



Tri-Level Anomaly Detection in Text-Attributed Graphs with Graph Foundation Model

^{1,2} A.D. Moiseykin, ORCID: 0009-0009-4546-5681 <a.moiseykin@ispras.ru>
^{1,3} M.D. Drobyshevskiy, ORCID: 0000-0002-1639-9154 <drobyshevsky@ispras.ru>
^{1,3} M.V. Soloviov, ORCID: 0009-0009-2243-273X <m.solovev@phystech.edu>

¹ Ivannikov Institute for System Programming of the RAS,
 25, Alexander Solzhenitsyn Str., Moscow, 109004, Russia.

² Lomonosov Moscow State University,
 GSP-1, Leninskie Gory, Moscow, 119991, Russia.

³ Moscow Institute of Physics and Technology,
 9, Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.

Abstract. Text-attributed graphs (TAGs) have become increasingly important in modern applications such as citation networks, knowledge graphs, and social platforms. As their scale and complexity continue to grow, anomaly detection in TAGs has attracted substantial attention. Despite recent progress, effectively exploiting the heterogeneous information available in these graphs remains a challenging task. At the same time, graph foundation models have recently emerged as a promising direction in graph learning, yet their potential for anomaly detection is still underexplored. In this paper, we propose the Tri-Level Anomaly Detection model (TriAD), a novel model for text-attributed graph anomaly detection that jointly leverages three levels of information: graph structure, node text, and representations obtained from a pretrained graph foundation model. TriAD aligns these levels in a shared representation space via cross-level contrastive learning, allowing it to capture inconsistencies between local graph context, semantic content, and higher-order structural patterns. By incorporating foundation-model-based graph representations, TriAD provides an additional anomaly signal that complements text and local graph structure. Across four benchmarks, TriAD obtains the best ROC-AUC on all four and the best Average Precision on three. In addition, we introduce an anomaly generation procedure to construct a new WordNet-domain dataset with a broader variety of synthetic anomalies. This dataset may be useful for future studies on anomaly detection in text-attributed graphs.

Keywords: graph anomaly detection; graph foundation models; text-attributed graphs; contrastive learning.

For citation: Moiseykin A.D., Drobyshevskiy M.D., Soloviov M.V. Tri-Level Anomaly Detection in Text-Attributed Graphs with Graph Foundation Model. Trudy ISP RAN/Proc. ISP RAS, 2026, vol. 38, issue 6, part 1, pp. 73-86. DOI: 10.15514/ISPRAS-2026-38(6)-5.

Acknowledgements. The results were obtained using the services of the Ivannikov Institute for System Programming (ISP RAS) Data Center.

Трёхуровневое обнаружение аномалий в графах с текстовыми атрибутами с использованием графовой фундаментальной модели

^{1,2} А.Д. Моисейкин, ORCID: 0009-0009-4546-5681 <a.moiseykin@ispras.ru>
^{1,3} М.Д. Дробышевский, ORCID: 0000-0002-1639-9154 <drobyshevsky@ispras.ru>
^{1,3} М.В. Соловьев, ORCID: 0009-0009-2243-273X <m.solovev@phystech.edu>

¹ Институт системного программирования им. В.П. Иванникова РАН,
 Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

² Московский государственный университет имени М.В. Ломоносова,
 Россия, 119991, Москва, Ленинские горы, д. 1.

³ НИУ Московский Физико-Технический институт,
 Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., 9.

Аннотация. Графы с текстовыми атрибутами (Text-Attributed Graphs, TAGs) приобретают всё большее значение в современных приложениях, таких как сети цитирования, графы знаний и социальные платформы. По мере увеличения их масштаба и сложности задача обнаружения аномалий в TAG-графах привлекает всё больше внимания. Несмотря на недавний прогресс, эффективное использование разнородной информации, содержащейся в таких графах, по-прежнему остаётся сложной задачей. В то же время фундаментальные модели для графов (graph foundation models) недавно стали перспективным направлением в области машинного обучения на графах, однако их потенциал для обнаружения аномалий пока исследован недостаточно. В данной работе мы предлагаем Tri-Level Anomaly Detection model (TriAD) — новую модель для обнаружения аномалий в графах с текстовыми атрибутами, которая совместно использует информацию на трёх уровнях: структуру графа, текст узлов и представления, полученные с помощью предварительно обученной фундаментальной модели для графов. TriAD согласовывает эти три уровня в общем пространстве представлений с помощью межуровневого контрастного обучения (cross-level contrastive learning), что позволяет модели выявлять несоответствия между локальным графовым контекстом, семантическим содержанием и структурными закономерностями более высокого порядка. Использование графовых представлений, полученных с помощью фундаментальной модели, предоставляет дополнительный сигнал об аномальности, дополняющий текстовую информацию и локальную структуру графа. На четырёх бенчмарках TriAD показывает наилучший результат по метрике ROC-AUC на всех четырёх наборах данных и наилучший результат по Average Precision на трёх из них. Кроме того, мы предлагаем процедуру генерации аномалий для построения нового набора данных на основе домена WordNet, содержащего более широкий спектр синтетических аномалий. Этот набор данных может быть полезен для будущих исследований в области обнаружения аномалий в графах с текстовыми атрибутами.

Ключевые слова: обнаружение аномалий в графах; фундаментальные модели для графов; графы с текстовыми атрибутами; контрастное обучение.

Для цитирования: Моисейкин А.Д., Дробышевский М.Д., Соловьев М.В. Трёхуровневое обнаружение аномалий в графах с текстовыми атрибутами с использованием графовой фундаментальной модели. Труды ИСП РАН, 2026, том 38, вып. 6, часть 4, стр. 73–86 (на английском языке). DOI: 10.15514/ISPRAS-2026-38(6)-5.

Благодарности. Результаты получены с использованием услуг Центра коллективного пользования Института системного программирования им. В.П. Иванникова РАН — ЦКП ИСП РАН.

1. Introduction

Text-attributed graphs (TAGs) are widely used in applications such as citation networks, recommendation systems, social platforms, and knowledge-centric domains [1]. In TAGs, nodes are connected by edges encoding interactions or semantic relations, while nodes contain textual attributes.

The growing use of TAGs has increased the importance of node anomaly detection [2-3]. Anomalous nodes may deviate from normal patterns in their structural context, textual content, or relational behavior.

Anomaly detection in TAGs is challenging because anomalies may not be visible from structure or text alone. Instead, they often appear as inconsistencies between the two modalities [4]. For example, in a citation graph, a paper may be linked to machine learning papers while its title and abstract discuss medieval history. Therefore, effective TAG anomaly detection requires modeling both local node-level signals and higher-order structural patterns [3-4].

In this work, we consider a label-aware setting in which labels are available for the predefined training split, while anomalous nodes form only a small minority of the labeled examples. Test labels are never used during model development, while validation labels are used only for checkpoint selection and are not used for gradient-based training. Since existing methods rely on supervision to different extents, we report unsupervised and label-aware methods separately and ensure that all label-aware methods use the same labeled training nodes and data splits [4-5].

Hypothesis. We hypothesize that aligning three distinct levels of node representation—textual semantics, local graph-aware structure, and pretrained foundation-level structural priors—provides a stronger anomaly signal than bi-level text–graph or single-view graph representations. We evaluate this hypothesis through comparisons with bi-level and single-view baselines, as well as ablations that remove the foundation branch or its cross-level alignment.

To test this hypothesis, we introduce Tri-Level Anomaly Detection (TriAD), a model for anomaly detection in text-attributed graphs. TriAD combines graph structure, textual node attributes, and representations from a pretrained graph foundation model. These three levels are projected into a shared space and aligned to capture cross-level consistency while highlighting anomalous deviations. By modeling irregularities within and across levels, TriAD provides a richer anomaly signal than bi-level text–graph or structure-only approaches.

We also address the limited availability of anomaly-oriented TAG benchmarks in semantically rich domains. Although recent work has expanded TAG benchmarks [4], existing resources remain limited in scale and diversity. To reduce this gap, we introduce a structurally and textually constrained anomaly generation algorithm and use it to construct a new WordNet-based dataset [6]. We expect this dataset to support both our evaluation and future research on anomaly detection in text-attributed graphs.

The main contributions of this work are as follows:

- We propose a new method for node anomaly detection that combines pretrained graph foundation representations with structural and textual information, enabling the model to capture anomalies arising from text-level, graph-level, and cross-level irregularities.
- We introduce an anomaly generation algorithm and a WordNet-based synthetic anomaly benchmark for text-attributed graph anomaly detection, providing a diverse testbed for evaluating four contextual and four structural anomaly patterns.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed method. Section 4 describes the anomaly generation algorithm and the construction of the WordNet dataset. Section 5 reports the experimental results. Section 6 concludes the paper.

2. Related work

2.1 Graph Anomaly Detection

Early graph anomaly detection approaches relied mainly on classical methods. LOF [7] identifies nodes with abnormal local density, SCAN [8] detects structurally inconsistent vertices, and Radar [9] models anomalies through residual analysis.

Deep learning methods later became dominant. DOMINANT [10] detects anomalies from reconstruction errors of node attributes and graph structure, while GAD-NR [11] reconstructs node

neighborhoods. Self-supervised methods further improved this direction: CoLA [12] formulates anomaly detection as a contrastive task, ANEMONE [13] introduces multi-scale contrastive learning, SL-GAD [14] combines generative and contrastive objectives, and GRADATE [15] uses multi-scale anomaly scoring. AEGIS [16] studies adversarial inductive anomaly detection. However, most of these methods are designed for attributed graphs and mainly rely on structure or generic node features, while textual semantics are not the main focus.

2.2 Anomaly Detection in Text-Attributed Graphs

In text-attributed graphs, anomalies may appear in graph structure, textual content, or their cross-modal relation. CMUCL [4] uses cross-modal and uni-modal contrastive learning, while GUARD [5] employs a graph-informed language-model framework.

These methods can be evaluated under different supervision regimes. CMUCL is primarily a contrastive TAG anomaly detection framework, whereas GUARD can incorporate supervision during downstream adaptation. Since these settings are not identical, we distinguish unsupervised and label-aware protocols in our experiments and use the same labeled training nodes for all methods that exploit anomaly labels.

Nevertheless, existing TAG anomaly detection methods are still mostly evaluated on relatively narrow anomaly settings, such as graph–text inconsistency or limited structural and semantic perturbations. They have not been extensively studied across a broader range of anomaly types in text-attributed graphs.

2.3 Graph Foundation Models for Graph Representation Learning

Graph foundation models and graph pretraining methods aim to learn transferable graph representations rather than optimize only for a single downstream task. They capture general structural regularities that can transfer across graph learning problems, and recent surveys summarize their concepts, opportunities, and challenges [17].

AnomalyGFM [18] extends this idea to graph anomaly detection through zero-shot and few-shot adaptation across unseen graphs. It learns graph-agnostic normal and abnormal prototypes and aligns them with node residual representations. This shows that anomaly detection can benefit from foundation-level structural priors. However, AnomalyGFM mainly targets structural anomaly generalization, while our setting involves text-attributed graphs, where anomalies may also arise from structure–text inconsistency.

We use GFT [19], which learns transferable graph- and node-level representations through a pretrained tree-based vocabulary. It provides higher-order structural information that can complement graph–text anomaly detection models.

3. Model Architecture

In this section, we introduce our TriAD model. We first describe the model components used in the framework, and then explain how these components interact and how the overall model is trained.

3.1 Feature Extraction Modules

Our approach combines three components to model different aspects of node behavior.

Node embedding: To extract semantic features from node text, we use a Transformer encoder. Given the textual attribute of a node, the encoder produces a text embedding vector $\mathbf{h}_v^t = \text{Transformer}(x_v^t)$, where x_v^t denotes the textual attribute of node v , and $\mathbf{h}_v^t$ is its semantic representation.

Graph embedding: To capture graph topology, a GCN propagates the text-derived node feature matrix X through adjacency structure A , producing the graph-aware embedding $\mathbf{h}_v^g = \text{GCN}(X, A)_v$.

Thus, the graph branch refines textual node features using local topology rather than operating on purely structural inputs.

GFT-based node representations.

To obtain pretrained structural representations of nodes, we use GFT, which is pretrained on multiple graph datasets. Given the node features X and the graph adjacency structure A , GFT produces a foundation-level embedding for each node: $\mathbf{h}_v^f = \text{GFT}(X, A)_v$, where $\mathbf{h}_v^f \in \mathbb{R}^d$ denotes the pretrained node representation of node v .

The text, graph, and foundation embeddings capture semantic information, local topological context, and pretrained structural information, respectively, and are aligned in a shared representation space.

3.2 Tri-level interaction

After extracting the text, graph, and foundation embeddings for each node v , we additionally construct neighborhood-level representations for all three levels. Let $\mathcal{N}_K(v)$ denote the set of K sampled neighbors of node v . Then, for each level $m \in \{t, g, f\}$, we define the neighborhood representation as $\bar{\mathbf{h}}_v^m = \frac{1}{K} \sum_{u \in \mathcal{N}_K(v)} \mathbf{h}_u^m$. Thus, for each node, the model operates on six representations: $\mathbf{h}_v^t, \bar{\mathbf{h}}_v^t, \mathbf{h}_v^g, \bar{\mathbf{h}}_v^g, \mathbf{h}_v^f, \bar{\mathbf{h}}_v^f$.

Before computing the loss, we normalize all representations

$$\mathbf{z}_v^m = \frac{\mathbf{h}_v^m}{\|\mathbf{h}_v^m\|_2}, \bar{\mathbf{z}}_v^m = \frac{\bar{\mathbf{h}}_v^m}{\|\bar{\mathbf{h}}_v^m\|_2}, m \in \{t, g, f\}.$$

Given two sets of embeddings, $\{\mathbf{a}_i\}_{i=1}^B$ and $\{\mathbf{b}_i\}_{i=1}^B$, where B is the batch size and $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^d$ are d -dimensional embedding vectors, we define their similarity as $S_{ij}^{a,b} = \tau \mathbf{a}_i^\top \mathbf{b}_j$, where $\tau = 1/T$ is a learnable inverse-temperature scale.

Based on this similarity, we define the symmetric contrastive loss between two embedding sets as

$$\mathcal{L}(a, b) = -\frac{1}{2B} \sum_{i=1}^B \left[\log \frac{\exp(S_{ii}^{a,b})}{\sum_{j=1}^B \exp(S_{ij}^{a,b})} + \log \frac{\exp(S_{ii}^{a,b})}{\sum_{j=1}^B \exp(S_{ji}^{a,b})} \right].$$

This loss follows the general InfoNCE-style contrastive learning paradigm also used in CMUCL, but we adopt a symmetric formulation that averages the row-wise and column-wise matching directions.

Table 1. Six pairwise relations used in the text–graph interaction module.

#	Compared representations	Loss term
1	node graph vs. node text	$\mathcal{L}(\mathbf{z}^g, \mathbf{z}^t)$
2	node graph vs. neighborhood text	$\mathcal{L}(\mathbf{z}^g, \bar{\mathbf{z}}^t)$
3	neighborhood graph vs. node text	$\mathcal{L}(\bar{\mathbf{z}}^g, \mathbf{z}^t)$
4	neighborhood graph vs. neighborhood text	$\mathcal{L}(\bar{\mathbf{z}}^g, \bar{\mathbf{z}}^t)$
5	node graph vs. neighborhood graph	$\mathcal{L}(\mathbf{z}^g, \bar{\mathbf{z}}^g)$
6	node text vs. neighborhood text	$\mathcal{L}(\mathbf{z}^t, \bar{\mathbf{z}}^t)$

The six relations in Table 1 define the text–graph interaction module inherited from CMUCL. The first four terms align graph and text representations across node-level and neighborhood-level views, comparing each node with both its own representation and the representation of its local neighborhood. This allows the model to capture not only direct graph–text consistency, but also whether the node remains compatible with its surrounding semantic and structural context. The last two terms preserve intra-modal consistency between node-level and neighborhood-level representations within the graph and text branches.

By combining all six relations, we obtain the bi-level text–graph objective

$$\mathcal{L}_{\text{TG}} = \mathcal{L}(\mathbf{z}^g, \mathbf{z}^t) + \mathcal{L}(\mathbf{z}^g, \bar{\mathbf{z}}^t) + \mathcal{L}(\bar{\mathbf{z}}^g, \mathbf{z}^t) + \mathcal{L}(\bar{\mathbf{z}}^g, \bar{\mathbf{z}}^t) + \gamma (\mathcal{L}(\mathbf{z}^g, \bar{\mathbf{z}}^g) + \mathcal{L}(\mathbf{z}^t, \bar{\mathbf{z}}^t)),$$

where γ is a weighting coefficient that downweights the intra-modal consistency terms relative to the main cross-modal alignment terms.

The objective $\mathcal{L}_{\text{TG}}$ models text–graph interaction. TriAD extends it with the GFT representation as a third level, aligning task-specific representations with a higher-level structural prior.

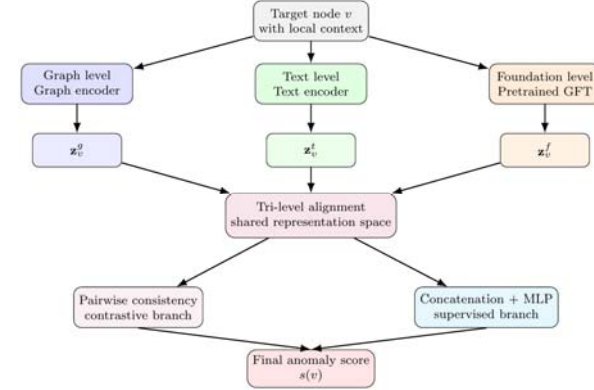


Fig. 1. Overview of the proposed TriAD model.

3.3 Foundation branch

The third level is obtained from GFT, a graph foundation model pretrained on multiple graph datasets. In our framework, GFT provides a foundation embedding $\mathbf{h}_v^f$ for each node, which serves as an additional structural view to the task-specific graph and text representations.

Similarly, to the text–graph interaction, the foundation branch is aligned with both the text and graph modalities through node-level and neighborhood-level representations. In particular, we define the text–foundation objective as $\mathcal{L}_{\text{TF}} = \mathcal{L}(\mathbf{z}^t, \mathbf{z}^f) + \mathcal{L}(\mathbf{z}^t, \bar{\mathbf{z}}^f) + \mathcal{L}(\bar{\mathbf{z}}^t, \mathbf{z}^f) + \mathcal{L}(\bar{\mathbf{z}}^t, \bar{\mathbf{z}}^f)$, and the graph–foundation objective as $\mathcal{L}_{\text{GF}} = \mathcal{L}(\mathbf{z}^g, \mathbf{z}^f) + \mathcal{L}(\mathbf{z}^g, \bar{\mathbf{z}}^f) + \mathcal{L}(\bar{\mathbf{z}}^g, \mathbf{z}^f) + \mathcal{L}(\bar{\mathbf{z}}^g, \bar{\mathbf{z}}^f)$. In addition, we enforce consistency between the node-level and neighborhood-level foundation representations: $\mathcal{L}_{\text{Fself}} = \mathcal{L}(\mathbf{z}^f, \bar{\mathbf{z}}^f)$.

During fine-tuning, we apply two regularizers to the foundation branch. The commitment regularizer, $\mathcal{L}_{\text{com}}$, encourages GFT encoder outputs to remain close to their codebook vectors, preserving consistency between the learned foundation representations and their quantized counterparts. The embedding-uniformity regularizer, $\mathcal{L}_{\text{uni}}$, promotes a balanced use of the foundation embedding space and helps prevent embedding collapse.

Following the contrastive formulation of CMUCL, we learn consistency between node-level and neighborhood-level text and graph representations. We extend this formulation with the GFT branch and, in the label-aware setting, compute the resulting tri-level contrastive objective only over training nodes labeled as normal. Let $\mathcal{I}_{\text{norm}}$ denote the set of normal-node indices in the current mini-batch: $\mathcal{I}_{\text{norm}} = \{i | y_i = 0\}$. Unlike the original unsupervised CMUCL objective, our label-aware objective excludes labeled anomalies from contrastive alignment, because aligning them would encourage irregular examples to match the normal consistency pattern. Restricting alignment to normal nodes therefore yields a cleaner tri-level representation of normality.

The resulting tri-level loss is written as

$$\mathcal{L}_{\text{tri}} = \mathcal{L}_{\text{TG}}^{\text{norm}} + \lambda_{\text{TF}} \mathcal{L}_{\text{TF}}^{\text{norm}} + \lambda_{\text{GF}} \mathcal{L}_{\text{GF}}^{\text{norm}} + \lambda_{\text{GF}} \gamma \mathcal{L}_{\text{Fself}}^{\text{norm}} + \lambda_{\text{uni}} \mathcal{L}_{\text{uni}} + \lambda_{\text{com}} \mathcal{L}_{\text{com}}.$$

In addition to the CMUCL-derived contrastive branch, we introduce a supervised anomaly classification head for the label-aware setting. For each source node v , we concatenate its normalized graph, text, and foundation representations: $\mathbf{r}_v = [\mathbf{z}_v^g \parallel \mathbf{z}_v^t \parallel \mathbf{z}_v^f]$. The classification head maps this representation to an anomaly logit: $o_v = \text{MLP}(\mathbf{r}_v)$. The supervised anomaly loss, computed over the labeled nodes in the training split, is $\mathcal{L}_{\text{sup}} = \text{BCEWithLogits}(o_v, y_v)$.

Figure 1 illustrates the architecture. The contrastive and supervised branches are jointly optimized using $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{tri}} + \lambda_{\text{sup}} \mathcal{L}_{\text{sup}}$.

For inference, let $c_r(v)$ denote the node-level contrastive inconsistency under the r -th independently sampled neighborhood, using the same relations as in training. We summarize R rounds by their mean and spread:

$$\bar{c}(v) = \frac{1}{R} \sum_{r=1}^R c_r(v), \text{sd}(v) = \sqrt{\frac{1}{R} \sum_{r=1}^R (c_r(v) - \bar{c}(v))^2},$$

$c(v) = \bar{c}(v) + \text{sd}(v)$, where the standard deviation uses the population $(1/R)$ normalization, so that nodes whose inconsistency is both high on average and unstable across sampled neighborhoods receive a larger score.

Let mm_{tr} denote min-max calibration whose minimum and maximum are estimated only from the training split. Let o_v be the logit of the supervised head and σ the logistic function. The final anomaly score is $s(v) = \text{mm}_{\text{tr}}(c(v)) + \beta \text{mm}_{\text{tr}}(\sigma(o_v))$, with β the inference-time weight of the supervised branch and larger values indicating higher anomaly likelihood. In the variant without the BCE head we set $\lambda_{\text{sup}} = \beta = 0$, which removes the second term and yields $s(v) = \text{mm}_{\text{tr}}(c(v))$.

4. Dataset Generation

For synthetic dataset generation on text-attributed graphs, we build on the anomaly injection scheme introduced in CMUCL and extend it with contextual and structural perturbations. Our dataset is constructed on the WordNet-based WN18RR graph, which contains $N = 40,943$ nodes and $|E| = 93,003$ directed relation edges. The original clean graph and node texts are used for GFT pretraining; no anomaly labels, contextual text perturbations, or injected structural edges are present at that stage. After anomaly injection, the benchmark contains 93,291 directed edges. Node texts are WordNet glosses, with mean and median lengths of 90 and 80 characters. The mean degree increases from 2.27 to 2.28, while the maximum degree is 494.

Let the source graph be $G = (V, E, T)$, where $T = \{t_v | v \in V\}$ denotes node texts, and let $\tilde{G} = (V, \tilde{E}, \tilde{T})$ be the perturbed graph. Sentence-transformer embeddings guide donor-node selection for contextual anomalies.

Let $N = |V|$. The number of anomalous nodes is $M = \lfloor \alpha N \rfloor$, where $\alpha \in (0, 1)$ controls the anomaly ratio. For the WN18RR benchmark, $\alpha = 0.01$, yielding 409 anomalous nodes. Anomalies span eight contextual and structural strategies: insertion, replacement, fusion, truncation, dense clique, cross-community edges, isolation, and hub amplification.

Following CMUCL, we retain four anomaly types: *contextual insertion*, *contextual replacement*, *dense clique injection*, and *cross-community random edge injection*. We add four new types: *contextual fusion*, *contextual truncation*, *structural isolation*, and *structural hub injection*. Thus, the pipeline includes four contextual and four structural perturbation strategies.

4.1 Contextual anomalies

Contextual anomalies modify only node text while preserving graph connectivity. The first two strategies, *contextual insertion* and *contextual replacement*, are inherited from CMUCL. For a target

node v , we sample a candidate donor set $\mathcal{C}(v)$ and select the node u with the most dissimilar text embedding: $u^* = \arg \min_{u \in \mathcal{C}(v)} \cos(\mathbf{z}_v, \mathbf{z}_u)$, where $\mathbf{z}_v$ and $\mathbf{z}_u$ are normalized text embeddings. In *contextual insertion*, a random contiguous donor fragment is inserted into the text of v . In *contextual replacement*, a randomly selected target fragment is replaced with donor content.

We additionally introduce two contextual perturbations. In *contextual fusion*, fragments from two semantically distant donors are inserted into the target text, producing a mixed description with incompatible semantic contexts. This models metadata corruption in which content from different entities is accidentally merged. In *contextual truncation*, a contiguous target fragment is removed, yielding an incomplete or abnormally shortened description and modeling partially scraped or damaged metadata. These perturbations increase dataset diversity by covering semantic contamination, fragmentation, and information loss.

4.2 Structural anomalies

Structural anomalies modify only graph connectivity while leaving textual attributes unchanged. The first two strategies, *dense clique injection* and *cross-community random edge injection*, are inherited from CMUCL. In *dense clique injection*, anomalous nodes are grouped into dense subgraphs of size $q = 5$, with edges added between all node pairs to create locally over-dense regions. In *cross-community random edge injection*, each target node is connected to several previously unconnected nodes, producing unusual long-range links.

We add two further structural anomaly types. In *structural isolation*, a fraction sampled from $[0.5, 1.0]$ of the edges incident to a target node is removed, modeling missing or broken relations that leave an otherwise valid node abnormally weakly connected. In *structural hub injection*, a target node is connected to many new nodes using a degree amplification factor of 1.5, modeling graph construction errors that create abnormally high connectivity despite ordinary textual content.

5. Experimental results

To evaluate the proposed framework, we use pretrained GFT representations as a structural prior aligned with the text and local graph branches, rather than as a standalone detector.

5.1 Implementation Details

We use the same architecture, optimization settings, and evaluation protocol for all TriAD variants. The graph, text, and GFT branches are projected into a shared 128-dimensional space. The graph branch is a two-layer GCN with hidden and output dimensions of 128 and dropout 0.2. The raw-text branch uses frozen `intfloat/e5-large-v2` embeddings (maximum sequence length 128) followed by a trainable projection. The GFT branch takes 1024-dimensional text features and uses a two-layer GraphSAGE-style encoder [20] with hidden and code dimensions of 768, batch normalization, and dropout 0.15. Its pretrained encoder checkpoint is taken from pretraining epoch 50. During downstream training the complete GFT encoder and its projection are fine-tuned, whereas the vector quantizer is frozen. All variants are trained for 20 epochs with Adam and batch size 64. The learning rate is 10^{-4} for the graph branch, supervised head, and projections, and 10^{-5} for GFT. We use $K = 3$, $R = 32$, $T_0 = 0.07$, $\gamma = 0.5$, $\lambda_{\text{TF}} = \lambda_{\text{GF}} = 0.3$, $\lambda_{\text{uni}} = \lambda_{\text{com}} = 0.01$, $\lambda_{\text{sup}} = 0.5$, and $\beta = 0.5$. The four-head codebook has 128 entries per head, code dimension 768, decay 0.8, and commitment weight 0.25. BCE uses $w_+ = N_{\text{train},0} / \max(N_{\text{train},1}, 1)$, computed on the training split only. We report the mean and sample standard deviation over five runs with seeds 1–5. Validation is performed after each epoch using one neighborhood-sampling round. We select the checkpoint maximizing validation AP + ROC-AUC, ignore improvements below 10^{-4} , use the first five epochs as warm-up, and restore the best weights before testing.

5.2 Datasets

The datasets used during GFT pretraining are listed in Table 2¹. GFT is pretrained in a self-supervised manner and does not use anomaly labels, downstream train/validation/test masks, or labels introduced by our anomaly-generation procedure. It learns shared encoder parameters and a vector-quantized vocabulary of computation-tree patterns rather than node-specific anomaly predictions.

Table 2. All datasets used in GFT pretraining.

Dataset	Nodes	Edges
cora [19]	2 708	10 556
pubmed [19]	19 717	88 648
arxiv [19]	169 343	2 315 598
wikics [19]	11 701	431 726
whoiswho [21]	147 895	34 448 378
history [4]	41 551	369 252
WN18RR [19]	40 943	93 291
FB15K237 [19]	14 541	310 116
chemhiv [19]	1 049 163	2 259 376
chemblpre [19]	8 845 648	19 123 034
chempcba [19]	11 349 235	24 566 048
oewn[22]	120 630	244 036

We evaluate on Pubmed, History, WN18RR, and WhoIsWho. Pubmed and History are used directly and in full from the official CMUCL release [4], including graph structures, texts, labels, anomaly-generation protocol, and predefined splits. Pubmed is a citation graph and History an e-commerce co-purchasing graph. WN18RR is constructed in this work, while WhoIsWho is a real-world academic name-disambiguation benchmark [21].

In WhoIsWho, each node is a paper–author assignment and edges connect assignments sharing an author or paper. A label of 1 denotes assignment to the wrong author profile; no synthetic anomalies or edges are added. The full graph has 81,976 nodes and 8,198 anomalies (10.00%). Due to computational limits, we use one fixed 5,000-node induced subgraph obtained by undirected Forest Fire sampling [23] with $p = 0.35$ and seed 42. It contains 585 anomalies, giving a ratio of 0.117. This ratio results from topology-driven sampling rather than a manually imposed target. Original split membership is preserved, yielding 3,460/526/1,014 disjoint train/validation/test nodes, and the splits are not regenerated from test labels. We exclude GUARD datasets because their exact training versions are unavailable and open reconstructions differ from the reported statistics.

5.3 Baseline methods

We test a diverse set of baselines. The unsupervised baselines include LOF, SCAN, Radar, AEGIS, MLPAE, DOMINANT, GAD-NR, CoLA, ANEMONE, SL-GAD, GRADATE and CMUCL. We also include label-aware neural baselines, including GraphSAGE [20] and supervised variants of representation-based methods, where a lightweight MLP classifier is trained on their learned representations. GUARD is treated as a label-aware text-rich graph baseline because it can use anomaly labels during downstream adaptation. AnomalyGFM is also evaluated in its label-aware

¹ GFT pretraining uses all datasets in Table 2, including the graphs later used for downstream evaluation. Our additional control experiments indicate that this overlap does not materially affect downstream performance; we therefore omit these auxiliary experiments for brevity.

fine-tuning setting. Finally, we report fixed-feature controls based on frozen text-only, graph-only, GFT-only, and concatenated text–graph–GFT representations.

5.4 Evaluation Protocol

We follow train/validation/test splits distributed by CMUCL for Pubmed and History. For WN18RR, we use the predefined 60/10/30 split generated with seed 42. For WhoIsWho, we use the intersection of benchmark’s original split membership with the fixed Forest Fire sample described above. All label-aware methods use the same labeled splits, while unsupervised methods serve as reference baselines. Model selection uses validation data only, and ROC-AUC and AP are reported on the held-out test split.

5.5 Results

Table 3. Performance comparison on Pubmed, History, WordNet, and WhoIsWho under different supervision regimes. The best result is shown in bold and the second-best result is underlined.

Method	Pubmed		History		WordNet		WhoIsWho ↑	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP
<i>Unsupervised and self-supervised methods</i>								
LOF	64.10±0.00	8.39±0.00	55.71±0.00	5.22±0.00	63.77±0.00	2.44±0.00	52.75±0.00	12.05±0.00
SCAN	60.16±0.52	10.41±0.64	57.27±0.89	5.00±0.70	49.86±0.91	1.00±0.67	49.26±0.83	11.39±0.46
Radar	51.88±0.35	3.92±0.24	48.48±1.05	3.60±0.76	45.62±0.23	0.87±0.42	48.56±0.39	11.25±0.46
AEGIS	49.08±1.10	3.66±0.50	52.28±1.97	4.23±0.75	50.18±1.34	1.10±0.23	54.52±1.38	12.26±0.41
MLPAE	50.56±0.26	3.93±0.39	47.54±0.16	3.67±0.53	72.76±0.46	5.29±0.21	46.52±0.36	10.52±0.48
DOMINANT	67.32±0.61	7.54±0.81	63.46±0.23	6.17±0.15	57.51±0.05	1.25±0.01	48.70±0.34	10.42±0.53
GAD-NR	66.25±0.44	6.34±0.04	65.86±0.13	5.89±0.22	64.56±2.84	4.14±0.10	45.11±0.08	15.10±0.23
CoLA	78.83±0.55	18.93±0.62	78.72±0.09	26.48±0.55	65.96±1.31	11.02±1.56	59.79±1.63	<u>21.35±1.58</u>
ANEMONE	78.81±0.02	19.43±0.54	79.14±0.02	27.20±0.66	67.98±0.72	21.29±0.03	48.16±0.13	11.48±0.03
SL-GAD	75.19±0.35	16.71±0.05	74.08±0.28	20.22±0.41	64.78±0.23	11.19±0.09	47.09±0.28	10.74±0.04
GRADATE	67.49±0.09	12.13±0.18	73.25±0.10	14.50±0.55	75.34±0.20	13.12±0.43	49.97±0.78	12.37±0.19
CMUCL	<u>81.78±0.27</u>	<u>30.96±0.15</u>	<u>79.69±0.31</u>	<u>36.80±0.66</u>	<u>84.89±0.21</u>	48.91±0.28	<u>69.31±0.19</u>	19.84±0.37
<i>Supervised baselines</i>								
SAGEConv	63.89±0.83	7.91±0.61	67.54±1.63	16.15±2.15	83.70±0.64	9.18±0.88	51.46±0.14	11.70±0.22
GUARD	65.32±0.53	28.19±0.83	68.15±0.27	5.27±0.16	62.88±0.93	17.87±0.67	64.04±0.43	21.04±0.15
AnomalyGFM	61.37±0.92	7.64±0.19	68.04±0.35	7.44±0.12	65.57±0.61	11.51±0.66	62.48±0.74	21.28±1.02
<i>Proposed supervised method</i>								
TriAD	90.63±0.14	43.20±0.35	84.42±0.12	37.92±0.65	94.42±0.43	<u>47.92±0.29</u>	85.18±0.10	60.53±0.48

† Modified version of the original WhoIsWho graph constructed in this work.

We conduct experiments on four datasets, with full results reported in Table 3. We use ROC-AUC and AP, where ROC-AUC measures ranking quality across thresholds and AP is especially important for imbalanced anomaly detection. Classical unsupervised baselines such as LOF, SCAN, Radar, and AEGIS perform weakly, while neural and contrastive methods are generally stronger. CMUCL is particularly competitive on Pubmed, History, and WordNet, whose synthetic anomaly-generation procedures are identical or closely related to the perturbation setting for which CMUCL was originally designed. However, its performance decreases on WhoIsWho, where the anomaly distribution and graph construction differ from its original synthetic setting. This suggests that CMUCL is effective for the anomaly patterns captured by its contrastive inconsistency objective, but may generalize less reliably to substantially different graph domains and anomaly types.

TriAD achieves the best ROC-AUC on all four datasets and the best AP on three of them. On WordNet AP, CMUCL is slightly higher, but TriAD still obtains the highest ROC-AUC. TriAD also maintains a substantially larger advantage on WhoIsWho, suggesting better robustness when the evaluation setting differs from the synthetic anomaly patterns used by CMUCL. Overall, the results show that combining textual, structural, and foundation-level information provides useful anomaly

signals. The ablation study below further tests whether these gains arise from tri-level interaction rather than only from supervision or pretrained GFT features.

5.6 Ablation Study

We conduct component-wise ablations by removing the GFT branch, tri-level alignment, neighborhood-level views, normal-only contrastive training, and the supervised BCE head. Normal-only contrastive training computes the contrastive loss only over normal, non-anomalous nodes. We also include fixed-feature controls with frozen text, graph, GFT, and concatenated text–graph–GFT representations to test whether the gains come from joint alignment rather than simply from adding more feature channels.

The “w/o GFT branch” variant is the directly matched bi-level text–graph control: it retains the same labeled training nodes, text and graph encoders, neighborhood sampling, normal-only contrastive objective, supervised BCE head, and optimization protocol, while removing only the GFT branch and its alignment terms. In the “w/o tri-level alignment” variant, we retain all three encoders and the supervised head but remove the text–foundation and graph–foundation contrastive alignment terms. The “w/o BCE head” variant remains label-aware because normal labels are still used to construct J_{norm} .

Table 4 shows that the GFT branch, tri-level alignment, and neighborhood views are useful, although their effects are not uniform across datasets and metrics. Removing GFT lowers performance on Pubmed, WordNet, and WhoIsWho, while disabling tri-level alignment causes a consistent drop, showing that feature concatenation alone is insufficient. The largest decline occurs without neighborhood-level views. Normal-only contrastive training improves most dataset–metric pairs, but its removal raises History AP from 37.92 to 40.49. Removing the BCE head further raises History AP to 45.61, while reducing History ROC-AUC and AP elsewhere. Since this trend appears only on History, it may reflect dataset-specific interactions between the supervised and contrastive objectives. Fixed-feature controls remain below TriAD, supporting joint tri-level interaction and identifying loss balancing on History as a limitation.

Table 4. Ablation study and fixed-feature control experiments.

Variant	Pubmed		History		WordNet		WhoIsWho	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP
TriAD	90.63±0.14	43.20±0.35	84.42±0.12	37.92±0.65	94.42±0.43	47.92±0.29	85.18±0.10	60.53±0.48
w/o GFT branch	82.36±2.95	29.34±4.44	82.66±2.11	37.18±0.91	82.10±1.24	31.57±2.47	82.11±1.11	41.43±3.82
w/o tri-level alignment	82.88±1.65	26.31±3.68	80.66±0.48	29.58±2.05	<u>83.61±2.33</u>	22.89±1.77	80.62±2.25	34.92±3.15
w/o neighborhood-level views	62.81±1.93	11.43±2.04	66.74±1.78	10.97±0.89	78.66±2.94	36.37±0.12	53.73±2.58	13.02±1.00
w/o normal-only contrastive training	<u>85.58±1.39</u>	34.45±1.55	<u>84.29±1.05</u>	<u>40.49±0.56</u>	81.46±2.39	31.83±3.03	82.56±1.60	38.49±4.21
w/o BCE head (label-aware)	83.31±0.74	<u>38.77±1.12</u>	79.91±0.27	45.61±0.43	80.26±0.91	<u>40.13±5.96</u>	<u>85.12±1.51</u>	<u>46.40±4.22</u>
GFT-only	79.52±1.72	21.15±2.64	73.94±0.60	18.43±1.70	62.72±1.51	2.28±0.33	51.54±1.53	13.76±0.71
MLP(text)	63.68±0.78	17.28±0.35	63.84±0.88	13.70±1.28	48.56±0.67	1.08±0.08	45.38±2.68	10.61±0.88
MLP(graph)	65.97±0.57	18.93±3.38	66.90±0.98	27.12±0.58	56.73±1.27	6.50±2.48	55.61±0.63	12.98±0.19
MLP(GFT)	57.02±0.94	6.37±0.35	60.24±0.61	6.79±0.23	52.35±0.87	1.19±0.07	55.81±3.23	14.84±1.37
MLP(text+graph+GFT)	70.54±0.94	18.78±0.86	68.96±0.92	14.94±1.60	54.25±1.10	1.19±0.10	58.79±1.44	14.39±1.30

6. Conclusion

We studied anomaly detection in text-attributed graphs from a tri-level perspective. Unlike other approaches that rely only on graph structure or combine graph and text information in a limited manner, TriAD incorporates an additional foundation-level structural view. The experimental results confirm that the proposed approach achieves improvements over competitive baselines, indicating that graph foundation representations provide a useful signal for this task. We also introduced a new WordNet-based dataset that can be used for evaluating anomaly detection methods in text-attributed graph domains. The dataset is available on Hugging Face [24].

References

- [1] Zhang D.C., Yang M., Ying R., Lauw H.W. Text-Attributed Graph Representation Learning: Methods, Applications, and Challenges. In Proceedings of The ACM Web Conference 2024, 2024. DOI: 10.1145/3589335.3641255.
- [2] Akoglu L., Tong H., Koutra D. Graph-based Anomaly Detection and Description: A Survey. Data Mining and Knowledge Discovery, 2015. DOI: 10.1007/s10618-014-0365-y.
- [3] Qiao H., Tong H., An B., King I., Aggarwal C., Pang G. Deep Graph Anomaly Detection: A Survey and New Perspectives. IEEE Transactions on Knowledge and Data Engineering, 2025. DOI: 10.1109/TKDE.2025.3581578.
- [4] Y. Xu, X. Hua, Z. Peng, B. Shi, J. Chen, X. Fu, S. Wang и B. Dong. Text-Attributed Graph Anomaly Detection via Multi-Scale Cross- and Uni-Modal Contrastive Learning. arXiv preprint, 2025. DOI: 10.48550/arXiv.2508.00513.
- [5] Pang Y., Chen B., Zhang F., Rao Y., Kharlamov E., Tang J. GuARD: Effective Anomaly Detection through a Text-Rich and Graph-Informed Language Model. arXiv preprint, 2024. DOI: 10.48550/arXiv.2412.03930.
- [6] Miller G.A. WordNet: A Lexical Database for English. Communications of the ACM, 1995, vol. 38, no. 11, pp. 39–41. DOI: 10.1145/219717.219748.
- [7] Breunig M.M., Kriegel H.-P., Ng R.T., Sander J. LOF: Identifying Density-Based Local Outliers. B Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104. DOI: 10.1145/335191.335388.
- [8] Xu X., Yuruk N., Feng Z., Schweiger T.A.J. SCAN: A Structural Clustering Algorithm for Networks. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 824–833. DOI: 10.1145/1281192.1281280.
- [9] Li J., Dani H., Hu X., Liu H. Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), 2017, pp. 2152–2158. DOI: 10.24963/ijcai.2017/299.
- [10] Ding K., Li J., Bhanushali R., Liu H. Deep Anomaly Detection on Attributed Networks. In Proceedings of the 2019 SIAM International Conference on Data Mining, 2019, pp. 594–602. DOI: 10.1137/1.9781611975673.67.
- [11] Roy A., Shu J., Li J., Yang C., Elshocht O., Smeets J., Li P. GAD-NR: Graph Anomaly Detection via Neighborhood Reconstruction. In Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024, pp. 576–585. DOI: 10.1145/3616855.3635767.
- [12] Liu Y., Li Z., Pan S., Gong C., Zhou C., Karypis G. Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning. IEEE Transactions on Neural Networks and Learning Systems, 2022, vol. 33, no. 6, pp. 2378–2392. DOI: 10.1109/TNNLS.2021.3068344.
- [13] Jin M., Liu Y., Zheng Y., Chi L., Li Y.-F., Pan S. ANEMONE: Graph Anomaly Detection with Multi-Scale Contrastive Learning. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management, 2021. DOI: 10.1145/3459637.3482057.
- [14] Zheng Y., Jin M., Liu Y., Chi L., Phan K.T., Chen Y.-P.P. Generative and Contrastive Self-Supervised Learning for Graph Anomaly Detection. IEEE Transactions on Knowledge and Data Engineering, 2023, vol. 35, no. 12, pp. 12220–12233. DOI: 10.1109/TKDE.2021.3119326.
- [15] Duan J., Wang S., Zhang P., Zhu E., Hu J., Jin H., Liu Y., Dong Z. Graph Anomaly Detection via Multi-Scale Contrastive Learning Networks with Augmented View. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, vol. 37, no. 6, pp. 7459–7467. DOI: 10.1609/aaai.v37i6.25907.

- [16]. Ding K., Li J., Agarwal N., Liu H. Inductive Anomaly Detection on Attributed Networks. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20), 2020, pp. 1288-1294. DOI: 10.24963/ijcai.2020/179.
- [17]. Liu J., Yang C., Lu Z., Chen J., Li Y., Zhang M., Bai T., Fang Y., Sun L., Yu P.S., Shi C. Graph Foundation Models: Concepts, Opportunities and Challenges. arXiv preprint, 2023. DOI: 10.48550/arXiv.2310.11829.
- [18]. Qiao H., Niu C., Chen L., Pang G. AnomalyGFM: Graph Foundation Model for Zero/Few-shot Anomaly Detection. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2025, pp. 2326-2337. DOI: 10.1145/3711896.3736843.
- [19]. Wang Z., Zhang Z., Chawla N.V., Zhang C., Ye Y. GFT: Graph Foundation Model with Transferable Tree Vocabulary. arXiv preprint, 2024. DOI: 10.48550/arXiv.2411.06070.
- [20]. Hamilton W.L., Ying R., Leskovec J. Inductive Representation Learning on Large Graphs. In Advances in Neural Information Processing Systems 30 (NeurIPS 2017), 2017.
- [21]. Chen B., Zhang J., Zhang F., Han T., Cheng Y., Li X., Dong Y., Tang J. Web-Scale Academic Name Disambiguation: The WholsWho Benchmark, Leaderboard, and Toolkit. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 3817-3828. DOI: 10.1145/3580305.3599930.
- [22]. McCrae J.P., Rademaker A., Bond F., Rudnicka E., Fellbaum C. English WordNet 2019. An Open-Source WordNet for English. In Proceedings of the 10th Global Wordnet Conference, 2019, pp. 245-252.
- [23]. Leskovec J., Faloutsos C. Sampling from Large Graphs. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 631-636. DOI: 10.1145/1150402.1150479.
- [24]. Hugging Face. WN18RR_anomaly. Available at: <https://huggingface.co/datasets/username/dataset-name>, accessed: 18.09.2026.

Information about authors

Андрей Денисович МОИСЕЙКИН – разработчик и исследователь в области машинного обучения и искусственного интеллекта. Сфера его научных интересов включает графовые нейронные сети, генеративные модели и крупные языковые модели. Его работа направлена на изучение современных архитектур и методов, предназначенных для анализа сложных данных, извлечения закономерностей и построения интеллектуальных систем нового поколения, а также на разработку новых графовых фундаментальных моделей для решения междоменных задач.

Andrey Denisovich MOISEYKIN – software engineer and researcher in the field of machine learning and artificial intelligence. His research interests include graph neural networks, generative models, and large language models. His work focuses on the study of modern architectures and methods designed for analyzing complex data, discovering patterns, and building next-generation intelligent systems, as well as on the development of new graph foundation models for solving cross-domain tasks.

Михаил Дмитриевич ДРОБЫШЕВСКИЙ – кандидат физико-математических наук, научный сотрудник ИСП РАН. Сфера научных интересов: доверенный ИИ, объяснимый ИИ, графовые нейронные сети.

Mikhail Dmitrievich DROBYSHEVSKIY – Cand. Sci. (Phys.-Math.), Researcher at ISP RAS. Research interests: trustworthy AI, explainable AI, graph neural networks.

Михаил Валентинович СОЛОВЬЕВ – программист-исследователь в области машинного обучения и искусственного интеллекта. Специализируется на разработке и оптимизации графовых нейронных сетей, трансформерных графовых моделей и системах объяснимого искусственного интеллекта (XAI). Его научные интересы охватывают сферу крупных языковых моделей (LLM) и систем Retrieval-Augmented Generation (RAG), таких как LightRAG и GraphRAG. Исследует различные методы защиты и атак на модели машинного

обучения. Активно занимается повышением интерпретируемости и прозрачности моделей в контексте сложных данных.

Mikhail Valentinovich SOLOVIOV – software engineer and researcher in the field of machine learning and artificial intelligence. He specializes in the development and optimization of graph neural networks, transformer-based graph models, and explainable artificial intelligence (XAI) systems. His research interests cover large language models (LLM) and Retrieval-Augmented Generation (RAG) systems, such as LightRAG and GraphRAG. He explores various methods for model protection and attacks on machine learning models. He is actively working on enhancing the interpretability and transparency of models in the context of complex data.