Fusion-Rewired One Class Graph Autoencoder

1st Marcos Gôlo
University of São Paulo
São Carlos, São Paulo, Brazil
marcosgolo@usp.br

2nd João Gama
University of Porto
Porto, Portugal
jgama@fep.up.pt

3rd Edgard Marx
Byondis
Nijmegen, Netherlands
digamarx@gmail.com

4th Ricardo Marcacini
University of São Paulo
São Carlos, São Paulo, Brazil
ricardo.marcacini@icmc.usp.br

***Abstract*—Heterogeneous graphs provide a representation for real-world systems that involve multiple entity types and relations. In many of these applications, the practical goal is to detect one class of interest while treating the remaining instances as outliers, which naturally leads to one-class learning (OCL) on graph neural networks (GNNs). However, most approaches either rely on unsupervised methods, decouple representation learning from the OCL objective in two-step pipelines, or focus on homogeneous graphs and do not explicitly leverage heterogeneity in generic settings. We propose FOLGA (Fusion-rewired One-cLass Graph Autoencoder), an end-to-end method for OCL node classification on heterogeneous graphs. FOLGA enriches heterogeneous structure through complementary rewiring strategies, fuses information from multiple rewired graph views via a fusion GNN encoder, and combines graph reconstruction with a state-of-the-art sphere loss to learn interpretable 3D embeddings. We also show that FOLGA naturally extends to dynamic data streams for evolving dynamic heterogeneous graphs in online learning. Experiments with 12 baselines in three datasets show that FOLGA outperforms them while providing real-time interpretability of learning dynamics and competitive performance.**

***Index Terms*—One-Class Classification, Graph Rewiring**

I. Introduction

Heterogeneous graphs have become a natural representation for complex real-world systems where multiple entity types interact through diverse relations [1], [2]. By modeling nodes and edges with explicit types, heterogeneous graphs preserve semantic structure that is often lost in homogeneous abstractions, enabling learning algorithms to exploit richer dependencies across entities and relations [1]. This expressiveness has made heterogeneous graphs central to a wide range of applications, including protein interaction networks (e.g., proteins, compounds, and functional relations), social networks (users, content, and interactions), academic networks (authors, venues, and publications), knowledge graphs (entities, predicates, and facts), and event detection and monitoring (events linked to who/what/when/where/why/how components) [3].

In many of these scenarios, the practical objective is not to recognize multiple classes, but to detect a specific class defined by the user, e.g., a target event category, a particular behavior, or a critical biological interaction, while treating the remaining nodes as non-interest (outliers) [4]. This setting naturally leads to one-class learning (OCL), where only instances of interest are reliably labeled to train the algorithm, and the remaining new data are unlabeled and may contain both examples of interest and outliers [5]. In heterogeneous graphs, this problem becomes one-class node classification in a typed relational

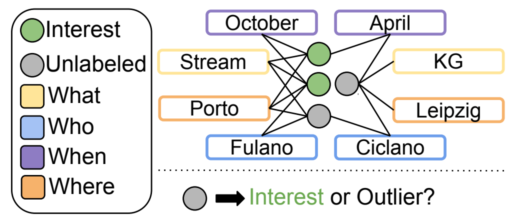


Fig. 1. OCL on heterogeneous graphs. Primary nodes are circles and secondary nodes are squares. Only a subset of main nodes has one label (green), while the remaining are unlabeled (gray). The task is to classify unlabeled nodes as interest or outliers using only labels of the interest class.

context: the model must learn from the labeled interest nodes of a primary type while leveraging secondary node types and cross-type relations to separate interest nodes from outliers [6]. Figure 1 shows the OCL in heterogeneous graphs task. Recent work has increasingly addressed this challenge with Graph Neural Networks (GNNs), which can propagate information through the graph structure to learn robust representations [7].

Existing one-class GNN approaches can be broadly grouped into three families. First, unsupervised anomaly detection methods learn representations via reconstruction objectives and then rely on scores and thresholds to flag anomalies [8], [9]. While effective for generic anomaly detection, they often struggle when the goal is explicit interest-versus-outlier classification without supervised calibration. Second, two-step pipelines learn graph embeddings in an unsupervised way and then apply a one-class algorithm [10]–[12]. These methods explore embeddings that are not explicitly shaped by the one-class objective, which can limit separability in challenging settings. Third, end-to-end one-class GNNs jointly optimize representation learning and one-class discrimination, typically combining hypersphere-based objectives with graph reconstruction to preserve topology [6], [13]–[18]. Although studies such as [17] explore the heterogeneity of graphs, most of these methods were developed for homogeneous graphs [13]–[16], [18] or hypergraphs [6], do not creating and exploring additional cross-type structure tailored to the one-class task.

To address these gaps, we propose FOLGA: Fusion-rewired One-cLass Graph Autoencoder, a new end-to-end method for one-class node classification. FOLGA builds on the strong foundation of graph reconstruction and sphere-based one-class objectives [18], but crucially introduces rewiring mechanisms designed to exploit heterogeneity by creating informative relations for the interest class and strengthening message passing among both primary and secondary node types [19]–[21]. The resulting rewired graphs are processed by a dedicated GNN layer that fuses multi-graph representations, and the model learns 3D embeddings to provide direct, real-time interpretability of the training dynamics and the learned decision boundary [22]. The core components of FOLGA do not require assumptions related to a fixed graph size. Thus, FOLGA can be run or updated repeatedly as the graph evolves, offering a robust and flexible solution for real-world deployments. In this sense, FOLGA is also naturally suited to both offline and online scenarios, i.e., it can be trained on a static heterogeneous graph and applied to data streams via dynamic graphs, where new primary and secondary nodes arrive continuously [23], [24]. Our contributions are:

1) We introduce a new OCL method for heterogeneous graphs that leverages GNNs and graph rewiring.
2) FOLGA provides interpretable 3D representations and offers a practical path toward online learning in dynamic, streaming settings, such as event detection, where new events arrive continuously.
3) We collect three new event datasets and enrich them with 5W1H components to construct naturally one-class heterogeneous graphs centered on events of interest.
4) We conducted a comprehensive evaluation against 12 representative baselines, yielding a robust experimental benchmark and an implementation framework for comparing one-class methods across heterogeneous graphs.

II. Preliminaries and Background

We consider a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node-type mapping $\phi : \mathcal{V} \rightarrow \mathcal{T}_\mathcal{V}$, node features $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$, and edges $\mathcal{E}$. Let $\tau \in \mathcal{T}_\mathcal{V}$ denote the *primary* node type to be classified, with $\mathcal{V}_\tau = \{v \in \mathcal{V} \mid \phi(v) = \tau\}$, while the remaining types form the *secondary* set $\mathcal{V}_{\text{sec}} = \mathcal{V} \setminus \mathcal{V}_\tau$ that provides heterogeneity. In the one-class learning, only a subset of primary nodes is labeled as interest, $\mathcal{V}_I \subset \mathcal{V}_\tau$, whereas the remaining primary nodes are unlabeled, $\mathcal{V}_U = \mathcal{V}_\tau \setminus \mathcal{V}_I$. Thus, $\mathcal{V}_U$ contains both interest and outlier nodes to be classified. The goal is to learn a classifier function using only $\mathcal{V}_I$ together with the heterogeneous structure induced by $\mathcal{V}_{\text{sec}}$ and $\mathcal{E}$, so that nodes in $\mathcal{V}_U$ can be classified as interest or outlier nodes. In practice, this is a common scenario for graph neural networks (GNNs) [7], often instantiated as graph autoencoders (GAEs) [25] trained with reconstruction objectives and hypersphere-based one-class losses [16], [17].

A GNN encoder iteratively aggregates neighborhood information to produce node embeddings [7]:

$$\boldsymbol{H}^{(t+1)} = gnn(\boldsymbol{H}^{(t)}, \boldsymbol{A}; \boldsymbol{W}^{(t)}), \tag{1}$$

where $\boldsymbol{A}$ are the adjacency matriz representing the relations $\mathcal{E}$, $\boldsymbol{H}$ are the learned embeddings for each GNN layer t, $\boldsymbol{H}^{(0)} = \boldsymbol{X}$, and $\boldsymbol{W}$ are learnable weights of the neural network.

A Graph Convolutional Network (GCN), considering N_{v_i} as the v_i neighborhood and the activation function σ [26]:

$$\boldsymbol{h}^t_{v_i} = \sigma(\boldsymbol{W}^{(t)} \cdot MEAN\{\boldsymbol{h}^{t-1}_{v_j} : v_j \in \{N_{v_i} \cup v_i\}\}). \tag{2}$$

A Graph Autoencoder (GAE) uses an encoder to compute embeddings $\boldsymbol{Z} \in \mathbb{R}^{|\mathcal{V}| \times p}$ [25]:

$$\boldsymbol{Z} = gnn(\boldsymbol{X}, \boldsymbol{A}; \boldsymbol{W}), \tag{3}$$

and reconstructs the adjacency matrix with an inner-product:

$$\hat{\boldsymbol{A}} = \sigma\big(\boldsymbol{Z}\boldsymbol{Z}^\top\big). \tag{4}$$

The common reconstruction objective for GAEs is the binary cross-entropy between $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ [25]:

$$\mathcal{L}_{\text{rec}}(\boldsymbol{W}) = -\sum_{i=1}^{|\mathcal{V}|}\sum_{j=1}^{|\mathcal{V}|}\Big(A_{ij}\log\hat{A}_{ij} + (1-A_{ij})\log(1-\hat{A}_{ij})\Big), \tag{5}$$

After learning embeddings with this loss, a common first strategy is to treat these reconstruction errors as scores to detect anomalies and, consequently, identify interesting classes [8], [9]. Another strategy explores the learned embeddings $\boldsymbol{z}_{v_i}$ for $v_i \in \mathcal{V}_\tau$ that capture both features and structure with another step evolving a OCL algorithm, such as One-Class Support Vector Machine (OCSVM), which fits a boundary around the interest embeddings [10]–[12]. End-to-end methods, to explicitly explore the OCL compactness, adopt a DeepSVDD-style hypersphere loss on interest nodes $v \in \mathcal{V}_I$, defining a center $\boldsymbol{c} \in \mathbb{R}^p$, learning a radius $R \geq 0$, and using $\nu \in [0, 1)$ as an upper bound on the fraction of training errors [13]:

$$\mathcal{L}_{\text{svdd}}(\boldsymbol{W}, R) = \frac{1}{|\mathcal{V}_I|}\sum_{v_i \in \mathcal{V}_I}\max\big(0,\ \|\boldsymbol{z}_{v_i} - \boldsymbol{c}\|_2^2 - R^2\big). \tag{6}$$

Most end-to-end one-class GNN approaches optimize only $\mathcal{L}_{\text{svdd}}$, a combination of $\mathcal{L}_{\text{svdd}}$ and $\mathcal{L}_{\text{rec}}$, or add auxiliary terms such as feature reconstruction, contrastive objectives, adversarial regularization, or node-type prediction losses [14]–[17]. This leads to a weighted multi-term objective of the form:

$$\mathcal{L} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{svdd}}\mathcal{L}_{\text{svdd}} + \sum_k \lambda_k \mathcal{L}_k, \tag{7}$$

where $\lambda_\cdot$ controls the relative contribution of each component, and $\mathcal{L}_k$ can be any additional learning loss function.

III. Related Work

Unsupervised GNNs. Unsupervised one-class GNNs typically learn node representations by reconstructing the graph (and/or contrasting multiple structural views) through a GAE and then use a reconstruction-based score as an anomaly indicator, classifying interest nodes by thresholding this score. For instance, SAMCL [8] proposes a subgraph-aligned multiview contrastive framework that samples 1–2-hop ego-subgraphs and optimizes intra- and inter-view alignment, complemented by a reconstruction objective to preserve topology. FIAD [9] instead perturbs latent representations through feature injection and trains structural and attribute decoders to reconstruct both topology and features while regularizing the injected and original embeddings. Although effective for generic anomaly detection, these approaches commonly require unsupervised threshold selection, often rely on artificially injected anomalies during evaluation, and are not explicitly designed to handle with OCL.

Two-Step One-Class GNNs. Two-step one-class GNNs decouple representation learning from one-class decision making: they first learn embeddings in a fully unsupervised manner (e.g., via adjacency reconstruction or contrastive objectives) and then apply a classical one-class algorithm in the learned embeddings. GCN+OCSVM [10] follows this paradigm for heterogeneous graphs by training a GAE with a GCN-based encoder to reconstruct the adjacency matrix and subsequently fitting an OCSVM on the learned embeddings. EAGLE [11] adopts a heterogeneous setting in a domain-specific problem (tax evasion) and learns embeddings with a contrastive objective guided by graph walks before applying a one-class classifier. GAT+OCSVM [12] extends [10] by exploring alternative GNN layers (e.g., GAT) and systematically analyzing early-fusion operators, showing that explicit fusion may not yield statistically significant gains over simpler choices. Despite their practicality, these pipelines learn representations that are agnostic to the downstream one-class objective, which can limit optimality because the embedding space is not customized for the one-class decision boundary.

End2End One-Class GNNs. End-to-end one-class GNNs jointly optimize representation learning and one-class learning, typically combining an unsupervised objective over all nodes (e.g., reconstruction) with a one-class loss applied to the interest data (e.g., SVDD-style hyperspheres [27]). OCGNN [13] introduces a DeepSVDD-inspired hypersphere loss to compact normal nodes in attributed graphs, while DualSVDAE [14] and STEAM [15] couple graph and feature autoencoders, combining hypersphere compactness with topology and attribute reconstructions. JAANE [16] further integrates adversarial regularization by fusing structural and attribute embeddings and training a discriminator alongside SVDD and reconstruction objectives. MULDUALGNN [17] augments the global structure, focusing on heterogeneity, with a learned local view and employs both global and cluster-specific hypersphere objectives to capture multi-modal normality. More recently, OLGA [18] emphasizes interpretability by learning low-dimensional (3D) embeddings within an end-to-end GAE framework, using a hypersphere objective that keeps in-class samples near the center even when they lie within the radius, with a focus on homogeneous graphs. ECHOLGA [6] extends OLGA to heterogeneous hypergraphs by enriching nodes/edges and adding a node-type cross-entropy loss, preserving the interpretability goal while shifting to a different prediction target (edge classification).

Limitations and Challenges. Overall, the recent state-of-the-art one-class GNN literature still exposes several practical gaps and challenges when the target problem is one-class node classification. First, a large fraction of evaluations relies on proxy one-class settings obtained by collapsing multi-class benchmarks or by injecting synthetic anomalies, which may not capture the distributional and structural challenges of OCL. Second, while heterogeneous graphs are inherent in practice, effectively exploiting node-type interactions in one-class learning remains a key challenge, since OCL is typically defined at the level of a single primary node category, whereas heterogeneity arises as a secondary relational structure that must be integrated without ignoring the distinction between inliers and outliers. Furthermore, the end-to-end work that focuses on heterogeneity uses the deepSVDD loss, which naturally stops pulling nodes of interest toward the center of the hypersphere once they are already inside it. This can hinder the entry of new nodes of interest. With the exception of OLGA and ECHOLGA, all end-to-end methods use the deepSVDD loss, which can degrade performance.

IV. FOLGA: Fusion-rewired One cLass Graph Autoencoder

Heterogeneous graphs naturally encode multiple, complementary signals through typed interactions, and ignoring such heterogeneity often wastes the very structure that motivated the heterogeneous modeling in the first place [2], [3]. In OCL node classification, this is particularly important since the decision is defined over a *primary* node type (e.g., events), yet the evidence that separates interest from outliers is frequently expressed via *secondary* node types and cross-type dependencies (e.g., who/where/when/what) [10], [12]. Therefore, an effective OCL method should not only model interactions between primary and secondary nodes, but also exploit relations *among* secondary nodes whenever they strengthen information flow [21]. At the same time, interpretability is desirable for practical real-world scenarios, as it enables direct inspection of the model's evolving notion of the interest class [18], [22]. This requirement becomes even more relevant in online (data-stream) settings, where the graph evolves over time, and the model must be updated incrementally as new nodes and edges arrive [23], [24]. Interpretability in such scenarios helps practitioners monitor distribution shifts and detect when the learned boundary may need to be adapted [23], [24].

Motivated by these points, we propose **FOLGA**, which explicitly leverages heterogeneity through graph rewiring, learns node representations with a dedicated fusion-rewired GNN encoder, and performs OCL via the sphere objective combined with GAE reconstruction to preserve structural constraints. The workflow of FOLGA follows: (i) generates additional graphs from the original heterogeneous graph via two complementary rewiring strategies; (ii) encodes nodes by aggregating information across all rewired graphs, producing embeddings suitable for both prediction and interpretation; and (iii) optimizes a sphere loss together with a GAE loss, yielding competitive performance while maintaining interpretability through low-dimensional. Figure 2 illustrates our proposal.

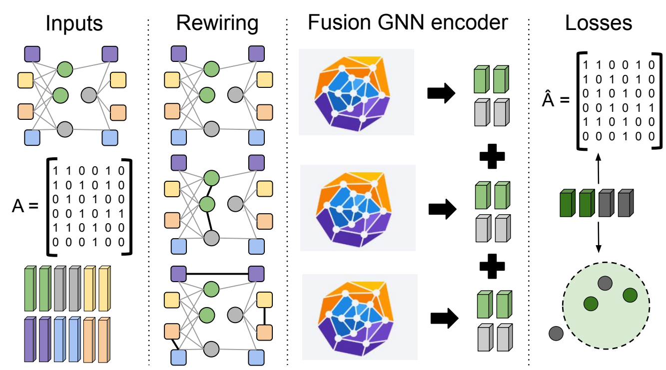


Fig. 2. FOLGA first stage illustrates the graph rewiring process, in which multiple graphs are generated by preserving the same set of nodes while adding new edges. In the second stage, a GNN learns node representations from all rewired graphs and fuses them into a single unified embedding space. The final stage presents the sphere-based loss and a reconstruction loss.

We first introduce two rewiring mechanisms that create additional relations to better exploit heterogeneity [19], [21]. The first rewiring is a probabilistic, distance-based mechanism that connects primary nodes using shortest-path distances computed on the full heterogeneous graph [20]. For a primary node $v_i \in \mathcal{V}_\tau$, let $d_\mathcal{G}(v_i, v_j)$ be the BFS distance from v_i to another primary node $v_j \in \mathcal{V}_\tau$. At the rewiring step $\ell \in \{1, \ldots, L-1\}$, we sample a Bernoulli variable for each pair (v_i, v_j) with probability [20]:

$$p_\ell(v_i, v_j) = \frac{1}{d_\mathcal{G}(v_i, v_j)^\alpha \, |\mathcal{V}_\tau|^\beta \, \ell^\gamma}, \tag{8}$$

where α, β, and γ are hyperparameters to control the rewiring, and add an edge (v_i, v_j) when $p(v_i, v_j)$ is greater than or equal to a random value. The key point is that $d_\mathcal{G}(v_i, v_j)$ is computed through paths that may traverse *secondary* nodes. Thus, secondary structure directly influences which primary-to-primary relations are formed, thereby introducing heterogeneous evidence into the rewired topology.

The second rewiring is similarity-based and explicitly creates additional relations *within* node-type partitions. Given $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and a similarity metric (e.g., cosine), we build two kNN graphs: one among primary nodes $\mathcal{V}_\tau$ and another among secondary nodes $\mathcal{V}_{\text{sec}}$.

Denoting the embedding of node v_i by $\boldsymbol{x}_i$, we connect each node to its k nearest neighbors under $\mathrm{sim}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, producing additional edges $\mathcal{E}_\tau^{\text{knn}}$ and $\mathcal{E}_{\text{sec}}^{\text{knn}}$. This yields a collection of three graphs:

$$\{\mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \mathcal{G}^{(2)}\} = \{\mathcal{G},\ (\mathcal{V}, \mathcal{E} \cup \mathcal{E}_\tau^{\text{knn}}),\ (\mathcal{V}, \mathcal{E} \cup \mathcal{E}_{\text{sec}}^{\text{knn}})\}, \quad (9)$$

which enriches message passing with complementary similarity relations. Importantly, the secondary-node rewiring enables interactions that are not explicit in the original schema. For example, if an event node connects to `Where:UnitedStates` and `Who:USPresident`, then adding similarity-driven links among secondary nodes can create a direct `Where`$\leftrightarrow$`Who` bridge (when supported by feature similarity), strengthening information flow and helping the GNN propagate across heterogeneous components.

Given the rewired graphs, we introduce a fusion-rewired GNN encoder that explores all rewired graphs and fuses their representations into a unified, enriched embedding, thereby generating a new architecture tailored to our setting [20]. Let $\{\mathcal{G}^{(\ell)}\}_{\ell=0}^{L-1}$ be the list of graphs produced by the rewiring procedure, with adjacency matrices $\{\boldsymbol{A}^{(\ell)}\}_{\ell=0}^{L-1}$. In FOLGA, each layer aggregates messages *across all rewired graphs* by adding the outputs of L GNN sublayers [20]:

$$\boldsymbol{H}^{(t+1)} = folga\left(\sum_{\ell=0}^{L-1} gnn\Big(\boldsymbol{H}^{(t)}, \boldsymbol{A}^{(\ell)}; \boldsymbol{W}^{(t,\ell)}\Big)\right), \quad (10)$$

where $gnn(\cdot)$ can be the tradicional GCN as defined in Equation 2.

FOLGA is instantiated as a GAE whose encoder receives node features $\boldsymbol{X}$ and the list of rewired adjacencies $\{\boldsymbol{A}^{(\ell)}\}_{\ell=0}^{L-1}$:

$$\boldsymbol{Z} = folga\left(\boldsymbol{X}, \{\boldsymbol{A}^{(\ell)}\}_{\ell=0}^{L-1}\right), \quad (11)$$

and whose decoder reconstructs the original graph structure via an inner product, as in Equation 4. We optimize the standard GAE reconstruction loss as in Equation 5.

In addition to the reconstruction loss, we explore a one-class sphere loss, following the OLGA premise that interest embeddings should be compact while avoiding representation collapse [18]. Let $\mathcal{V}_I$ be the set of labeled interest nodes (of the primary type), and let $\boldsymbol{z}_{v_i}$ be the embedding of node $v_i \in \mathcal{V}_I$. We define:

$$o_i = \|\boldsymbol{z}_{v_i} - \boldsymbol{c}\|_2^2 - R^2, \quad (12)$$

where $\boldsymbol{c}$ and R are the center and radius. The one-class loss is [18]:

$$\mathcal{L}_{\mathrm{ocl}}(\boldsymbol{W}) = \frac{1}{|\mathcal{V}_I|} \sum_{i=1}^{|\mathcal{V}_I|} \begin{cases} o_i + 1, & \text{if } o_i > 0, \\ \exp(o_i), & \text{otherwise.} \end{cases} \quad (13)$$

Using only $\mathcal{L}_{\mathrm{ocl}}$ would encourage all embeddings to collapse toward $\boldsymbol{c}$. The reconstruction term $\mathcal{L}_{\mathrm{rec}}$ acts as a structural constraint that preserves informative differences required to explain the observed topology, thereby preventing degenerate solutions and yielding embeddings that are simultaneously compact for interest nodes and meaningful for graph modeling [14]–[16], [18].

Finally, we propose interpretability by setting the latent dimensionality to $p = 3$, i.e., $\boldsymbol{z}_v \in \mathbb{R}^3$. This enables direct visualization of node trajectories and the learned sphere boundary over training (and across time in streaming scenarios), supporting real-time inspection of how interest nodes concentrate and how outliers separate. The overall training is the sum of the one-class and reconstruction terms:

$$\mathcal{L} = \mathcal{L}_{\mathrm{ocl}} + \mathcal{L}_{\mathrm{rec}}. \quad (14)$$

At inference time, nodes can be ranked by the hypersphere score (e.g., o_i from Eq. 12) and classified accordingly, while the 3D embeddings provide an immediate geometric explanation of the decision.

Building on this geometric view of the learned space, FOLGA also naturally extends to online learning in heterogeneous graph data streams. In such settings, the input is no longer a single static $\mathcal{G}$, but a sequence of graph snapshots $\{\mathcal{G}_m\}_{m=1}^{M}$, where new primary nodes (and their associated secondary nodes and edges) arrive over time. We explore a sliding-window strategy: at each time step m, we construct a windowed graph $\mathcal{G}_m$ that retains the most recent primary nodes and the secondary nodes that remain connected to them, while discarding older primary nodes and any secondary nodes that become isolated once those outdated primaries are removed. This pruning is consistent with event-driven heterogeneous graphs, where secondary components (e.g., who/where/when/what) are meaningful insofar as they support the current set of active primary instances. Under this protocol, FOLGA can be trained on each window, enabling the model to track concept drifts, distribution shifts, and evolving interest patterns while preserving interpretability through the 3D latent space.

V. Experimental Evaluation

This section presents the experimental evaluation of our method, FOLGA. We first describe the datasets and experimental settings, then report the results and discuss them to assess effectiveness against state-of-the-art baselines for one-class node classification in heterogeneous graphs. Our evaluation is proposed to (i) demonstrate that FOLGA outperforms state-of-the-art one-class graph neural network methods, (ii) show that FOLGA learns 3-dimensional representations that enable real-time interpretability of the learning process while outperforming the predictive performance of high-dimensional methods, and (iii) provide a basis for exploring FOLGA in heterogeneous graph online scenarios, where we demonstrate competitive results in a dedicated data stream study case. Finally, to support reproducibility, all code and datasets used in our experiments are publicly available[1].

A. Datasets

Prior work on one-class learning with graph neural networks typically relies on recasting standard multi-class benchmarks into a one-class setting, either by selecting a single class as the class of interest or by injecting synthetic anomalies and treating the original classes as normal. While this practice enables controlled experiments, it can mitigate the challenges in naturally one-class problems. In this work, we instead focus on datasets that are inherently one-class, yet can be modeled as heterogeneous graphs with a clear notion of interest class. We found that existing two-class heterogeneous datasets used in some studies are often highly domain-specific, e.g., knowledge-graph-based [28] or tailored to particular graph problems, e.g., heterophilic and large-scale graphs [29], which makes fair comparisons difficult with a broad range of baselines and limits the generality of methodological insights. For these reasons, we collected new, naturally one-class datasets that admit straightforward heterogeneous graph constructions, enabling both principled evaluation and reproducible benchmarking for OCL in heterogeneous graphs.

A particularly natural one-class scenario arises in event sensing, where a user, government agency, or organization is typically interested in only a small subset of event types within a much broader stream of heterogeneous happenings. This domain also lends itself to heterogeneous graph modeling, since events can be modeled using 5W1H elements: the event itself serves as the central node, while who, what, when, where, why, and how form auxiliary node types that can be shared across multiple events, naturally inducing a relational structure. Based on this motivation, we collected event data from the GDELT Project[2], focusing on three target categories (agriculture, immigration, and terrorism) identified by filtering events whose GDELT titles contain the corresponding keywords in four years (2021, 2022, 2023, and 2024). For each event, we use the associated news URL to retrieve the textual content by scraping the

[1]https://github.com/GoloMarcos/FOLGA

[2]http://data.gdeltproject.org/events/

TABLE I
SUMMARY STATISTICS OF THE ONE-CLASS HETEROGENEOUS GRAPH EVENT DATASETS.

Dataset	#Nodes	#Edges	#Events	#Interest	#Outliers	#What	#Where	#When	#Who	#Why	#How
Agriculture	33132	152474	26527	23266	3261	1822	915	179	735	1363	1591
Immigration	8076	32312	5590	1974	3616	650	356	72	273	553	582
Terrorism	8078	32312	5590	1287	4303	652	356	73	273	553	581

website. We use all content returned by the URL, whether complete, incomplete, or noisy, since this strategy is similar to a real-world scenario in which events arrive continuously with noisy content. Finally, we apply a large language model, Gemma 3, to extract the event's 5W1H components from the article text, enabling the construction of heterogeneous graphs.

Each dataset is represented as a heterogeneous graph composed of (i) all events from the target category (the interest class) and (ii) a sampled set of events from other categories, which are treated as outliers. To construct the secondary node types induced by the extracted 5W1H fields, we apply agglomerative clustering independently within each component type (who/what/when/where/why/how). Each resulting cluster defines a single secondary node, which both helps keep the graph well-connected and merges semantically equivalent components that appear in different forms. For initial node features, we encode event texts and 5W1H content with *ibm-granite* multilingual embedding model. For each clustered secondary node, its feature vector is computed as the mean of the embeddings of the component strings assigned to that cluster. Table I presents the dataset details.

B. Experimental Settings

We evaluate all methods using the class f_1-score (i.e., interest and outlier) and the $f1$-macro (the arithmetic mean of the two class f_1-scores). All reported results are computed on a hold-out test split containing 10% of the events of interest and all outlier events. The remaining 90% of the interest events are used to train the one-class graph neural network models. We compare FOLGA against twelve representative baselines from the literature. Since several of these methods do not provide public implementations, and, when available, are often tailored to specific datasets or assumptions that do not directly transfer to generic heterogeneous graphs, we implemented all baselines for this study. To ensure a fair and meaningful comparison in our setting, we carefully adapted each method where necessary (e.g., to handle heterogeneous schemas and our naturally one-class protocol) while preserving the original training objectives and design choices described in the respective papers. Even though we introduced slight variations in the method in some cases, we always maintained the loss functions and training strategies, as they are the core of comparison in this research. We explore the baselines:

1) **SAMCL [8]**: an unsupervised method based on subgraph contrastive learning. SAMCL optimizes three losses: (i) intra-view alignment between an anchor and its subgraph nodes, (ii) inter-view alignment between the two views of the same anchor, and (iii) a reconstruction loss that predicts the anchor representation from its subgraph context.
2) **FIAD [9]**: an unsupervised injection-based method that perturbs encoder embeddings via feature-level operations. Two decoders (structural and attribute) are used for the original and injected embeddings, yielding reconstruction and MSE losses, and an additional loss on the difference of the two latent spaces.
3) **GCN-OCSVM [10] and GAT-OCSVM [12]**: a two-step pipeline. First, a GAE with a GCN/GAT encoder learns node embeddings by reconstructing the adjacency matrix. Second, an OCSVM is used.
4) **EAGLE [11]**: a two-step method that first trains a GCN encoder with a contrastive objective using positive and negative pairs. We employ random walks to generate positive pairs. Second, an OCSVM is used.
5) **OCGNN [13]**: an end-to-end method that applies GNN layers on the graph and optimizes the Equation 6.
6) **DUAL-SVDAE [14]**: an end-to-end approach combining a GAE and AE. It jointly optimizes four losses: Equation 6 on embeddings produced by the GAE and the AE, plus Equation 5 and an MSE by the AE.
7) **STEAM [15]**: an end-to-end model combining a GAE and an AE, where the GAE employs GAT layers and the AE is an MLP whose decoder reconstructs features via embedding interactions. The objective combines three terms: Equations 6 and 5 with an MSE loss.
8) **JAANE [16]**: a GAN-inspired one-class with a dual-view encoder that shares parameters to produce structural and attribute embeddings, which are fused into a latent representation. A discriminator then regularizes the latent space. This method also uses a GAE and a decoder MLP. The final objective combines Equations 6 and 5 with an MSE loss.
9) **MULDUALGNN [17]**: a dual-graph approach that augments the graph with a learned local graph built from node similarity and local topology. The model processes both graphs across layers and optimizes hypersphere-based objectives, including a global hypersphere and multiple cluster-specific hyperspheres.
10) **OLGA [18]**: an end-to-end GAE-based one-class method trained with (i) adjacency reconstruction and (ii) a dedicated new hypersphere objective different from the previous ones.
11) **ECHOLGA [6]**: an extension of OLGA that adds a third objective, a cross-entropy loss over node types.

We adopt a unified hyperparameter configuration across methods:

- **Training**: epochs $=100$, learning rate $=0.001$, number of GNN layers $=2$, $\lambda_{rec}=0.5$, $\lambda_{svdd}=0.5$, $\lambda_k=0.5$;
- **GNN Layer**: All methods used GCNConv, except for methods that have native GATConv in the pipeline (MULDUALGNM, STEAM, and GAT+OCSVM) and methods that implement their own encoder, such as JAANE.
- **Encoder/latent**: hidden dimension $=128$, output dimension $=64$, attention heads $=1$ (when applicable).
- **One-class parameters**: $\nu=0.2$, kernel = `rbf`.
- **FOLGA**: radius $=0.2$, $L=3$ for rewiring bfs with $\alpha=0.2$, $\beta=0.5$, and $\gamma=2$ (immigration and terrorism). radius $=0.3$, $k=3$ for rewiring knn (agriculture). radius $=0.4$ for the data-stream experiments on the slide-window (agriculture).

C. Results and Discussion

Table II presents the results of this article. We report f_1-macro, the f_1 score for the interest class (f_1-int), and the f_1 score for the non-interest class (f_1-out) for the twelve baselines and FOLGA across the three datasets. Each row corresponds to a method, and each column set reports metrics for a single dataset. The best results are in bold, the second best are underlined, and the third best are in italics.

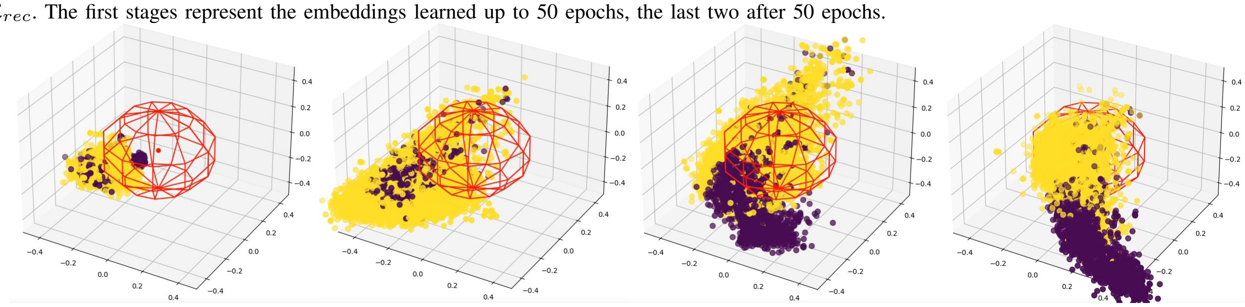
Overall, FOLGA gets the best performance, achieving the highest f_1-macro, f_1-int, and f_1-out. The second-best results are obtained by two-step methods, GCN-OCSVM and GAT-OCSVM, on Terrorism and Agriculture, whereas STEAM ranks second on Immigration. The third-best results are obtained by GCN-OCSVM, JAANE, and OCGNN on Agriculture, Immigration, and Terrorism, respectively.

TABLE II

f_1-MACRO, f_1 OF THE CLASS OF INTEREST (f_1-INT), AND f_1 OF THE CLASS OF NON-INTEREST (f_1-OUT) FOR ALL METHODS IN THE THREE DATASETS. BEST VALUES ARE IN BOLD, SECOND BEST UNDERLINED, AND THIRD BEST IN ITALICS.

Methods	Reference	Agriculture f_1-macro	Agriculture f_1-int	Agriculture f_1-out	Immigration f_1-macro	Immigration f_1-int	Immigration f_1-out	Terrorism f_1-macro	Terrorism f_1-int	Terrorism f_1-out
OCGNN	**Wang et al. (2021) [13]**	0.791	0.767	0.816	0.663	0.397	0.929	*0.954*	*0.910*	*0.997*
DUAL-SVDAE	**Zhang et al. (2022) [14]**	0.865	0.837	0.894	0.878	0.770	0.986	0.935	0.873	*0.997*
STEAM	**Zhang et al. (2022) [15]**	0.330	0.491	0.168	0.889	0.790	0.988	0.939	0.881	*0.997*
SAMCL	**Hu et al. (2023) [8]**	0.516	0.427	0.604	0.416	0.096	0.736	0.402	0.061	0.744
GCN-OCSVM	**Gôlo et al. (2023) [10]**	*0.868*	*0.838*	*0.898*	0.837	0.694	0.979	0.959	0.920	0.998
EAGLE	**Shi et al. (2023) [11]**	0.536	0.594	0.479	0.699	0.456	0.943	0.934	0.871	*0.997*
GAT-OCSVM	**Gôlo et al. (2024) [12]**	0.892	0.869	0.916	0.860	0.736	0.984	0.936	0.876	*0.997*
JAANE	**Fan et al. (2024) [16]**	0.858	0.829	0.887	*0.879*	*0.771*	*0.988*	0.931	0.865	0.996
MULDUALGNN	**Li et al. (2024) [17]**	0.756	0.723	0.790	0.695	0.444	0.946	0.913	0.831	0.995
OLGA	**Gôlo et al. (2025) [18]**	0.856	0.829	0.882	0.802	0.631	0.973	0.887	0.781	0.992
FIAD	**Chen et al. (2025) [9]**	0.671	0.556	0.785	0.474	0.152	0.796	0.409	0.070	0.748
ECHOLGA	**Gôlo et al. (2025) [6]**	0.452	0.169	0.735	0.820	0.664	0.976	0.926	0.857	0.995
FOLGA		**0.933**	**0.924**	**0.942**	**0.902**	**0.815**	**0.989**	**0.962**	**0.926**	**0.998**

Fig. 3. Interpretability of FOLGA on the Agriculture dataset through its 3D latent space. Interest nodes are shown in **yellow** and outliers (non-interest) in **purple**. The **red sphere** represents the learned sphere boundary (center $\boldsymbol{c} = [0, 0, 0]$ and radius $R = 0.3$) induced by the one-class loss $\mathcal{L}_{ocl}$ and reconstruction loss $\mathcal{L}_{rec}$. The first stages represent the embeddings learned up to 50 epochs, the last two after 50 epochs.



Other methods, such as DUAL-SVDAE, MULDUALGNN, and OLGA, remain competitive and close to the top three, although they do not place in the top one, two, or three. Regarding our heterogeneous competitor, MULDUALGNN, even when exploiting graph heterogeneity, the method did not outperform FOLGA, indicating that the rewiring technique explored here is a better way to leverage graph heterogeneity than the method in [17].

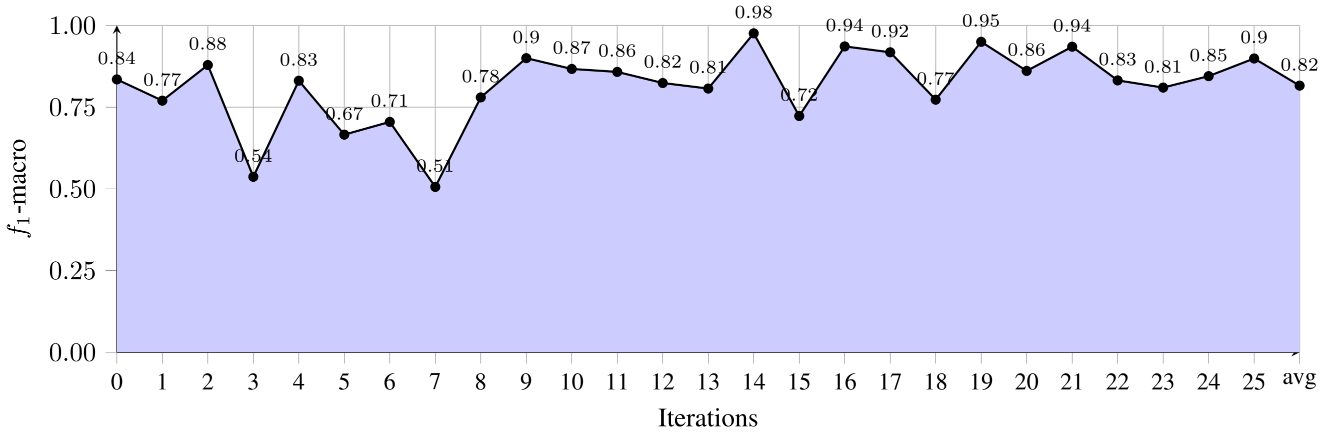
The worst results are observed from SAMCL and FIAD. The consistently weak performance of the unsupervised approaches in this one-class learning node classification protocol indicates that relying solely on reconstruction-based scores and selecting thresholds without supervision is insufficient to accurately recover the interest class, which is reflected in particularly low f_1-int values. The poor results of STEAM, EAGLE, and ECHOLGA in Agriculture reveal limitations that become more pronounced at larger scales. Although STEAM incorporates a hypersphere objective during training, it does not directly use the learned hypersphere for prediction. Instead, it combines multiple scores and applies a thresholding rule. While this strategy works on smaller datasets, its very low performance on Agriculture suggests that threshold-based calibration becomes unreliable given the class distribution and the complexity of the larger graph. EAGLE, despite its two-step approach, fails to separate classes during its contrastive stage, suggesting that its random-walk-based positive and negative sampling strategies may not yield sufficiently informative pairs in large heterogeneous graphs. Finally, ECHOLGA, an extension of OLGA that adds an auxiliary node-type objective, improves performance on the Immigration and Terrorism datasets but degrades on the Agriculture dataset, indicating that the node-type separation term may require additional adaptations when each node type is represented by a very large number of instances.

An additional surprising observation is the competitive performance of the two-step baselines. Although these pipelines learn embeddings that are not explicitly tailored to the one-class objective, they remain competitive across datasets. Our interpretation is that the RBF kernel in OCSVM mitigates part of this limitation by implicitly mapping embeddings into a feature space where the decision boundary can be learned more flexibly. Thus, the constraints of agnostic embeddings are more evident in one-class methods that do not rely on kernelized decision functions.

Overall, Table II supports the effectiveness of combining a reconstruction objective with a hypersphere-based one-class loss: FOLGA ranks first, while DUAL-SVDAE, STEAM, and JAANE are among the strongest competitors in this design family. Despite sharing the same high-level ingredients, the key distinction of FOLGA lies in its relation-rewiring-driven enrichment and its fusion mechanism that aggregates information across multiple rewired graphs. These results indicate that jointly leveraging rewiring (to strengthen heterogeneous edges), a fusion encoder, and a one-class objective coupled with reconstruction yields embeddings that are both more discriminative and inherently interpretable, thereby improving one-class learning for node classification in heterogeneous graphs.

On the Agriculture dataset, FOLGA achieves the best performance (f_1-macro: 0.933), clearly outperforming the strongest baselines (GAT-OCSVM: 0.892 and GCN-OCSVM: 0.868). The gain is consistent across both classes (f_1-int: 0.924; f_1-out: 0.942), suggesting that in large graphs with rich heterogeneity, exploiting additional relations via rewiring enhances message passing and improves separation between interest nodes and outliers. On Immigration, the margin is lower: FOLGA reaches an f_1-macro of 0.902, ahead of STEAM (0.889) and JAANE (0.879). Here, the main advantage is maintaining a high f_1-int (0.815) without decreasing f_1-out (0.989), indicating a more stable one-class boundary even in a smaller

Fig. 4. Data-stream evaluation on Agriculture using a sliding-window protocol (2000). At each iteration m, FOLGA is trained with 1000 old interest nodes and evaluated on 1000 newly arriving primary nodes. We report f_1-macro.



dataset where performance is typically more sensitive to loss design. On Terrorism, many methods already achieve f_1-out values close to 1.0, which reduces the scenario for improvement in f_1-macro. Nevertheless, FOLGA achieves the best result (f_1-macro: 0.962), slightly surpassing the second-best method (GCN-OCSVM: 0.959). The improvement is driven primarily by f_1-int, since FOLGA reaches 0.926, reflecting better recovery of the interest class in a setting where the non-interest class is easily identified by several approaches.

Although f_1-macro is a useful balanced metric, in OCL, another important metric is f_1-int, since the practical goal is to accurately recover the instances of interest. FOLGA achieves the highest f_1-int on all datasets, which is particularly important in OCL. On Agriculture, FOLGA attains f_1-int of 0.924, substantially outperforming the best baseline (GAT-OCSVM at 0.869). On Immigration, FOLGA reaches 0.815, outperforming STEAM (0.790). On Terrorism, it again leads with 0.926, exceeding GCN-OCSVM (0.920). These gains in f_1-int indicate that FOLGA consistently improves the detection of the interest class. Moreover, Table II shows that several methods obtain very high f_1-out but comparatively low f_1-int, which can mask an inability to detect what matters most. In contrast, FOLGA maintains both high f_1-out and consistently superior f_1-int, reflecting a more effective separation of the interest class.

Figure 3 illustrates the interpretability results on the Agriculture dataset by visualizing the 3D embeddings learned by FOLGA for the primary nodes of the heterogeneous graph. Because the latent space is three-dimensional, we can directly produce a real-time video of the training dynamics without additional post-processing. We report four representative stages of training to highlight the evolution of the FOLGA embeddings: (1) initialization, (2) early intermediate, (3) late intermediate, and (4) the final stage.

The visualization shows how FOLGA balances reconstruction and one-class compactness throughout training. Initially, the model primarily optimizes adjacency reconstruction, and the sphere constraint has only a limited influence on the global geometry. As training progresses, the one-class loss $\mathcal{L}_{ocl}$ becomes increasingly influential, pulling interest nodes toward the sphere center, while the reconstruction loss $\mathcal{L}_{rec}$ continues to regularize the space and prevents representation collapse. Consequently, interest nodes (yellow) progressively concentrate inside or near the sphere, whereas outliers (purple) form a separated area outside the boundary, especially in stages (3) and (4). This behavior provides a direct geometric explanation of the score o_i, which measures the distance to the center relative to the radius. We also observe outliers near the sphere boundary and a few nodes of interest outside it, which may correspond to noise events. These instances can be inspected qualitatively due to the explicit 3D space.

Since we have the video embedding trajectories over training time, the exact visualization can be leveraged in streaming or windowed settings to monitor concept drift: if the interest cluster shifts substantially or outliers begin to invade the sphere, it signals a distributional change and motivates updating the model or its parameters. In the next section, we present the Data Stream Case Study of FOLGA, using a sliding window strategy on the agriculture dataset.

D. Data Stream Case Study

In our data-stream study case, we assess whether FOLGA can operate on an evolving heterogeneous graph in which new events and their associated secondary nodes/edges arrive over time. We adopt a sliding-window protocol on our large dataset (Agriculture) with a window size $2,000$: at iteration m, we keep $1,000$ old interest nodes (used for training) and incorporate $1,000$ newly arriving primary nodes (the prediction batch). The heterogeneous snapshot is formed by these primary nodes together with the secondary nodes that remain connected through their typed relations, yielding a sequence of dynamic graphs (25 windows in total). Figure 4 reports the resulting f_1-macro per iteration, measuring the model's ability to maintain balanced interest/outlier performance as the heterogeneous graph evolves. Figure 5 shows the confusion matrices of iterations 3 and 7, in which FOLGA had the worst results.

Fig. 5. Confusion matrices for the data-stream performance. Rows denote true labels and columns denote predicted labels.

		Predicted	
		Out	**Int**
True	**Out**	5	61
	Int	16	918

Iteration 3

		Predicted	
		Out	**Int**
True	**Out**	2	74
	Int	0	924

Iteration 7

Overall, FOLGA achieves competitive, mostly stable performance across the stream, with an average f_1-macro of 0.82 and several iterations exceeding 0.85, indicating that the method can be applied repeatedly to an online stream without requiring post-processing beyond standard inference. The main degradations occur at iterations 3 and 7, which is consistent with concept drift and/or changes in class composition and difficulty: the confusion matrices show that the failures are dominated by false positives for the interest class (it. 3: 61 outliers predicted as interest; it. 7: 74 outliers predicted as interest), while the interest class itself remains well recovered (it. 7 has 0 false negatives). This pattern suggests that, in those windows, the outlier distribution becomes closer to the learned interest region, causing outliers to be absorbed into the interest boundary. Even under these challenging windows, the remaining iterations maintain strong performance, supporting the view that FOLGA is suitable as a foundation for online heterogeneous one-class learning and motivating drift-aware updates in long-running deployments.

VI. Conclusion and Future Work

We introduced FOLGA, a new end-to-end approach to one-class node classification in heterogeneous graphs. By explicitly leveraging heterogeneity through rewiring strategies and fusing multiple rewired graphs into a dedicated GNN encoder, FOLGA learns representations that preserve topology and achieve one-class compactness. The resulting model combines a GAE reconstruction objective with a sphere-based one-class loss, producing 3D embeddings that enable direct, real-time interpretability of the decision boundary and training dynamics without additional post-processing.

Our study on three one-class event datasets shows that FOLGA consistently achieves state-of-the-art results compared to 12 representative baselines. In addition to f_1-macro, FOLGA got the highest f_1-int across all datasets, highlighting its practical ability to recover the class of interest. Beyond static evaluation, we demonstrated that FOLGA is applicable to data-stream settings, achieving competitive average performance while exposing drift-related failure modes that can be monitored through the model's geometric interpretability.

As future work, given the strong results achieved through rewiring on heterogeneous graphs, we plan to evaluate FOLGA in homogeneous graphs to better characterize when heterogeneity-specific components are beneficial versus new rewiring strategies for homogeneous graphs. We also intend to explore alternative message-passing operators, including attention-based layers (e.g., GAT) and explicitly heterogeneous GNN layers, to assess whether richer type-aware aggregation can further improve interest/outlier separation. Finally, we intend to develop a new data-stream version of FOLGA with drift-aware adaptation, in which the model dynamically updates its learning behavior when concept drift is detected, to maintain performance under evolving graph distributions.

Declaration on Generative AI

The authors utilized generative AI to correct grammar and enhance scientific writing. The authors take responsibility for the generated content and affirm that all scientific references were added manually and appropriately to support the AI-enhanced content.

Acknowledgment

We were supported by CAPES (Process No. 88887.671481/2022-00), and FAPESP (Grants 2025/07160-0, 2024/08485-8, 2024/15430-5, 2023/10100-4, 2022/09091-8). We also acknowledge support from the Google Latin America PhD Fellowship.

References

[1] G. Gkarmpounis, C. Vranis, N. Vretos, and P. Daras, "Survey on graph neural networks," *IEEE Access*, 2024.

[2] D. Yang, Q. Zhang, J. Gou, and P. Chen, "Gafe: Enhancing fraud detection performance in multi-relational heterogeneous graphs through grouping aggregation and feature extraction," in *International Joint Conference on Neural Networks*, 2025, pp. 1–8.

[3] R. Bing, G. Yuan, M. Zhu, F. Meng, H. Ma, and S. Qiao, "Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications," *Artificial Intelligence Review*, vol. 56, no. 8, 2023.

[4] N. Seliya, A. Abdollah Zadeh, and T. M. Khoshgoftaar, "A literature review on one-class classification and its potential applications in big data," *Journal of Big Data*, vol. 8, 2021.

[5] F. Emmert-Streib and M. Dehmer, "Taxonomy of machine learning paradigms: A data-centric perspective," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 5, 2022.

[6] M. P. S. Gôlo and R. M. Marcacini, "One-class edge classification through heterogeneous hypergraph for causal discovery," *Scientific Reports*, 2025.

[7] W. Ju, Z. Fang, Y. Gu, Z. Liu, Q. Long, Z. Qiao, Y. Qin, J. Shen, F. Sun, Z. Xiao *et al.*, "A comprehensive survey on deep graph representation learning," *Neural Networks*, 2024.

[8] J. Hu, B. Xiao, H. Jin, J. Duan, S. Wang, Z. Lv, S. Wang, X. Liu, and E. Zhu, "Samcl: Subgraph-aligned multiview contrastive learning for graph anomaly detection," *Transactions on Neural Networks and Learning Systems*, 2023.

[9] A. Chen, J. Wu, and H. Zhang, "Fiad: Graph anomaly detection framework based feature injection," *Expert Systems with Applications*, vol. 259, p. 125216, 2025.

[10] M. P. S. Gôlo, M. I. De Moraes, R. Goularte, and R. M. Marcacini, "On the use of early fusion operators on heterogeneous graph neural networks for one-class learning," in *Brazilian Symposium on Multimedia and the Web*, 2023.

[11] B. Shi, B. Dong, Y. Xu, J. Wang, Y. Wang, and Q. Zheng, "An edge feature aware heterogeneous graph neural network model to support tax evasion detection," *Expert Systems with Applications*, vol. 213, 2023.

[12] M. P. S. Gôlo, M. I. de Moraes Junior, R. Goularte, and R. M. Marcacini, "Unsupervised heterogeneous graph neural networks for one-class tasks: Exploring early fusion operators," *Journal on Interactive Systems*, vol. 15, no. 1, 2024.

[13] X. Wang, B. Jin, Y. Du, P. Cui, Y. Tan, and Y. Yang, "One-class graph neural networks for anomaly detection in attributed networks," *Neural computing and applications*, vol. 33, 2021.

[14] F. Zhang, H. Fan, R. Wang, Z. Li, and T. Liang, "Deep dual support vector data description for anomaly detection on attributed networks," *International Journal of Intelligent Systems*, vol. 37, no. 2, 2022.

[15] J. Zhang, X. Wang, Z. Yu, M. Huang, L. Duan, and J. Zuo, "Steam: Svdd-based anomaly detection on attributed networks via multi-autoencoders," in *Intenational Conference on High Performance Computing & Communications*. IEEE, 2022.

[16] H. Fan, R. Wang, X. Huang, F. Zhang, Z. Li, and S. Su, "Deep joint adversarial learning for anomaly detection on attribute networks," *Information Sciences*, vol. 654, 2024.

[17] Q. Li, G. Wu, H. Ni, and T. You, "Anomaly detection with dual-channel heterogeneous graph based on hypersphere learning," *Information Sciences*, vol. 681, p. 121242, 2024.

[18] M. P. S. Gôlo, J. G. B. de Medeiros Junior, D. F. Silva, and R. M. Marcacini, "One-class graph autoencoder: A new end-to-end, low-dimensional, and interpretable approach for node classification," *Information Sciences*, 2025.

[19] J. Zhou, C. Xie, S. Gong, Z. Wen, X. Zhao, Q. Xuan, and X. Yang, "Data augmentation on graphs: A technical survey," *ACM Computing Surveys*, vol. 57, no. 11, 2025.

[20] G. H. Messias, S. Iasulaitis, and A. D. B. Valejo, "Graph neural networks for positive and unlabeled learning: a rewiring approach," *Knowledge and Information Systems*, vol. 67, no. 11, 2025.

[21] J. Guo, L. Du, W. Bi, Q. Fu, X. Ma, X. Chen, S. Han, D. Zhang, and Y. Zhang, "Homophily-oriented heterogeneous graph rewiring," in *Proceedings of the ACM web conference 2023*, 2023.

[22] Y. Yang, P. Wang, X. He, and D. Zou, "Gram: An interpretable approach for graph anomaly detection using gradient attention maps," *Neural Networks*, 2024.

[23] Q. Li, T. You, J. Chen, Y. Zhang, and C. Du, "Biodyngrap: Biomedical event prediction via interpretable learning framework for heterogeneous dynamic graphs," *Expert Systems with Applications*, vol. 244, 2024.

[24] X. Chen, J. Liang, and H. Yu, "Semi-supervised graph neural networks with incremental learning for fraud detection," in *International Joint Conference on Neural Networks*, 2025, pp. 1–8.

[25] T. Kipf and M. Welling, "Variational graph auto-encoders," *stat*, 2016.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[27] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018.

[28] Y. Liu, H. Zhang, C. Yang, A. Li, Y. Ji, L. Zhang, T. Li, J. Yang, T. Zhao, J. Yang *et al.*, "Datasets and interfaces for benchmarking heterogeneous graph neural networks," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023.

[29] J. Lin, X. Guo, S. Zhang, D. Zhou, Y. Zhu, and J. Shun, "When heterophily meets heterogeneity: New graph benchmarks and effective methods," *CoRR*, vol. abs/2407.10916, 2024.