# N2NSC Node-to-Neighborhood Semantic Consistency: Text-Topology Alignment for TAGs Anomaly Detection

**Bochen Lin**[1], **Jianxiang Yu**[1], **Jiayi Wu**[1], **Lin Qi**[2], **Huang Lu**[2], **Xiang Li**[1,*]
[1]School of Data Science and Engineering, East China Normal University, Shanghai, China
[2]WeChat, Tencent, Guangzhou, China
bochenlin@stu.ecnu.edu.cn

## Abstract

Graph anomaly detection (GAD) on text-attributed graphs (TAGs) is vital for applications such as fraud detection and academic integrity verification. Existing approaches generally fall into two paradigms. GNN-based methods capture structural patterns but struggle to capture fine-grained textual semantics. Methods integrating LLMs with graphs improve semantic understanding yet fail to fully comprehend topological relationships among neighbors. Moreover, both paradigms overlook the correspondence between textual semantics and graph topological relationships, limiting their ability to identify nodes whose semantics are inconsistent with their neighborhoods. In this paper, we formalize TAG anomaly detection as a node-to-neighborhood semantic consistency problem, where anomalies may arise from either textual semantic mismatch or topological deviation between a node and its neighbors. We propose N2NSC (Node-to-Neighborhood Semantic Consistency), a framework that captures the correspondence between graph topology and textual semantics through two complementary fusion paths. The two pathways work synergistically, enabling the LLM to jointly use textual and structural neighborhood information for anomaly detection. Extensive experiments across eight benchmark datasets, with additional validation on two real-world fraud datasets, demonstrate the effectiveness of N2NSC. Our code is available.[3]

## 1 Introduction

Anomaly detection on graphs has become increasingly important in graph mining (Chandola et al., 2009; Ding et al., 2019; Liu et al., 2021a; Tang et al., 2022; Xu et al., 2025b), driven by its broad applicability to fraud detection and scientific integrity analysis. While substantial progress has been made on attributed graphs with fixed-dimensional node features (Liu et al., 2021a; Roy et al., 2024; Ju et al., 2025), real-world networks increasingly exhibit richer representations (Yan et al., 2023). In citation networks (Hu et al., 2020), nodes carry paper abstracts. In e-commerce platforms, product descriptions serve as node attributes. For anomaly detection on such text-attributed graphs (TAGs) (Su et al., 2025), the anomaly signal depends not only on a node's own textual content, but also on the textual content of its neighbors and the topological relationships among them.

We refer to this phenomenon as *node-to-neighborhood semantic inconsistency* (Song et al., 2007), where a node's semantics are inconsistent with the semantic and topological context of its neighborhood. Such inconsistency may manifest as textual semantic deviation, where a node's text diverges from what its neighbors discuss, or as structural topological anomaly, where a node's connectivity pattern deviates from its local graph context. This is illustrated in Figure 1. Consider a paper on deep learning for NLP. If all its citing neighbors are also NLP papers, the neighborhood is semantically consistent and the node is normal. If instead the same paper is surrounded by chemistry publications, the semantic mismatch with its neighborhood makes it anomalous. The node text remains identical in both cases. The anomaly signal arises from node-to-neighborhood semantic inconsistency rather than any intrinsic textual defect.

Existing methods have yet to address this problem adequately. GNN-based methods (Tang et al., 2022; Liu et al., 2021a; Dou et al., 2020; Roy et al., 2024; Ma et al., 2023) model graph topology through message passing but reduce text to frozen embedding vectors, making fine-grained textual reasoning unavailable at decision time. Methods integrating LLMs with graphs (Chen et al., 2024b; Tang et al., 2024; Li et al., 2024) bring language understanding to graph tasks, yet

*Corresponding author.
[3]https://github.com/aibert2/N2NSC



*Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 18574–18589
October 24-29, 2026 ©2026 Association for Computational Linguistics

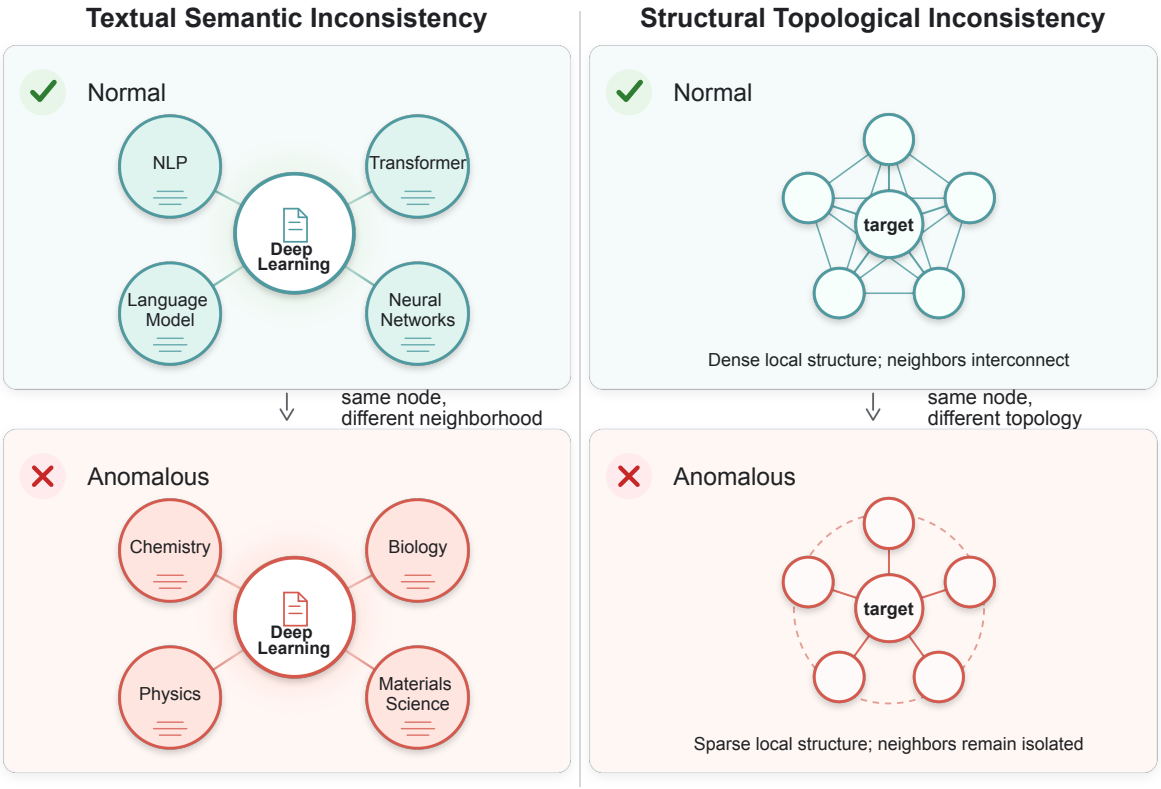


Figure 1: **Neighborhood semantic inconsistency in text-attributed graphs.** Left: the same target node ("deep learning") becomes anomalous when its neighbors shift from the NLP domain to unrelated domains. Right: the same textual domain becomes anomalous when the local topology shifts from dense to sparse. In both cases, the anomaly signal comes from node-to-neighborhood inconsistency rather than intrinsic node defects.

existing approaches either use LLMs as feature generators (He et al., 2024), jointly train text and graph encoders with contrastive objectives (Xu et al., 2025b), or employ multi-LLM collaboration for evidence aggregation (Xu et al., 2025a). In these approaches, the model either has no access to what neighboring nodes actually say, has no awareness of graph-topological relationships, or receives both forms of information yet lacks the capacity to reason over them jointly.

In this work, we introduce N2NSC (*Node-to-Neighborhood Semantic Consistency*), a framework that addresses this limitation through two complementary fusion paths, each capturing the correspondence between graph topology and textual semantics through a different mechanism. The **explicit fusion path** directly shapes what the LLM reads by fusing textual neighborhood semantics and graph-structural neighborhood semantics into a unified representation, enabling the LLM to reason over the full graph-text context of each node. The **implicit fusion path** transforms neighborhood semantics into node-adaptive modulation signals that guide the language model to capture the correspondence between graph topology and textual semantics for each node. Together, the two paths enable the language model to jointly use both textual and structural neighborhood information for anomaly detection.

Extensive experiments on eight benchmark TAG anomaly detection datasets (Xu et al., 2025b), spanning citation networks, e-commerce networks, and large-scale bibliographic data, show that N2NSC achieves strong performance compared with GAD and LLM-based baselines, with additional validation on two real-world fraud datasets. Key contributions of this work are as follows.

- We formalize TAG anomaly detection as a node-to-neighborhood semantic consistency problem and propose N2NSC, a framework that deeply integrates graph-structural semantics and textual semantics into LLMs through two complementary fusion paths.
- We design an explicit fusion path that integrates textual and topological neighborhood semantics into a unified representation, alongside an im-

plicit fusion path that transforms neighborhood semantics into node-adaptive parameter modulation, enabling the LLM to dynamically adapt its computation to each node's graph context.
- Extensive experiments on eight benchmark datasets and two real-world fraud datasets validate that N2NSC achieves consistent improvements over 17 baselines spanning both GNN-based and LLM-integrated methods.

## 2 Related Work

**Graph anomaly detection.** Graph anomaly detection has attracted broad research interest (Ma et al., 2026; Xu and Ding, 2024). Existing methods generally fall into two paradigms. The first uses GNN-based classifiers operating on fixed node embeddings. Representative approaches include reconstruction-based autoencoders such as DOMINANT (Ding et al., 2019), spectral-wavelet filters (BWGNN (Tang et al., 2022)), imbalance-aware neighbor sampling (PC-GNN (Liu et al., 2021a)), camouflage-resistant relational learning (CARE-GNN (Dou et al., 2020)), and neighborhood-reconstruction objectives (GAD-NR (Roy et al., 2024)). Contrastive self-supervised methods such as CoLA (Liu et al., 2021b), ANEMONE (Jin et al., 2021), and GCCAD (Chen et al., 2022) enrich structural representations. These methods treat text solely as frozen pretrained embedding vectors and therefore fail to capture fine-grained textual semantic patterns. The second paradigm integrates textual semantics with graph structure through language models. TAPE (He et al., 2024) uses LLM-generated explanations as enhanced node features for downstream classification. CMU-CL (Xu et al., 2025b) jointly trains text and graph encoders with contrastive consistency. GuARD (Pang et al., 2025) combines GNN embeddings with LM features through multi-modal instruction tuning. CoLL (Xu et al., 2025a) employs multi-LLM collaboration for evidence-augmented generation. In a related direction, UNPrompt (Niu et al., 2024) learns unified graph prompts from neighborhood structure for zero-shot anomaly detection without relying on language models. TAGAD (Liu et al., 2025) compares ego-graph and text-graph representations to detect contextual anomalies, yet its dual modules operate independently without joint reasoning. Overall, these methods treat graph information as auxiliary signals rather than exploiting the correspondence between graph topology and textual semantics.

**Graph-text semantic fusion in language models.** Existing approaches to integrating graph-structural semantics into language models (Li et al., 2024; Xu and Ding, 2024) follow two strategies. Some inject graph signals into the model input as additional text or tokens, such as structural descriptions (Ye et al., 2024) or LLM-generated explanations (He et al., 2024). Others align graph and text representations in the embedding space through contrastive objectives (Xu et al., 2025b) or cross-modal projection (Tang et al., 2024; Chen et al., 2024b; Wang et al., 2024). All these approaches confine graph semantics to either the input or the embedding space, without capturing the correspondence between graph topology and textual semantics. In contrast, N2NSC achieves semantic fusion through two complementary levels simultaneously. The explicit path fuses textual neighborhood semantics and graph-structural neighborhood semantics at the input level, while the implicit path transforms neighborhood semantics into node-adaptive modulation signals that guide the model to capture the correspondence between graph topology and textual semantics. This dual-path design provides a distinctive capability for TAG anomaly detection.

## 3 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{T})$ denote a text-attributed graph (TAG), where $\mathcal{V}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the node feature matrix with each row $\mathbf{x}_v \in \mathbb{R}^d$ being the feature vector of node $v$ obtained from its text via a pretrained language encoder, and $\mathbf{T} = \{T_v\}_{v \in \mathcal{V}}$ is the collection of raw node texts. We denote the adjacency matrix as $\mathbf{A} \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}|}$, where each entry $\mathbf{A}_{vu}$ indicates the link between nodes $v$ and $u$. Each node $v \in \mathcal{V}$ is associated with a binary label $y_v \in \{0,1\}$ indicating whether the node is normal or anomalous. Given a set of labeled nodes $\mathcal{V}_L \subset \mathcal{V}$, the goal is to accurately identify anomalous nodes in $\mathcal{V}_U = \mathcal{V} \setminus \mathcal{V}_L$.

## 4 Methodology

### 4.1 Overview

Detecting node-to-neighborhood semantic inconsistency requires deeply fusing graph-structural semantics with textual semantics. N2NSC achieves this through two complementary fusion paths that operate synergistically. The **explicit fusion path** fuses textual neighborhood semantics and graph-structural neighborhood semantics into a unified

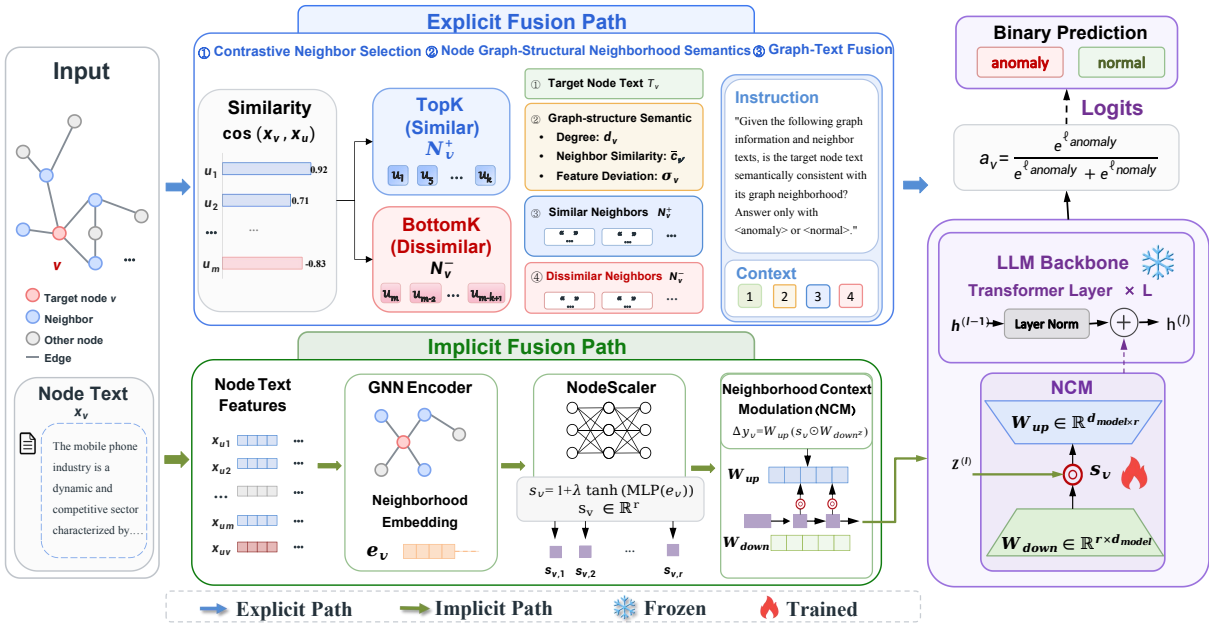


Figure 2: N2NSC framework overview. The explicit fusion path fuses textual neighborhood semantics and graph-structural neighborhood semantics into a unified representation, while the implicit fusion path (NCM) transforms neighborhood semantics into node-adaptive modulation signals that guide the LLM to capture the correspondence between graph topology and textual semantics.

representation, providing the LLM with complete neighborhood semantics. The **implicit fusion path** transforms neighborhood semantics into node-adaptive modulation signals that guide the language model to capture the correspondence between graph topology and textual semantics for each node. Figure 2 illustrates the framework.

### 4.2 Explicit Fusion Path

The explicit path fuses graph-structural neighborhood semantics and textual neighborhood semantics into a unified representation. It captures the node's topological profile through neighborhood-derived statistics, namely degree, neighbor similarity, and feature deviation, along with contrastive neighbor texts from the node's graph neighborhood $\mathcal{N}(v) = \{u : \mathbf{A}_{vu} = 1\}$.

**Textual Neighborhood Semantics.** For each target node $v$, we compute pairwise cosine similarity between $\mathbf{x}_v$ and all graph neighbors:

$$c_{vu} = \cos(\mathbf{x}_v, \mathbf{x}_u), \quad u \in \mathcal{N}(v). \tag{1}$$

Two contrastive sets are selected. $\mathcal{N}_v^+ = \mathrm{TopK}_u\ c_{vu}$ contains semantically similar neighbors, forming a *local prototype*, and $\mathcal{N}_v^- = \mathrm{BottomK}_u\ c_{vu}$ contains semantically dissimilar neighbors, exposing *local conflicts*.

**Graph-Structural Neighborhood Semantics.** We characterize the node's topological context through three neighborhood-derived statistics. Let $\bar{\mathbf{x}}_v = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \mathbf{x}_u$ denote the mean feature vector of node $v$'s neighbors. We define the degree $d_v = |\mathcal{N}(v)|$, the average neighbor similarity $\bar{c}_v = \cos(\mathbf{x}_v, \bar{\mathbf{x}}_v)$, and the feature deviation $\delta_v = \|\mathbf{x}_v - \bar{\mathbf{x}}_v\|_2$. A normal node typically has high $\bar{c}_v$ and low $\delta_v$, while an anomalous node exhibits the opposite pattern.

**Semantic Fusion.** The fused representation is assembled into a structured input comprising the raw text $T_v$ of the target node, the neighborhood-derived statistics $d_v$, $\bar{c}_v$, $\delta_v$, and extractive text snippets from $\mathcal{N}_v^+$ and $\mathcal{N}_v^-$. The LLM is then asked a binary consistency question: *is the node text semantically consistent with its graph neighborhood?*

By jointly presenting graph-structural neighborhood semantics and textual neighborhood semantics in a unified representation, the explicit path enables the LLM to reason over the full graph-text context of each node without relying on hand-crafted rules or LLM-generated summaries.

### 4.3 Implicit Fusion Path

The implicit path fuses graph-text semantics into the LLM's parameter space through **Neighborhood Context Modulation (NCM)**. The core idea is to

transform each node's neighborhood semantics into a node-adaptive modulation signal that guides the language model to understand the correspondence between graph topology and textual semantics for each node.

Concretely, a GNN encoder first aggregates each node's textual feature $\mathbf{x}_v$ over its neighborhood structure via multi-hop message passing, producing a *neighborhood embedding* that jointly encodes text semantics and graph context. A learned NodeScaler then maps this embedding to a per-node scaling vector:

$$\mathbf{s}_v = \mathbf{1} + \lambda \tanh(\mathrm{MLP}(\mathrm{GNN}(\mathbf{x}_v))), \quad (2)$$

where $\lambda$ controls the maximum deviation from identity and $\mathbf{1}$ denotes the all-ones vector. This scaling vector modulates the intermediate representation of a low-rank parameter update $\Delta W = W_{\mathrm{up}} W_{\mathrm{down}}$ applied to the frozen LLM weights:

$$\Delta \mathbf{y}_v = W_{\mathrm{up}}(\mathbf{s}_v \odot W_{\mathrm{down}}\mathbf{z}), \quad (3)$$

where $\mathbf{z}$ is the layer's hidden state, $\Delta \mathbf{y}_v$ is the modulated output for node $v$, and $\odot$ denotes elementwise multiplication. At initialization, $\mathbf{s}_v = \mathbf{1}$, which recovers the standard parameter update. During training, NodeScaler learns to selectively amplify or suppress individual channels per node, creating a complementary path through which neighborhood semantics guide the LLM to understand the correspondence between graph topology and textual semantics beyond what the explicit input alone conveys. Although this form is related to feature-wise modulation and hypernetwork-style adapters (Perez et al., 2018; Lv et al., 2024), NCM differs in its conditioning signal and granularity. The modulation vector is generated separately for each node from that node's GNN neighborhood embedding, so different nodes in the same batch receive different parameter updates according to their local graph-text context.

### 4.4 Training and Inference

The LLM backbone is frozen. Only the low-rank weights $\{W_{\mathrm{up}}, W_{\mathrm{down}}\}$ and NodeScaler parameters are trainable, optimized jointly via cross-entropy over two answer tokens (`anomaly`, `normal`). At inference, the score for node $v$ is:

$$a_v = \frac{\exp(\ell_{\texttt{anomaly}})}{\exp(\ell_{\texttt{anomaly}}) + \exp(\ell_{\texttt{normal}})}, \quad (4)$$

where $\ell_{\texttt{anomaly}}$ and $\ell_{\texttt{normal}}$ are the logits of the two answer tokens given the assembled input $\mathrm{ctx}_v$, which comprises the node text $T_v$, graph-structural neighborhood semantics, and contrastive neighbor texts as described above.

All input information is constructed without validation or test labels. The GNN encoder trains only on labeled nodes $\mathcal{V}_L$. The explicit path uses only graph topology, node features, and raw text.

| Dataset | Nodes | Edges | Avg. len. | Anomalies | Ratio (%) |
|---|---|---|---|---|---|
| Citeseer | 3,186 | 3,432 | 153.94 | 128 | 4.02 |
| Pubmed | 19,717 | 90,368 | 256.08 | 788 | 4.00 |
| History | 41,551 | 369,252 | 228.36 | 1,662 | 4.00 |
| Photo | 48,362 | 512,933 | 150.25 | 1,934 | 4.00 |
| Computers | 87,229 | 742,792 | 93.16 | 3,490 | 4.00 |
| Children | 76,875 | 1,574,664 | 209.12 | 3,076 | 4.00 |
| ogbn-Arxiv | 169,343 | 1,210,112 | 179.70 | 6,774 | 4.00 |
| CitationV8 | 1,106,759 | 6,396,265 | 148.77 | 44,270 | 4.00 |

Table 1: Dataset statistics. Avg. len. = whitespace-tokenised document length. All attributes are 768-dim. frozen BGE embeddings. Anomaly ratio fixed at ≈4% (50% attribute-perturbed + 50% structure-perturbed, no overlap).

## 5 Experiments

### 5.1 Datasets

We evaluate on eight text-attributed graph anomaly detection benchmarks (Xu et al., 2025b): Citeseer, Pubmed, History, Photo, Computers, Children, ogbn-Arxiv, and CitationV8. The datasets cover citation and e-commerce graphs, spanning diverse text domains and graph scales from 3K to over 1M nodes. Full statistics are given in Table 1.

### 5.2 Baselines

To verify the effectiveness of our proposed method, we select several baseline models for comparison.

**GNN-based methods.** GCN (Kipf and Welling, 2016), GAT (Veličković et al., 2017), GIN (Xu et al., 2018), GraphSAGE (Hamilton et al., 2017), PC-GNN (Liu et al., 2021a), BWGNN (Tang et al., 2022), GHRN (Gao et al., 2023), XGBGraph and RFGraph (Chen and Guestrin, 2016), GAAP (Duan et al., 2025), and PMP (Zhuo et al., 2024). These methods classify on fixed embeddings without language model reasoning.

**Integrating LLMs with graphs.** TAPE (He et al., 2024), CoLL (Xu et al., 2025a), UNPrompt (Niu et al., 2024), GraphGPT (Tang et al., 2024), InstructGLM (Ye et al., 2024), and ConsisGAD (Chen et al., 2024a). These methods integrate language understanding with graph structure for node-level tasks.

| Method | Citeseer | Pubmed | History | Photo | Computers | Children | ogbn-Arxiv | CitationV8 | Avg. Rank |
|---|---|---|---|---|---|---|---|---|---|
| *GNN-based methods* | | | | | | | | | |
| GCN | 55.81 | 45.32 | 54.58 | 48.31 | 57.19 | 38.36 | 59.90 | 62.20 | 7.25 |
| GAT | 34.48 | 35.41 | 46.63 | 48.21 | 52.47 | 45.49 | 59.59 | 48.31 | 9.62 |
| GIN | 58.54 | 33.68 | 51.93 | 54.47 | **62.86** | **54.31** | 42.90 | 62.96 | 7.00 |
| GraphSAGE | **63.70** | **72.37** | 43.46 | 39.53 | 48.71 | 34.29 | 92.12 | 84.38 | **5.50** |
| PC-GNN | 42.42 | 61.99 | 45.88 | 31.74 | 45.63 | 30.90 | 90.39 | 82.27 | 7.25 |
| BWGNN | **61.54** | 47.11 | 21.96 | 18.52 | 34.71 | 19.90 | **94.52** | **85.16** | 8.00 |
| GHRN | 38.89 | 43.12 | 26.04 | 20.02 | 33.66 | 19.94 | **92.97** | **84.28** | 9.25 |
| XGBGraph | 52.89 | 59.44 | **60.88** | **48.97** | **57.34** | **38.18** | 89.99 | 72.24 | **5.00** |
| RFGraph | 47.06 | 42.09 | **59.64** | 45.60 | 54.44 | 34.17 | 78.35 | 63.24 | 7.62 |
| GAAP | 18.95 | 51.80 | 19.31 | 17.72 | 25.50 | 16.17 | 79.23 | 61.86 | 12.25 |
| PMP | 19.44 | 58.27 | 21.06 | 21.33 | 24.89 | 11.81 | 73.45 | 49.63 | 12.38 |
| *Integrating LLMs with graphs* | | | | | | | | | |
| TAPE | 58.54 | **76.97** | 49.68 | **57.82** | 53.44 | 48.36 | 65.28 | 68.21 | 5.25 |
| CoLL | 25.07 | 19.71 | 25.70 | 16.85 | 22.91 | 15.56 | 36.58 | 47.88 | 14.62 |
| UNPrompt | 23.18 | 16.14 | 26.73 | 11.86 | 14.30 | 11.82 | 37.65 | 35.63 | 15.12 |
| GraphGPT | 14.78 | 11.52 | 8.96 | 8.59 | 10.66 | 8.65 | 16.81 | 15.03 | 17.75 |
| InstructGLM | 9.21 | **61.34** | 12.95 | 21.70 | 27.51 | 12.91 | 64.60 | 63.81 | 12.00 |
| ConsisGAD | 13.48 | 43.88 | 17.10 | 21.11 | 26.94 | 9.93 | 73.40 | 17.28 | 14.12 |
| *Ours* | | | | | | | | | |
| **N2NSC** | **87.50** | **93.46** | **84.70** | **66.76** | **85.78** | **70.40** | **98.35** | **98.53** | **1.00** |
| Improve. | +23.80 | +16.49 | +23.82 | +8.94 | +22.92 | +16.09 | +3.83 | +13.37 | – |

Table 2: Anomaly-F1 (%) on eight TAG benchmarks. Performance comparison of GNN-based methods and LLM-graph methods. **First**, **second**, and **third** best results are color-coded.

### 5.3 Experimental Setup

We employ a stratified 60/20/20 split to partition the training, validation, and test sets, and all experiments are conducted on this shared partition. For GNN-based methods, we utilize the 768-dim node features generated by embedding the raw text of all nodes using LM-based BGE (Xiao et al., 2024). For methods that integrate LLMs with graph structures, we provide the raw text of each node as input along with the same BGE embeddings and graph topology. Given that integrating LLMs for TAG anomaly detection remains an emerging field, most LLM-graph baselines were originally designed for other tasks (such as node classification or link prediction) or different feature spaces (such as bag-of-words). We adapt them to a unified evaluation protocol to provide the most informative comparison possible (adaptation details in Appendix B).

Regarding our proposed method, we select Qwen3-8B (Yang et al., 2025) as the frozen LLM backbone and employ LoRA (Hu et al., 2022) with a rank of 64 for parameter-efficient training. The GNN encoder adopts a two-layer GAT architecture with 8 attention heads and an output dimension of 256. In the contrastive neighbor selection phase, both TopK and BottomK are set to $K=2$. During inference, N2NSC uses a single forward pass and reads the logits of the two answer tokens, without autoregressive generation.

### 5.4 Main Results

As shown in Table 2, N2NSC ranks 1st on all eight datasets with an average F1 of 85.69, surpassing the best baseline on each dataset by an average of +16.16 points. The largest gains appear on History (+23.82) and Computers (+22.92), where detecting anomalies requires jointly reasoning over textual and structural neighborhood semantics. Even on ogbn-Arxiv, where spectral-based GNN methods already achieve above 90, N2NSC still improves by +3.83. The F1 gains should be interpreted together with the threshold-free metrics in Appendix C. On highly imbalanced graphs with about 4% anomalies, AUROC can saturate once most normal nodes are ranked correctly; AUPRC focuses more directly on the anomaly-relevant region of the ranking and shows larger gains on most datasets. We report mean and standard deviation over three random seeds in Appendix F.

Both existing paradigms exhibit fundamental limitations when compared with N2NSC. **(1) GNN-based methods show high instability across datasets**, with average ranks ranging from 5.0 to 12.4. For example, BWGNN ranks 2nd on ogbn-Arxiv but drops to 13th on History, while XGBGraph ranks 2nd on History but only 6th on ogbn-Arxiv. This instability stems from their reliance on static embeddings, which discard the fine-grained textual cues needed for assessing node-to-

neighborhood semantic consistency. **(2) Methods integrating LLMs perform even worse overall.** Despite having access to language understanding, these methods inject graph information only as shallow auxiliary signals, failing to capture the correspondence between graph topology and textual semantics. **(3)** In contrast, **N2NSC achieves stable rank-1 performance** through deep fusion of both modalities via the explicit and implicit paths.

### 5.5 Ablation

We validate each component of N2NSC by progressively adding them and observing the effect. Table 3 reports results on Citeseer and Pubmed, and Table 11 reports full eight-dataset results.

**Problem. Text alone cannot reveal neighborhood inconsistency.** A text-only LLM achieves only 70.15 AUROC on Citeseer and 73.78 on Pubmed, because anomalous nodes are textually plausible in isolation. Adding neighborhood-derived statistics lifts AUROC to 92.74 and 97.91, respectively, demonstrating that graph-structural context is essential for detecting node-to-neighborhood semantic inconsistency. Feature deviation alone is not always reliable: on heterogeneous neighborhoods such as Citeseer, a large deviation may indicate either an anomaly or simply a diverse local context. Its benefit appears when combined with degree and neighbor similarity, where the full statistics recover this ambiguity.

**Cause 1. Contrastive neighbor text is effective.** Neighborhood statistics tell the LLM *that* something is inconsistent, but not *what* the inconsistency is. Adding contrastive neighbor text enables direct semantic comparison. To confirm this, we compare against two controls where the neighbor text is replaced with randomly sampled or shuffled content while keeping NCM active. As shown in Table 11, both controls degrade substantially, confirming that the quality of neighbor selection matters. Furthermore, using only dissimilar neighbors outperforms using only similar neighbors on most datasets, because dissimilar neighbors directly expose the semantic conflicts that characterize anomalies.

**Cause 2. NCM provides complementary gains.** The explicit path captures what can be expressed in text, but some anomaly patterns manifest as distributional shifts across the neighborhood that text alone cannot convey. NCM addresses this by transforming neighborhood semantics into node-adaptive modulation signals. As shown in Tables 9 and 10, adding NCM consistently improves threshold-free metrics across all eight datasets, with the largest gains on large-scale graphs where distributional patterns are richer. Figure 3 provides mechanistic confirmation on ogbn-Arxiv: Panel (a) shows that normal nodes receive consistently larger per-channel scaling deviations than anomaly nodes, Panel (b) reveals that the discriminative power concentrates in mid-to-late Transformer blocks, and Panel (c) confirms the separation at the per-node level. An analysis is provided in Appendix D.

### 5.6 Sensitivity Analysis

We analyze the sensitivity of N2NSC along its two fusion paths on Citeseer and Pubmed. All results are reported as mean$\pm$std across the tested values of each hyperparameter.

**Explicit fusion path. Neighbor count $K$.** The explicit path selects the $K$ most similar and $K$ most dissimilar neighbors as contrastive evidence. Across $K \in \{1, 2, 3, 4\}$, AUROC is $91.34\pm0.50$ on Citeseer and $99.57\pm0.19$ on Pubmed, indicating that performance is highly stable regardless of the number of selected neighbors. $K$=2 achieves the best overall balance and is used as the default.

**Implicit fusion path. Scaling strength $\lambda$.** NCM modulates the parameter space via $\mathbf{s}_v = \mathbf{1} + \lambda \tanh(\cdot)$, where $\lambda$ controls the maximum deviation from identity. Across $\lambda \in \{0.1, 0.5, 1.0, 2.0\}$, AUROC is $92.01\pm0.31$ on Citeseer and $99.62\pm0.09$ on Pubmed. On Citeseer, all four values yield identical F1 of 89.80. This confirms that the residual initialization provides a stable optimization landscape regardless of the maximum modulation magnitude.

**Implicit fusion path. Modulation dimension $r$.** The dimension $r$ determines the number of channels that NCM modulates. Across $r \in \{16, 32, 64, 128\}$, performance on Pubmed is stable with AUROC $99.57\pm0.11$ and F1 $93.27\pm1.07$. On Citeseer, $r$=128 overfits with F1 dropping to 80.0, consistent with limited training data of approximately 1,900 nodes. Excluding this outlier, the remaining three values yield AUROC $92.57\pm1.68$ and F1 $90.09\pm1.47$. This motivates the default $r$=64 as a trade-off between expressiveness and generalization across graph scales.

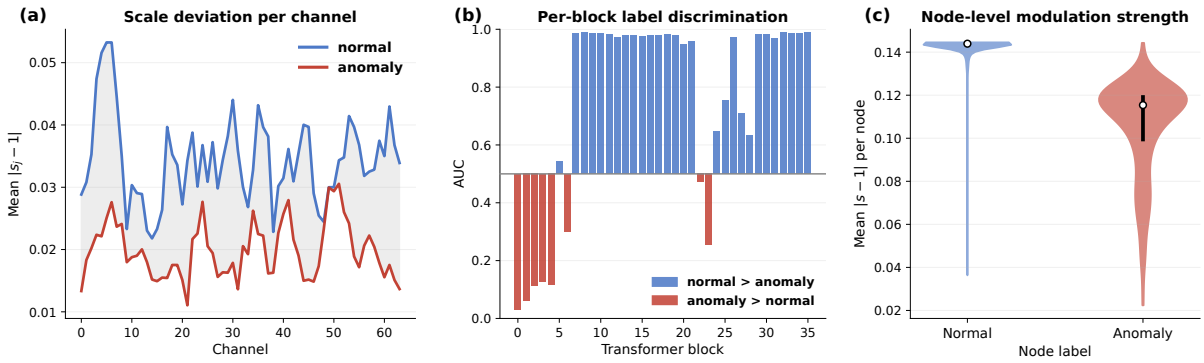


Figure 3: Mechanistic analysis of NCM on ogbn-Arxiv. **(a)** Per-channel mean $|s_j - 1|$ averaged over the top-5 most discriminative layers. Normal nodes receive consistently different modulation from anomaly nodes. **(b)** Per-block discriminative power. Bars extend upward from the 0.5 midline for blocks where normal nodes have larger deviation, downward where anomaly nodes do. **(c)** Violin plot of per-node mean $|s-1|$, showing that normal nodes receive systematically stronger modulation than anomaly nodes.

| | | | | | Citeseer | | Pubmed | |
|---|---|---|---|---|---|---|---|---|
| Variant | Deg | Cos | Feat | NCM | AUROC | AUPRC | AUROC | AUPRC |
| Text only | × | × | × | × | 70.15 | 44.55 | 73.78 | 52.94 |
| Text + Degree | ✓ | × | × | × | 91.74 | 86.54 | 95.28 | 67.46 |
| Text + CosSim | × | ✓ | × | × | 90.94 | 71.24 | 84.71 | 58.81 |
| Text + FeatDev | × | × | ✓ | × | 65.83 | 44.96 | 80.03 | 53.87 |
| Text + Deg&Cos | ✓ | ✓ | × | × | 90.62 | 85.52 | 98.20 | 86.79 |
| Text + All | ✓ | ✓ | ✓ | × | 92.74 | 85.93 | 97.91 | 86.70 |
| Text + All + NCM | ✓ | ✓ | ✓ | ✓ | **93.28** | **86.06** | **98.44** | **88.19** |

Table 3: Feature ablation on Citeseer and Pubmed without contrastive neighbor text. All variants include node text as the base input. Deg = degree, Cos = neighbor similarity, Feat = feature deviation, NCM = Neighborhood Context Modulation. All values in %.

### 5.7 Discussion

**Two fusion paths, one consistency objective.** The ablation validates the two-path design. Graph-neighborhood semantic cues yield +22 AUROC over text-only on Citeseer in Table 3. Textual neighborhood semantics outperform random and shuffled controls by +4 to +15 F1 in Table 11. NCM adds +7 to +20 AUPRC on large graphs in Table 10. Removing any single component causes degradation, confirming that the three components are complementary. NCM is most valuable when explicit evidence is absent or insufficient, reinforcing its role as a complementary fusion path.

**Why extractive rather than aggregate?** We compare with a variant summarizing neighbor text statistically. On Pubmed, the extractive approach achieves 93.46 F1 vs. 89.23 for aggregation, because neighbor text preserves fine-grained semantic patterns that summaries discard.

**Distinction from retrieval-augmented generation.** Standard RAG retrieves globally relevant documents to supply missing knowledge. N2NSC retrieves only from the target node's graph neighborhood to assess node-to-neighborhood semantic consistency. The Random and Shuffled controls confirm this distinction.

**Scalability.** N2NSC scales to graphs with over one million nodes. On CitationV8 (1.1M nodes, 6.4M edges), the framework achieves 98.53 F1. The GNN encoder processes neighborhoods locally, and the LLM evaluates each node with fixed-length input and no autoregressive generation. With batch size 8 on a single GPU, per-node inference latency is 49.7ms on Citeseer, 78.3ms on Pubmed, and 74.6ms on ogbn-Arxiv. Total test-set inference takes about 32s, 5.1min, and 42min on the three datasets, respectively. Extrapolating from the same measurement, the CitationV8 test split (221,353 nodes) takes about 4.6h on one GPU. Training time and peak memory are reported in Appendix G.

| Dataset | Method | AUROC | AUPRC | Macro-F1 |
|---|---|---|---|---|
| YelpReviews | DGP | 84.39 | 49.69 | 69.30 |
| | **N2NSC** | **89.47** | **55.34** | **71.19** |
| Amazon Video | DGP | 77.43 | 34.87 | 67.04 |
| | **N2NSC** | **83.16** | **41.52** | **73.28** |

Table 4: Results on real-world fraud datasets.

**Generalization to real-world fraud.** We further evaluate N2NSC on two real-world fraud datasets, Amazon Video and YelpReviews (Dou et al., 2020), where anomalies arise naturally rather than through synthetic injection. As shown in Table 4, N2NSC outperforms DGP (Li et al., 2026) by $+5$ to $+6$ AUROC and $+5$ to $+7$ AUPRC. These results provide preliminary evidence beyond the synthetic protocol, while broader evaluation on naturally occurring anomalies remains an important direction.

## 6 Conclusion

We formalize TAG anomaly detection as a node-to-neighborhood semantic consistency problem and propose N2NSC, a framework addressing it through two complementary fusion paths. The explicit path fuses textual and graph-structural neighborhood semantics into a unified representation. The implicit path transforms neighborhood semantics into node-adaptive modulation signals that guide the model to capture the correspondence between graph topology and textual semantics for each node. Meanwhile, our work reveals that capturing the correspondence between graph topology and textual semantics is essential for reasoning over text-attributed graphs. Experiments across eight benchmark datasets and two real-world fraud datasets validate the effectiveness of N2NSC.

## Limitations

This work introduces a framework that deeply fuses graph-structural semantics with textual semantics for anomaly detection on text-attributed graphs. However, the eight main benchmarks use a controlled synthetic anomaly protocol, which cannot fully cover the diversity of naturally occurring anomaly distributions. Moreover, the framework primarily focuses on anomaly detection and cannot be directly applied to generative tasks such as graph captioning, nor has it been evaluated on other discriminative tasks like node classification. Additionally, the explicit fusion path depends on the availability of neighborhood context, and its effectiveness is reduced for isolated nodes with very few graph neighbors. For truly isolated nodes, the method falls back to text-only classification. Leveraging this framework to construct a graph foundation model presents a challenging yet valuable area for future exploration.

## Ethics Considerations

This work addresses graph anomaly detection for applications such as fraud detection and academic integrity verification. All experiments are conducted on publicly available benchmark datasets. The datasets do not contain personally identifiable information, and no human subjects are involved in this research. Deployments of anomaly detection systems should include appropriate oversight to mitigate risks such as surveillance and discriminatory profiling.

## Acknowledgments

This work is sponsored by CCF-Tencent Rhino-Bird Open Research Fund (CCF-Tencent RAGR20250128).

## References

Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

Bo Chen, Jing Zhang, Xiaokang Zhang, Yuxiao Dong, Jian Song, Peng Zhang, Kaibo Xu, Evgeny Kharlamov, and Jie Tang. 2022. Gccad: Graph contrastive coding for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8037–8051.

Nan Chen, Zemin Liu, Bryan Hooi, Bingsheng He, Rizal Fathony, Jun Hu, and Jia Chen. 2024a. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In *The twelfth international conference on learning representations*.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. 2024b. LLaGA: Large language and graph assistant. In *International Conference on Machine Learning*.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM international conference on data mining*, pages 594–602. SIAM.

Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*.

Mingjiang Duan, Da He, Tongya Zheng, Lingxiang Jia, Mingli Song, Xinyu Wang, and Zunlei Feng. 2025. Global attribute-association pattern aggregation for graph fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11616–11624.

Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM web conference 2023*, pages 1528–1538.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *International conference on learning representations*, volume 2024, pages 5711–5732.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

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*, 33:22118–22133.

Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. 2021. Anemone: Graph anomaly detection with multi-scale contrastive learning. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 3122–3126.

Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. 2025. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Yuan Li, Jun Hu, Bryan Hooi, Bingsheng He, and Cheng Chen. 2026. Dgp: A dual-granularity prompting framework for fraud detection with graph-enhanced llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 15171–15179.

Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Victor W Chan, and Jia Li. 2024. Glbench: A comprehensive benchmark for graph with large language models. *Advances in Neural Information Processing Systems*, 37:42349–42368.

Xudong Liu, Yanan Ren, Hengtong Zhang, Run-An Wang, Shenghe Zheng, and Zhaonian Zou. 2025. Towards anomaly detection on text-attributed graphs. *OpenReview preprint*.

Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021a. Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference 2021*, pages 3168–3177.

Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. 2021b. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33(6):2378–2392.

Chuancheng Lv, Lei Li, Shitou Zhang, Gang Chen, Fanchao Qi, Ningyu Zhang, and Hai-Tao Zheng. 2024. HyperLoRA: Efficient cross-task generalization via constrained low-rank adapters generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. 2023. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12012–12038.

Xiaoxu Ma, Dong Li, Minglai Shao, Xintao Wu, and Chen Zhao. 2026. Llm-enhanced energy contrastive learning for out-of-distribution detection in text-attributed graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24308–24316.

Chaoxi Niu, Hezhe Qiao, Changlu Chen, Ling Chen, and Guansong Pang. 2024. Zero-shot generalist graph anomaly detection with unified neighborhood prompts. *arXiv preprint arXiv:2410.14886*.

Yunhe Pang, Bo Chen, Fanjin Zhang, Yanghui Rao, Evgeny Kharlamov, and Jie Tang. 2025. Guard: Effective anomaly detection through a text-rich and graph-informed language model. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2222–2233.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Amit Roy, Juan Shu, Jia Li, Carl Yang, Olivier Elshocht, Jeroen Smeets, and Pan Li. 2024. Gad-nr: Graph anomaly detection via neighborhood reconstruction.

In *Proceedings of the 17th ACM international conference on web search and data mining*, pages 576–585.

Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. 2007. Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering*, 19(5):631–645.

Guangxin Su, Hanchen Wang, Jianwei Wang, Wenjie Zhang, Ying Zhang, and Jian Pei. 2025. Large language models meet text-attributed graphs: A survey of integration frameworks and applications. *arXiv preprint arXiv:2510.21131*.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 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*, pages 491–500.

Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. 2023. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. *Advances in Neural Information Processing Systems*, 36:29628–29653.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. 2022. Rethinking graph neural networks for anomaly detection. In *International conference on machine learning*, pages 21076–21089. PMLR.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Yaoke Wang, Yun Zhu, Wenqiao Zhang, Yueting Zhuang, Liyunfei Liyunfei, and Siliang Tang. 2024. Bridging local details and global context in text-attributed graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14830–14841.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Ruiyao Xu and Kaize Ding. 2024. Large language models for anomaly and out-of-distribution detection: A survey. *arXiv preprint arXiv:2409.01980*.

Yiming Xu, Jiarun Chen, Zhen Peng, Zihan Chen, Qika Lin, Lan Ma, Bin Shi, and Bo Dong. 2025a. Court of LLMs: Multi-LLM collaboration for text-attributed graph anomaly detection. *arXiv preprint arXiv:2508.00507*.

Yiming Xu, Xu Hua, Zhen Peng, Bin Shi, Jiarun Chen, Xingbo Fu, Song Wang, and Bo Dong. 2025b. Text-attributed graph anomaly detection via multi-scale cross- and uni-modal contrastive learning. *arXiv preprint arXiv:2508.00513*.

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

An Yang, Anfeng Li, Baosong Yang, et al. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2024. Language is all a graph needs. In *Findings of the association for computational linguistics: EACL 2024*, pages 1955–1973.

Wei Zhuo, Zemin Liu, Bryan Hooi, Bingsheng He, Guang Tan, Rizal Fathony, and Jia Chen. 2024. Partitioning message passing for graph fraud detection. *arXiv preprint arXiv:2412.00020*.

## A Dataset Descriptions

We evaluate on eight text-attributed graph anomaly detection benchmarks introduced by Xu et al. (2025b), spanning citation networks and Amazon e-commerce networks. All datasets share the same anomaly injection protocol (4% ratio, 50% attribute-perturbed + 50% structure-perturbed, no overlap) and are represented as 768-dimensional frozen BGE embeddings (Xiao et al., 2024).

**Citeseer.** A citation network of 3,186 computer science publications connected by 3,432 citation links. Each node carries the paper title and abstract as its text attribute, with an average document length of 154 tokens. The graph is sparse (avg. degree $\approx$2.2), making neighborhood evidence limited for low-degree nodes.

**Pubmed.** A biomedical citation network of 19,717 publications and 90,368 citation edges. Node texts are paper titles and abstracts from the PubMed database, averaging 256 tokens. The domain-specific vocabulary provides strong lexical cues for neighborhood consistency.

**History.** An Amazon e-commerce graph of 41,551 products in the Books→History category, connected by 369,252 co-purchase edges. Node texts are book titles and descriptions, averaging 228 tokens. The graph is denser than citation networks (avg. degree $\approx$17.8).

**Photo.** An Amazon e-commerce graph of 48,362 products in the Electronics→Photo category with 512,933 co-purchase edges. Node texts are high-rated reviews and product summaries, averaging 150 tokens.

**Computers.** An Amazon e-commerce graph of 87,229 products in the Electronics→Computers category with 742,792 co-purchase edges. Node texts are high-rated reviews and product summaries, averaging 93 tokens (the shortest among all datasets).

**Children.** An Amazon e-commerce graph of 76,875 products in the Books→Children category with 1,574,664 co-purchase edges. Node texts are book titles and descriptions, averaging 209 tokens. This is the densest graph in our benchmark (avg. degree ≈41.0), providing rich neighborhood context.

**ogbn-Arxiv.** A large-scale citation network of 169,343 computer science papers from the Open Graph Benchmark (Hu et al., 2020) with 1,210,112 citation edges. Node texts are paper titles and abstracts averaging 180 tokens. The graph covers multiple CS sub-fields, creating natural topical clusters.

**CitationV8.** The largest dataset in our benchmark, containing 1,106,759 papers and 6,396,265 citation edges from the DBLP Citation Network V8. Node texts are paper titles and abstracts averaging 149 tokens. This dataset tests scalability to million-node graphs.

## B Baseline Descriptions

We compare against 17 baseline methods from two families. Since the benchmark (Xu et al., 2025b) is recent and no prior work has reported results on all eight datasets under identical conditions, we adapt each baseline to the unified evaluation protocol (same 768-dim BGE node features, same 60/20/20 split, same threshold calibration on validation). For methods originally designed for different tasks or feature spaces, we describe the adaptation below. For recent methods without reproducible code or complete benchmark results, we include comparisons where directly comparable results are available.

### B.1 GNN-Based Methods

These methods classify on fixed 768-dim BGE node embeddings and train on the same split as N2NSC. We use official implementations where available and tune hyperparameters on the validation set.

**GCN (Kipf and Welling, 2016).** Graph Convolutional Network with symmetric normalized aggregation. We use a two-layer architecture and tune hidden dimension on the validation set.

**GAT (Veličković et al., 2017).** Graph Attention Network using multi-head attention for adaptive neighbor weighting. Two-layer architecture with hyperparameters tuned per dataset.

**GIN (Xu et al., 2018).** Graph Isomorphism Network with sum aggregation and learnable epsilon. Designed to be as expressive as the Weisfeiler-Leman graph isomorphism test.

**GraphSAGE (Hamilton et al., 2017).** Inductive representation learning with mean aggregation and neighbor sampling. Uses mini-batch training suitable for large graphs.

**PC-GNN (Liu et al., 2021a).** Pick and Choose GNN designed for imbalanced graph classification. Uses label-balanced neighbor sampling to address the severe class imbalance (4% anomaly ratio). We use the official implementation and adapt it to BGE features.

**BWGNN (Tang et al., 2022).** Beta Wavelet GNN with spectrally localized band-pass filters. Originally designed for lower-dimensional features. We adapt it to 768-dim BGE embeddings and tune hyperparameters per dataset.

**GHRN (Gao et al., 2023).** Graph Heterophily-aware Residual Network that prunes inter-class edges by emphasizing high-frequency components. Adapted to the benchmark with the same features and split.

**XGBGraph and RFGraph (Chen and Guestrin, 2016).** Gradient-boosted trees (XGBoost) and Random Forest. We concatenate node features with hand-crafted graph statistics (degree, clustering coefficient, PageRank) as input, following the GADBench protocol (Tang et al., 2023).

**GAAP (Duan et al., 2025).** Global Attribute-Association Pattern aggregation for graph fraud detection. Captures high-order attribute association patterns among nodes for anomaly scoring. We adapt it to BGE features on the benchmark split.

**PMP (Zhuo et al., 2024).** Partitioning Message Passing for graph fraud detection. Partitions the message-passing process to separate benign and fraudulent propagation patterns. Adapted to the benchmark with the same features and split.

### B.2 Methods Integrating LLMs with Graphs

These methods were originally designed for different datasets or tasks. We adapt them to the benchmark by providing the same BGE features and graph topology under the unified split. Since TAG anomaly detection with LLMs is a nascent area, most methods below are adapted from related graph-LLM tasks rather than reproduced from native TAG-GAD settings.

**TAPE (He et al., 2024).** Originally proposed for node classification. Uses LLM-generated explanations as enhanced node features for downstream GNN training. We adapt it by replacing the classification objective with binary anomaly detection on the benchmark split using official source code.

**CoLL (Xu et al., 2025a).** Multi-LLM collaboration for TAG anomaly detection. This is one of the few methods natively designed for this task. We use official source code with the same benchmark data and split.

**UNPrompt (Niu et al., 2024).** Learns unified graph prompts for zero-shot GAD. Originally pretrains a GCN backbone on Facebook and transfers to target graphs. We follow the original protocol (pre-train on Facebook, fine-tune prompts on the benchmark train split). Note that UNPrompt reduces features to 8-dim via SVD, discarding most of the 768-dim BGE information.

**GraphGPT (Tang et al., 2024).** Graph instruction-tuning framework originally for node classification and link prediction. We adapt it to anomaly detection by reformulating the task as binary classification.

**InstructGLM (Ye et al., 2024).** Instruction-tuned graph language model that converts graph structures into natural language descriptions. Adapted to the anomaly detection task on the benchmark data.

**ConsisGAD (Chen et al., 2024a).** Consistency-based graph anomaly detection that leverages consistency between multiple augmented views of graph data for anomaly scoring. We use the provided implementation on the benchmark features.

## C Additional Results

Tables 7 and 8 report threshold-free AUROC and AUPRC for N2NSC and all baseline methods, providing a complementary view to the threshold-dependent Anomaly-F1 comparison in Table 2.

## D NodeScaler Analysis

The three-panel NodeScaler visualization (Figure 3 in the main body) provides mechanistic evidence that NCM learns meaningful, label-discriminative modulation rather than collapsing to the standard parameter update. Panel (a) plots mean $|s_j - 1|$ per channel averaged over the top-5 most discriminative layers. Normal nodes exhibit consistently higher deviation than anomaly nodes across all 64 channels, indicating that NCM selectively amplifies channel activations for normal nodes while leaving anomaly nodes closer to the default behavior. Panel (b) shows the per-block discriminative power of NodeScaler deviations. The majority of mid-to-late Transformer blocks are strongly discriminative, with normal nodes receiving larger modulation, while a small number of early blocks show the opposite direction, revealing a structured, depth-dependent modulation pattern. Panel (c) presents the violin plot of per-node mean $|s-1|$, confirming that normal nodes receive systematically stronger modulation than anomaly nodes.

## E Qualitative Case Study

We present a case from Pubmed illustrating how N2NSC detects an anomaly that the text-only variant misses.

**Setup.** Node 7229 is a structure-perturbed anomaly (degree 6, avg. neighbor similarity 0.30). Its text is a meta-analysis of the Gly482Ser variant in the PPARGC1A gene in type 2 diabetes. The two most similar neighbors (similarity 0.51 and 0.47) discuss closely related PPARGC1A genetics. However, the two most dissimilar neighbors (similarity $-0.10$ and $-0.08$) discuss cerebral blood flow measurement and herbal extracts for dyslipidemia, topics unrelated to the target node's genetics focus.

**Text-only prediction.** Without neighborhood context, the LLM assigns a near-zero anomaly score (0.0032) and predicts *normal*, because the abstract is well-formed and topically coherent in isolation.

| Dataset | Train | Train Mem. | Infer Mem. | Latency | Test |
|---|---|---|---|---|---|
| Citeseer | 21.7min | 42.5GB | 20.3GB | 49.7ms | 32s |
| Pubmed | 3.5h | 42.6GB | 21.8GB | 78.3ms | 5.1min |
| ogbn-Arxiv | 28.5h | 44.7GB | 22.0GB | 74.6ms | 42min |
| CitationV8 | 184.3h | 45.8GB | 21.5GB | 71.7ms | 4.6h |

Table 5: Efficiency statistics measured on GPUs.

| Dataset | AUROC | AUPRC | Anomaly-F1 |
|---|---|---|---|
| Citeseer | 91.91±0.38 | 86.11±0.09 | 86.36±1.84 |
| Pubmed | 99.63±0.09 | 97.66±0.58 | 93.02±0.65 |
| History | 96.06±0.20 | 85.99±0.59 | 84.30±0.75 |
| Photo | 95.43±0.10 | 73.03±0.57 | 66.54±0.28 |
| Computers | 97.84±0.15 | 89.23±0.42 | 85.12±1.21 |
| Children | 93.57±0.33 | 71.19±0.68 | 69.49±1.45 |
| ogbn-Arxiv | 99.92±0.01 | 99.71±0.01 | 98.19±0.23 |
| CitationV8 | 99.96±0.02 | 99.74±0.08 | 98.29±0.35 |

Table 6: Training stability of N2NSC across three random seeds. All values are in %.

**N2NSC prediction.** With the explicit fusion path, the LLM receives both similar and dissimilar neighbor texts alongside graph statistics. The low average neighbor similarity (0.30) combined with the distant dissimilar neighbors enables the model to identify that this node's PPARGC1A genetics content is inconsistent with its neighborhood context. The model assigns a high anomaly score (0.9844) and correctly predicts *anomaly*.

## F Training Stability

This case demonstrates that anomalies arising from neighborhood semantic inconsistency require neighborhood-level semantic reasoning that is unavailable to methods operating on node text alone.

### F.1 Explicit Fusion Path Ablation

Table 11 reports Anomaly-F1 for all ablation variants across the eight datasets. The core variants (Text, Text+NCM, Stats, Stats+NCM, Explicit, and N2NSC) and the contrastive-neighbor controls (SimOnly+NCM, DissimOnly+NCM, Random+NCM, Shuffled+NCM) all have results on the full set of eight datasets.

## G Efficiency Details

Table 5 reports training and inference costs on representative datasets. The reported training time is for the full training schedule, estimated from one measured epoch. Inference is measured with batch size 8 on 200 sampled nodes and uses a single forward pass without generation. All entries use 185M trainable parameters, accounting for 2.21% of the Qwen3-8B backbone.

Table 6 reports mean and standard deviation across three random seeds. The contrastive neighbor selection is deterministic given the graph and frozen BGE embeddings, so the variation mainly comes from parameter initialization and mini-batch optimization. The AUROC standard deviation is below 0.38 on all eight datasets.

## H The Use of Large Language Models

For manuscript preparation, large language models were used only for grammatical polishing and stylistic refinement, aimed at enhancing the clarity and readability of our presentation of results and conclusions.

| Method | Citeseer | Pubmed | History | Photo | Computers | Children | ogbn-Arxiv | CitationV8 | Avg. Rank |
|---|---|---|---|---|---|---|---|---|---|
| *GNN-based methods* | | | | | | | | | |
| GCN | **90.04** | 91.61 | 78.20 | 78.48 | 81.81 | 75.65 | 95.00 | 90.01 | 7.38 |
| GAT | 78.30 | 92.43 | 78.24 | 79.07 | 79.70 | 76.04 | 98.09 | 78.22 | 8.00 |
| GIN | 88.38 | 81.97 | 73.80 | 74.17 | 76.61 | 75.58 | 96.22 | 91.98 | 9.50 |
| GraphSAGE | 90.21 | **96.18** | 81.09 | 80.29 | **85.97** | **79.10** | 99.77 | 98.46 | **3.50** |
| PC-GNN | 86.68 | 92.28 | **84.04** | **79.80** | **86.69** | **79.18** | 99.43 | 98.25 | **4.75** |
| BWGNN | 84.63 | 84.08 | 75.10 | 73.74 | 79.44 | 71.31 | **99.78** | **98.74** | 7.75 |
| GHRN | 74.53 | 78.60 | 74.85 | 73.39 | 77.64 | 70.27 | 99.68 | **98.76** | 9.12 |
| XGBGraph | 88.72 | 92.83 | 79.34 | 78.66 | 85.59 | 77.64 | 99.56 | 96.49 | 5.38 |
| RFGraph | 88.20 | 82.84 | 79.39 | 76.90 | 81.50 | 75.54 | 98.50 | 91.59 | 7.75 |
| GAAP | 63.16 | 82.19 | 67.36 | 65.66 | 69.92 | 65.23 | 87.10 | 85.34 | 13.38 |
| PMP | 61.21 | 84.37 | 66.83 | 66.63 | 72.69 | 60.79 | 86.52 | 82.83 | 13.62 |
| *Integrating LLMs with graphs* | | | | | | | | | |
| TAPE | **91.87** | **98.30** | **87.38** | **88.84** | 89.24 | 84.53 | 90.65 | 97.42 | 3.62 |
| CoLL | 71.44 | 74.41 | 76.95 | 72.88 | 75.33 | 68.79 | 84.39 | 82.61 | 12.62 |
| UNPrompt | 73.90 | 67.67 | 74.09 | 59.62 | 63.92 | 64.09 | 82.55 | 80.21 | 14.88 |
| GraphGPT | 68.84 | 60.31 | 56.11 | 54.54 | 60.45 | 54.90 | 68.71 | 66.67 | 17.50 |
| InstructGLM | 50.77 | 72.62 | 59.04 | 62.63 | 64.55 | 60.72 | 73.65 | 74.93 | 16.50 |
| ConsisGAD | 63.32 | 77.89 | 66.93 | 67.24 | 70.01 | 58.42 | 86.59 | 69.59 | 14.75 |
| *Ours* | | | | | | | | | |
| **N2NSC** | **92.08** | **99.70** | **96.19** | **95.49** | **97.93** | **93.85** | **99.93** | **99.98** | **1.00** |
| Improve. | +0.21 | +1.40 | +8.81 | +6.65 | +8.69 | +9.32 | +0.15 | +1.22 | – |

Table 7: AUROC comparison across the eight benchmark datasets. **First**, **second**, and **third** best results are color-coded.

| Method | Citeseer | Pubmed | History | Photo | Computers | Children | ogbn-Arxiv | CitationV8 | Avg. Rank |
|---|---|---|---|---|---|---|---|---|---|
| *GNN-based methods* | | | | | | | | | |
| GCN | **67.37** | 54.63 | 50.68 | 48.86 | 54.24 | 34.78 | 69.11 | 65.73 | 6.12 |
| GAT | 27.49 | 50.24 | 42.09 | 45.66 | 52.11 | **39.00** | 81.94 | 43.04 | 8.12 |
| GIN | 62.39 | 45.16 | 49.87 | **51.39** | **56.88** | **51.79** | 75.55 | 68.21 | 5.88 |
| GraphSAGE | **68.10** | **76.82** | 45.38 | 38.46 | 50.76 | 33.77 | **98.19** | **91.09** | **4.75** |
| PC-GNN | 46.19 | 65.88 | 42.58 | 28.16 | 45.66 | 28.76 | **95.72** | **88.48** | 7.00 |
| BWGNN | 61.62 | 46.49 | 16.28 | 15.93 | 29.44 | 13.34 | 98.41 | 91.11 | 8.25 |
| GHRN | 39.81 | 40.64 | 17.50 | 16.37 | 27.83 | 13.49 | 97.58 | 90.84 | 9.38 |
| XGBGraph | 56.30 | 61.74 | **53.98** | **45.69** | **56.96** | 35.51 | 96.33 | 78.94 | **4.50** |
| RFGraph | 60.47 | 33.68 | **53.34** | 41.32 | 52.34 | 29.92 | 86.46 | 66.28 | 7.38 |
| GAAP | 12.30 | 51.73 | 11.68 | 11.16 | 20.03 | 9.13 | 75.73 | 61.65 | 12.25 |
| PMP | 9.45 | 57.95 | 11.80 | 13.94 | 18.51 | 6.27 | 71.66 | 47.67 | 13.12 |
| *Integrating LLMs with graphs* | | | | | | | | | |
| TAPE | 63.26 | **83.61** | 40.20 | 40.55 | 45.30 | 27.12 | 31.67 | 73.09 | 7.88 |
| CoLL | 17.49 | 11.38 | 17.56 | 10.59 | 14.32 | 8.52 | 27.44 | 39.65 | 14.50 |
| UNPrompt | 17.47 | 9.97 | 22.10 | 7.23 | 9.69 | 8.07 | 36.63 | 34.62 | 14.88 |
| GraphGPT | 7.06 | 5.82 | 4.75 | 4.61 | 5.69 | 4.56 | 9.84 | 8.62 | 17.88 |
| InstructGLM | 4.24 | 51.52 | 7.51 | 15.74 | 20.55 | 7.15 | 53.47 | 54.08 | 13.75 |
| ConsisGAD | 9.93 | 42.74 | 9.90 | 14.31 | 20.89 | 4.83 | 71.38 | 15.11 | 14.38 |
| *Ours* | | | | | | | | | |
| **N2NSC** | **86.18** | **98.18** | **86.25** | **73.45** | **89.41** | **71.53** | **99.71** | **99.78** | **1.00** |
| Improve. | +18.08 | +14.57 | +32.27 | +22.06 | +32.45 | +19.74 | +1.30 | +8.67 | – |

Table 8: AUPRC comparison across the eight benchmark datasets. **First**, **second**, and **third** best results are color-coded.

| Variant | Citeseer | Pubmed | History | Photo | Computers | Children | ogbn-Arxiv | CitationV8 |
|---|---|---|---|---|---|---|---|---|
| Text | 70.24 | 74.68 | 71.21 | 75.27 | 71.63 | 70.50 | 74.09 | 74.87 |
| Text+NCM | 69.69 | 77.34 | 71.26 | 75.59 | 72.14 | 90.27 | 99.33 | 99.60 |
| Stats | 91.79 | 98.04 | 94.21 | 91.36 | 94.08 | 84.16 | 99.07 | 99.58 |
| Stats+NCM | 92.77 | 98.23 | 94.15 | 90.68 | 97.28 | 91.54 | 99.79 | 99.91 |
| Explicit | 91.98 | 99.68 | 96.12 | 95.32 | 96.82 | 93.05 | 99.88 | 99.97 |
| **N2NSC** | **92.08** | **99.70** | **96.19** | **95.49** | **97.93** | **93.85** | **99.93** | **99.98** |
| SimOnly+NCM | 92.27 | 98.49 | 94.22 | 91.27 | 97.25 | 91.69 | 99.82 | 99.95 |
| DissimOnly+NCM | 91.23 | 99.41 | 96.15 | 95.24 | 97.98 | 93.67 | 99.91 | 99.98 |
| Random+NCM | 90.96 | 98.74 | 95.12 | 95.94 | 98.01 | 91.66 | 99.85 | 99.97 |
| Shuffled+NCM | 91.09 | 98.11 | 92.98 | 90.71 | 97.66 | 91.05 | 99.76 | 99.93 |

Table 9: Full ablation results (AUROC). Same variant naming as Table 11.

| Variant | Citeseer | Pubmed | History | Photo | Computers | Children | ogbn-Arxiv | CitationV8 |
|---|---|---|---|---|---|---|---|---|
| Text | 44.62 | 52.94 | 41.87 | 49.16 | 46.53 | 39.88 | 53.19 | 53.99 |
| Text+NCM | 44.57 | 53.40 | 42.65 | 48.80 | 46.06 | 65.35 | 97.05 | 97.45 |
| Stats | 85.89 | 86.46 | 72.77 | 60.18 | 66.03 | 50.56 | 90.65 | 94.46 |
| Stats+NCM | 86.34 | 87.67 | 73.10 | 58.98 | 86.32 | 67.12 | 98.45 | 99.15 |
| Explicit | 85.84 | 97.79 | 85.42 | 73.43 | 78.67 | 71.38 | 99.67 | 99.71 |
| **N2NSC** | **86.18** | **98.18** | **86.25** | **73.45** | **89.41** | **71.53** | **99.71** | **99.78** |
| SimOnly+NCM | 84.69 | 89.48 | 73.26 | 60.09 | 87.35 | 66.72 | 98.61 | 99.25 |
| DissimOnly+NCM | 80.19 | 97.22 | 83.58 | 72.23 | 87.41 | 69.97 | 99.74 | 99.84 |
| Random+NCM | 84.99 | 93.29 | 81.33 | 77.57 | 88.61 | 67.30 | 99.00 | 99.52 |
| Shuffled+NCM | 86.28 | 87.51 | 70.42 | 58.09 | 86.87 | 64.15 | 98.48 | 99.28 |

Table 10: Full ablation results (AUPRC). Same variant naming as Table 11.

| Variant | Citeseer | Pubmed | History | Photo | Computers | Children | ogbn-Arxiv | CitationV8 |
|---|---|---|---|---|---|---|---|---|
| Text | 56.41 | 65.25 | 52.08 | 59.22 | 57.00 | 48.79 | 65.34 | 65.89 |
| Text+NCM | 56.41 | 65.25 | 51.64 | 58.82 | 55.78 | 64.95 | 93.35 | 94.38 |
| Stats | 89.00 | 82.06 | 70.31 | 59.29 | 62.15 | 52.14 | 83.90 | 87.89 |
| Stats+NCM | 89.77 | 84.41 | 71.53 | 59.29 | 82.54 | 65.65 | 95.81 | 97.00 |
| Explicit | 86.54 | 92.83 | 83.71 | 65.49 | 72.25 | 68.10 | 98.22 | 98.41 |
| **N2NSC** | **87.50** | **93.46** | **84.70** | **66.76** | **85.78** | **70.40** | **98.35** | **98.53** |
| SimOnly+NCM | 89.80 | 85.62 | 71.19 | 56.34 | 83.89 | 65.39 | 96.21 | 96.84 |
| DissimOnly+NCM | 84.62 | 93.38 | 80.59 | 66.38 | 83.22 | 68.74 | 98.64 | 98.74 |
| Random+NCM | 88.00 | 88.10 | 79.20 | 72.12 | 86.41 | 64.63 | 96.28 | 97.81 |
| Shuffled+NCM | 89.80 | 83.83 | 69.10 | 57.24 | 82.23 | 63.48 | 95.89 | 97.06 |

Table 11: Full ablation results (Anomaly-F1). Text = text-only input. Stats = graph-structural neighborhood semantics only. Explicit = full explicit path without NCM. N2NSC = full system (explicit + NCM). +NCM indicates Neighborhood Context Modulation is active.