveness over uni-modal baselines across multiple benchmark dataset in realistic threat settings. Notably, it does not require model gradients, surrogates, or data modification. This makes GRAPHTEXTACK uniquely suited to realistic attack scenarios.

3 Preliminaries

In this section, we introduce key concepts for our work.

Graph Definitions. We consider a text-attributed graph, or TAG, $G = (V, E, T, Y)$, where V denotes the set of nodes, E the set of edges, $T : V \rightarrow \mathcal{T}$ a mapping from each node to an associated textual attribute, and $Y \in \mathcal{Y}^{|V|}$ the node labels. Each model defines an encoding function $\phi : \mathcal{T} \rightarrow \mathbb{R}^d$ that processes the raw text into node feature embeddings. Thus, the node feature matrix is $\mathbf{X} = [\phi(T(v))]_{v \in V} \in \mathbb{R}^{|V| \times d}$, where ϕ may correspond to frozen LLM encoders, task-specific fine-tuned models, or other text representation methods depending on the application.

LLM-Enhanced GNNs. Recent approaches have integrated LLMs into GNNs to improve representation learning on TAGs (Ye et al. 2024; Tang et al. 2024; Liu et al. 2024). In LLM-enhanced GNNs, textual attributes $T(v)$ are first processed through an LLM encoder to generate rich semantic embeddings, which are then aggregated using graph message-passing architectures. This design enables the model to leverage both semantic and structural information for downstream tasks. Empirical studies show that they often achieve accuracy surpassing traditional GNNs operating on shallow embeddings (Liu et al. 2024).

Adversarial Attack Settings. Uni-modal graph adversarial attacks seek to create “imperceptible” perturbations that deteriorate model performance. These attacks can be categorized as *evasion* attacks, which perturb the graph data at test time, and *poisoning* attacks, which perturb the graph data during training. Attackers may target specific nodes (*targeted* attacks), or aim to reduce overall model performance (*global* attacks). In the following sections, we consider sev-

eral adversarial attack scenarios targeting LLM-enhanced GNNs: (1) **Textual perturbation attacks:** adversarial modifications to the raw text $T(v)$ associated with existing nodes, without altering the graph structure; (2) **Structural perturbation attacks:** adversarial modifications to the edge set E (e.g., adding or removing edges between existing nodes), without changing node features; (3) **Node injection attacks:** injection of new nodes V' with associated edges E' and features derived from text $T'(v')$.

In textual and structural attacks, adversaries directly modify T or E , resulting in perturbed graphs $G_{\text{text}} = (V, E, T^{\text{adv}}, Y)$ or $G_{\text{struct}} = (V, E^{\text{adv}}, T, Y)$, where T^{adv} and E^{adv} denote adversarially modified text and edge sets.

In the node injection setting, the attacker constructs an augmented graph $G' = (V \cup V', E \cup E', T \cup T', Y \cup Y')$, where T' and Y' denote the text and labels associated with the injected nodes, respectively. The goal is to inject nodes such that the model’s predictions are significantly disrupted, without modifying existing nodes or edges. Throughout the paper, G refers to the clean graph, G' to the graph after node injection, and f to the target model under attack.

4 Adversarial Setting and Threat Model

LLM-enhanced GNNs leverage language models to enrich node representations with semantic information extracted from textual descriptions. While this added expressivity improves performance, it also introduces new vulnerabilities: perturbations to LLM-derived features or their interactions with the graph structure can significantly influence neighborhood aggregation and classification outcomes.

Motivation: Limitations of uni-modal attacks. Recent work has begun to study adversarial attacks against TAGs in LLM-GNN joint models (Lei et al. 2024; Guo et al. 2024), typically treating structure and text as disjoint modalities. As a result, such studies evaluate structural attacks and textual attacks separately. However, structure and semantic features are inherently entangled, and real-world adversaries may exploit both modalities simultaneously.

To better understand these vulnerabilities, we empirically evaluate the robustness of LLM-enhanced GNNs to uni-modal attacks, including textual perturbations (BERT-Attack (Li et al. 2020), HotFlip (Ebrahimi et al. 2018), VIPER (Eger et al. 2019)) and structural perturbations (random edge modifications, PRBCD (Zügner, Akbarnejad, and Günnemann 2018)). We discuss the detailed empirical setup in Sec. 6 and the detailed results in appendix in (Ma, Trivedi, and Koutra 2025), and only give the key observations that motivate our approach here. Uni-modal attacks result in *only modest* performance degradation: textual attacks typically reduce accuracy by less than 5%, and structural attacks require randomly altering over 10% of edges or white-box access to achieve comparable effects. These findings highlight the need for stronger, more realistic attacks, particularly ones that simultaneously manipulate structure and semantic features, without modification of existing graphs.

Problem Setup. Unlike prior uni-modal perturbations, we focus on a more restricted and practical threat model: injecting new nodes without altering existing nodes or edges. Under our **threat model**, the attacker can inject a set of

adversarial nodes V' , with associated edges E' and textual descriptions T' , leading to an augmented graph $G' = (V \cup V', E \cup E', T \cup T', Y \cup Y')$.

We study a *poisoning attack* setting, where adversarial nodes are injected prior to model training to influence the learning process. The attacker seeks to strategically design (V', E', T') to maximally degrade the model’s predictive performance on the unlabeled nodes, while operating under realistic constraints. Specifically, we consider a *black-box attack* setting: the attacker has no access to the model’s parameters, gradients, or internal architecture, and can only observe model predictions through queries. In practice, these queries are lightweight inferences, and results can be cached to avoid redundant calls. This mirrors real-world adversarial scenarios where deployed models are typically opaque beyond input-output access. We focus on *global attacks* aimed at degrading the model’s overall performance across the graph, rather than targeting specific nodes individually.

5 GRAPHTEXTACK: Our Attack Framework

To effectively search the space of possible node injections under the black-box threat model defined in Sec. 4, we propose GRAPHTEXTACK, a black-box evolutionary optimization framework designed for node injection poisoning attacks on text-attributed graphs.

Motivation for an evolutionary attack. Optimizing node injections under a black-box model is uniquely challenging: the search space is high-dimensional, discrete, combinatorial, and non-differentiable. Traditional black-box optimization techniques, such as Bayesian optimization, struggle due to reliance on surrogate models. In contrast, evolutionary algorithms (EAs) offer a pragmatic, assumption-light approach, requiring only model queries and can explore complex solution spaces without gradient information.

While EAs have been applied in adversarial machine learning, prior work primarily targeted structural perturbations under white-box or surrogate-assisted settings. GRAPHTEXTACK addresses a significantly harder problem: realistic, global node injection attacks on LLM-enhanced GNNs, where injected nodes influence both graph structure and LLM-derived features. By jointly evolving structural connections and feature assignments, GRAPHTEXTACK effectively navigates the combinatorial complexity introduced by multi-modal vulnerabilities, outperforming prior approaches constrained to single-modality attacks.

As outlined in Algorithm 1, GRAPHTEXTACK operates iteratively over multiple generations. It first initializes a population of candidate injection strategies and explores the combinatorial search space through iterative selection, crossover, mutation, and fitness evaluation. It identifies an optimal injection strategy via this adaptive evolutionary process.

The core challenge of selecting effective edge connections and feature representations is a discrete, high-dimensional combinatorial problem. Gradient-based optimization methods struggle in this setting, particularly under a black-box threat model where surrogate gradients may poorly approximate the target model’s decision boundaries. Exhaustive search is infeasible due to the exponential number of perturbation possibilities. Furthermore, each injection alters the

Algorithm 1: GRAPHTEXTACK

Input: Graph $G = (V, E, T, Y)$, model f , injection budget r

Output: Poisoned graph

$$G' = (V \cup V', E \cup E', T', Y')$$

for $t = 1$ **to** r **do**

Initialize population $\mathcal{P}_0 = \{s_i\}_{i=1}^{N_p}$;

foreach candidate $s_i \in \mathcal{P}_0$ **do**

Randomly initialize structural connections and feature generation strategy;

for $g = 1$ **to** T_{gen} **do**

foreach candidate $s_i \in \mathcal{P}_g$ **do**

Fitness(s_i) = $\alpha \cdot \Delta_{conf}(s_i) + \beta \cdot PR(s_i)$;

Rank candidates by fitness scores;

Select top N_e elite candidates to form \mathcal{P}_g^{elite} ;

foreach pair $(s_1, s_2) \in \mathcal{P}_g^{elite}$ with probability

$p_{crossover}$ **do**

Create s_{new} by combining connections and feature strategies from s_1 and s_2 ;

Add s_{new} to \mathcal{P}_{g+1} ;

foreach candidate $s_i \in \mathcal{P}_{g+1}$ with

probability p_{mutate} **do**

Randomly modify edge connections and/or feature assignment;

Select best candidate s^* from final generation

$\mathcal{P}_{T_{gen}}$;

Inject nodes and edges from s^* into graph;

return G'

graph topology, influencing downstream classification and necessitating adaptive re-optimization.

To address these challenges, GRAPHTEXTACK maintains a population of candidate solutions, each encoding a potential injection strategy, and iteratively refines them through evolutionary operators guided by our novel multi-objective fitness function. Our attack has four key components.

(C1) Candidate Representation and Initialization. We represent the population at generation t as a set of N_p candidate node injections $\mathcal{P}_t = \{s_i\}_{i=1}^{N_p}$, where each candidate s_i encodes: (1) a set of injected nodes V'_i and their connections to existing nodes $E'_i \subseteq V' \times V$; and (2) a feature generation strategy specifying how node features $T'(v')$ are sampled for each injected node $v' \in V'_i$.

We randomly sample the initial population \mathcal{P}_0 by selecting edge connections and feature strategies according to uniform distributions, ensuring diverse exploration of the search space. The number of edges assigned to each injected node (budget) is sampled from the empirical degree distribution of nodes in G , ensuring that injected nodes exhibit realistic connectivity patterns and remain less detectable.

(C2) Semantic Feature Generation. Each injected node must also be assigned a textual description consistent with the graph’s semantic space. Rather than directly optimizing feature embeddings, which would involve a complex, con-

tinuous search space (and risk generating unrealistic node attributes that do not fit in the feature space), we adopt a class-conditioned sampling strategy.

Each candidate s_i specifies a class label $c \in \mathcal{Y}$ for its injected nodes. Features are then sampled from the empirical distribution of nodes belonging to class c . Formally, for each injected node v' , its feature embedding is sampled as $X'(v') \sim p(X(v) \mid Y(v) = c)$, where $p(\cdot)$ denotes the empirical distribution estimated from labeled nodes V_L , which can be constructed by querying the target model to obtain pseudo labels. The class label assignment is part of the candidate’s representation and is subject to evolutionary optimization during the search process.

Unlike prior attacks that perturb the structure or inject nodes with handcrafted features (Wang et al. 2020; Zou et al. 2021), GRAPHTEXTACK introduces class-conditioned semantic feature generation, this enables injected nodes to blend into the graph’s textual feature space in a way that preserves semantic consistency and improves imperceptibility, preventing injected nodes from appearing anomalous.

(C3) Multi-modal Fitness Evaluation. Since the effects of poisoning attacks cannot be directly measured, we design the fitness function as a multi-modal multi-objective optimization problem, balancing semantic-level disruption (measured by local prediction shifts) and structural-level influence (measured by global graph centrality).

Formally, fitness is defined: $\text{Fitness}(s_i) = \alpha \cdot \Delta_{\text{conf}}(s_i) + \beta \cdot \text{PR}(s_i)$, where $\Delta_{\text{conf}}(s_i)$ measures the prediction shift induced by an injected node, $\text{PR}(s_i)$ measures its structural influence, and $\alpha, \beta \geq 0$ balance the two components.

- *Local prediction shift (Δ_{conf}).* For target model f that outputs SoftMax probabilities over class labels, let $C_v = \max(f_v(G))$ denote node’s $v \in V$ maximum confidence score. After injecting node v' according to s_i , the new graph is G'_i and the updated confidence is $C'_v = \max(f_v(G'_i))$. Thus, we define the local prediction shift as: $\Delta_{\text{conf}}(s_i) = \frac{1}{|\mathcal{N}_2(v')|} \sum_{v \in \mathcal{N}_2(v')} |C_v - C'_v|$, where $\mathcal{N}_2(v')$ is the two-hop neighborhood of node v' .

- *Global, PageRank influence (PR).* To capture the global influence of the injected node, we compute its PageRank score after injection into graph G'_i . The influence term is then defined as: $\text{PR}(s_i) = \text{PR}(v')$, where v' is the node injected according to candidate s_i . Nodes with higher centrality scores are likely to affect more nodes via message passing, so high PageRank indicates greater potential influence.

This multi-modal fitness function ensures that the attack effectiveness is evaluated across both feature-based and topology-based vulnerabilities. Moreover, we combine prediction shift and PageRank influence to jointly optimize both localized adversarial impact and global structural importance. This ensures that the evolutionary search process favors candidates that not only perturb model predictions in their local neighborhoods but are also central in the graph.

Unlike prior adversarial attacks on graphs that primarily focus on structural modifications or localized prediction changes in isolation, GRAPHTEXTACK jointly optimizes semantic perturbations and structural influence.

(C4) Multi-modal Evolutionary Operations. At each gen-

eration, we apply the following evolutionary operations to refine the candidate population.

Selection: Candidate injection strategies are ranked based on fitness scores, and the top N_e elite individuals $\mathcal{P}_t^{\text{elite}} \subseteq \mathcal{P}_t$ are retained to form the basis of the next generation.

Crossover: We select pairs of elite candidate injection strategies, and combine them to produce new candidates. Given parent candidates s_1 and s_2 , a new candidate s_{new} is created by splitting and recombining these components. Specifically, for the edge connections, a random crossover point j is selected, and s_{new} inherits the first j connections from s_1 and the remaining connections from s_2 : $E'_{\text{new}} = E'_{s_1}[:j] \parallel E'_{s_2}[j:]$. And the class sampling strategy is randomly chosen from one of the two parents. This *multi-modal crossover* enables the evolutionary process to explore interactions between graph structure and node semantics, which is critical for effective attacks on LLM-enhanced GNNs.

Mutation: To maintain population diversity, each candidate is mutated with probability p_{mut} by randomly altering edge connections or feature assignments.

5.1 Search Space and Computational Complexity for Multi-modal Injection

Proposition 5.1 (Search Space of Multi-Modal Injection). *Optimizing multi-modal node injection attacks in $G = (V, E)$ exactly requires searching over a space of size $O(|V|^{r \cdot d_{\text{max}}} \times |\mathcal{F}|^r)$, which is exponentially large in the number of injected nodes r , the maximum degree per injected node d_{max} , and the feature set size $|\mathcal{F}|$.*

The large search space motivates the evolutionary optimization design of GRAPHTEXTACK, which maintains population diversity and uses mutation and crossover to explore the search space efficiently without exhaustive enumeration.

Lemma 5.2 (Polynomial-Time Evolutionary Approximation). *Under fixed population size N_p and generation budget T_{gen} , the runtime complexity of GRAPHTEXTACK per injection step is $O(N_p \cdot T_{\text{gen}} \cdot (r \cdot d_{\text{max}}^2 + |E|))$, where r is the number of injected nodes, d_{max} is maximum degree per injected node, and $|E|$ is the original number of edges.*

In practice, d_{max} is chosen to be small to ensure stealthy attacks. Since N_p and T_{gen} are tunable parameters, the overall runtime of GRAPHTEXTACK scales effectively linearly in graph size ($|V|, |E|$), and so it offers a scalable and tunable trade-off between search space and runtime. We empirically validate the efficiency of our attack in the next section. In addition to the above complexity analysis, we formally characterize the synergy between structure and feature perturbations in multi-modal attacks, provide theoretical analysis of the advantages of multi-modal node injections, and analyze how local prediction shifts arise with node injection perturbations, in appendix in (Ma, Trivedi, and Koutra 2025).

6 Experiments

We now present an extensive empirical evaluation of GraphTextack. Our analysis aims to answer the following key research questions: **(RQ1)** Can our GRAPHTEXTACK framework effectively degrade the node classification perfor-

mance of LLM-enhanced GNN models through node injection? How does our surrogate-free framework in a black-box setting compare to strong baselines? **(RQ2)** How does GRAPHTEXTACK compare to baselines in terms of efficiency? **(RQ3)** How do different components of our framework contribute to the overall effectiveness?

6.1 Empirical Setup

Datasets. We conduct experiments on five widely used text-attributed graph benchmark datasets for node classification: Cora(McCallum et al. 2000), PubMed (Sen et al. 2008), WikiCS (Mernyei and Cangea 2022), ogbn-arxiv(Hu et al. 2020), and ogbn-products (Hu et al. 2020). For ogbn-products, we use the sampled subset from (He et al. 2024). For all datasets, we use the default splits, and obtain the raw textual attributes from (Liu et al. 2024) and (He et al. 2024). We summarize the dataset statistics in appendix in (Ma, Trivedi, and Koutra 2025).

Target Models. We evaluate two representative paradigms of LLM-enhanced GNNs: (1) **representation-level enhancer**: One-for-all (Liu et al. 2024) is a universal graph representation learning framework that incorporates LLM-based text encodings to create a task-agnostic graph representation; (2) **node-level enhancer**: In this setup, we directly perform node feature enhancement by encoding node attributes using the pretrained e5-large-v2 model (Wang et al. 2024), and pass the resulting embeddings into a GCN model (Luo, Shi, and Wu 2024).

Attack Baselines & Hyperparameter Tuning. Since black-box node injection attack is an emerging area, there are few existing methods directly applicable for comparison. We compare our attack against 12 representative global attack methods and adaptations of general graph attacks. Specifically, we consider three textual attacks: (1) BERT-Attack (Li et al. 2020), (2) HotFlip (Ebrahimi et al. 2018), and (3) VIPER (Eger et al. 2019); (4) a gradient-based structural modification attack, PRBCD (Geisler et al. 2021); (5) a Preferential attack (Barabási and Albert 1999), a heuristics-based approach; popular attacks that we adapted to our node injection setup: (6) Nettack (Zügner, Akbarnejad, and Günnemann 2018): widely used targeted modification attack; (7) PSO (Zang et al. 2020), a population-based targeted attack; (8) TDGIA (Zou et al. 2021), a targeted evasion attack; (9) AFGSM (Wang et al. 2020), a targeted poisoning attack; (10) G^2A2C (Ju et al. 2023), an evasion injection attack based on direct model queries; (11) GANI (Fang et al. 2024), a global poisoning injection attack; and (12) WTGIA (Lei et al. 2024), a text-level node injection attack.

For our approach, we tune the population size, crossover and mutation probabilities, and the coefficients α and β (control the balance between local prediction disruption and global influence) through hyperparameter search. We give more details in appendix in (Ma, Trivedi, and Koutra 2025).

Evaluation Metrics. For different node injection budgets, we report the average node classification accuracy after attack and stdev over five runs and attack generation runtime (wall-clock time). More effective methods lead to bigger accuracy drop relative to the clean graph.

6.2 (RQ1) Overall Attack Performance

First, we seek to assess the overall performance of GRAPHTEXTACK by evaluating the classification accuracy drop across multiple datasets and target models. We vary the node injection budget r in $r = \{0.01, 0.03, 0.05\}$, intentionally keeping it low to ensure imperceptibility and avoid distribution shifts that would make the attack more detectable by defense mechanisms. We give the post-attack classification accuracy results for the representation-level enhancer in Table 2, and for the feature-level enhancer in appendix in (Ma, Trivedi, and Koutra 2025). Among the three textual attacks, which modify *all* the textual features, we provide the best-performing result per dataset (*‘Best text*’*); since the budget does not apply in this case, we report a single number. In the results table, the best performance (i.e., the lowest accuracy) is bolded and the second best underlined. We discuss our key observations below.

Observations. GRAPHTEXTACK generally achieves more significant degradation in classification accuracy compared to baseline attacks, across datasets and injection budgets, demonstrating strong generalization and scalability. This is achieved through the multi-modal, multi-objective fitness evaluation and evolutionary operations. Jointly optimizing for prediction impact and structural centrality allows the attack to more effectively explore the search space, leading to more effective injections. Notably, even the strongest textual attacks, despite modifying all node features, still underperform compared to GRAPHTEXTACK, highlighting the advantage of multi-modal attacks.

GRAPHTEXTACK directly queries the target model, avoiding reliance on surrogate model approximations. This allows it to better adapt to the decision boundaries of the target model, regardless of architectural differences (representation-level in Table 2 vs. feature-level enhancer in appendix (Ma, Trivedi, and Koutra 2025)). That makes it more effective in settings with a significant mismatch between the surrogate and target models, such as the representation-level enhancer.

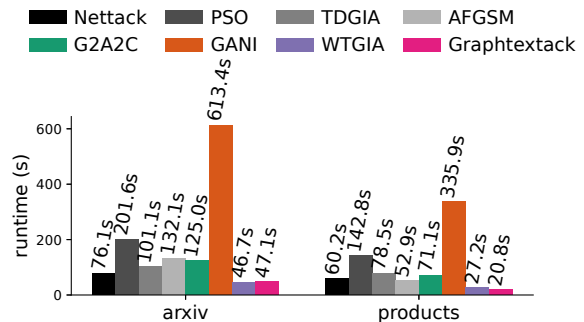


Figure 1: Runtime per injection on the representation-level enhancer target model.

(RQ2) Runtime In addition to attack effectiveness, computational efficiency is another important practical consideration for node injection attacks. Figure 1 compares the time required to generate each injection attack for the base-

Dataset (Clean Acc)	Methods	Classification Accuracy		
		r=0.01	r=0.03	r=0.05
Cora (80.95 ±0.87)	Pref.	77.34±0.58	73.49±0.52	72.39±0.96
	Best text*	-----	74.36±1.68	-----
	PRBCD	79.33±2.08	77.37±1.03	74.75±0.54
	Nettack	76.83±0.27	68.47±0.20	<u>62.59±0.30</u>
	PSO	77.83±0.60	73.14±0.52	68.07±0.39
	TDGIA	75.79±0.12	71.26±0.26	66.26±0.49
	AFGSM	<u>75.47±0.37</u>	69.75±0.75	63.89±0.22
	$G^2 A2C$	77.20±0.82	<u>67.79±0.67</u>	63.74±0.69
	GANI	77.18±0.77	70.98±0.96	66.06±0.70
	WTGIA	76.35±0.38	69.63±0.43	65.80±0.36
Ours	73.99±0.78	65.75±0.81	62.02±0.64	
PubMed (71.65 ±0.77)	Pref.	67.48±0.57	59.34±0.60	50.97±0.85
	Best text*	-----	71.78±0.69	-----
	PRBCD	71.13±0.18	68.60±1.03	65.04±0.90
	Nettack	64.28±0.53	51.00±0.45	45.14±0.27
	PSO	65.62±0.35	49.61±0.38	43.58±0.71
	TDGIA	64.70±0.56	51.33±0.39	43.37±0.43
	AFGSM	62.96±0.59	53.98±0.77	44.11±0.24
	$G^2 A2C$	<u>62.10±0.41</u>	54.96±0.24	43.78±0.76
	GANI	64.33±0.74	54.59±1.23	45.61±0.48
	WTGIA	65.89±0.96	<u>48.96±0.46</u>	<u>43.10±0.74</u>
Ours	60.78±0.25	48.43±0.63	42.05±0.68	
WikiCS (76.31 ±0.94)	Pref.	73.33±1.66	72.02±0.54	69.95±0.61
	Best text*	-----	74.34±0.21	-----
	PRBCD	72.96±1.04	70.13±0.52	67.38±0.51
	Nettack	72.92±0.70	65.29±0.92	62.68±0.67
	PSO	73.89±1.67	67.39±0.64	64.77±1.00
	TDGIA	72.33±0.65	65.67±0.83	62.23±0.78
	AFGSM	72.47±0.71	67.25±0.27	61.71±1.16
	$G^2 A2C$	72.90±0.36	65.82±0.50	62.46±0.71
	GANI	73.22±0.86	69.03±0.89	63.83±0.55
	WTGIA	<u>71.87±1.17</u>	64.17±0.84	60.95±0.26
Ours	71.69±1.04	<u>64.58±0.69</u>	<u>61.35±1.33</u>	
ogbn- arxiv (75.44 ±1.09)	Pref.	74.86±0.44	72.53±0.37	69.27±0.75
	Best text*	-----	72.13±0.65	-----
	PRBCD	74.98±0.49	73.38±0.70	72.38±0.79
	Nettack	73.96±0.89	69.93±0.33	<u>66.76±0.55</u>
	PSO	<u>72.03±0.46</u>	69.82±1.14	67.87±0.89
	TDGIA	73.63±0.47	70.05±0.32	66.97±0.21
	AFGSM	73.09±0.90	71.24±0.20	68.82±0.45
	$G^2 A2C$	72.63±0.34	<u>69.50±0.68</u>	66.67±0.24
	GANI	73.61±0.91	70.31±0.42	68.17±0.36
	WTGIA	72.38±0.72	70.04±0.76	67.61±0.98
Ours	71.95±0.38	68.23±0.85	66.61±0.78	
ogbn- products (83.51 ±1.05)	Pref.	80.32±1.46	77.92±0.56	75.29±0.95
	Best text*	-----	75.45±0.23	-----
	PRBCD	81.09±0.27	79.28±0.29	77.85±0.14
	Nettack	80.09±0.87	76.55±0.44	70.23±0.29
	PSO	78.91±0.95	75.16±0.45	70.71±0.40
	TDGIA	80.22±0.95	75.45±0.99	72.11±0.93
	AFGSM	79.76±0.68	75.05±0.51	70.33±0.94
	$G^2 A2C$	79.05±0.23	74.90±0.17	71.61±0.32
	GANI	<u>78.93±0.37</u>	<u>74.83±0.99</u>	<u>69.51±0.64</u>
	WTGIA	79.37±0.54	75.55±0.28	71.83±0.21
Ours	78.99±0.63	74.26±0.64	69.05±0.51	

Table 2: [Representation-level enhancer model] Classification accuracy after injection attacks.

Variants	r=0.01	r=0.05
GRAPHTEXTACK	71.69±1.04	61.35±1.33
Without crossover	73.50±0.71	64.42±0.59
Without mutation	73.39±0.85	65.19±0.41
Without pred. shift	73.77±0.67	66.83±0.54
Without PageRank	71.92±0.78	61.97±1.00

Table 3: Ablation study for GRAPHTEXTACK on WikiCS: Accuracy after injection on the representation-level enhancer model.

line methods and our proposed GRAPHTEXTACK across the largest datasets. We provide results for all datasets in appendix in (Ma, Trivedi, and Koutra 2025).

Observations. GRAPHTEXTACK achieves the lowest average runtime, showing clear efficiency improvement over baselines. By avoiding gradient-based optimization and surrogate modeling, it eliminates major computational bottlenecks, enabling more scalable attacks.

(RQ3) Ablation study To verify the effectiveness of our approach, we analyze the influence of each main component of the optimization framework through an ablation study. We systematically remove crossover, mutation, and both components of the fitness function. For each ablation setting, we evaluate the post-attack classification accuracy under different injection budgets and compare it to the full model.

Observations. Table 3 shows that removing any of the main mechanisms of the evolutionary optimization process noticeably reduces attack effectiveness. This highlights the importance of population diversity and search space exploration in the optimization process. We observe consistent trends across different datasets and budgets (see appendix in (Ma, Trivedi, and Koutra 2025)).

7 Conclusion

We introduced GRAPHTEXTACK, a novel black-box node injection attack against LLM-enhanced GNNs. Unlike existing approaches that rely on surrogate models, GRAPHTEXTACK directly queries the target model during the optimization process, leading to a more effective and efficient strategy. We demonstrated that our method outperforms 12 baselines, including heuristics-based methods and state-of-the-art evolutionary optimization models. We also showed that utilizing direct model queries is more computationally efficient than surrogate model-based approaches. Our work provides a foundation for future work to explore defense mechanisms tailored for LLM-enhanced GNNs, such as joint adversarial training and data filtering.

Limitations. GRAPHTEXTACK assumes the attacker can inject nodes with controllable structure and text, which may not be feasible in some applications or domains. Additionally, our current approach follows a simple class-conditioned feature sampling strategy, which, while effective, may be further improved by learning more expressive or task-adaptive feature generation strategies.

Ethical Statement

Our work studies attacks against LLM-enhanced GNNs. All experiments were performed on public datasets and models. While our methods expose potential vulnerabilities, the goal is to highlight real risks in order to promote the development of more robust and secure models, as well as defense mechanisms.

Acknowledgments

This material in this work is supported by the National Science Foundation under IIS 2504090, IIS 2212143 and CAREER Grant No.IIS 1845491. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding parties.

References

- Barabási, A.-L.; and Albert, R. 1999. Emergence of scaling in random networks. *science*, 286(5439): 509–512.
- Chien, E.; Chang, W.; Hsieh, C.; Yu, H.; Zhang, J.; Milenkovic, O.; and Dhillon, I. S. 2022. Node Feature Extraction by Self-Supervised Multi-scale Neighborhood Prediction. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Duan, K.; Liu, Q.; Chua, T.-S.; Yan, S.; Ooi, W. T.; Xie, Q.; and He, J. 2023. SimTeG: A Frustratingly Simple Approach Improves Textual Graph Learning. arXiv:2308.02565.
- Ebrahimi, J.; Rao, A.; Lowd, D.; and Dou, D. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 31–36.
- Eger, S.; Sahin, G. G.; Rücklé, A.; Lee, J.; Schulz, C.; Mesgar, M.; Swarnkar, K.; Simpson, E.; and Gurevych, I. 2019. Text Processing Like Humans Do: Visually Attacking and Shielding NLP Systems. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 1634–1647. Association for Computational Linguistics.
- Eiben, A. E.; and Smith, J. E. 2003. *Introduction to Evolutionary Computing*. Springer.
- Fang, J.; Wen, H.; Wu, J.; Xuan, Q.; Zheng, Z.; and Tse, C. K. 2024. GANI: Global Attacks on Graph Neural Networks via Imperceptible Node Injections. *IEEE Trans. Comput. Soc. Syst.*, 11(4): 5374–5387.
- Fatemi, B.; Halcrow, J.; and Perozzi, B. 2024. Talk like a Graph: Encoding Graphs for Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Geisler, S.; Schmidt, T.; Şirin, H.; Zügner, D.; Bojchevski, A.; and Günnemann, S. 2021. Robustness of Graph Neural Networks at Scale. In *Neural Information Processing Systems, NeurIPS*.
- Guo, K.; Liu, Z.; Chen, Z.; Wen, H.; Jin, W.; Tang, J.; and Chang, Y. 2024. Learning on Graphs with Large Language Models (LLMs): A Deep Dive into Model Robustness. arXiv:2407.12068.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.
- He, X.; Bresson, X.; Laurent, T.; Perold, A.; LeCun, Y.; and Hooi, B. 2024. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33: 22118–22133.
- Jin, W.; Li, Y.; Xu, H.; Wang, Y.; Ji, S.; Aggarwal, C.; and Tang, J. 2020. Adversarial Attacks and Defenses on Graphs: A Review, A Tool and Empirical Studies. arXiv:2003.00653.
- Ju, M.; Fan, Y.; Zhang, C.; and Ye, Y. 2023. Let graph be the go board: gradient-free node injection attack for graph neural networks via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4383–4390.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Lei, R.; Hu, Y.; Ren, Y.; and Wei, Z. 2024. Intruding with Words: Towards Understanding Graph Injection Attacks at the Text Level. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Li, J.; Li, H.; He, J.; and Dou, T. 2023. Lapa: Multi-label-flipping adversarial attacks on graph neural networks. In *2023 International Seminar on Computer Science and Engineering Technology (SCSET)*, 29–32. IEEE.
- Li, L.; Ma, R.; Guo, Q.; Xue, X.; and Qiu, X. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*.
- Li, Y.; Huang, H.; Zhao, Y.; Ma, X.; and Sun, J. 2025. Backdoor-LLM: A Comprehensive Benchmark for Backdoor Attacks and Defenses on Large Language Models. arXiv:2408.12798.
- Liu, H.; Feng, J.; Kong, L.; Liang, N.; Tao, D.; Chen, Y.; and Zhang, M. 2024. One For All: Towards Training One Graph Model For All Classification Tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Luo, Y.; Shi, L.; and Wu, X.-M. 2024. Classic gnn are strong baselines: Reassessing gnn for node classification. *Advances in Neural Information Processing Systems*, 37: 97650–97669.
- Ma, J.; Trivedi, P.; and Koutra, D. 2025. GRAPHTEXTACK: A Realistic Black-Box Node Injection Attack on LLM-Enhanced GNNs. arXiv:2511.12423.
- Ma, Y.; Wang, S.; Derr, T.; Wu, L.; and Tang, J. 2021. Graph Adversarial Attack via Rewiring. In Zhu, F.; Ooi, B. C.; and Miao, C., eds., *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, 1161–1169. ACM.
- McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3: 127–163.
- Mernyei, P.; and Cangea, C. 2022. Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks. arXiv:2007.02901.

- Mizrahi, M.; Kaplan, G.; Malkin, D.; Dror, R.; Shahaf, D.; and Stanovsky, G. 2024. State of What Art? A Call for Multi-Prompt LLM Evaluation. *Trans. Assoc. Comput. Linguistics*.
- Sciar, M.; Choi, Y.; Tsvetkov, Y.; and Suhr, A. 2024. Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting. In *Proc. Int. Conf. on Learning Representations (ICLR)*.
- Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.
- Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; and Huang, C. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 491–500.
- Tao, S.; Cao, Q.; Shen, H.; Huang, J.; Wu, Y.; and Cheng, X. 2021. Single Node Injection Attack against Graph Neural Networks. In Demartini, G.; Zuccon, G.; Culpepper, J. S.; Huang, Z.; and Tong, H., eds., *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, 1794–1803. ACM.
- Wang, J.; Luo, M.; Suya, F.; Li, J.; Yang, Z.; and Zheng, Q. 2020. Scalable attack on graph data by injecting vicious nodes. *Data Mining and Knowledge Discovery*, 34: 1363–1389.
- Wang, L.; Yang, N.; Huang, X.; Jiao, B.; Yang, L.; Jiang, D.; Majumder, R.; and Wei, F. 2024. Text Embeddings by Weakly-Supervised Contrastive Pre-training. arXiv:2212.03533.
- Wei, A.; Haghtalab, N.; and Steinhardt, J. 2023. Jailbroken: How Does LLM Safety Training Fail? In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.
- Ye, R.; Zhang, C.; Wang, R.; Xu, S.; and Zhang, Y. 2024. Language is All a Graph Needs. arXiv:2308.07134.
- Zang, Y.; Qi, F.; Yang, C.; Liu, Z.; Zhang, M.; Liu, Q.; and Sun, M. 2020. Word-level Textual Adversarial Attacking as Combinatorial Optimization. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 6066–6080. Association for Computational Linguistics.
- Zhu, J.; Jin, J.; Loveland, D.; Schaub, M. T.; and Koutra, D. 2022. How does Heterophily Impact the Robustness of Graph Neural Networks?: Theoretical Connections and Practical Implications. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*.
- Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In *Proc. Conf. Neural Information Processing Systems, NeurIPS*.
- Zou, X.; Zheng, Q.; Dong, Y.; Guan, X.; Kharlamov, E.; Lu, J.; and Tang, J. 2021. TDGIA: Effective Injection Attacks on Graph Neural Networks. In Zhu, F.; Ooi, B. C.; and Miao, C., eds., *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, 2461–2471. ACM.
- Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2847–2856.
- Zügner, D.; Borchert, O.; Akbarnejad, A.; and Günnemann, S. 2020. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(5): 1–31.