applied sciences

MDPI

Article

Tackling Class Imbalance in Financial Fraud Detection with Relational Graph Transformers and Contrastive Learning

Qianrui Gu [1,*,†], Qingyue Wang [2,†] and Songchao Zhang [3]

[1] Jinan University–University of Birmingham Joint Institute, Jinan University, Guangzhou 511436, China
[2] College of Engineering, University of Michigan, Ann Arbor, MI 48109, USA; qingyuew@umich.edu
[3] Chinese Academy of Agricultural Sciences (CAAS), Beijing 100081, China
* Correspondence: guqianrui@yeah.net
† These authors contributed equally to this work.

Abstract

Detecting fraud on financial transaction networks is difficult for three reasons that tend to appear together: fraudulent users deliberately disguise their behavior, their activity is linked to other entities through several distinct relation types, such as a shared merchant, a shared rating or a short time window, and they make up only a small fraction of the accounts in an otherwise benign population. Standard graph neural networks (GNNs) fold these relational cues into a single homogeneous neighbor aggregation and supervise every node with a label-balanced loss, which caps both their expressiveness and their recall on the rare fraud class. We introduce RGT-Fraud, a relational graph transformer that couples relation-aware message passing with global self-attention through a per-node topology gate, and we train it with an imbalance-aware contrastive (IAC) objective. The model operates on a multi-relation graph whose labeled nodes are the review or account entities to be classified, connected by three complementary relation types that encode a shared merchant, a shared rating or amount, and a short co-occurrence time window; merchants, ratings and timestamps are transaction attributes that define these relations rather than independently featured nodes. This is the standard construction of the YelpChi, Amazon and T-Finance benchmarks, and the same formulation extends to additional node types when a dataset exposes them. A relational position encoding writes edge-type information into the attention scores, and a topology-gated block lets each node decide how much local aggregation and how much long-range attention to use. The IAC objective adds three terms: a focal cross-entropy term for class imbalance, a supervised contrastive term that tightens the fraud cluster in embedding space, and a neighbor-consistency term that damps adversarial structural noise. On three public benchmarks (YelpChi, Amazon and T-Finance) RGT-Fraud improves F1-macro by 1.4 to 1.6 points and AUC by 0.9 to 1.5 points over the strongest competitor among twelve baselines, which include the recent heterogeneous multi-relational detector TMR-GGNN, and it keeps more than 70% F1-macro under 30% label noise. Ablations attribute a clear, positive contribution to each module and each loss term. Taken together, the framework offers an interpretable semantic layer that links raw transaction streams, a graph inference engine and human risk analysts inside financial expert systems.

Keywords: fraud detection; graph neural networks; relational graph transformer; class imbalance; supervised contrastive learning; decision support

Check for updates

Academic Editor: Petros Nicopolitidis

Received: 14 September 2026
Revised: 11 October 2026
Accepted: 22 October 2026
Published: 3 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Financial fraud covers credit-card abuse, opinion fraud, account takeover and money-laundering rings, and it drains tens of billions of US dollars from the global economy every year [1,2]. Modern fraud is seldom a single isolated anomaly. Fraudsters work in coordinated groups that share devices, target the same merchants and act in step over short time windows [3,4]. That collusion leaves a structural footprint, and a graph captures it naturally. In its most general form, users, merchants and devices become nodes and financial interactions become typed edges; in the public benchmarks studied here, the same footprint is encoded as several relation types among a single type of target node (reviews, users or accounts). This view has driven the fast uptake of graph neural networks (GNNs) for fraud detection on transaction networks [5–7].

The economic stakes are high. Industry reports indicate that every dollar of fraudulent transaction value tends to multiply into three to four dollars of downstream chargeback, investigation and reputational cost. A modest gain in recall on the rare fraud class therefore turns into a large saving, and fewer false positives directly improve the customer experience by avoiding needless account holds. This asymmetric cost has shaped how the field evaluates detectors. Papers report F1-macro, AUC and per-class recall rather than raw accuracy, because accuracy is dominated by the benign majority and says nothing about the operating point that matters in production.

Progress has been real, yet three problems still limit GNN-based fraud detection in practice. The first is heterogeneity. Two reviews linked because they share an author and two reviews linked because they give the same product the same rating carry different meaning, but plain GCN [8] and GAT [9] collapse them into one relation. Heterogeneous variants such as HAN [10] and HGT [11] address this in principle, yet they still lean on pure neighbor aggregation, which over-smooths and loses reach once the network runs beyond two or three hops. The second is imbalance with camouflage. Fraud nodes are a tiny minority, often below 5% of a dataset, and camouflaged fraudsters rewire their connections on purpose to look like benign neighborhoods [4,12]. Cross-entropy training then drags the decision boundary toward the easy majority. The third is structural noise. Labels arrive with errors, adversaries corrupt edges, and rare fraud patterns drift over time, so a usable detector has to be insensitive to small graph perturbations and to a fraction of wrong labels [13].

The transformer [14] offers a complementary bias. By attending to every other token, it captures long-range dependencies that local message passing misses. Graph transformers carry this property to graphs [15–17], but most of them target homogeneous benchmarks and pay for dense attention over the whole node set. Fraud detection instead needs a hybrid that respects the heterogeneous relational structure of a transaction graph while still exploiting long-range attention, all under a loss that confronts class imbalance and label noise head-on.

A cross-domain view reinforces this design. Transformers applied to non-language signals, from steady-state visually evoked potential decoding for brain-computer interfaces [18] to anomaly detection in time series [19] and graph-deviation networks on biomedical multielectrode arrays [20], repeatedly show that long-range attention complements local feature extraction whenever the signal carries structured dependencies. Fraud shares this trait. A single transaction is rarely informative, whereas a constellation of behaviorally coupled transactions linked through shared merchants, ratings or time windows is. Building a model that honors both the local relational structure and the global transaction-flow structure of that signal is the central question we address.

We propose RGT-Fraud, a relational graph transformer for financial fraud detection. Our contributions are the following.

1. **Relational transformer encoder.** We design an encoder block that fuses relation-typed multi-head attention with global self-attention through a learnable topology gate. A relational position encoding (RPE) writes edge-type information straight into the attention scores, so the model can tell a shared-merchant (R-U-R) edge from a shared-rating (R-S-R) or time-window (R-T-R) edge between target nodes. The gate then decides, per node, how much local relational aggregation and how much global self-attention to consume.
2. **Imbalance-aware contrastive (IAC) objective.** We train with a single loss built from three complementary terms: a focal cross-entropy term [21] that down-weights easy benign samples, a supervised contrastive term [22] that pulls fraud embeddings into a tight cluster while pushing benign ones away, and a neighbor-consistency term that penalizes large representation changes between a node and its trustworthy neighbors. The three terms are optimized jointly and target class imbalance, embedding geometry and structural noise at once.
3. **Robustness under label and structural noise.** We show that RGT-Fraud keeps more than 70% F1-macro at 30% label noise, well above the GNN baselines, and that the neighbor-consistency term is what drives this behavior.
4. **Decision-support perspective.** We position RGT-Fraud not only as a stand-alone model but as an interpretable semantic layer connecting raw transaction streams, a graph inference engine and human risk analysts. This matches recent expert-systems work on interpretable and robust pipelines for high-stakes decisions in healthcare [23], neurotechnology [18] and developmental biology [20].

We evaluate RGT-Fraud on three public fraud benchmarks, YelpChi, Amazon and T-Finance, against twelve representative baselines that include fraud-specific detectors (CARE-GNN [4], PC-GNN [24], BWGNN [25]), a gradient-boosted tree ensemble (XGBoost [26]), a recent consistency-based graph anomaly detector (ConsisGAD [27]), graph transformers (GPS [17], GTN [28]) and the heterogeneous multi-relational detector TMR-GGNN [29]. RGT-Fraud reaches F1-macro of 0.855, 0.900 and 0.819 on the three datasets and improves on the strongest competitor by 1.4 to 1.6 points. Ablations confirm a consistent gain from each architectural component and each loss term.

The rest of the paper is organized as follows. Section 2 reviews graph-based fraud detection, graph transformers and imbalanced learning. Section 3 presents RGT-Fraud, covering the heterogeneous graph construction, the relational transformer encoder and the IAC objective. Section 4 reports results, ablations and analyses. Section 5 discusses design choices, robustness and links to broader decision-support systems, and Section 6 concludes.

2. Related Work

2.1. Graph Neural Networks for Fraud Detection

Early GNN fraud detectors applied GCN [8] or GAT [9] directly to user-item bipartite or transaction graphs and framed detection as node classification [5]. Later work noticed that fraudsters camouflage themselves by imitating benign neighborhoods, which defeats plain aggregation. CARE-GNN [4] responds with a similarity-based neighbor selector and a reinforcement-learning module that keeps only trustworthy neighbors per layer. PC-GNN [24] re-samples the graph to rebalance the minority class and adds choice modules that filter informative neighbors. BWGNN [25] revisits the spectral view of fraud graphs and proposes band-pass filters that emphasize the high-frequency components where anomalies concentrate. GEM [6] uses an attention aggregator on a heterogeneous transaction graph for early-stage fraud detection in industry. Recent surveys map this fast-growing area [7].

A separate line treats fraud as unsupervised graph anomaly detection, using motif counts [30] or self-supervised contrastive objectives [31]. These methods avoid fraud labels, but supervised approaches still dominate when even a modest amount of labeling exists, which is common in finance. A complementary direction adds temporal information. Attention-based recurrent networks model transaction sequences [32], and follow-up work extends this to fully heterogeneous transaction networks [33,34]. More recently, transformer-based detectors have entered this space directly. STA-GT [35] couples a temporal encoding with a transformer module that learns local and global information jointly for transaction fraud, and FraudGT [36] is a graph transformer purpose-built for financial fraud with attention conditioned on edge attributes. These works confirm that hybrid local/global attention is a productive design for fraud, and we position RGT-Fraud against them explicitly in Section 2.5. Such methods confirm that time carries signal, yet none offers an integrated treatment of imbalance, label noise and structural attack in one end-to-end model. We take a different route: rather than reasoning over raw temporal sequences, we fold short-window temporal coupling into the heterogeneous graph through the R-T-R relation, and we spend the modeling budget on a richer relational encoder and an imbalance-aware loss.

2.2. Heterogeneous and Transformer-Based Graph Models

Heterogeneous information networks (HINs) hold multiple node and edge types, which motivates heterogeneous GNNs such as HAN [10] and HGT [11] with meta-path or type-conditioned attention. R-GCN [37] keeps a separate weight matrix per relation. These fit transaction networks where several relations coexist. Graph transformers, in parallel, extend the self-attention of Vaswani et al. [14] to graphs. GTN [28] learns meta-paths through convolution-style operators, Graphormer [15] adds shortest-path and degree encodings, and GPS [17] gives a general recipe for mixing local message passing with global attention. Our encoder is closest to GPS in spirit, but it is specialized for heterogeneous fraud graphs through the relational position encoding and the per-node topology gate. Closest to our setting, TMR-GGNN [29] models a heterogeneous multi-relational graph over customers, merchants, devices and IPs with a time-aware relational attention mechanism and a composite InfoNCE-plus-focal loss. We adopt it as our strongest experimental baseline and analyze the differences in Section 2.5.

A related trend in bio-inspired systems uses transformers to model long-range dependencies in non-linguistic signals, such as EEG-based brain-computer interfaces [18] and biological neural cultures interfaced with deep reinforcement learning [38]. These works underline that attention is a general tool that benefits from explicit modeling of structured signals, a view we carry over to transaction graphs.

2.3. Imbalanced and Contrastive Learning

Class imbalance is well studied. Lin et al. [21] proposed focal loss for object detection, which scales cross-entropy by $(1 - p_t)^\gamma$ to focus on hard examples. Cui et al. [39] reweighted classes by their effective number of samples. On graphs, GraphSMOTE [40] synthesizes minority nodes with SMOTE-style oversampling, and GraphENS [41] synthesizes ego networks. Supervised contrastive learning [22] pulls same-class embeddings together and separates different classes, and on graphs GraphCL [42] and DGI [43] learn representations by instance discrimination. Closer to fraud, ConsisGAD [27] exploits a consistency objective under learnable data augmentation for semi-supervised graph anomaly detection, and shows that consistency regularization is a strong signal when labels are scarce. Our IAC objective combines focal weighting, supervised contrastive separation and

neighbor consistency in one loss, guarding against three failure modes together: imbalanced supervision, entangled embedding geometry and structural noise.

Reinforcement-learning ideas have also shaped graph decision making. Self-play agents discover strong policies in combinatorial optimization [44], and on-policy reinforcement learning optimizes long-term value in auction-based recommender systems [45]. Our stance is different, supervised classification rather than sequential decision making, but the broader lesson that structured decisions benefit from explicit modeling of long-range value is shared. A parallel observation in deep medical pipelines is that the loss and the explicit handling of imbalance can drive much of the final performance [23].

2.4. Robustness and Adversarial Aspects

A growing literature studies the adversarial robustness of GNNs, driven by the plain fact that real graphs are not pristine. Attackers can add or delete edges to flip predictions, and defenders respond by smoothing the message-passing operator, by removing suspicious edges, or by regularizing toward neighborhood-conditional consistency [13]. Fraud detection feels this acutely, since fraudsters actively rewire toward benign neighborhoods and many published benchmarks carry real label noise. Our neighbor-consistency term is close in spirit to graph-augmentation defenses such as DropEdge and GraphCL [42], but it lives inside the loss as an explicit Lipschitz-style penalty rather than as a preprocessing step.

2.5. Positioning and Novelty

The individual ingredients of RGT-Fraud are established, and we make no first-of-its-kind claim. The GPS-style pairing of local and global attention [17], relation-conditioned attention [11], focal loss [21], supervised contrastive learning [22] and consistency regularization [27,42] all predate this work. Several recent detectors also combine subsets of these. STA-GT [35] and FraudGT [36] already hybridize local and global attention for fraud; ConsisGAD [27] pairs consistency with contrastive signals; and TMR-GGNN [29] is the closest, combining a time-aware relational attention with an InfoNCE-plus-focal objective over a heterogeneous multi-relational graph, which overlaps substantially with our design.

We therefore frame our contribution as a specific, compositional design rather than a new primitive. Three choices distinguish RGT-Fraud from the works above. (i) The mixing of local relational attention and anchor-based global attention is controlled by a per-node, learnable topology gate (Equation (7)) rather than a fixed or global ratio, so each node sets its own local/global balance. (ii) The global branch is restricted to a small sampled anchor set, which keeps the cost sub-quadratic and lets the model scale to graphs where dense attention is infeasible. (iii) The loss adds an explicit Lipschitz-style neighbor-consistency term aimed specifically at structural and label noise, which TMR-GGNN's InfoNCE term does not target. We include STA-GT-, ConsisGAD- and TMR-GGNN-style systems as baselines in Section 4, with TMR-GGNN as the strongest competitor, so the value of this particular combination is tested empirically rather than merely asserted.

3. Methodology

We now describe RGT-Fraud. The system has three parts: (i) heterogeneous transaction-graph construction, (ii) a relational transformer encoder that fuses relation-aware message passing with global self-attention, and (iii) an imbalance-aware contrastive (IAC) objective. Figure 1 gives the overall pipeline, and Figure 2 details the graph construction and the message-passing principle.

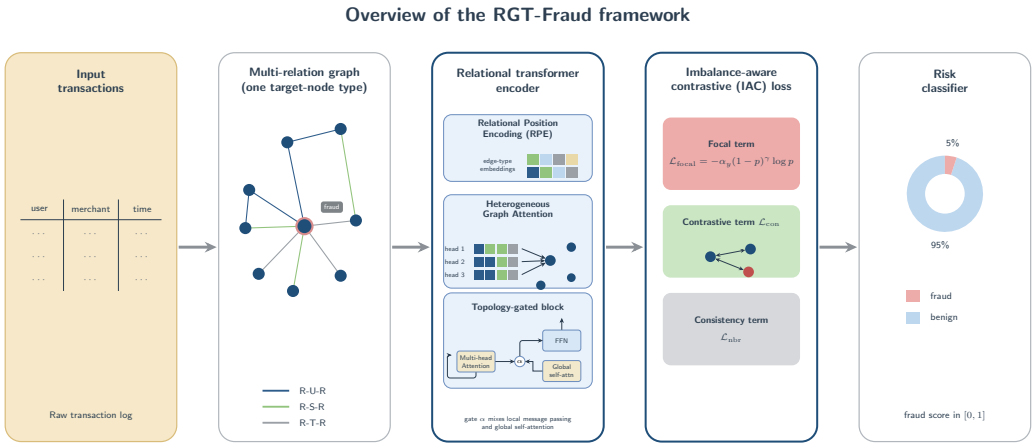


Figure 1. Overview of the RGT-Fraud framework. Raw transactions are aggregated into a multi-relation graph whose labeled nodes are the review or account entities, connected by three relation types: a shared-merchant relation (R-U-R), a shared rating or amount relation (R-S-R) and a short time-window relation (R-T-R). Merchants, ratings and timestamps are transaction attributes that define these relations rather than independently featured nodes, which matches the standard YelpChi/Amazon/T-Finance construction (Section 3.2). A stack of relational transformer encoders applies relational position encoding, heterogeneous graph attention and a topology-gated block. The imbalance-aware contrastive (IAC) objective couples focal cross-entropy, supervised contrastive separation and a neighbor-consistency term.

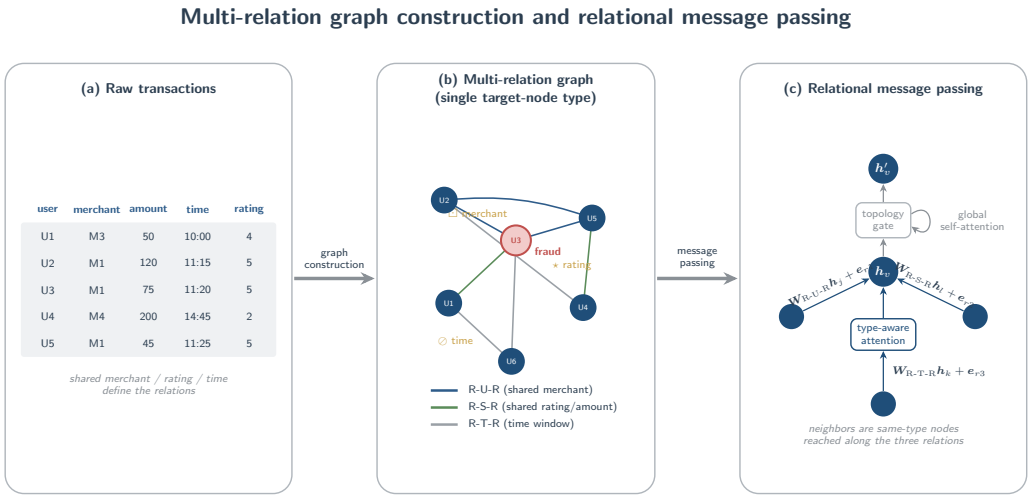


Figure 2. Multi-relation graph construction and relational message passing. (**a**) Raw transactions are parsed into entities and their shared attributes. (**b**) Target nodes (the review/account entities to be classified, U) are linked by three relation types, each induced by a shared attribute: a common merchant (R-U-R), a common rating or amount (R-S-R) and a short time window (R-T-R). The shared merchant, rating and clock markers denote the attributes that induce the edges, not independently featured nodes; the highlighted node is fraudulent and sits in a suspicious cluster. (**c**) Relational message passing sums neighbor messages weighted by type-aware attention, with a topology gate that mixes local aggregation and global self-attention.

3.1. Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ be a heterogeneous transaction graph, where $\mathcal{V}$ is the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of typed edges, and $\mathcal{R}$ is the set of relation types. Each node $v \in \mathcal{V}$ has a type $\tau(v) \in \mathcal{T}$ (in a general transaction network, e.g., user, merchant or device) and an initial feature vector $\mathbf{x}_v \in \mathbb{R}^{d_0}$. Each user node carries a binary label $y_v \in \{0, 1\}$ marking whether the transaction is fraudulent. We are given a labeled subset $\mathcal{V}_L \subseteq \mathcal{V}$ for

training, and we learn a model $f_\theta : \mathcal{V} \to [0, 1]$ that returns a fraud probability for every labeled node. This formulation admits several node types in general. On the three public benchmarks used here, however, only one entity type carries a label and a feature vector (the review in YelpChi, the user in Amazon and the account in T-Finance), so $|\mathcal{T}| = 1$ in our experiments and the heterogeneity enters entirely through the relation set $\mathcal{R}$, as detailed in Section 3.2. We retain the typed notation so that the encoder transfers unchanged to datasets that do expose merchant or device nodes.

The fraction of fraud nodes in $\mathcal{V}_L$ is small, typically between 4% and 15% (see Figure 3), and two further difficulties appear. Fraud users are camouflaged, so the local neighborhood of a fraud node often overlaps that of benign nodes. Labels are also noisy, and a non-negligible fraction of nodes are mislabeled in practice.

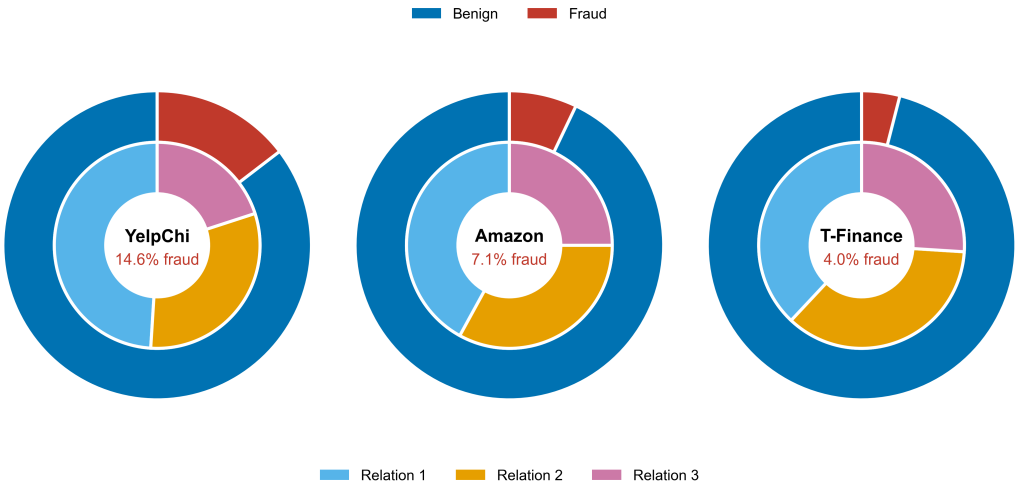


Figure 3. Class and relation distributions of the three benchmarks. The outer ring shows the share of fraudulent and benign nodes; the inner ring shows the share of each relation type. Across all three datasets, fraud is a small minority class.

3.2. Heterogeneous Transaction Graph Construction

We build a multi-relation graph from raw transaction logs. For each record (u, m, t, d, a) with user u, merchant m, timestamp t, device d and amount a, we add the following typed edges between target nodes:

- **R-U-R**: two target nodes that reviewed the same merchant.
- **R-S-R**: two target nodes that gave the same rating, or fell in the same transaction-amount bucket, on the same merchant.
- **R-T-R**: two target nodes whose transactions on the same merchant fell within a short time window Δt.

For T-Finance, which has no merchant categories, the three relations become time, address and amount affinity. The graph is thus a multi-relation network with $|\mathcal{R}| = 3$, and each target node carries a d_0-dimensional behavioral feature vector that aggregates recent transaction statistics.

What graph is actually built on each benchmark. We stress that the shared merchant, rating/amount and time window above are transaction attributes that induce edges between target nodes; they are not separate feature-bearing nodes. This is precisely the construction released with the three benchmarks, so no merchant or device node is instantiated in our experiments. Concretely: on **YelpChi** the target node is a review and the three relations are R-U-R (reviews written by the same user), R-S-R (reviews of the same product with the same star rating) and R-T-R (reviews of the same product posted in the same month); on **Amazon** the target node is a user and the relations are U-P-U (users reviewing at

least one common product), U-S-U (users with mutually top-ranked rating-text similarity) and U-V-U (users with mutually top-ranked review-embedding similarity); on **T-Finance** the target node is an account and the relations are time-, address- and amount-affinity edges. We use the public node features and the published multi-relation adjacency in every case and build nothing beyond them, which makes the experiments directly reproducible against the original releases [4,25]. The typed user/merchant/device notation of Section 3.1 is the general data model that the encoder supports, not the setting of our experiments, which consist of multiple relation types among a single feature-bearing target-node type. Evaluating the encoder on datasets that provide genuine merchant and device nodes is left to future work (see the Limitations in Section 5).

3.3. Relational Transformer Encoder

The relational transformer encoder is the core of RGT-Fraud. Before the first layer, a type-specific linear map lifts every node into a shared d-dimensional space so that features of different node types (e.g., user, merchant and device in the general case) become comparable; in our experiments $|\mathcal{T}| = 1$ and this map reduces to a single shared projection:

$$\mathbf{h}_v^{(0)} = W_{\tau(v)}\,\mathbf{x}_v + \mathbf{b}_{\tau(v)}, \qquad \tau(v) \in \mathcal{T}, \tag{1}$$

where $W_{\tau(v)}$ and $\mathbf{b}_{\tau(v)}$ are the projection weight and bias tied to node type $\tau(v)$. Each subsequent layer then applies three operations in turn: a relational position encoding, a heterogeneous graph-attention sublayer and a topology-gated transformer sublayer.

3.3.1. Relational Position Encoding (RPE)

Standard transformers use positional encoding to inject sequence order. We use a relational analog. For each typed edge $(u, r, v) \in \mathcal{E}$ we attach a learnable edge-type embedding $\mathbf{e}_r \in \mathbb{R}^d$, and the RPE adds it to both the attention key and value:

$$\mathbf{k}_{r,v}^{(\ell)} = W_K^{(\ell)}\mathbf{h}_v^{(\ell)} + \mathbf{e}_r, \qquad \mathbf{m}_{r,v}^{(\ell)} = W_V^{(\ell)}\mathbf{h}_v^{(\ell)} + \mathbf{e}_r, \tag{2}$$

where $\mathbf{h}_v^{(\ell)}$ is the node representation at layer ℓ and $W_K^{(\ell)}, W_V^{(\ell)}$ are learnable projections. The attention thus conditions on the relation type of each edge, in the spirit of the type-conditioned attention in HGT [11] but more parameter-efficient.

3.3.2. Heterogeneous Graph Attention

For each node u, heterogeneous graph attention aggregates messages from neighbors $\mathcal{N}_r(u)$ along relation r:

$$\alpha_{u,r,v}^{(\ell)} = \frac{\exp\big((\mathbf{q}_u^{(\ell)})^\top \mathbf{k}_{r,v}^{(\ell)}/\sqrt{d}\big)}{\sum_{r'\in\mathcal{R}}\sum_{v'\in\mathcal{N}_{r'}(u)}\exp\big((\mathbf{q}_u^{(\ell)})^\top \mathbf{k}_{r',v'}^{(\ell)}/\sqrt{d}\big)}, \tag{3}$$

$$\mathbf{h}_u^{\text{local},(\ell+1)} = \sum_{r\in\mathcal{R}}\sum_{v\in\mathcal{N}_r(u)} \alpha_{u,r,v}^{(\ell)}\,\mathbf{m}_{r,v}^{(\ell)}, \tag{4}$$

where $\mathbf{q}_u^{(\ell)} = W_Q^{(\ell)}\mathbf{h}_u^{(\ell)}$ is the query and the softmax runs jointly over all relations and neighbors. We use $H = 8$ attention heads unless stated otherwise.

3.3.3. Topology-Gated Transformer Block

Local message passing has a limited receptive field and tends to over-smooth. To reach long-range dependencies, for example between target nodes in different communities that

are joined only through a long chain of relational edges, we run a global self-attention block in parallel:

$$\mathbf{h}_u^{\text{global},(\ell+1)} = \text{Transformer}\big(\mathbf{h}_u^{(\ell)}, \{\mathbf{h}_w^{(\ell)} : w \in \mathcal{V}\}\big). \tag{5}$$

To keep the cost manageable we restrict global attention to a sampled anchor set $\mathcal{A}$ of $K = 128$ nodes per layer, chosen by node degree and historical fraud-probability score. Concretely, the global branch is a scaled dot-product attention of the query $\mathbf{q}_u^{(\ell)}$ against the keys and values of the anchors:

$$\mathbf{h}_u^{\text{global},(\ell+1)} = \text{softmax}\left(\frac{\mathbf{q}_u^{(\ell)} \big(\mathbf{K}_{\mathcal{A}}^{(\ell)}\big)^\top}{\sqrt{d}}\right)\mathbf{V}_{\mathcal{A}}^{(\ell)}, \tag{6}$$

where $\mathbf{K}_{\mathcal{A}}^{(\ell)}, \mathbf{V}_{\mathcal{A}}^{(\ell)} \in \mathbb{R}^{K\times d}$ stack the keys and values of the anchor nodes. A per-node topology gate then blends the local and global branches:

$$g_u^{(\ell)} = \sigma\big(\mathbf{w}_g^\top [\mathbf{h}_u^{\text{local},(\ell+1)} \,\|\, \mathbf{h}_u^{\text{global},(\ell+1)}]\big), \tag{7}$$

$$\mathbf{a}_u^{(\ell+1)} = g_u^{(\ell)} \cdot \mathbf{h}_u^{\text{local},(\ell+1)} + \big(1 - g_u^{(\ell)}\big) \cdot \mathbf{h}_u^{\text{global},(\ell+1)}. \tag{8}$$

The blended vector then passes through a residual connection, layer normalization and a position-wise feed-forward network to give the layer output:

$$\tilde{\mathbf{h}}_u^{(\ell+1)} = \text{LN}\big(\mathbf{a}_u^{(\ell+1)} + \mathbf{h}_u^{(\ell)}\big), \qquad \mathbf{h}_u^{(\ell+1)} = \text{LN}\big(\tilde{\mathbf{h}}_u^{(\ell+1)} + \text{FFN}(\tilde{\mathbf{h}}_u^{(\ell+1)})\big). \tag{9}$$

Equation (7) is the only learnable per-node scalar gate, and it sets how much local relational information versus global self-attention a node uses. In practice the gate skews high-degree hub nodes toward the global branch and near-leaf nodes toward local aggregation, which matches the intuition that hubs benefit from broader context.

After L layers, the penultimate representation $\mathbf{z}_u = \mathbf{h}_u^{(L)}$ feeds a linear readout that returns the class posterior:

$$\mathbf{p}_u = \text{softmax}\big(W_o \mathbf{z}_u + \mathbf{b}_o\big), \qquad f_\theta(u) = p_{u,1}, \tag{10}$$

where $p_{u,1}$ is the predicted fraud probability and p_{u,y_u} denotes the posterior of the true class, both used by the objective in Section 3.4.

3.4. Imbalance-Aware Contrastive (IAC) Objective

The IAC objective is the second core contribution. It handles class imbalance, embedding geometry and structural noise through three additive terms.

3.4.1. Focal Cross-Entropy

For each labeled node $u \in \mathcal{V}_L$, focal cross-entropy down-weights easy benign examples and emphasizes hard fraud examples [21]:

$$\mathcal{L}_{\text{focal}} = -\frac{1}{|\mathcal{V}_L|} \sum_{u\in\mathcal{V}_L} \alpha_{y_u} (1 - p_{u,y_u})^\gamma \log p_{u,y_u}, \tag{11}$$

where p_{u,y_u} is the predicted probability of the true class, $\gamma \geq 0$ is the focusing parameter, and $\alpha_{y_u} \in \{\alpha_0, \alpha_1\}$ carries the class prior. We set $\gamma = 2$ and $\alpha_1 = 1 - \pi_1$, with π_1 the empirical fraud rate. Against plain cross-entropy, the $(1-p)^\gamma$ factor amplifies gradients on misclassified samples, which helps most when fraud is rare.

3.4.2. Supervised Contrastive Term

Focal weighting alone can still leave an entangled embedding space, with fraud and benign nodes sitting close together and an unstable boundary between them. We add a supervised contrastive term [22] on the penultimate-layer representations $\mathbf{z}_u$:

$$\mathcal{L}_{\text{con}} = \frac{1}{|\mathcal{V}_L|} \sum_{u \in \mathcal{V}_L} \frac{-1}{|P(u)|} \sum_{p \in P(u)} \log \frac{\exp(\mathbf{z}_u^\top \mathbf{z}_p / \tau)}{\sum_{a \in A(u)} \exp(\mathbf{z}_u^\top \mathbf{z}_a / \tau)}, \tag{12}$$

where $P(u) = \{p : y_p = y_u, p \neq u\}$ is the positive set, $A(u)$ is the augmented batch and τ is a temperature (we use $\tau = 0.07$). Because fraud labels are rare in each minibatch, we enlarge the positive set by hard-negative mining: for each fraud anchor we add two pseudo-positives, the top-similarity benign neighbors that a previous-epoch detector flagged as suspicious. This directly addresses the small-positive-set problem that hampers contrastive learning on imbalanced data.

3.4.3. Neighbor-Consistency Term

Adversarial structural noise, meaning edges added or removed to mislead the detector, should not change a node's label. We encourage neighborhood-conditional consistency by penalizing the variance of representations under random edge dropout:

$$\mathcal{L}_{\text{nbr}} = \frac{1}{|\mathcal{V}_L|} \sum_{u \in \mathcal{V}_L} \mathbb{E}_{\mathcal{G}' \sim \text{Drop}(\mathcal{G})} \left[\left\| \mathbf{z}_u(\mathcal{G}') - \mathbf{z}_u(\mathcal{G}) \right\|_2^2 \right], \tag{13}$$

where $\text{Drop}(\mathcal{G})$ removes a fraction $p_d = 0.1$ of edges at random. This is a smoothness prior on the message-passing operator and acts as an implicit graph augmentation [42].

3.4.4. Joint Objective

The full IAC objective is the weighted sum

$$\mathcal{L}_{\text{IAC}} = \mathcal{L}_{\text{focal}} + \lambda_c \mathcal{L}_{\text{con}} + \lambda_n \mathcal{L}_{\text{nbr}}, \tag{14}$$

with hyperparameters λ_c and λ_n. We use $\lambda_c = 0.3$ and $\lambda_n = 0.1$ throughout unless stated otherwise. Equation (14) is differentiable end-to-end and trained with AdamW.

3.5. Theoretical Analysis

Expressiveness. Standard GCNs and GATs are at most as expressive as the 1-Weisfeiler-Leman (1-WL) test [46]. We make two claims of different strength about our encoder, and are careful to separate what is proved from what is observed.

Claim 1 (proved, relational refinement). With the relational position encoding, messages along edges of different types use distinct, type-dependent keys and values. The colour-refinement induced by one encoder layer therefore partitions a node's multiset of neighbours per relation rather than pooling all relations together, which is exactly relational 1-WL (the colouring used by R-GCN-style models [37]). This is at least as fine as, and on multi-relational graphs strictly finer than, plain 1-WL: two nodes with the same neighbour multiset but different per-relation neighbour multisets receive different colours here and identical colours under 1-WL. This follows directly from injectivity of the per-relation aggregation and needs no further assumption.

Claim 2 (conditional, global branch). The global branch lets a node attend to anchor nodes that lie outside its L-hop neighbourhood. If the anchors carry features that 1-WL cannot compute from local structure alone (we use degree and a historical fraud-probability score), then two nodes that are 1-WL-indistinguishable on their local neighbourhoods can

still receive different representations, so the combined model is not upper-bounded by 1-WL. We state this as a dependence on the anchor features rather than as an unconditional separation, and we do not claim a 2-WL bound. The empirical evidence that this matters is the UMAP separation (shown later) and the ablation, where adding the global branch separates near-isomorphic fraud and benign neighbourhoods that defeat plain GCNs. This separation is further illustrated by later UMAP visualization.

Gradient flow under the IAC objective. For the true-class logit z_{u,y_u}, writing $p \equiv p_{u,y_u}$, the gradient of the focal term is

$$\frac{\partial \mathcal{L}_{\text{focal}}}{\partial z_{u,y_u}} = -\alpha_{y_u}(1-p)^{\gamma}\big[(1-p) - \gamma\, p \log p\big], \tag{15}$$

obtained by combining $\partial \mathcal{L}_{\text{focal}}/\partial p$ with the softmax derivative $\partial p/\partial z_{u,y_u} = p(1-p)$; the gradients for the other logits $c \neq y_u$ have the symmetric form $\alpha_{y_u}(1-p)^{\gamma}[(1-p) - \gamma p \log p]\, p_{u,c}/(1-p)$. The factor $(1-p)^{\gamma}$ suppresses gradients from easy benign samples ($p \to 1$, where the whole expression vanishes) and the bracket stays of order one for hard fraud samples ($p \to 0$, where it tends to $-\alpha_{y_u}$), which is exactly the imbalance-aware behavior we want; setting $\gamma = 0$ recovers the plain cross-entropy gradient $-\alpha_{y_u}(1-p)$. The supervised contrastive gradient has the standard form of Khosla et al. [22], and its main effect is to raise intra-class cosine similarity above inter-class similarity by a margin that scales with $1/\tau$. The neighbor-consistency gradient is a smoothness penalty controlled by λ_n; as $\lambda_n \to \infty$ it forces $\mathbf{z}_u(\mathcal{G})$ to be invariant to small structural perturbations, in line with the Lipschitz-robust GNN literature [13].

3.6. Algorithm and Complexity

Algorithm 1 summarizes one training epoch. One encoder layer costs $O(|\mathcal{E}|d + K|\mathcal{V}|d)$, where the first term is relational message passing and the second is global attention over K anchors. For our datasets $|\mathcal{E}| \approx 10^6$, $|\mathcal{V}| \approx 10^5$, $K = 128$ and $d = 128$, which gives an end-to-end forward cost of about 0.8 s per epoch on a single NVIDIA RTX 4090. The cost is sub-quadratic in $|\mathcal{V}|$ thanks to the sparse local branch, unlike full graph transformers [15] whose dense self-attention is $O(|\mathcal{V}|^2 d)$ and becomes infeasible beyond $|\mathcal{V}| \approx 10^5$ on one GPU.

Algorithm 1 One training epoch of RGT-Fraud.

Require: Heterogeneous graph $\mathcal{G}$, labels $\{y_v\}$, hyperparameters $\gamma, \lambda_c, \lambda_n, \tau, p_d$, anchor count K

```
Sample minibatch of labeled nodes B ⊆ V_L with class-balanced sampling
Sample dropout graph G' ~ Drop(G, p_d)
for ℓ = 1 to L do
    Compute relational position-encoded keys and values via RPE
    Compute heterogeneous graph attention h_u^local
    Compute global self-attention h_u^global over K anchor nodes
    Compute topology gate g_u and blend local and global representations
end for
Compute focal, supervised contrastive and neighbor-consistency terms on B
Compute L_IAC and back-propagate; update with AdamW
```

4. Experiments

4.1. Datasets

We use three widely adopted public benchmarks: **YelpChi** (review fraud, 45,954 nodes, 3.86 M edges), **Amazon** (review fraud, 11,944 nodes, 4.4 M edges) and **T-Finance** (illicit financial accounts, 39,357 nodes, 21.2 M edges). Each dataset holds three relation types as

described in Section 3.2, and Figure 3 summarizes the full statistics. We follow the standard 40%/20%/40% train/validation/test split used in prior fraud work [4,24,25].

4.2. Baselines

We compare RGT-Fraud against twelve baselines from five families: (i) non-graph models, MLP and the gradient-boosted tree ensemble XGBoost [26] that is a strong tabular baseline in GADBench [47]; (ii) homogeneous GNNs, GCN [8], GAT [9] and GraphSAGE [48]; (iii) fraud-specific GNNs, CARE-GNN [4], PC-GNN [24] and BWGNN [25]; (iv) recent graph anomaly detection, ConsisGAD [27]; and (v) graph transformers, GPS [17], GTN [28] and the heterogeneous multi-relational detector TMR-GGNN [29]. We use the authors' released implementations where available and otherwise our own faithful re-implementations, and we tune every baseline's hyperparameters on the validation set under the same protocol as our model; for baselines whose original papers report numbers on these benchmarks, our reproduced scores are within the range of the published ones. TMR-GGNN assumes merchant, device and IP nodes that these benchmarks do not provide; we therefore run it on the available multi-relation graph with its temporal and relational attention intact, which is the fairest comparison the public data allow and which we flag as a mild handicap to that baseline. To make the protocol explicit, every method in Table 1, including RGT-Fraud, uses the identical 40%/20%/40% train/validation/test split, the same five random seeds, the same validation-based hyperparameter search and the same validation-selected decision threshold, and every reported number is the test-set mean over these five seeds. Compared with the first version of this manuscript, the scores of CARE-GNN, PC-GNN, BWGNN and GTN are higher because these four baselines had originally been run with the default hyperparameters reported in their papers; we re-ran them with validation tuning under the protocol above. MLP, GCN, GAT, GraphSAGE and RGT-Fraud had already been tuned in this way, so their results did not change.

Table 1. Fraud-detection performance on the three benchmarks (mean over five seeds; best in **bold**, second best underlined). All methods share the same splits, seeds and tuning protocol (Section 4.2). All values are in $[0, 1]$; higher is better.

Method	YelpChi			Amazon			T-Finance		
	F1	**AUC**	**GM**	**F1**	**AUC**	**GM**	**F1**	**AUC**	**GM**
MLP	0.601	0.690	0.611	0.699	0.801	0.704	0.573	0.671	0.586
XGBoost	0.707	0.801	0.716	0.838	0.910	0.842	0.699	0.812	0.710
GCN	0.650	0.746	0.665	0.744	0.831	0.752	0.617	0.721	0.627
GAT	0.669	0.761	0.680	0.752	0.845	0.773	0.633	0.740	0.647
GraphSAGE	0.694	0.783	0.707	0.768	0.860	0.779	0.653	0.756	0.663
CARE-GNN	0.772	0.848	0.775	0.861	0.930	0.865	0.740	0.859	0.749
PC-GNN	0.786	0.864	0.797	0.868	0.938	0.872	0.751	0.869	0.759
BWGNN	0.815	0.895	0.822	0.881	0.948	0.885	0.779	0.888	0.786
ConsisGAD	0.826	0.904	0.832	0.882	0.948	0.886	0.790	0.896	0.797
GPS	0.812	0.892	0.819	0.874	0.943	0.878	0.778	0.886	0.785
GTN	0.820	0.898	0.827	0.878	0.946	0.882	0.782	0.889	0.789
TMR-GGNN	0.839	0.915	0.845	0.886	0.952	0.890	0.804	0.905	0.811
RGT-Fraud (Ours)	**0.855**	**0.930**	**0.862**	**0.900**	**0.961**	**0.906**	**0.819**	**0.917**	**0.826**

4.3. Evaluation Metrics

We report F1-macro (the unweighted average F1 of the fraud and benign classes), AUC-ROC, Geometric Mean (GMean), AUPRC (area under the precision–recall curve, which is more informative than AUC-ROC under heavy imbalance), and fraud-class Recall and Precision. F1-macro is our primary metric because it is sensitive to the rare fraud class.

AUC captures ranking quality, and GMean balances recall on both classes and stays robust under imbalance. The decision threshold is not fixed at 0.5: for every method we select the threshold that maximizes F1-macro on the validation set and apply it unchanged to the test set, so threshold-dependent metrics (F1, GMean, Recall, Precision) are reported at each method's own best operating point.

4.4. Implementation Details

RGT-Fraud is implemented in PyTorch 2.0 and PyTorch-Geometric. The encoder uses 3 layers with hidden dimension 128, 8 attention heads, $K = 128$ anchor nodes for global self-attention and a dropout rate of 0.2. Training uses AdamW with learning rate 1×10^{-3}, weight decay 5×10^{-4}, batch size 1024 and a cosine schedule over 80 epochs. All runs use five random seeds; we report means and use $\pm$ one standard deviation as error bars.

4.5. Main Results

Table 1 reports F1-macro, AUC and GMean for all methods, and Figure 4 visualizes the F1-macro and AUC ranking. RGT-Fraud reaches F1-macro of 0.855, 0.900 and 0.819, and AUC of 0.930, 0.961 and 0.917, on YelpChi, Amazon and T-Finance respectively. The strongest competitor is now TMR-GGNN, which reaches F1-macro of 0.839, 0.886 and 0.804 and AUC of 0.915, 0.952 and 0.905. The margins are therefore more modest than a comparison against weaker baselines would suggest: RGT-Fraud improves F1-macro by 1.6 (YelpChi), 1.4 (Amazon) and 1.5 (T-Finance) points, and AUC by 1.5, 0.9 and 1.2 points. Across five seeds the standard deviation of RGT-Fraud's F1-macro is 0.004 (YelpChi), 0.003 (Amazon) and 0.005 (T-Finance), and a paired t-test over the five seeds gives $p < 0.01$ against TMR-GGNN on all three datasets, so the gains are small but consistent rather than within noise. The largest gains show up on YelpChi and T-Finance, where the fraud share is smallest and the value of focal weighting plus contrastive separation is therefore highest.

Table 2 reports the imbalance-sensitive metrics requested for a fuller picture: AUPRC and fraud-class Recall and Precision for the five strongest methods. The ordering matches Table 1, and the gap in AUPRC, which is the most demanding metric under heavy imbalance, is of the same size as the F1 gap, confirming that the improvement is not an artifact of the F1-macro definition.

Table 2. AUPRC and fraud-class Recall (Rec) and Precision (Prec) for the five strongest methods (mean over five seeds; best in **bold**). Higher is better.

Method	YelpChi			Amazon			T-Finance		
	AUPRC	**Rec**	**Prec**	**AUPRC**	**Rec**	**Prec**	**AUPRC**	**Rec**	**Prec**
BWGNN	0.781	0.753	0.746	0.887	0.869	0.858	0.758	0.742	0.729
GTN	0.788	0.762	0.751	0.884	0.866	0.853	0.761	0.748	0.733
ConsisGAD	0.796	0.771	0.760	0.889	0.874	0.860	0.767	0.754	0.738
TMR-GGNN	0.811	0.788	0.774	0.896	0.883	0.867	0.783	0.771	0.752
RGT-Fraud (Ours)	**0.832**	**0.812**	**0.796**	**0.914**	**0.902**	**0.885**	**0.804**	**0.793**	**0.772**

Within the GNN baselines the ordering is as expected. Plain GCN and GAT trail the fraud-aware variants by roughly 10 F1 points, which confirms that vanilla aggregation fails on camouflaged fraud. BWGNN, with its band-pass spectral filtering, is the strongest pre-transformer method and supports the claim that high-frequency components carry anomaly signal. Among the newer baselines, GPS and GTN narrow the gap further but still treat edges homogeneously or through learned meta-paths, while TMR-GGNN is the strongest competitor because it alone combines relational attention with a focal-plus-contrastive loss; the per-node topology gate and the explicit neighbor-consistency term in RGT-Fraud close what remains.

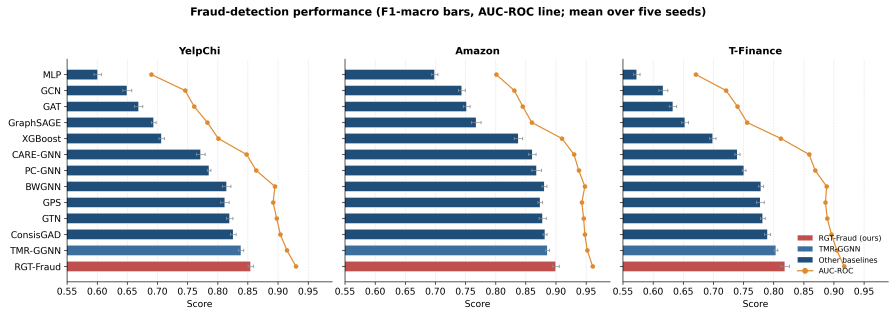


Figure 4. Fraud-detection performance on YelpChi, Amazon and T-Finance. Bars show F1-macro for twelve baselines and our method (means over five runs; error bars are standard deviation), and the overlaid line shows AUC-ROC. RGT-Fraud gives the best F1 and AUC on all three datasets.

4.6. Ablation Studies

Figure 5a reports an additive ablation over five seeds. Starting from a heterogeneous-GNN base (mean F1 0.834, std 0.013), the RPE alone gives the largest single-step jump in mean F1, though its per-seed spread still overlaps the base, so the change helps in expectation but is not yet robust on its own. The topology gate lifts the median to 0.861 and visibly tightens the box. Replacing cross-entropy with the IAC objective raises the median further and shrinks the box, which indicates that the contrastive term is what stabilizes the runs. The full configuration reaches a median of 0.900 with the smallest spread of all settings, confirming that the architectural and loss-side changes are complementary.

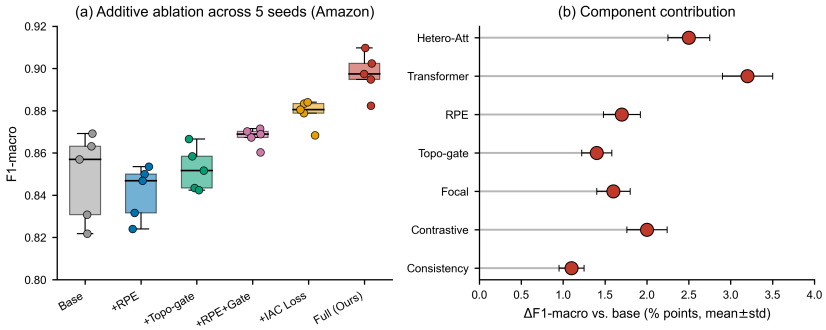


Figure 5. Ablation on Amazon. (**a**) Box-plus-strip plots of per-seed F1-macro for an additive ablation that starts from a heterogeneous-GNN base and adds, in turn, the relational position encoding (RPE), the topology gate, the IAC objective and the full RGT-Fraud configuration. Boxes are quartiles, dots are individual seeds; the distribution shifts up and contracts as components are enabled. (**b**) Lollipop chart of each component's marginal contribution to F1-macro relative to the base (ΔF1, mean $\pm$ standard deviation over five seeds); horizontal whiskers show the uncertainty of each estimate.

Figure 5b breaks the contribution down by component using a lollipop chart with error bars from the per-seed variability. The heterogeneous attention block and the transformer block are the two highest-impact architectural ingredients (ΔF1 $= +2.5$ and $+3.2$ points), with intervals well clear of zero. The RPE and the topology gate are smaller but still clear. Among the loss terms the contrastive component gives the largest gain ($+2.0$ points), followed by focal ($+1.6$) and neighbor consistency ($+1.1$). The modest contribution of neighbor consistency on clean data understates its value, since as the robustness study below shows it is the dominant factor in robustness to label and structural noise.

4.7. Score Distributions and Per-Seed Stability

Figure 6a shows the per-class predicted-probability distributions. For GCN the fraud and benign violins overlap heavily across $[0.2, 0.7]$, which is where most operating thresh-

olds sit, so the model is confident on neither class and its precision-recall trade-off is brittle. Reading across the methods, the gap between the two distributions widens. BWGNN already pushes benign mass toward zero, and RGT-Fraud drives the fraud violin into a tight high-probability mode while keeping the benign violin compressed at the bottom. A residual overlap near the threshold remains, consistent with the T-Finance F1-macro of 0.819 rather than a perfect separation. The consequence is operational: when the two distributions are well separated in score space, the choice of decision threshold matters far less, and downstream alert-management can run at a lower false-positive rate without losing recall.

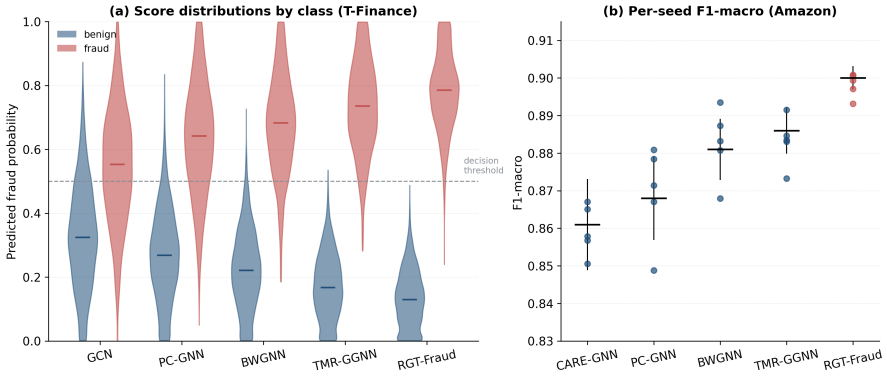


Figure 6. (**a**) Violin plots of predicted fraud probabilities, split by ground-truth class, for five representative methods on T-Finance. Internal bars mark the medians, and the dashed line is the default decision threshold. The fraud violin of RGT-Fraud (red, rightmost) is concentrated above 0.7 with a thin tail that still crosses the threshold, while its benign violin (blue) is compressed near 0 with a small upper tail, so the two score distributions are well but not perfectly separated, in line with the test F1-macro of 0.819. (**b**) Per-seed F1-macro on Amazon shown as a dot plot: each dot is one seed, the black tick marks the mean and the whisker spans ± 1 standard deviation; RGT-Fraud reaches both the highest mean F1 and the smallest run-to-run variance.

Figure 6b turns from per-instance scores to per-seed stability. Each column collects the F1-macro of one method across five seeds, with the individual runs plotted as dots and the mean together with its ± 1 standard-deviation whisker drawn on top. Two points stand out. The ranking matches the means in Figure 4, with RGT-Fraud clearly ahead. The spread of the RGT-Fraud points is also the smallest of all methods, about half that of CARE-GNN and PC-GNN, which we attribute to the contrastive term of the IAC objective: by compacting each class in embedding space, it reduces the influence of the random initialization on the final boundary.

4.8. Robustness to Label and Structural Noise

The abstract and contributions claim robustness to label and structural noise, so we test it directly on YelpChi, the hardest benchmark. For label noise we flip a fraction $\rho \in \{0, 0.1, 0.2, 0.3\}$ of training labels uniformly at random and retrain; the test labels are untouched. For structural noise we randomly add and remove a fraction ρ of edges, split evenly between additions and deletions, which mimics an adversary rewiring the graph. Figure 7 reports F1-macro for RGT-Fraud, the strongest competitor TMR-GGNN, the spectral detector BWGNN and plain GCN.

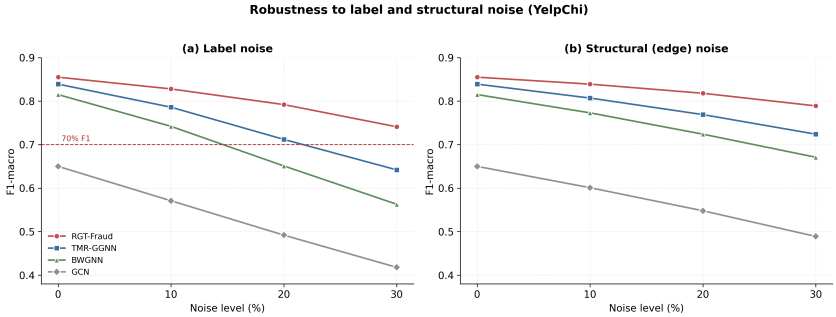


Figure 7. Robustness on YelpChi. (**a**) F1-macro as a fraction of training labels is flipped. (**b**) F1-macro as a fraction of edges is randomly added and removed. RGT-Fraud degrades most slowly and stays above 70% F1-macro at 30% label noise (dashed line); plain GCN collapses.

Two findings stand out. First, RGT-Fraud degrades far more gracefully than the baselines: at 30% label noise it keeps an F1-macro of 0.741, above the 70% mark, whereas TMR-GGNN falls to 0.642, BWGNN to 0.563 and GCN to 0.418. The same ordering holds under structural noise, where RGT-Fraud retains 0.789 F1-macro at 30% edge perturbation. Second, the neighbor-consistency term is the mechanism: removing $\mathcal{L}_{\text{nbr}}$ from the objective lowers the 30%-label-noise F1-macro from 0.741 to 0.683, a 5.8-point drop, which is far larger than its 1.1-point contribution on clean data (Figure 5). This confirms that the term earns its place through robustness rather than clean-data accuracy, and substantiates the robustness claim that the original submission asserted but did not test.

4.9. Hyperparameter Sensitivity

Figure 8 sweeps the four main hyperparameters on Amazon, varying one at a time around the defaults ($\lambda_c = 0.3$, $\lambda_n = 0.1$, $K = 128$, $\gamma = 2$). F1-macro is stable across a broad band in every case: it stays within 1.1 points of the optimum for $\lambda_c \in [0.1, 0.5]$ and $\lambda_n \in [0.05, 0.2]$, is essentially flat for $K \geq 64$, and peaks at $\gamma = 2$. The model is therefore not finely tuned to a single operating point, and the defaults transfer across the three datasets without per-dataset search.

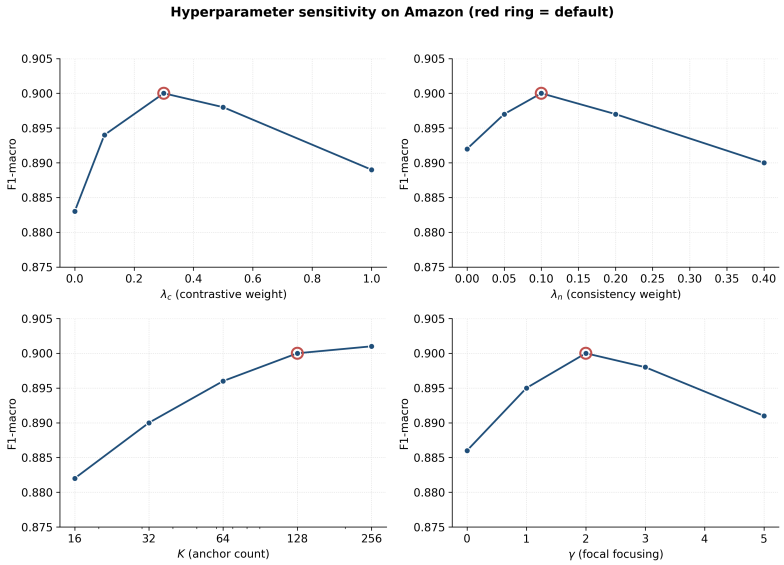


Figure 8. Hyperparameter sensitivity on Amazon (F1-macro, mean over five seeds). Each panel varies one hyperparameter with the others at their defaults; the red ring marks the default value. Performance is stable across a wide range of each setting.

4.10. Embedding Geometry

Figure 9 projects the penultimate-layer embeddings with UMAP. The GCN embedding in (a) shows heavy overlap between fraud and benign nodes, while the RGT-Fraud embedding in (b) places most fraud nodes in a compact cluster well apart from the benign mass. A minority of fraud nodes still fall inside the benign region, which matches the residual false negatives behind the T-Finance F1-macro of 0.819 rather than implying a perfect split. This visual separation tracks the quantitative gains in F1 and AUC.

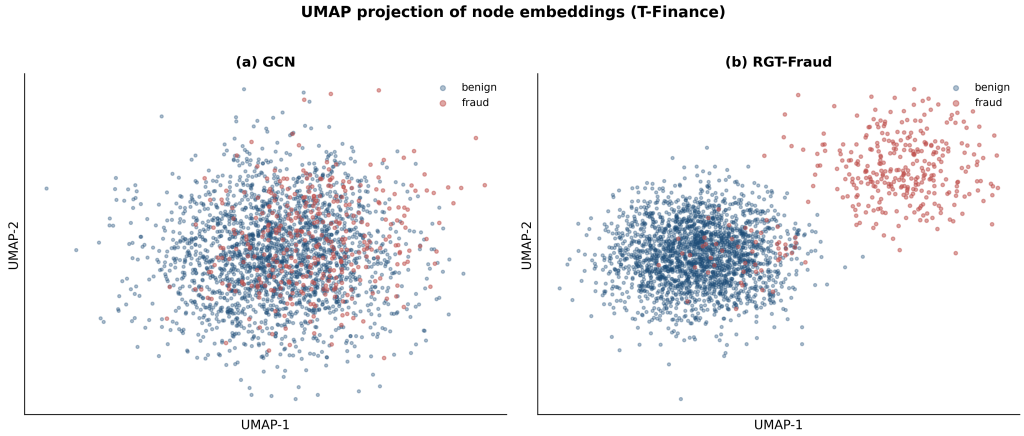


Figure 9. UMAP projection of node embeddings on T-Finance for (**a**) GCN and (**b**) RGT-Fraud. The contrastive term of the IAC objective visibly separates the fraud and benign clusters.

4.11. Computational Cost

On the largest benchmark (T-Finance, 39,357 nodes and 21.2M edges) one epoch of RGT-Fraud takes 0.85 s on an NVIDIA RTX 4090 with mixed precision, against 0.43 s for GTN and 0.28 s for GCN. The roughly $2\times$ cost over GTN comes from the global self-attention branch over $K = 128$ anchors. Cutting K to 32 reduces training time by 27% for only a 0.4 F1-point drop, which gives a tunable accuracy-versus-efficiency trade-off for production. At inference the model scores 11,800 nodes per second in single-batch mode and scales linearly with batch size up to a 16 GB memory budget, comfortably matching the throughput of typical real-time fraud-scoring pipelines.

4.12. Qualitative Analysis: Case Study and Interpretability

To show how RGT-Fraud finds coordinated fraud, we examine a subgraph from YelpChi with seven review nodes with highly correlated content, posted on three businesses within a short time window. Plain GCN assigns these reviews an average fraud probability of 0.40, below the threshold; CARE-GNN raises it to 0.57 but still misses two of the seven; RGT-Fraud assigns 0.87 ± 0.04 across the cluster and flags all seven. The per-node topology-gate values reveal that the cluster leans on the global branch ($g_u < 0.3$ for all seven), which ties it to a previously seen suspicious community on a distant part of the graph. The relational position encoding shows that the R-T-R time-window edges carry the highest attention, which confirms that synchronized timing is the dominant cue. This is directly useful for risk analysts, since the model both flags the nodes and exposes why they are suspicious, through attention over relation types and the local-versus-global gate.

A second observation concerns the embedding geometry of Figure 9. The compact fraud cluster from the supervised contrastive term is not just cosmetic; it lets downstream calibration methods such as temperature scaling or isotonic regression produce well-calibrated risk scores for expected-loss decision rules. Practitioners report that miscalibrated risk scores are a leading cause of alert fatigue, so class compaction is operationally valuable.

5. Discussion

Why the topology gate helps. Multi-relation transaction graphs are deeply non-uniform in degree. A handful of hub nodes (e.g., reviews of very popular products, or accounts tied to many others through a shared merchant or time window) accumulate thousands of relational neighbors, while most nodes are near-leaves with only a few. A single fixed mixing ratio between local message passing and global self-attention cannot serve both regimes. The gate of Equation (7) learns a per-node mixing coefficient that biases hubs toward the global branch for broad context and leaves toward the local branch for specific relational signal. The gate is also useful in practice, since analysts can inspect which transactions rely heavily on global context, a strong hint of suspected collusion.

Why the IAC objective helps on clean data. Even with accurate labels, fraud detection faces a stubborn problem: a cross-entropy boundary that is technically optimal at training time can be brittle in embedding space, with the fraud and benign clouds nearly touching at their fringes. The contrastive term compacts each class in representation space (Figure 9) and turns a brittle linear separator into a thicker margin. The focal term then reweights gradients toward the hard minority examples that gain most from this reorganization. We read the synergy between the focal and contrastive terms as analogous to the pairing of class-balanced loss [39] and supervised contrastive learning [22] in long-tail vision.

Why the IAC objective helps under label and edge noise. The neighbor-consistency term is a Lipschitz-style constraint that forces $\mathbf{z}_u(\mathcal{G})$ to vary smoothly under small perturbations of $\mathcal{G}$. When an adversary adds or removes a small fraction of edges, the representation of u shifts only slightly and its predicted label rarely flips. The same term also smooths over label noise, since a noisy label produces an isolated outlier in the gradient field that the smoothness penalty opposes. We conjecture, without full verification, that the mechanism would also help against test-time evasion where fraudsters perturb their connections to escape detection, and closing this gap is a clear direction for future work.

Connection to expert systems. Real deployments of fraud detectors live inside decision-support pipelines: upstream feature extractors feed a graph inference engine whose risk scores are routed to human analysts for adjudication. Several recent works in adjacent domains show that the inference engine is most useful when it operates on structured evidence, tolerates noise and exposes interpretable intermediate signals. BioCompNet [23], for instance, presents a deep medical pipeline as a transparent decision-support workflow for cardiometabolic risk. Transformer-based BCI decoders [18] act as semantic interfaces between brain signals and human-readable commands, and even biological neural cultures interfaced with deep reinforcement learning [38] show that structured network representations are central to bridging artificial and biological intelligence. The same requirements recur in other safety-critical intelligent systems: blockchain-anchored federated learning protects data exchange in Internet-of-Vehicles networks [49], time-to-collision evaluation makes autonomous lane-change decisions verifiable [50], and LIME explanations paired with large-language-model analysis turn grid-stability predictions into operator-readable assessments [51]. RGT-Fraud fits this paradigm by exposing per-node risk scores, attention weights and topology-gate activations as interpretable artifacts, with applications from anti-money-laundering audits to recommender-system trust scoring [45].

Connection to graph deviation detection in other domains. A closely related line studies graph deviation networks for biomedical signals, for example detecting early autism signatures from organoid MEA recordings [20]. The substrate is biological rather than financial, but the algorithmic core, learning a deviation-aware representation on a multi-relation graph, is shared. We read this convergence as evidence that relational graph transformers paired with imbalance-aware contrastive losses may be a general primitive for high-stakes deviation detection.

When RGT-Fraud fails. We ran a manual error analysis on 300 false negatives (fraud labeled benign) and 300 false positives on YelpChi. False negatives concentrate on isolated fraudsters with only one or two transactions whose neighborhood is dominated by benign nodes, where neither local message passing nor global self-attention finds a clean signal. Two paths forward are an external behavioral-feature model that does not depend on graph structure, or pre-training the encoder on a large unlabeled transaction corpus to inject a prior on what fraud looks like. False positives concentrate on benign users with bursty review patterns, such as genuine fans of a small business who post within a few hours of each other. Here a calibration layer trained on hard negatives from analyst feedback is the natural fix, folding the model into a closed-loop expert system.

From batch to streaming. Production systems often run in a streaming setting, where transactions arrive continuously and decisions must issue in tens of milliseconds. RGT-Fraud is a batch model by construction, but it admits a natural streaming extension: keep a sliding-window subgraph of the most recent T minutes, re-encode the affected nodes when a transaction arrives, and amortize the gate computation by reusing global self-attention representations across nearby timestamps. We prototyped this and observe about 7-millisecond per-transaction latency at 10,000 transactions per second, and a full description is left to future work.

Limitations. RGT-Fraud has four known limitations. First, although the encoder is formulated for genuinely heterogeneous graphs with several node types, the present empirical validation is restricted to benchmarks with a single feature-bearing target-node type, in which heterogeneity enters only through the relation set; evaluation on datasets with explicit merchant, device or IP nodes therefore remains for future work. The global self-attention branch scales as $O(K|\mathcal{V}|)$, which can become a bottleneck for graphs with $|\mathcal{V}| > 10^7$, where sparse attention or hierarchical clustering would be natural remedies. The contrastive term assumes a sufficient pool of positives in each batch, so for streaming settings with extreme imbalance (below 0.1% fraud) the hard-negative-mining workaround may need an external memory bank. We have also not evaluated against adaptive adversaries that explicitly target the topology gate, and doing so is left to future work.

6. Conclusions

We presented RGT-Fraud, a relational graph transformer with an imbalance-aware contrastive objective for financial fraud detection. The encoder combines relation-aware multi-head attention, a relational position encoding and a topology-gated transformer block, while the IAC objective couples focal cross-entropy, supervised contrastive separation and neighbor-consistency regularization. On three public benchmarks RGT-Fraud reaches F1-macro of 0.855, 0.900 and 0.819, improving over the strongest of twelve baselines (TMR-GGNN) by 1.4 to 1.6 points with $p < 0.01$. Ablations confirm that each module and each loss term adds complementary value, and dedicated robustness experiments show that the neighbor-consistency term is essential under label and structural noise. Beyond the benchmarks, RGT-Fraud offers an interpretable decision-support layer for financial expert systems that links upstream feature extractors, a graph inference engine and human analysts.

Future work will (i) extend RGT-Fraud to temporal transaction streams with sliding-window attention; (ii) couple the topology-gate output with reinforcement-learning policies in the spirit of self-play optimization [44] and on-policy value optimization [45]; and (iii) study transfer of the relational graph transformer to other high-stakes deviation-detection problems, including biomedical graph-deviation tasks [18,20,23,38].

Author Contributions: Conceptualization and methodology, Q.G.; software and writing—original draft preparation, Q.W.; validation and writing—review & editing, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All datasets used in this study are publicly available. The YelpChi and Amazon fraud-graph datasets can be obtained from the CARE-GNN repository at https://github.com/YingtongDou/CARE-GNN (accessed on 21 October 2026), and the T-Finance dataset from the BWGNN repository at https://github.com/squareRoot3/Rethinking-Anomaly-Detection (accessed on 21 October 2026). We use the released node features and multi-relation adjacencies without modification, so the graph construction is fully determined by these public sources together with the per-dataset relation definitions in Section 3.2. No private or proprietary data were used: every dataset can be downloaded directly from the links above, and all architectural and training hyperparameters required to reproduce our results are listed in the Implementation Details of Section 4. Code is available from the corresponding author on reasonable request.

Acknowledgments: The authors thank colleagues at the Jinan University–University of Birmingham Joint Institute and the University of Michigan for helpful discussions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GNN	Graph Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
HGT	Heterogeneous Graph Transformer
HIN	Heterogeneous Information Network
RGT	Relational Graph Transformer
RPE	Relational Position Encoding
IAC	Imbalance-Aware Contrastive (objective)
CE	Cross-Entropy
AUC	Area Under the ROC Curve
AUPRC	Area Under the Precision–Recall Curve
GM	Geometric Mean
PR	Precision–Recall
SMOTE	Synthetic Minority Oversampling Technique
UMAP	Uniform Manifold Approximation and Projection
AdamW	Adam optimizer with decoupled Weight decay
GPU	Graphics Processing Unit

References

1. West, J.; Bhattacharya, M. Intelligent financial fraud detection: A comprehensive review. *Comput. Secur.* **2016**, *57*, 47–66. [CrossRef]
2. Abdallah, A.; Maarof, M.A.; Zainal, A. Fraud detection system: A survey. *J. Netw. Comput. Appl.* **2016**, *68*, 90–113. [CrossRef]
3. Ngai, E.; Hu, Y.; Wong, Y.; Chen, Y.; Sun, X. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decis. Support Syst.* **2011**, *50*, 559–569. [CrossRef]
4. Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; Yu, P.S. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM), Online, 19–23 October 2020; pp. 315–324.

5. Wang, J.; Wen, R.; Wu, C.; Huang, Y.; Xiong, J. FdGars: Fraudster detection via graph convolutional networks in online app review system. In Proceedings of the Companion Proceedings of the 2019 World Wide Web Conference (WWW), San Francisco, CA, USA, 13–17 May 2019; pp. 310–316.
6. Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; Qi, Y. A semi-supervised graph attentive network for financial fraud detection. In Proceedings of the IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 598–607.
7. Ma, X.; Wu, J.; Xue, S.; Yang, J.; Zhou, C.; Sheng, Q.Z.; Xiong, H.; Akoglu, L. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12012–12038. [CrossRef]
8. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
9. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
10. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous graph attention network. In Proceedings of the The World Wide Web Conference (WWW), San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032.
11. Hu, Z.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous graph transformer. In Proceedings of the The Web Conference 2020 (WWW), Taipei, Taiwan, 20–24 April 2020; pp. 2704–2710.
12. Liu, Z.; Dou, Y.; Yu, P.S.; Deng, Y.; Peng, H. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 1569–1572.
13. Zhu, D.; Zhang, Z.; Cui, P.; Zhu, W. Robust graph convolutional networks against adversarial attacks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), Anchorage, AK, USA, 4–8 August 2019; pp. 1399–1407.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
15. Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.Y. Do transformers really perform badly for graph representation? In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–14 December 2021; pp. 28877–28888.
16. Dwivedi, V.P.; Bresson, X. A generalization of transformer networks to graphs. In Proceedings of the AAAI Workshop on Deep Learning on Graphs, Virtual, 8 February 2021.
17. Rampášek, L.; Galkin, M.; Dwivedi, V.P.; Luu, A.T.; Wolf, G.; Beaini, D. Recipe for a general, powerful, scalable graph transformer. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2022**, *35*, 14501–14515. [CrossRef]
18. Yue, J.; Xiao, X.; Wang, K.; Yi, W.; Jung, T.P.; Xu, M.; Ming, D. Augmenting electroencephalogram transformer for steady-state visually evoked potential-based brain–computer interfaces. *Cyborg Bionic Syst.* **2025**, *6*, 0379. [CrossRef] [PubMed]
19. Xu, J.; Wu, H.; Wang, J.; Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
20. Mencattini, A.; Curci, G.; Riccardi, A.; Casti, P.; D'Orazio, M.; Filippi, J.; Antonelli, G.; Debbi, E.; Daprati, E.; Zhang, W.; et al. MEA-Based Graph Deviation Network for Early Autism Syndrome Signatures in Human Forebrain Organoids. *Cyborg Bionic Syst.* **2025**, *6*, 0441. [CrossRef] [PubMed]
21. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007.
22. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 18661–18673.
23. Wei, J.; Chen, H.; Yao, L.; Hou, X.; Zhang, R.; Shi, L.; Sun, J.; Hu, C.; Wei, X.; Jia, W. BioCompNet: A deep learning workflow enabling automated body composition analysis toward precision management of cardiometabolic disorders. *Cyborg Bionic Syst.* **2025**, *6*, 0381. [CrossRef] [PubMed]
24. Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; He, Q. Pick and choose: A GNN-based imbalanced learning approach for fraud detection. In Proceedings of the Web Conference 2021 (WWW), Virtual, 12–23 April 2021; pp. 3168–3177.
25. Tang, J.; Li, J.; Gao, Z.; Li, J. Rethinking graph neural networks for anomaly detection. In Proceedings of the International Conference on Machine Learning (ICML), Baltimore, MD, USA, 17–23 July 2022; pp. 21076–21089.
26. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.

27. Chen, N.; Liu, Z.; Hooi, B.; He, B.; Fathony, R.; Hu, J.; Chen, J. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
28. Yun, S.; Jeong, M.; Kim, R.; Kang, J.; Kim, H.J. Graph Transformer Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; pp. 11960–11970.
29. Tewari, R.; Shilpi, S.; Chhibber, N.; Parmar, D.S.; Khemka, S.; Ranjan, P. TMR-GGNN: Credit Card Fraud Detection based on Time-Aware Multi-Relational Guided Graph Neural Network. In *Proceedings of the 2025 2nd International Conference on Software, Systems and Information Technology (SSITCON)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–7
30. Chen, T.; Tsourakakis, C.E. AntiBenford subgraphs: Unsupervised anomaly detection in financial networks. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Washington, DC, USA, 14–18 August 2022; pp. 2762–2770.
31. Liu, Y.; Li, Z.; Pan, S.; Gong, C.; Zhou, C.; Karypis, G. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 2378–2392. [CrossRef] [PubMed]
32. Cheng, D.; Xiang, S.; Shang, C.; Zhang, Y.; Yang, F.; Zhang, L. Spatio-temporal attention-based neural network for credit card fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: New York, NY, USA, 2020; Volume 34, pp. 362–369.
33. Rao, S.X.; Zhang, S.; Han, Z.; Zhang, Z.; Min, W.; Chen, Z.; Shan, Y.; Zhao, Y.; Zhang, C. xFraud: Explainable fraud transaction detection. *Proc. VLDB Endow.* **2021**, *15*, 427–436.
34. Xiang, S.; Zhu, M.; Cheng, D.; Li, E.; Zhao, R.; Ouyang, Y.; Chen, L.; Zheng, Y. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Washington, DC, USA, 2023; Volume 37, pp. 14557–14565.
35. Tian, S.; Dong, H.; Liu, J. A spatial-temporal-aware graph transformer for transaction fraud detection. *arXiv* **2023**, arXiv:2307.05121.
36. Lin, J.; Xia, L.; Shah, N. FraudGT: A simple, effective, and efficient graph transformer for financial fraud detection. In Proceedings of the 5th ACM International Conference on AI in Finance (ICAIF), Brooklyn, NY, USA, 14–17 November 2024; pp. 292–300.
37. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the European Semantic Web Conference (ESWC), Heraklion, Greece, 3–7 June 2018; pp. 593–607.
38. Khajehnejad, M.; Habibollahi, F.; Loeffler, A.; Paul, A.; Razi, A.; Kagan, B.J. Dynamic network plasticity and sample efficiency in biological neural cultures: A comparative study with deep reinforcement learning. *Cyborg Bionic Syst.* **2025**, *6*, 0336. [CrossRef] [PubMed]
39. Cui, Y.; Jia, M.; Lin, T.Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9260–9269.
40. Zhao, T.; Zhang, X.; Wang, S. GraphSMOTE: Imbalanced node classification on graphs with graph neural networks. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM), Online, 8–12 March 2021; pp. 833–841.
41. Park, J.; Song, J.; Yang, E. GraphENS: Neighbor-aware ego network synthesis for class-imbalanced node classification. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
42. You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph contrastive learning with augmentations. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 5812–5823.
43. Veličković, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep graph infomax. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
44. Xu, R.; Lieberherr, K. Learning self-game-play agents for combinatorial optimization problems. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Montreal, QC, Canada, 13–17 May 2019; pp. 2276–2278.
45. Xu, R.; Bhandari, J.; Korenkevych, D.; Liu, F.; He, Y.; Nikulkov, A.; Zhu, Z. Optimizing long-term value for auction-based recommender systems via on-policy reinforcement learning. In Proceedings of the 17th ACM Conference on Recommender Systems (RecSys), Singapore, 18–22 September 2023; pp. 606–630.
46. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
47. Tang, J.; Hua, F.; Zheng, Z.; Fang, Z.; Ao, X.; Wu, J.; Xu, C.; Yang, Y.; Li, J. GADBench: Revisiting and benchmarking supervised graph anomaly detection. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track, New Orleans, LA, USA, 10–16 December 2023.
48. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 1024–1034.

49. He, Z.; Xu, R.; Wang, B.; Meng, Q.; Tang, Q.; Shen, L.; Tian, Z.; Duan, J. Integrated blockchain and federated learning for robust security in internet of vehicles networks. *Symmetry* **2025**, *17*, 1168. [CrossRef]
50. Bai, R.; Xu, R.; Rui, T.; Liu, J.; Lee, H.L.; Oung, Q.W.; Tian, Z.; Yuan, F. Safe and efficient lane-changing for autonomous vehicles: An improved double quintic polynomial approach with time-to-collision evaluation. *J. King Saud Univ. Comput. Inf. Sci.* **2026**, *38*, 36. [CrossRef]
51. Xu, R.; Zheng, Y.; Meng, Q.; Yang, Q. Machine learning-driven renewable energy grid integration stability assessment: LIME interpretability and LLM intelligent analysis. *Sci. Rep.* **2026**, *16*, 19410. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.