Journal on Interactive Systems, 2024, 15:1, doi: 10.5753/jis.2024.4109

Unsupervised Heterogeneous Graph Neural Networks for One-Class Tasks: Exploring Early Fusion Operators

Marcos Paulo Silva Gôlo [**University of São Paulo** | ***marcosgolo@usp.br***]
Marcelo Isaias de Moraes Junior [**University of São Paulo** | ***marcelo.junior@usp.br***]
Rudinei Goularte [**University of São Paulo** | ***rudinei@icmc.usp.br***]
Ricardo Marcondes Marcacini [**University of São Paulo** | ***ricardo.marcacini@icmc.usp.br***]

Institute of Mathematics and Computer Sciences, University of São Paulo, Av. Trabalhador São Carlense, 400, Centro, São Carlos, Sãp Paulo, Brazil, 13566-590.

Received: 07 February 2024 • **Accepted:** 28 May 2024 • **Published:** 30 May 2024

Abstract Heterogeneous graphs are an essential structure that models real-world data through different types of nodes and relationships between them, including multimodality, which comprises different types of data such as text, image, and audio. Graph Neural Networks (GNNs) are a prominent graph representation learning method that takes advantage of the graph structure and its attributes that, when applied to the multimodal heterogeneous graph, learn a unique semantic space for the different modalities. Consequently, it allows multimodal fusion through simple operators such as sum, average, or multiplication, generating unified representations considering the supplementary and complementarity relationships between the modalities. In multimodal heterogeneous graphs, the labeling process tends to be even more costly due to the multiple modalities analyzed, in addition to the imbalance of classes inherent to some applications. In order to overcome these problems in applications that comprise a class of interest, One-Class Learning (OCL) is used. Given the lack of studies on multimodal early fusion in heterogeneous graphs for OCL tasks, we proposed a method based on unsupervised GNN for heterogeneous graphs and evaluated different early fusion operators. In this paper, we extend another work by evaluating the behavior of the main GNN convolutions in the method. We highlight that using operators such as average, addition, and subtraction were the best early fusion operators. In addition, GNN layers that do not use an attention mechanism performed better. In this way, we argue for heterogeneous graph neural networks in multimodal using early fusion simple operators instead of well-often-used concatenation and less complex convolutions.

Keywords: Heterogeneous Early Fusion, One-Class Learning, Heterogeneous Graphs, Multimodal Graphs

1 Introduction

Early fusion data generates new multimodal, robust, and unified representations considering supplementary and complementary modalities of different data types, such as audio, image, and text (Baltrušaitis *et al.*, 2018). These heterogeneous and multimodal data can be modeled using heterogeneous graphs. Graphs offer a powerful structure for modeling real-world problems by explicitly capturing the relations between entities since graphs better deal with abstract concepts such as relations and interactions (Rahman, 2017). Heterogeneous graphs model real-world problems with a natural structure that enriches the task resolution that solves these problems (Zhou *et al.*, 2020; Xia *et al.*, 2021). In addition, heterogeneous graphs allow the modeling of different relations between different graph nodes, which enriches the representation through graphs, modeling more information and modalities (Wang *et al.*, 2022).

In the scenario of heterogeneous graphs, each type of node can be considered a modality. In this way, we can exploit the modalities fusion to generate better representations for the problem. Early fusion studies for multimodal data explore mainly concatenation (Beserra, 2022) of the modalities. Concatenation increases the dimensionality of vectors (doubling in the case of two modalities, tripling in the case of three, and so on). On the other hand, few studies explore different early fusion strategies, such as vector operators between feature vectors in the latent spaces of each modality (Beserra *et al.*, 2020; Beserra and Goularte, 2023). For instance, addition, average, subtraction, and minimum, among others (Beserra, 2022).

Existing studies do not investigate multimodal fusion operators for heterogeneous graphs. This research gap is particularly promising as many multimodal applications are now being modeled using heterogeneous graphs (Guo *et al.*, 2019). For instance, multimodal document classification (which may include text, images, and audio) (Liu *et al.*, 2019), recommendation systems, in which multimodal fusion in heterogeneous graphs can enhance accuracy by considering different types of information (e.g., browsing history and personal preferences) (Guo *et al.*, 2020), event detection in multimedia-based social networks (Schinas *et al.*, 2015), and content retrieval based on its multimodal content (Kumar *et al.*, 2013).

In different scenarios of data modeled through heterogeneous graphs, there is an interest class, such as the detection of hit songs (da Silva *et al.*, 2022), detection of fake news (de Souza *et al.*, 2022), recommendation (Gôlo *et al.*, 2022), and detection of interest events (Nguyen and Grishman, 2018). In these scenarios, the studies model the data through heterogeneous graphs and explore One-Class Learning (OCL) (Emmert-Streib and Dehmer, 2022; Tax, 2001). Studies use the OCL because they can learn to classify the interest class only with labels of this class available. Therefore, OCL reduces the user's labeling effort, is more appropriate for imbalanced classification scenarios, and does not need to cover the scope of the non-interest class (or classes) (Khan and Madden, 2014; Alam *et al.*, 2020).

In the one-class heterogeneous graph learning literature, studies learn representations for nodes through Graph Neural Networks (GNNs), methods considered state-of-the-art for learning representation in graphs (Wu *et al.*, 2020). GNNs are neural networks applied to data modeled through graphs capable of learning new, more robust representations that capture structural features given the graph node relations and node features given the initial representations of the nodes. At the end of learning, the GNNs learn representations for the different types of nodes at the same semantic level, which makes it possible to use simple operators, such as addition, average, or multiplication, to combine the representations of the different types of nodes considering the early fusion and improve the task solved by one-class learning (Atrey *et al.*, 2010; Jakob *et al.*, 2021; Gôlo *et al.*, 2021). On the other hand, studies in the one-class heterogeneous graph learning literature concatenate the learned representations (Huang *et al.*, 2022; Zhou and Mao, 2022; Gôlo *et al.*, 2022) or only use one type of node (da Silva *et al.*, 2022; Ganz *et al.*, 2023) in the one-class learning step.

This article is as extended version of Gôlo *et al.* (2023a) that proposes a graph neural network (GNN) method for heterogeneous graphs that explores different types of early fusion operators to deal with multiple modalities. We perform an extensive empirical evaluation of fusion operators for the representations of different types of nodes learned by GNNs and by graph regularization for one-class tasks. We propose a generic pipeline with the learning of representation in heterogeneous graphs through a Graph Autoencoder (GAE) (Kipf and Welling, 2016), considering the classification through the algorithms One-Class Support Vector Machines (OCSVM) (Schölkopf *et al.*, 2001). In this extended version we explore three GNN layers in our results: Graph Convolutional Network (GCN) (Kipf and Welling, 2017), Graph SAmpling and aggreGatE (GraphSAGE) (Hamilton *et al.*, 2017), and Graph Attention Network (GAT) (Velickovic *et al.*, 2018). We add more analysis and disccusions. Our pipeline represents any type of heterogeneous graph and solves any one-class problem. Based on the experiments conducted, we answered the following research questions:

1. Does the early fusion of the regularized representations or learned by GNN improve the performance of one-class tasks?
2. Which early fusion operator generates better representations for one-class problems?
3. What is the best representation obtained through graphs to apply early fusion in one-class tasks?
4. Which Graph Neural Network layer obtains better representations for one-class tasks considering unsupervised representation learning?

We performed an extensive experimental evaluation to answer these research questions. We consider four one-class problems using four different datasets: hit song detection, movie recommendation, events of interest detection, and fake news detection. We represent the nodes of all heterogeneous graphs naturally modeled through an unsupervised GAE considering the GCN, GraphSAGE, and GAT layers and classify the representations of interest using OCSVM. We use seven fusion operators to generate the representations of interest: addition, subtraction, average, multiplication, concatenation, minimum, and maximum. We consider the k-fold cross-validation to run our experiments, the t-Distributed Stochastic Neighbor Embedding algorithm to reduce the dimension of the representations to visualize the embeddings, and the f_1-macro to evaluate the fusion operators. In summary, our contributions are:

- We present a model that incorporates additional graph modalities to the target nodes of classification, facilitating the exploration of graph heterogeneity through their combination;
- While most of the existing methods explore concatenation operator for modality fusion, we investigate and evaluate the impact of alternative types of early fusion operators (addition, subtraction, multiplication, maximum, minimum, and average) to advance studies on heterogeneous scenarios modeled through graphs in one-class tasks;
- We introduce the application of different GNN layers for unsupervised learning in one-class tasks, enabling progress in selecting appropriate layers for various one-class problems.

The early fusion of the representations improved the performance in one-class learning in all datasets considering representations generated by the graph regularization and the GAE. The GAE representations performed better than the regularized representations in most evaluation scenarios. We highlight the Average, Addition, and Subtraction operators as the fusion operators that obtained the best results for one-class tasks considering data modeled through heterogeneous graphs. Two-dimensional projections showed the effectiveness of the fusion operators. We highlight the two-dimensional projection of the representations of the Addition, Average, and Subtraction operators.

We divide the remainder of the article: Section 2 presents the background for the paper. Section 3 presents early fusion in heterogeneous graphs on one-class problems related work. Section 4 presents our proposal for learning representation in heterogeneous graphs, early fusion, and one-class learning. Section 5 presents the experimental evaluation with information about the datasets, experimental setup, results, and discussion. Finally, Section 6 presents the study's conclusions and future work.

2 Background

In graphs without initial representation for all nodes, we need to obtain initial representations for all nodes. Thus, we use a regularization framework to obtain a vector of attributes for each graph node, enabling Graph Autoencoders to learn representation in the regularized graph. We present the regularization in Section 2.1. After, we can apply a graph neural network to learn new, more robust representations for the nodes from the representations obtained. We present the graph autoencoders in Section 2.2 as an unsupervised graph neural network.

2.1 Regularization

We denote $G = (V, E)$ as a graph in which V is the set of vertices, and E is the set of edges. In addition, we associate G with an array of attribute vectors of f-dimensional nodes F. However, there are scenarios where only a subset of nodes of the V_F graph has an associated attribute vector, which makes the use of graph neural networks impracticable. Thus, a solution is using a regularization framework for learning graph representations (do Carmo and Marcacini, 2021). Equation 1 defines the objective function to be minimized by the process:

$$Q(\mathbf{X}) = \frac{1}{2} \sum_{u,v \in E} (x_u - x_v)^2 + \mu \sum_{k \in V_F} (x_k - f_k)^2. \quad (1)$$

The first term determines that attribute vectors of neighboring nodes u and v are similar. At the same time, the second term, weighted by a factor $\mu \in \mathbb{R}$, indicates how much the initial attribute vector we want to preserve during the procedure. The described problem is an optimization problem that can be solved using an iterative label propagation method (Zhou and Schölkopf, 2004). At the end of the process, we have an array of node attributes $\mathbf{X} \in \mathbb{R}^{|V| \times f}$, in which all graph nodes have a vector with features. After obtaining a feature for each node, we can obtain robust and tuned representation for the graph nodes through the heterogeneous graph autoencoders that we present in the next section.

2.2 Graph Autoencoders

Graph neural networks are a learning method of graph representation that generalize convolutions to graphs. It consists of iterative updates of the node representation through neighborhood aggregation (Wu *et al.*, 2020). After k iterations, the GNN aggregates structural information of the k-hop neighborhood of nodes (Xu *et al.*, 2019). Formally defined as $\mathbf{Z} = \text{GNN}(\mathbf{A}, \mathbf{X})$, generating a matrix of latent representations $\mathbf{Z} \in \mathbb{R}^{|V| \times d}$, in which $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$ is the adjacency matrix. Among the GNN convolutions, the Graph Convolution Networks (GCN) (Kipf and Welling, 2017) is a spectral convolution method based on the Laplacian of a graph. The l-th layer of GCN is defined in Equation 2,

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\mathbf{W}^{(l)} \sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{1}{\sqrt{\tilde{d}_i \tilde{d}_j}} \mathbf{h}_j^{(l)} \right), \quad (2)$$

in which $\mathbf{h}_i^{(l)}$ and $\mathbf{W}^{(l)}$ are, respectively, node i representation and parameters of the l-th layer, $\mathcal{N}_i = \{j, \ (j,i) \in E\}$ is the 1-hop neighborhood of node i, $\tilde{d}_i = |\mathcal{N}_i| + 1$ is node degree with self-loop added, and σ a non-linear activation function like ReLU. It is worth noting that $\mathbf{h}_i^{(0)} = \mathbf{X}_i$ and $\mathbf{Z}_i = \mathbf{h}_i^{(k)}$. Eliminating the need for Laplacian computation, Graph SAmpling and aggreGatE (GraphSAGE)[1] (Hamilton *et al.*, 2017) is a non-spectral convolution that generalizes GCN to use trainable aggregation functions. The l-th layer of SAGE is formalized in Equation 3,

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\mathbf{W}^{(l)} \left[\mathbf{h}_i^{(l)} \middle\| \mathbf{h}_{\mathcal{N}_i}^{(l)} \right] \right), \quad (3)$$

in which concatenates the representation of the current node to the aggregated representation of the neighborhood of the node $\mathbf{h}_{\mathcal{N}_i}^{(l)}$. In the original paper, they evaluated SAGE with non-trainable aggregation functions such as the simple element-wise mean of the neighborhood representations defined in Equation 4,

$$\mathbf{h}_{\mathcal{N}_i}^{(l)} = \text{MEAN} \left(\left\{ \mathbf{h}_j^{(l)}, \ \forall j \in \mathcal{N}_i \right\} \right), \quad (4)$$

and aggregation functions with trainable parameters such as max-pooling described in Equation 5,

$$\mathbf{h}_{\mathcal{N}_i}^{(l)} = \text{MAX} \left(\left\{ \text{ReLU} \left(\mathbf{W}_{\text{agg}}^{(l)} \mathbf{h}_j^{(l)} + \mathbf{b}_{\text{agg}}^{(l)} \right), \ \forall j \in \mathcal{N}_i \right\} \right), \quad (5)$$

where $\mathbf{W}_{\text{agg}}^{(l)}$ and $\mathbf{b}_{\text{agg}}^{(l)}$ are the learnable weights. Using learnable aggregation functions, the Graph Attention Network (GAT) (Velickovic *et al.*, 2018) is a convolution incorporating the attention mechanism in aggregation. The mechanism assigns different weights (or importance levels) to each node in the node's neighborhood. GAT uses multiple independent heads, where each head pays attention to different particularities of the neighborhood. Equation 6 defines the l-th GAT layer,

$$\mathbf{h}_i^{(l+1)} = \Big\Vert_{h=1}^{H} \sigma \left(\sum_{j \in \mathcal{N}_i \cup \{i\}} \alpha_{ij}^{(l,h)} \mathbf{W}^{(l,h)} \mathbf{h}_j^{(l)} \right), \quad (6)$$

in which $\alpha_{ij}^{(l,h)}$ denotes the h-th head of l-th layer attention coefficient (or importance) of node j to i and concantenating the representations of the H heads. The dynamic attention (Brody *et al.*, 2022) $\alpha_{ij}^{(l,h)}$ is normalized through the neighborhood $\mathcal{N}_i$ is computed with trainable weights $\mathbf{W}^{(l,h)}$ and $\mathbf{a}^{(l,h)}$ as shown in Equation 7,

$$\alpha_{ij}^{(l,h)} = \text{Softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \quad (7)$$

$$e_{ij} = \mathbf{a}^{(l,h)} \text{LeakyReLU} \left(\mathbf{W}^{(l,h)} \left[\mathbf{h}_i^{(l)} \middle\| \mathbf{h}_j^{(l)} \right] \right). \quad (8)$$

In this paper, we propose to use Graph Autoencoder (GAE) to obtain representations in an unsupervised way (Kipf and Welling, 2016). The GAE is an unsupervised training framework for GNNs, whose objective is to compress the structural information of the graph into a lower dimensionality space through reconstructing the adjacency matrix, i.e., we use the encoder with the GNN layers. Which can

[1] We will refer to it as SAGE

be predicted through an inner product of the representations obtained, described as $\hat{\mathbf{A}} = \sigma(\mathbf{Z}\mathbf{Z}^T)$, i.e., we use the decoder with the inner product. However, in practice, as the adjacency matrix is sparse, we use the negative sampling of edges that do not exist in the original matrix. After obtaining representations through the GAE, we can apply different operators in the representations because the nodes' representations are at the same semantic level.

3 Related Work

This section presents related work to multimodal fusion on data modeled through heterogeneous graphs in the resolution of problems solved by one-class learning. Huang et al. (Huang *et al.*, 2022) proposed to detect intrusions in systems through one-class learning. The study model the graph considering process and file nodes. The authors propose a directed heterogeneous graph neural network to learn the representation of the process and file nodes. In graph modeling, the authors use the process **fork** process and process **access** files as the edges. In heterogeneous GNN, the authors proposed a new type of aggregation that considers the directionality of the graph since directionality influences the intrusion detection task in a graph of processes and files. The authors concatenate the node representations to detect anomalies. The authors use the Deep Support Vector Data Description (DeepSVDD) algorithm (Ruff *et al.*, 2018) to detect the anomalies. The authors used the real host data from the enterprise dataset and obtained better results than baselines, such as DeppSVDD, and other methods based on GNNs.

Gôlo *et al.* (2022) recommended movies through one-class learning. The authors proposed a framework that combined enriched modeling of a graph for the recommendation, representation learning through an unsupervised GNN, and a one-class learning algorithm. Naturally, movies are connected with users (user rating for movies). The authors added nodes for keywords, genres, and movie reviews for enriched modeling. The study used a link prediction strategy, i.e., prediction if an edge between the types of nodes existed or not, to learn the representations for nodes through the unsupervised GNN. After learning the representations, the authors concatenated the representations of rating 5 to train the one-class classifier (recommendation). The authors used One-Class Support Vector Machines (OCSVM) to classify the recommendations. The work used a movie recommendation dataset and enriched it with IMDB data. The proposal performed better than baselines such as BERT and GNNs end-to-end.

da Silva *et al.* (2022) proposed to detect hit songs through one-class learning. The authors modeled the songs through graphs and used a GNN to learn a robust node representation. In the modeling, da Silva *et al.* (2022) connect the songs with their respective artists and enrich the modeling with relations between artists. The authors use an unsupervised heterogeneous GNN that learns representations with a loss function that keeps nodes connected with similar representations and unconnected nodes with less similar representations. The study uses the OCSVM considering only music-type nodes to classify hit songs. The authors used a Spotify dataset and obtained better results than the baselines such as the BERT, and the concatenation of BERT and artist representation.

Ganz *et al.* (2023) detects backdoor software through one-class learning. The authors model code activities through collaborative graphs with commit nodes, branches, files, developers, and methods (functions). Ganz *et al.* (2023) represents the graph's nodes through an unsupervised heterogeneous GNN, specifically, a Variational graph autoencoder. After learning the representations for the different types of heterogeneous graph nodes, the authors use only the commit node to detect software backdoors through Deep SVDD. The authors used a dataset extracted from GitHub repositories. Ganz *et al.* (2023) generated anomalies synthetically for training. The study uses state-of-the-art from the literature and the OCSVM, Deep-SVDD, Local Outlier Factor, Elliptic Envelop, and Isolation Forest as baselines. The study has competitive results with the advantages of detecting anomalies in different nodes and proposing an interpretable model.

Zhou and Mao (2022) perform the extraction of arguments in events by classifying arguments of interest and non-interest. The authors proposed a new loss function based on hyperspheres. This function can be adapted for one-class learning and has been proposed as an adaptation of loss from Wang *et al.* (2021). The loss function penalizes nodes of interest outside the hypersphere and unsupervised nodes inside the hypersphere. In the data modeling, Zhou and Mao (2022) modeled the events through the texts and generated graphs with two types of nodes: sentences and entities. The work uses a Graph Attention Network to learn representations at the same semantic level and later concatenates the learned representations to generate a new one. The authors use a dataset and four variations, each with an argument of interest. The proposed method performed better than other state-of-the-art.

Studies with modeling through heterogeneous graphs for one-class tasks perform the concatenation of the learned representations at the same semantic level or use only one node type, even having more representations from other node types available. Therefore, the studies do not progress in relation to other heterogeneous representations or other fusion operators for representations on the same semantic level that can obtain better representation and consequently improve the results on one-class tasks. In this sense, the next section presents a method based on GNNs that considers different fusion operators on data modeled through heterogeneous graphs to solve one-class problems.

4 Early Fusion On Heterogeneous Graph Neural Networks For One-Class Learning

The first step of our pipeline was the regularization (Section 2.1), and the second was the representation learning through a Graph Autoencoder (Section 2.2). Later, the fusion operators aggregate information from the different node types in the graph into a single fused representation. Section 4.1 presents the early fusion process. We will submit these new representations to a one-class learning algorithm and then classify the instances as belonging to the interest class. Finally, Section 4.2 presents the one-class learning algorithm. Figure 1 summarizes the proposed method.

4.1 Heterogeneous Early Fusion

After obtaining the representations, we divide the nodes by node type. For instance, consider a graph with three node types, $\{V_F, V_a, V_b\} \in V$, in which a and b are the node types of the heterogeneous graph whose representations were obtained through regularization. V_F is the set of main nodes that have the initial representation. Considering our scenarios, we have sets of V_F: the news, events, music, and items. We employ early fusion operators by combining the nodes' representations $\boldsymbol{Z_F}$ with $\boldsymbol{Z_a}, \boldsymbol{Z_b}$, in which $\boldsymbol{Z_a}, \boldsymbol{Z_b}$ are the representations generated by the GNN for the node type a and b. Given an $v_i \in V_F$, the neighboring nodes of v_i are first grouped to a single representation for each node type through an average. We define this first step in Equation 9,

$$\boldsymbol{d_{a_{v_i}}} = average(\boldsymbol{Z_{a_{v_i}}}), \tag{9}$$

in which $\boldsymbol{d_{a_{v_i}}}$ is the representation generated for the nodes of type a generated by the representation average of all neighboring nodes of type a considering v_i ($\boldsymbol{Z_{a_{v_i}}}$). We apply this process to all neighboring node types of v_i, $\{\boldsymbol{d_{a_{v_i}}}, \boldsymbol{d_{b_{v_i}}}\} \in \boldsymbol{D_{v_i}}$.

Finally, we define an *op* operator, i.e., the early fusion operator. We use the operators: addition, subtraction, multiplication, minimum, maximum, average, and concatenation. It is worth mentioning that the concatenation will increase the new representation dimensionality, i.e. doubling in the case of two modalities, tripling in the case of three, and so on, while the other operators will maintain the modalities dimension. For a main node v_i, the fusion process consists of applying the *op* operator to all generated node type representations $\boldsymbol{D_{v_i}}$. We define the process of combining the representations in Equation 10,

$$\boldsymbol{\lambda_{v_i}} = op(\boldsymbol{z_{v_i}}, \boldsymbol{d_{j_{v_i}}}) \ \ \forall \boldsymbol{d_{j_{v_i}}} \in \boldsymbol{D_{v_i}}, \tag{10}$$

in which $\boldsymbol{\lambda_{v_i}}$ is the fused representation generated to the v_i node, and $\boldsymbol{z_{v_i}}$ is the representation generated through GAE to the node v_i. With the fused and new representations, we can apply one-class learning (OCL). We present OCL in the next section.

4.2 One-Class Learning

After obtaining a fused representation, we can apply one-class learning algorithms to classify interest instances. We use the One-Class Support Vector Machine (OCSVM) that is based on the Support Vector Machine (Schölkopf *et al.*, 2001). The Binary SVM aims to generate a hyperplane of maximum separation margin between the two classes. In the OCSVM, the algorithm generates fictitious instances close to the origin corresponding to the interest class's counterexamples to apply a maximum separation hyperplane (Schölkopf *et al.*, 2001). Formally, OCSVM uses Equation 11 to create the maximum separation hyperplane between the interest class and the origin instances,

$$\min_{c,\varepsilon,\rho} \frac{1}{2} \parallel c \parallel^2 + \frac{1}{\nu \cdot |\boldsymbol{\Lambda_{int}}|} \sum_{\boldsymbol{\lambda_i} \in \boldsymbol{\Lambda_{int}}} \varepsilon_{\boldsymbol{\lambda_i}} - \rho, \tag{11}$$

subject to:

$$(c \cdot \varphi(\boldsymbol{\lambda_i})) \geq \rho - \varepsilon_{\boldsymbol{\lambda_i}}, \varepsilon_{\boldsymbol{\lambda_i}} \geq 0, \tag{12}$$

in which $\boldsymbol{\Lambda_{int}}$ is a set of fused representations of interest, c are the coefficients of the separation hyperplane, $\nu \in [0, 1)$ is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors, $\varepsilon_{\boldsymbol{\lambda_i}}$ is the distance from a node $\boldsymbol{\lambda_i}$ to the separation hyperplane, ρ is the classification error threshold, and $\varphi(\boldsymbol{\lambda_i})$ is a kernel function to map the node into a linearly separable space. After creating the hyperplane, the function $f(\boldsymbol{\lambda_i})$ indicates if node $\boldsymbol{z}_i$ belongs to the interest class, returning $+1$ (the interest side of the hyperplane) or -1 (the origin side of the hyperplane). The function $f(\boldsymbol{\lambda_i})$ is given by Equation 13,

$$f(\boldsymbol{\lambda_i}) = sgn(c \cdot \varphi(\boldsymbol{\lambda_i}) - \rho), \tag{13}$$

in which $sgn()$ is a signal function that returns -1 when $c \cdot \varphi(\boldsymbol{\lambda_i}) - \rho$ is negative and returns $+1$ when greater than or equal to 0.

5 Experimental Evaluation

In the experimental evaluation, we propose to compare seven early fusion operators and the non-use of operators. We used the OCSVM algorithm to compare the operators. Our goal is to demonstrate that the fusion of node representations through other operators outperforms concatenation and the non-use of operators, which are commonly used in the literature for one-class learning on heterogeneous graphs. The next sections present the datasets used in the experimental evaluation, experimental settings, results, and discussion.

5.1 Datasets

We use four datasets to evaluate our proposal. The first is a fake news dataset commonly used in one-class studies (Gôlo *et al.*, 2023b). The dataset name is Fact Checked News (FCN)[2]. Interest instances are fake news, and outliers are real news. The second dataset is a recommender system dataset for movies used and enriched by (Gôlo *et al.*, 2022). In this dataset, interest instances are relations between users and items with a rating of five, and outliers are relations with a rating one[3]. The third is a hit song prediction dataset collected by (da Silva *et al.*, 2022)[4], in which interest nodes are hit songs and outliers are other songs. Finally, the fourth dataset is an event dataset used in the article (Mattos and Marcacini, 2021)[5]. The dataset name is GoldStd, a 5W1H event dataset generated from news text. The dataset has 13 classes related

[2] Source: `https://github.com/GoloMarcos/FKTC.git`.

[3] Source: `https://github.com/GoloMarcos/One-Class-Recommendation-GNN-LinkPrediciton.git`.

[4] Source: https://github.com/AngeloMendes/Unsupervised-Heterogeneous-Graph-Neural-Network-for-Hit-Song-Prediction-through-One-Class-Learning.git

[5] Source: `https://github.com/joaopedromattos/GNEE`.

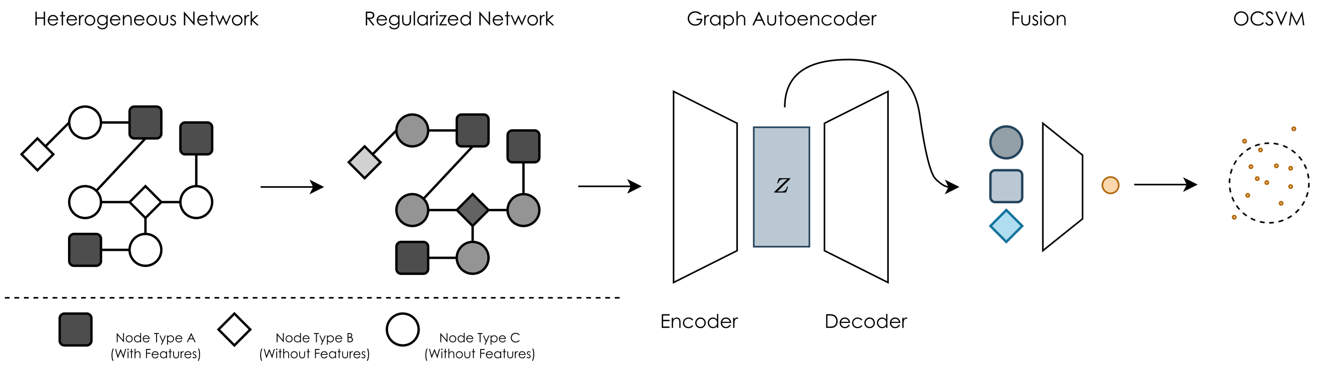
Heterogeneous Network
Regularized Network
Graph Autoencoder
Fusion
OCSVM
Z
Encoder
Decoder
Node Type A (With Features)
Node Type B (Without Features)
Node Type C (Without Features)

Figure 1. Proposed pipeline with five steps for early fusion for one-class learning on heterogeneous graph neural networks.

to the news. We use as criteria to choose the interest class the number of class instances. In this sense, we use interest class, the class with more instances. Outliers are the other 12 classes' instances.

We model the fake news dataset with a bipartite graph, in which the nodes are documents and terms (words). Edges are relations between documents and terms, i.e., if the document has the term, we add an edge. We model the recommender systems datasets with a heterogeneous graph with five nodes (users, items, keywords, genre, and review) and four edges (user-item, keyword-item, genre-item, and review-item) (Gôlo *et al.*, 2022). For the hit song dataset, we have artists and song nodes. The songs and artists are directly related, while pre-annotated data give the relations between artists (da Silva *et al.*, 2022). In this sense, we have edges between artists and artists and songs. Finally, we model the event dataset through a heterogeneous graph with seven nodes (event, what, who, when, where, why, how, IPTC code, and cluster code), and each event has edges with the nodes what, who, when, where, why, and how. Cluster nodes are to keep the graph connected, considering event nodes (Mattos and Marcacini, 2021). IPTC codes are nodes related to the event topic considering the media topics extracted from the International Press Telecommunications Council (IPTC). Thus, we also have edges between event and cluster nodes and event and IPTC nodes. Table 1 shows the datasets synthesis.

Table 1. Number of nodes, edges, and nodes with initial features for all datasets.

Datasets	$\lvert V \rvert$	$\lvert E \rvert$	$\lvert V_F \rvert$
Fake News	10348	318411	2064
Rec. Sys.	8774	30471	2397
Music	2125	5243	1529
Event	579	803	96

5.2 Experimental Settings

It is important to emphasize that we decided to use a new experimental setting to standardize the experimental evaluation in these four datasets. Therefore, even using the same datasets from (da Silva *et al.*, 2022) and (Gôlo *et al.*, 2022), we decided not to use the same experimental configurations of the one-class graph studies. We represent one node type in each dataset to create the V_F set and perform the regularization. After regularization, all nodes have a feature vector. The main nodes for all our datasets have textual content (text news, events description, music lyrics, and items overview (movies)).

To represent the textual contents, we use variations of the pre-trained model Bidirectional Encoder From Transformers (BERT) (Devlin *et al.*, 2019) since this model obtained state-of-the-art results for textual data (Otter *et al.*, 2020). BERT is a pre-trained neural network based on transformer architecture. This architecture has attention mechanics that focus on the main words in the sentence (Vaswani *et al.*, 2017). (Devlin *et al.*, 2019) trains the BERT model in a large textual corpus that represents sentences based on their context and outperforms other natural language pre-processing models in different tasks and languages (Otter *et al.*, 2020). BERT can extract semantic and syntactic characteristics from the text generating dynamic embeddings (Otter *et al.*, 2020).

We chose BERT variations according to each dataset. We used the multilingual BERT model for fake news because the news are in Portuguese[6]. We also use this model to represent events. For the songs, we used pre-trained BERT on lyrics[7]. Finally, we represent the overviews and reviews of the movies from the recommendation dataset with the *all-MiniLM-L6-v2* model that obtained the highest number of downloads considering the sentence similarity task.

For the GAE, Early Fusion, and One-Class Support Vector Machines, we have the following parameters:

- **Heterogeneous Early Fusion:** operators = {addition, subtraction, average, minimum, maximum, multiplication, and concatenation}.
- **GAE**: layers = {GCN, SAGE, GAT}, layer sizes = $\{[32], [64], [32, 32], [64, 64]\}$, patience = $\{50, 100\}$, activation functions = $\{relu\}$, and learning rates = $\{1^{-2}, 1^{-3}, 1^{-4}, \}$. For the SAGE layers we vary the aggregation functions = {mean, max-pooling} while for the GAT layers the number of heads = $\{4, 8\}$;
- **OCSVM**: kernel = {rbf, poly, sigmoid, linear}, $\nu = \{0.05 * a, 0.005 * a\}, a \in [1..19]$, and $\gamma = \{\frac{1}{n}\}, \frac{1}{(n \cdot o)}\}$,

[6] `https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1`

[7] `https://huggingface.co/juliensimon/autonlp-song-lyrics-18753417`

in which n is the dimension of the input data and o is the variance of the representations.

We use the procedure 5-Fold Cross-Validation in the one-class learning stage, i.e., when applying the OCSVM in the classification step of the pipeline. In this procedure, we apply a 5-Fold Cross-Validation considering only the interest class for each dataset. The procedure consists of dividing the interest class into folds and using 4 folds to train and the remaining fold to test iteratively. We also added the not-interest set to the test set. Finally, we use the macro f_1-score as the evaluation measure. f_1-macro is the arithmetic average between the classes. We present f_1-score in Equation 14,

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \tag{14}$$

$$Precision = \frac{TP}{TP + FP}, \tag{15}$$

$$Recall = \frac{TP}{TP + FN}, \tag{16}$$

in which TP (True Positives) is the number of positive instances that the algorithm has correctly classified; TN (True Negatives) is the number of negative instances that the algorithm has correctly classified; FP (False Positives) is the number of negative instances that have been classified as positive; and FN (False Negatives) is the number of positive instances classified as negative.

5.3 Results and Discussion

Tables 2, 3, 4, and 5 present the experimental evaluation results considering the regularized representations of the heterogeneous graph (REG) and the representations learned by the Graph Autoencoder (GAE) considering the GCN, GAT and SAGE layers for each of the four datasets. We present the values of f_1-macro and the respective standard deviation. Each column presents the result referring to an early fusion operator or the non-use of the operator (Without). Each line represents the results for the regularized representations and the representations learned by GCN, GAT, and SAGE layers. We bold the best results. In cases of ties, we highlight better results considering the smallest standard deviation.

5.4 Does the early fusion of the regularized or GNN representations improve the performance of one-class tasks?

The subtraction operator obtained the highest values of f_1-macro in the music and fakenews datasets. The minimum and multiply operators obtained the highest values of f_1-macro in the event dataset. On the other hand, the non-use of operator obtained the highest values of f_1-macro in the recommender systems dataset. Thus, in general, the use of early fusion operators improve the performance of one-class tasks. Even though not using fusion operators in the recommendation dataset generates better results in most representations, the concatenation operator obtain the highest $f1$-macro in this dataset.

Concatenation is the most used early fusion operator in the literature. Few studies (Beserra *et al.*, 2020; Beserra, 2022; Beserra and Goularte, 2023) use operators such as the ones presented in this research, and even fewer use operators to fuse features at a medium semantic level. However, the concatenation, in addition to increasing the dimensionality of the generated vector (doubling in the case of two modalities, tripling in the case of three, and so on.), and increasing the cost of the algorithm, did not obtain the best results, with the exception of one scenario. On the other hand, the other operators, in addition to obtaining the best results, are also space efficient since they generate a new representation with the same modalities' dimensions.

We can observe that early fusion does not benefit the classification performance in the recommender systems dataset. We believe that two factors together benefited the non-use of early fusion operators. The first factor is the number of modalities. The recommendation systems dataset has five modalities (number of nodes) in total (users, items, keywords, genre, and review), and the greater the number of modalities, the more challenging it is to combine these modalities. This was not the only factor since the event dataset has many modalities, and early fusion benefits the classification performance. This guides us to the second factor, which is a characteristic that differentiates these two datasets: how the classification is carried out. In the event dataset, we classify nodes of interest. On the other hand, in the recommendation systems dataset, we classified interactions between users and items, which did not benefit from using early fusion operators. It is worth mentioning that we carried out the early fusion of without weights in each modality, which may have resulted in the non-benefit of the use of early fusion operators.

5.5 Which early fusion operator generates better representations for one-class problems?

For the music dataset considering the regularized representations, the multiplication operator performed better than the others. For GCN and GAT layers on GAE, the subtraction operator outperformed the other operators. Finally, in the SAGE layer, the multiplication operator obtain the highest $f1$-macro. The minimum, without, without, and minimum obtained the worst results considering REG, GCN, GAT and SAGE representations, respectively.

For the fakenews dataset considering the regularized representations and GCN layer, the subtraction operator performed better than the others. For GAT layer, the maximum operator outperformed the other operators. Finally, in the SAGE layer, the multiplication operator obtain the highest $f1$-macro. The maximum, without, minimum, and minimum obtained the worst results considering REG, GCN, GAT and SAGE representations, respectively.

The average, minimum and multiply operators in the events dataset presented the best results for the regularized representations and those generated by GAE. We note that the GAT representations generate best results for various operators, but with smaller $f1$-macro than the GCN and SAGE

Table 2. Results for the seven operators considering f_1-macro in the **Music** dataset. We show results for regularized representations and graph autoencoder representations with GCN, GAT, and SAGE layers. Bold values indicate the best results.

Graph	**Without**	**Addition**	**Subtract**	**Average**	**Minimum**	**Maximum**	**Multiply**	**Concat**
REG	0.533 ± 0.016	0.530 ± 0.012	0.529 ± 0.015	0.530 ± 0.012	0.525 ± 0.018	0.534 ± 0.012	**0.557 ± 0.013**	0.532 ± 0.017
GGN	0.561 ± 0.011	0.586 ± 0.013	**0.591 ± 0.013**	0.586 ± 0.013	0.565 ± 0.006	0.582 ± 0.014	0.577 ± 0.013	0.591 ± 0.015
GAT	0,536 ± 0,011	0,567 ± 0,007	**0,568 ± 0,014**	0,567 ± 0,007	0,538 ± 0,023	0,559 ± 0,009	0,541 ± 0,018	0,559 ± 0,008
SAGE	0,557 ± 0,017	0,581 ± 0,023	0,572 ± 0,021	**0,581 ± 0,022**	0,553 ± 0,01	0,580 ± 0,013	0,562 ± 0,008	0,575 ± 0,018

Table 3. Results for the seven operators considering f_1-macro in the **Fake News** dataset. We show results for regularized representations and graph autoencoder representations with GCN, GAT, and SAGE layers. Bold values indicate the best results.

Graph	**Without**	**Addition**	**Subtract**	**Average**	**Minimum**	**Maximum**	**Multiply**	**Concat**
REG	0.913 ± 0.006	0.913 ± 0.006	**0.914 ± 0.009**	0.913 ± 0.006	0.879 ± 0.007	0.771 ± 0.012	0.781 ± 0.007	0.913 ± 0.006
GCN	0.618 ± 0.018	0.790 ± 0.012	**0.895 ± 0.007**	0.790 ± 0.012	0.619 ± 0.012	0.782 ± 0.015	0.788 ± 0.017	0.781 ± 0.015
GAT	0,909 ± 0,007	0,918 ± 0,006	0,908 ± 0,007	0,918 ± 0,006	0,894 ± 0,009	**0,918 ± 0,005**	0,903 ± 0,005	0,908 ± 0,007
SAGE	0,924 ± 0,004	0,929 ± 0,004	0,936 ± 0,005	0,929 ± 0,004	0,922 ± 0,005	0,926 ± 0,006	**0,943 ± 0,005**	0,923 ± 0,007

Table 4. Results for the seven operators considering f_1-macro in the **Event** dataset. We show results for regularized representations and graph autoencoder representations with GCN, GAT, and SAGE layers. Bold values indicate the best results.

Graph	**Without**	**Addition**	**Subtract**	**Average**	**Minimum**	**Maximum**	**Multiply**	**Concat**
REG	0.860 ± 0.085	0.814 ± 0.188	0.881 ± 0.102	**0.891 ± 0.062**	0.864 ± 0.116	0.759 ± 0.141	0.763 ± 0.164	0.881 ± 0.102
GCN	0.979 ± 0.041	0.979 ± 0.041	0.945 ± 0.070	0.979 ± 0.041	**1.000 ± 0.000**	0.979 ± 0.041	0.979 ± 0.041	0.949 ± 0.102
GAT	**0,979 ± 0,041**	**0,979 ± 0,041**	**0,979 ± 0,041**	**0,979 ± 0,041**	0,966 ± 0,068	**0,979 ± 0,041**	**0,979 ± 0,041**	**0,979 ± 0,041**
SAGE	0,979 ± 0,041	0,979 ± 0,041	0,959 ± 0,050	0,979 ± 0,041	0,979 ± 0,041	0,979 ± 0,041	**1.000 ± 0.000**	0,979 ± 0,041

Table 5. Results for the seven operators considering f_1-macro in the **Rec. Sys.** dataset. We show results for regularized representations and graph autoencoder representations with GCN, GAT, and SAGE layers. Bold values indicate the best results.

Graph	**Without**	**Addition**	**Subtract**	**Average**	**Minimum**	**Maximum**	**Multiply**	**Concat**
REG	**0.642 ± 0.004**	0.589 ± 0.006	0.555 ± 0.002	0.589 ± 0.006	0.504 ± 0.008	0.533 ± 0.002	0.595 ± 0.004	0.590 ± 0.005
GCN	0.689 ± 0.004	0.684 ± 0.008	0.602 ± 0.006	0.684 ± 0.008	0.653 ± 0.010	0.668 ± 0.004	0.616 ± 0.021	**0.694 ± 0.004**
GAT	**0,693 ± 0,006**	0,685 ± 0,005	0,678 ± 0,009	0,685 ± 0,005	0,638 ± 0,015	0,673 ± 0,016	0,647 ± 0,021	0,679 ± 0,013
SAGE	**0,650 ± 0,006**	0,642 ± 0,016	0,588 ± 0,008	0,641 ± 0,015	0,603 ± 0,024	0,572 ± 0,026	0,623 ± 0,012	0,612 ± 0,003

layers. The maximum, subtract, minimum, and subtract obtained the worst results considering REG, GCN, GAT and SAGE representations, respectively.

In the recommendation results, for regularized representations, GAT and SAGE, the non-use of operators improves the performance. However, for GCN, concatenating the representations generate the best results. The manimum, subtract, minimum, and maximum obtained the worst results considering REG, GCN, GAT and SAGE representations, respectively.

Generally, the subtraction operator generates better representations for one-class problems. On the other hand, we highlight some interesting particularities in the best results when comparing operators. In the fake news dataset, subtracting the terms representation from the document representation differentiated the document representations to improve the classification performance. When investigating the collection procedure of this dataset, we observe that the authors collect the dataset on the fake and real news of the same topic (politics). Therefore, when modeling the heterogeneous graph as a bipartite graph of terms and documents, documents of different classes will share words and subtract these shared representations, removing redundant information between documents of different classes, which improves the classification. In addition, the event dataset has few main nodes (see Table 1) in relation to other node types, and therefore some operators obtain the same results. This fact also influenced the result of f_1 macro 1 for all folds in the minimum and multuply operator.

The recommendation dataset was the only one with the best result considering the non-use of operators. We obtain these results in the regularized representation, GAT and SAGE layers. However, we the have the best result is this dataset with the concatenation operator in the learned GCN representations. This dataset also was the only one that obtained the best result with the concatenation operator. Thus, it is interesting to highlight that depending on the representation you explore, it may not be worth using an operator or using a more costly one such as concatenation.

In addition to the analysis of which fusion operators obtain higher f1-scores for each dataset, we carried out a general analysis of all operators in all scenarios. We provide a statistical significance analysis for the results. We show the critical difference diagram proposed by Demšar (2006). First, the Friedman test is performed to reject the null hypothesis, and then we proceed with a posthoc analysis based on the Wilcoxon-Holm method to generate the average ranking and the critical difference. Figure 2 presents the result of the

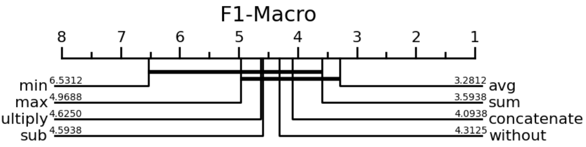

Figure 2. Friedman test with posthoc analysis based on the Wilcoxon-Holm through the critical difference diagram for all operators in all scenarios.

Friedman test with posthoc analysis based on the Wilcoxon-Holm through the critical difference diagram Ismail Fawaz *et al*. (2019). The diagram presents the methods' average rankings. Methods connected by a line do not present statistically significant differences between them. Average obtains the best average ranking, followed by Sum, Concatenation, Without, and Subtraction operators. Min, max, and Multiplication obtain the worst average rankings. Furthermore, the average operator obtained a statistically significant difference from the minimum operator.

5.6 What is the best representation obtained through graphs to apply early fusion in one-class tasks?

Representation learning through GAE generated better results in all datasets. Furthermore, most of the time, regardless of the chosen fusion operator, GAE representations generate better f_1-macro. We note exceptions in the fake news dataset, in which the GCN and GAT representation generated f_1-macro smaller than the regularized representation in some operators. An indication of these results is the type of dataset used. This dataset has fake and real news that is well-behaved in context, type, topic, and veracity. Therefore, good initial representations of fake news (BERT) already solve the separation of classes very well, as shown in the results of other studies of fake news detection through one-class learning that uses this dataset (de Souza *et al*., 2021; Gôlo *et al*., 2021; de Souza *et al*., 2022; Gôlo *et al*., 2023b). Thus, with a very robust initial representation, the regularization already adds enough information to generate good representations for the terms of the bipartite graph and does not need the graph autoencoder.

In addition to comparing the operators' performances, we performed another experiment to analyze the representations generated by the fused representation. Figures 3, 4, and 5 present two-dimensional projections of the fused representation considering each operator in the fake news dataset with the graph autoencoder representations. We choose this dataset because, in this scenario, the operators have the highest $f1$-macro and $|V_f|$. We generated the representations using the t-Distributed Stochastic Neighbor Embedding (t-SNE) for the analysis (Van der Maaten and Hinton, 2008).

In the TSNE results for the GCN layer, the non-use of fusion and minimum operators obtain the worst visual results. On the other hand, the other operators performed the separation of classes satisfactorily, showing good visual results. In operators with good results, we noticed that there are few real news closer to fake news and far from the real news region. In addition, we observe a few real news grouped in the fake news right-bottom region that could be the differential between operators for better or worse results, i.e., how operators represent this news group directly impacts their performance. We can observe this fact in the two-dimensional projection of the concatenation, addition, and average operators that project this real news in a smaller region than the subtraction operator, and the subtraction operator obtained the best f_1-macro result.

In the TSNE results for the GAT and SAGE layers, we note a different behavior. All the operators performed the separation of classes satisfactorily, showing good visual results. We highlight that these layers outperform the GCN layer and do not show a few real news grouped in the fake news right-bottom such as the GCN layer. This difference may be what made the one-class learning model obtain higher values of f_1. Therefore, we emphasize that adding attention to the edges and sampling the neighboring nodes that will be aggregated, improved the representation learning through graph neural networks to detect fake news through one-class learning.

We also analyzed GCN and OCSVM best parameters for the early fusion scenario for data modeled through heterogeneous graphs to solve one-class tasks. We present the best parameters to indicate that better parameters should be used in future studies for one-class learning and heterogeneous graphs. We highlight the GAE architecture with one layer containing 32 neurons, the learning rate 1^{-4}, and the patience 100. For the OCSVM, the polynomial and sigmoid kernels obtained the best results. Regarding ν, smaller values between 0.05 and 0.15 were better for the sigmoid kernel, while values between 0.40 and 0.65 were better for the polynomial kernel. Finally, $\gamma = \frac{1}{n \cdot o}$ gave the best results in most scenarios. Notably, the projected representations have curved separation, which indicates the advantage of the polynomial kernel over the others.

5.7 Which Graph Neural Network layer obtains better representations for one-class tasks considering unsupervised representation learning?

Figure 6 present the GCN, GAT and SAGE layers best results for each dataset independent of the operator, i.e., the best result of each layer without considering the same operator in the comparison. In general, GCN obtain the best values of $f1$ and GAT obtain the worst values. In the music and recommender systems datasets GCN outperformed the other layers. SAGE outperformed the other layers to detect fake news. On the other hand, we have a tie for GCN and SAGE layers in the event dataset.

Once again, the particularity of the the fake news dataset graph modeling influences the best result. The SAGE layer performs sampling when aggregating information, i.e., considering the bipartite modeling of documents and terms, in the sampled aggregation, only a portion of the terms will be selected, which improved representation learning and consequently the one-class learning to detect fake news. For the other one-class tasks with others graph modeling, we indicate the simple and tradicional GCN layer.

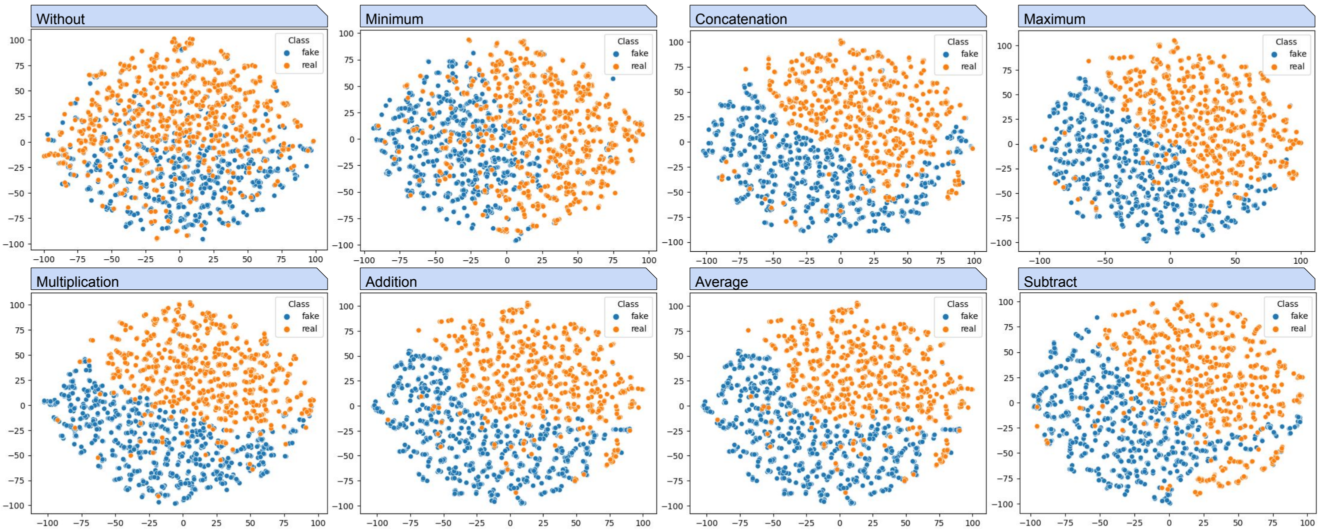

Figure 3. Two-dimensional projections (t-SNE) of each fused representation considering each operator and the non-use of an operator in the fake news dataset for the GCN layer. The colors indicate class real news (orange) and fake news (blue). Operators that show less overlap between classes are more promising for one-class learning.

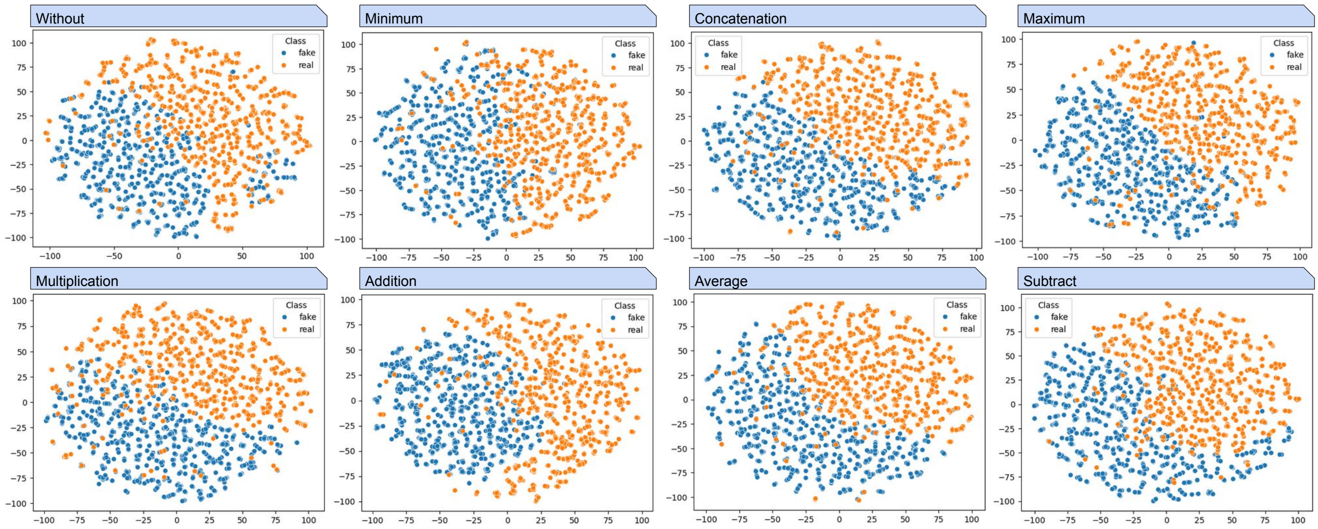

Figure 4. Two-dimensional projections (t-SNE) of each fused representation considering each operator and the non-use of an operator in the fake news dataset **for the GAT layer**. The colors indicate class real news (orange) and fake news (blue). Operators that show less overlap between classes are more promising for one-class learning.

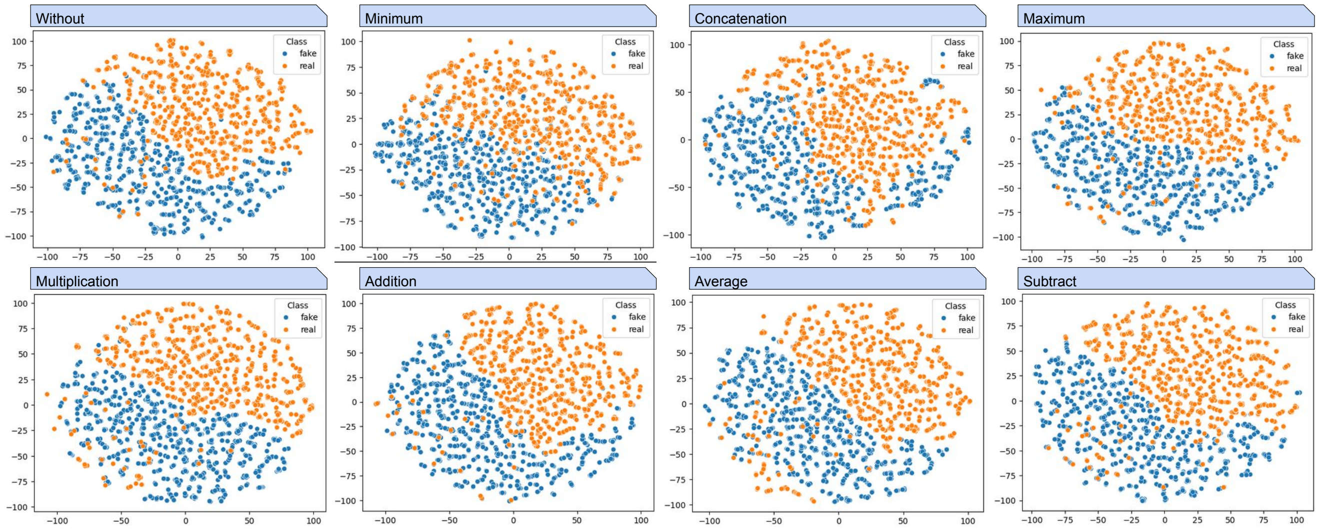

Figure 5. Two-dimensional projections (t-SNE) of each fused representation considering each operator and the non-use of an operator in the fake news dataset **for the SAGE layer**. The colors indicate class real news (orange) and fake news (blue). Operators that show less overlap between classes are more promising for one-class learning.

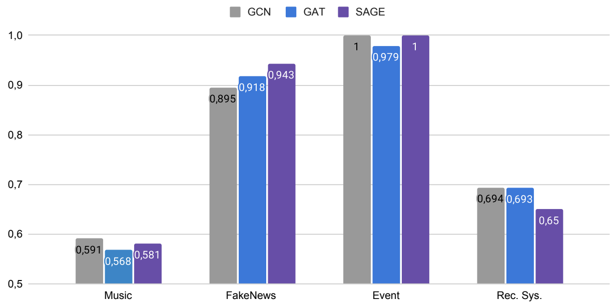

Figure 6. GCN, GAT and SAGE best results for each dataset.

6 Conclusions and Future Work

In this article, we aim to answer some research questions and significantly contribute to data modeled through heterogeneous graphs in one-class tasks. To answer the research questions ”Does pre-merging the representations regularized and learned by GNN improve the performance of one-class tasks?”, ”Which pre-merging operator generates better representations to solve one-class problems?”, and ”What is the best representation obtained through graphs to apply pre-merging in one-class tasks?”, we propose the use of a graph neural networks method considering different early fusion operators in four different one-class tasks. Our objective was to compare the performance of seven fusion operators, evaluate the impact of using the operators, and evaluate the impact of using graph neural networks in these scenarios.

The results presented by the study showed that the early fusion of the regularized and learned representations by GCN, GAT, and SAGE improved the performance of one-class learning in the four datasets modeled through heterogeneous graphs. The representations learned through the GCN and SAGE obtained better results. However, GCN representations obtained better results in most datasets. In twelve of sixteen scenarios, fusion operators had a positive impact, improving the classification performance in the four datasets used. We highlight the average, addition, and subtraction operators as the best early fusion operators for one-class tasks in which data is modeled using heterogeneous graphs. On the other hand, we highlight the non-use of operators to recommend interest movies.

In future work, we intend to propose a GNN that learns the representations for the different types of nodes while learning to combine the modalities biased by some task. In this sense, we intend to explore the one-class graph neural networks Wang *et al.* (2021), considering a heterogeneous version of the data in which the method will learn how to combine the heterogeneous data into a single fused representation through neurons. We intend to explore this pipeline in one-class edge classification in the homogeneous and heterogeneous scenarios. Furthermore, we intend to explore weights in the early fusion.

Declarations

Acknowledgements

The authors of this work would like to thank the Center for Artificial Intelligence (C4AI-USP) and the support from the São Paulo Research Foundation (FAPESP grant 2019/07665-4) and from the IBM Corporation.

Funding

This work was supported by *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) grant numbers 88887.671481/2022-00 and 88887.710875/2022-00. Also, this work was supported by LATAM Google Ph.D. Fellowship.

Authors’ Contributions

- **Conceptualization:** Marcos Gôlo, Marcelo Moraes, Rudinei Goularte and Ricardo Marcacini
- **Data curation:** Marcos Gôlo
- **Formal Analysis:** Marcos Gôlo and Marcelo Moraes
- **Funding acquisition:** Marcos Gôlo, Marcelo Moraes and Ricardo Marcacini
- **Investigation:** Marcos Gôlo, Marcelo Moraes, Rudinei Goularte and Ricardo Marcacini
- **Methodology:** Marcos Gôlo, Marcelo Moraes, Rudinei Goularte and Ricardo Marcacini
- **Project administration:** Marcos Gôlo and Ricardo Marcacini
- **Resources:** Ricardo Marcacini
- **Software:** Marcos Gôlo and Marcelo Moraes
- **Supervision:** Rudinei Goularte and Ricardo Marcacini
- **Validation:** Marcos Gôlo, Marcelo Moraes, Rudinei Goularte and Ricardo Marcacini
- **Visualization:** Marcos Gôlo, Marcelo Moraes, Rudinei Goularte and Ricardo Marcacini
- **Writing – original draft:** Marcos Gôlo and Marcelo Moraes
- **Writing – review & editing:** Rudinei Goularte and Ricardo Marcacini

Competing interests

The authors declare that they do not have competing interests.

Availability of data and materials

We made available all source codes and datasets for reproducibility reasons[8].

References

Alam, S., Sonbhadra, S. K., Agarwal, S., and Nagabhushan, P. (2020). One-class support vector classifiers: A survey. *Knowledge-Based Systems*, 196:105754. DOI: https://doi.org/10.1016/j.knosys.2020.105754.

Atrey, P. K., Hossain, M. A., El Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16:345–379. DOI: https://doi.org/10.1007/s00530-010-0182-0.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443. DOI: https://doi.org/10.1109/TPAMI.2018.2798607.

[8] `https://github.com/GoloMarcos/JIS-2024-HGNN-OCL.git`.

Beserra, A. A. and Goularte, R. (2023). Multimodal early fusion operators for temporal video scene segmentation tasks. *Multimedia Tools and Applications*, 82:1–18. DOI: https://doi.org/10.1007/s11042-023-14953-6.

Beserra, A. A., Kishi, R. M., and Goularte, R. (2020). Evaluating early fusion operators at mid-level feature space. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pages 113–120, online. ACM. DOI: https://doi.org/10.1145/3428658.3431079.

Beserra, A. A. R. (2022). Operadores de fusão prévia para segmentação temporal de vídeo em cenas. Master's thesis, Universidade de São Paulo.

Brody, S., Alon, U., and Yahav, E. (2022). How attentive are graph attention networks? In *International Conference on Learning Representations*. DOI: https://openreview.net/forum?id=F72ximsx7C1.

da Silva, A., Gôlo, M., and Marcacini, R. (2022). Unsupervised heterogeneous graph neural network for hit song prediction through one class learning. In *10th Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, pages –, Campinas, SP, Brazil. SBC. DOI: https://doi.org/10.5753/kdmile.2022.227954.

de Souza, M. C., Nogueira, B. M., Rossi, R. G., Marcacini, R. M., Dos Santos, B. N., and Rezende, S. O. (2022). A network-based positive and unlabeled learning approach for fake news detection. *Machine Learning*, 111(10):3549–3592. DOI: https://doi.org/10.1007/s10994-021-06111-6.

de Souza, M. C., Nogueira, B. M., Rossi, R. G., Marcacini, R. M., and Rezende, S. O. (2021). A heterogeneous network-based positive and unlabeled learning approach to detect fake news. In *Intelligent Systems: 10th Brazilian Conference, BRACIS 2021, Virtual Event, November 29–December 3, 2021, Proceedings, Part II*, pages 3–18, online. Springer. DOI: https://doi.org/10.1007/978-3-030-91699-2_1.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30. DOI: http://jmlr.org/papers/v7/demsar06a.html.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL 2019: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minnesota. Association for Computational Linguistics. DOI: https://doi.org/10.18653/v1/N19-1423.

do Carmo, P. and Marcacini, R. (2021). Embedding propagation over heterogeneous event networks for link prediction. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 4812–4821, online. IEEE. DOI: https://doi.org/10.1109/BigData52589.2021.9671645.

Emmert-Streib, F. and Dehmer, M. (2022). Taxonomy of machine learning paradigms: A data-centric perspective. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(5):e1470. DOI: https://doi.org/10.1002/widm.1470.

Ganz, T., Ashraf, I., Härterich, M., and Rieck, K. (2023). Detecting backdoors in collaboration graphs of software repositories. In *Proceedings of the Thirteenth Conference on Data and Application Security and Privacy*, pages 189–200, Charlotte, NC, USA. ACM. DOI: https://doi.org/10.1145/3577923.3583657.

Gôlo, M., Caravanti, M., Rossi, R., Rezende, S., Nogueira, B., and Marcacini, R. (2021). Learning textual representations from multiple modalities to detect fake news through one-class learning. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pages 197–204, Online. ACM. DOI: https://doi.org/10.1145/3470482.3479634.

Gôlo, M. P. S., De Moraes, M. I., Goularte, R., and Marcacini, R. M. (2023a). On the use of early fusion operators on heterogeneous graph neural networks for one-class learning. In *Proceedings of the 29th Brazilian Symposium on Multimedia and the Web*, pages 128–136. DOI: https://doi.org/10.1145/3617023.3617041.

Gôlo, M. P. S., de Souza, M. C., Rossi, R. G., Rezende, S. O., Nogueira, B. M., and Marcacini, R. M. (2023b). One-class learning for fake news detection through multimodal variational autoencoders. *Engineering Applications of Artificial Intelligence*, 122:106088. DOI: https://doi.org/10.1016/j.engappai.2023.106088.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., and He, Q. (2020). A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568. DOI: https://doi.org/10.1109/TKDE.2020.3028705.

Guo, W., Wang, J., and Wang, S. (2019). Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394. DOI: https://doi.org/10.1109/ACCESS.2019.2916887.

Gôlo, M., Moraes, L., Goularte, R., and Marcacini, R. (2022). One-class recommendation through unsupervised graph neural networks for link prediction. In *10th Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, pages –, campinas, SP, Brazil. SBC. DOI: https://doi.org/10.5753/kdmile.2022.227810.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems*, volume 30. DOI: https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

Huang, Z., Gu, Y., and Zhao, Q. (2022). One-class directed heterogeneous graph neural network for intrusion detection. In *6th International Conference on Innovation in Artificial Intelligence (ICIAI)*, pages 178–184, Guangzhou, China. ACM. DOI: https://doi.org/10.1145/3529466.3529480.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963. DOI: https://doi.org/10.1007/s10618-019-00619-1.

Jakob, P., Madan, M., Schmid-Schirling, T., and Valada, A. (2021). Multi-perspective anomaly detection. *Sensors*, 21(16):5311. DOI: https://doi.org/10.3390/s21165311.

Khan, S. S. and Madden, M. G. (2014). One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374. DOI: https://doi.org/10.1017/S026988891300043X.

Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*, pages 1–3, Barcelona, Spain. NIPS. DOI: http://bayesiandeeplearning.org/2016/papers/BDL_16.pdf.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, pages 1–14, Toulon, France. OpenReview. DOI: https://openreview.net/forum?id=SJU4ayYgl.

Kumar, A., Kim, J., Cai, W., Fulham, M., and Feng, D. (2013). Content-based medical image retrieval: a survey of applications to multidimensional and multimodality data. *Journal of digital imaging*, 26:1025–1039. DOI: https://doi.org/10.1007/s10278-013-9619-2.

Liu, X., Gao, F., Zhang, Q., and Zhao, H. (2019). Graph convolution for multimodal information extraction from visually rich documents. In *Proceedings of NAACL-HLT*, pages 32–39, Minneapolis, Minnesota. Association for Computational Linguistics. DOI: http://dx.doi.org/10.18653/v1/N19-2005.

Mattos, J. P. R. and Marcacini, R. M. (2021). Semi-supervised graph attention networks for event representation learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1234–1239, online. IEEE. DOI: https://doi.org/10.1109/ICDM51629.2021.00150.

Nguyen, T. and Grishman, R. (2018). Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 5900–5907, Vancouver, Canada. AAAI. DOI: https://doi.org/10.1609/aaai.v32i1.12039.

Otter, D., Medina, J., and Kalita, J. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624. DOI: https://doi.org/10.1109/TNNLS.2020.2979670.

Rahman, M. S. (2017). *Basic graph theory*, volume 9. Springer, online. DOI: https://doi.org/10.1007/978-3-319-49475-3.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018). Deep one-class classification. In *International Conference on Machine Learning (ICML)*, pages 4393–4402, Stockholm, SWEDEN. PMLR. DOI: https://proceedings.mlr.press/v80/ruff18a.html.

Schinas, M., Papadopoulos, S., Petkos, G., Kompatsiaris, Y., and Mitkas, P. A. (2015). Multimodal graph-based event detection and summarization in social media streams. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 189–192, Brisbane, Australia. ACM. DOI: https://doi.org/10.1145/2733373.2809933.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471. DOI: https://doi.org/10.1162/089976601750264965.

Tax, D. M. J. (2001). *One-class classification: Concept learning in the absence of counter-examples*. PhD thesis, Technische Universiteit Delft.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11):2579–2605. DOI: http://jmlr.org/papers/v9/vandermaaten08a.html.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30:1–12. DOI: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). Graph attention networks. *stat*, 1050:4. DOI: https://openreview.net/forum?id=rJXMpikCZ.

Wang, X., Bo, D., Shi, C., Fan, S., Ye, Y., and Philip, S. Y. (2022). A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*, 9:415 – 436. DOI: https://doi.org/10.1109/TBDATA.2022.3177455.

Wang, X., Jin, B., Du, Y., Cui, P., Tan, Y., and Yang, Y. (2021). One-class graph neural networks for anomaly detection in attributed networks. *Neural computing and applications*, 33(18):12073–12085. DOI: https://doi.org/10.1007/s00521-021-05924-9.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24. DOI: https://doi.org/10.1109/TNNLS.2020.2978386.

Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., and Liu, H. (2021). Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127. DOI: https://doi.org/10.1109/TAI.2021.3076021.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How powerful are graph neural networks? In *International Conference on Learning Representations*, pages 1–17, New Orleans. OpenReview. DOI: https://openreview.net/forum?id=ryGs6iA5Km.

Zhou, D. and Schölkopf, B. (2004). A regularization framework for learning from graph data. In *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields (SRL 2004)*, pages 132–137, Alberta, Canada. MPG Pure. DOI: https://www.microsoft.com/en-us/research/publication/regularization-framework-learning-graph-data/.

Zhou, H. and Mao, K. (2022). Document-level event argument extraction by leveraging redundant information and closed boundary loss. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3041–3052, Seattle, Washington. ACL. DOI: http://dx.doi.org/10.18653/v1/2022.naacl-main.222.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81. DOI: https://doi.org/10.1016/j.aiopen.2021.01.001.