symmetry

MDPI

Article

Graph Embedding with Similarity Metric Learning

Tao Tao *, Qianqian Wang, Yue Ruan, Xue Li and Xiujun Wang

School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243032, China; qian2.wang@foxmail.com (Q.W.); yue_ruan@163.com (Y.R.); lixue_angel@163.com (X.L.); xjwang@ahut.edu.cn (X.W.)

* Correspondence: taotao@ahut.edu.cn

Abstract: Graph embedding transforms high-dimensional graphs into a lower-dimensional vector space while preserving their structural information and properties. Context-sensitive graph embedding, in particular, performs well in tasks such as link prediction and ranking recommendations. However, existing context-sensitive graph embeddings have limitations: they require additional information, depend on community algorithms to capture multiple contexts, or fail to capture sufficient structural information. In this paper, we propose a novel Graph Embedding with Similarity Metric Learning (GESML). The core of GESML is to learn the optimal graph structure using an attention-based symmetric similarity metric function and establish association relationships between nodes through top-k pooling. Its primary advantage lies in not requiring additional features or multiple contexts, only using the symmetric similarity metric function and pooling operations to encode sufficient topological information for each node. Experimental results on three datasets involving link prediction and node-clustering tasks demonstrate that GESML significantly improves learning for all challenging tasks relative to a state-of-the-art (SOTA) baseline.

Keywords: similarity metric learning; graph embedding; similarity metric; top-k pooling

Citation: Tao, T.; Wang, Q.; Ruan, Y.; Li, X.; Wang, X. Graph Embedding with Similarity Metric Learning. *Symmetry* **2023**, *15*, 1618. https://doi.org/10.3390/sym15081618

Academic Editors: Christos Volos and Michel Planat

Received: 25 June 2023
Revised: 9 August 2023
Accepted: 20 August 2023
Published: 21 August 2023

1. Introduction

Graph embedding is a powerful technique for representing graphs with a wide range of applications in various domains. For instance, node-clustering tasks group together similar news articles [1], link prediction tasks anticipate associations between users in social networks [2], and product recommendation tasks reveal users' preferences on shopping platforms [3]. However, most current research on graph embedding focuses on context-free methods that rely solely on node representations based on local or global features [4–6]. For example, Perozzi et al. [7] introduced the DeepWalk method, which employs random walks to capture neighborhood structures and maximizes the probability of neighboring nodes appearing in the walk sequence. Sheikh et al. [8] investigated using attributes and structures to learn node representations. Pan et al. [9] proposed a method for jointly learning optimal node representations by leveraging network structure, node content, and node labels. Importantly, in numerous cases, particularly for sparse graphs, context-free graph-embedding methods may result in the omission of crucial details, consequently diminishing the performance of network analysis tasks.

Incomplete representations of a node's single context have prompted researchers to propose a context-sensitive method known as context-sensitive graph embedding. This method captures the contexts where the nodes exist and learns distinct representations of the nodes in various scenarios, i.e., the representations of the node change according to the target task. Existing context-sensitive representation methods depend on additional textual information, the graph structure itself, and the information of the neighbors. For example, Tu et al. [10] were the first to propose a mutual attention mechanism based on textual embedding. This mechanism emphasizes the interactions between a node

and its neighbors and advocates for separate embeddings for nodes during these interactions. Gracious et al. [11] combined mutual and topological attention mechanisms and proposed a diffusion graph for text graph embedding. This graph captures the semantic relatedness between texts by leveraging the global structural information of the graph. Epasto et al. [12] encoded multiple representations of a node by leveraging its roles in distinct local communities through ego-network decomposition. Kefato et al. [13] utilized an attentional pooling network to determine the personalized importance of a node's neighbors. Qin et al. [14] proposed using GCN to assist in obtaining high-quality node representations.

We aim to generate high-quality context-sensitive node representations by focusing solely on the structural information of nodes. Our work is based on the Graph Attentional Pooling network (GAP) [13], which could view as a generalization of it. The model suggests the source node's representation using a mutual attention mechanism that responds to changes in the target node, thereby achieving context sensitivity. To describe more clearly how GAP works, we use a graph with some points and edges to illustrate how it works. For example, in Figure 1, node 5 and node 6 have first-order neighbor sets of $\{3, 4, 6, 8, 9\}$ and $\{3, 4, 5, 7\}$, respectively. During interactions between nodes 5 and 6, the model primarily focuses on the same neighbors $\{3, 4\}$ while ignoring or paying little attention to other neighbors. Although the relationship between node 6's neighbor $\{7\}$ and node 5's neighbors $\{8, 9\}$ exists, it is not considered during the model training process. Therefore, the mutual attention mechanism between nodes could result in the model missing important information necessary for achieving accurate node representation, such as implicit topological structure information.

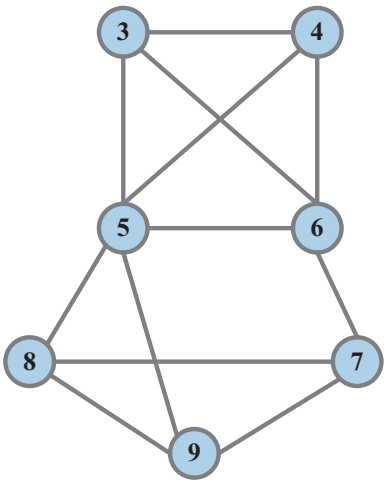

Figure 1. A graph with some edges and nodes.

We propose a novel Graph Embedding with Similarity Metric Learning (GESML), a more expressive mutual attention method to obtain more topological information about node neighborhoods. GESML employs an attention-based symmetric similarity measurement function in metric learning that can learn the similarity matrix between pairs of nodes. This method can capture the optimal graph structure and establish the relationship between node representation and the target task. GESML consists of three modules: a graph structure learning module, a graph representation module, and a prediction module. The graph structure learning module aims to obtain embedding representations of the source and target nodes by sampling from the graph and then generating the similarity matrix based on the attention-based symmetric similarity measurement function. The graph representation module encodes from the perspectives of the source node and the target node to generate the embedding representation of pairs of nodes. The prediction module calculates the soft alignment score using the embedding representation of the source and target nodes.

We achieved state-of-the-art performance in link prediction and node clustering tasks on three datasets. Furthermore, the results illustrated that introducing attention-based symmetric similarity measurement functions and the top-k pooling operation can significantly enhance the embedding ability. In summary, our contributions are two-fold:

- We propose a simple embedding representation method that combines an attention-based symmetric similarity measurement function and top-k pooling.
- Experimental results on three datasets illustrate that GESML is significantly better than SOTA methods.

2. Related Work

Most existing research on graph embedding still focuses on context-free graph-embedding methods, i.e., using only local or global features to accomplish node representation. For example, Perozzi et al. [7] introduced the DeepWalk method to address the challenge of defining neighborhood structures. This method utilizes random walks to capture neighbors' structures and enhances the likelihood of neighboring nodes appearing in each random walk sequence by implementing the Skip-Gram approach. Building upon the foundations of DeepWalk, Grover et al. [15] introduced the Node2Vec method, which establishes a versatile notion of node graph neighborhoods. Node2Vec formulates a second-order random walk strategy to sample neighboring nodes, enabling a seamless transition between breadth-first sampling (BFS) and depth-first sampling (DFS). In addition to addressing neighborhood structures, Tang et al. [16] proposed LINE, a large-scale network embedding model that preserves first-order and second-order proximities. To overcome the limitation of the expressive power of LINE, Wang et al. [17] presented a deep model called SDNE for graph embedding. It also endeavors to capture both first-order and second-order proximities. Additionally, numerous works concentrate on the fusion of graph embedding with topological structure. For example, Yang et al. [18] introduced TADW, which incorporates much information (such as text) from nodes while acquiring low-dimensional representations. Pan et al. [9] devised a coupled deep model encompassing graph structure, node attributes, and node labels within graph-embedding techniques. Huang et al. [19] put forth attribute graph embedding (for both nodes and features) with matrix factorization subject to joint constraints. Kipf et al. [4] developed a graph autoencoder that naturally integrates graphs' node attributes and structural information.

The accuracy of a single representation of context-free graph embeddings has been debated among recent works, leading to the proposal of context-sensitive approaches. Context-sensitive graph embedding captures different scenarios a node is in and learns the node's representation in that scenario. It means that for the same node, the representation changes as the goal task changes. For example, Epasto et al. [12] devised a method for acquiring multiple representations of nodes through self-network decomposition, which enables the encoding of node roles within distinct local communities. Kefato et al. [13] employed an attentional pooling network to learn the personalized importance of neighboring nodes. It allows the model to concentrate exclusively on relevant neighbor information and ignore irrelevant neighbor information during interaction processes. Li et al. [20] introduced a personalized PageRank method to acquire larger receptive fields and higher-quality representations of text embeddings. Several works employ side information to aid in the learning of node representations. Tu et al. [10] proposed a mutual attention mechanism that integrates both structural and textual information of nodes. This mechanism takes into account contextual information and interaction relationships in node representations. Zhang et al. [21] introduced a diffusion graph model for text graph embedding that utilizes the global structural information of the graph to capture semantic correlations between texts. Wang et al. [22] put forth an embedding graph form that concentrates on text. This form encompasses the modeling of text information and network topology. Expanding upon the research by Kefato et al. [13], we propose a novel graph-embedding model that employs a symmetric similarity measurement function that enables encoding a more significant amount of topological structural information of nodes, thereby obtaining high-quality node embedding representations.

3. Preliminaries

We first briefly overview the essential components of the prior work GAP [13] before delving into GESML detail. A graph $\mathcal{G}$ can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of nodes and $\mathcal{E}$ denotes the set of edges. The first step in the GAP method involves defining a neighborhood sampling function $f : \mathcal{V} \to 2^{\mathcal{V}}$. Here, f maps each node $u \in \mathcal{V}$ in the graph to a set of nodes $f_u \subseteq \mathcal{V}$. One straightforward way of implementing f_u is by sampling the first-order neighbors of node u ($f_u = [v : (u, v) \in \mathcal{E} \vee (v, u) \in \mathcal{E}]$). Given the neighbors' sequences $N_{\text{source}} = (u_1, \cdots, u_m)$ and $N_{\text{target}} = (v_1, \cdots, v_m)$ corresponding to node pairs $(u_s, v_t) \in \mathcal{E}$, where m denotes the number of sampled neighbors. Applying word embedding techniques, we can generate the embedding vectors. $\boldsymbol{S} = (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m)$ and $\boldsymbol{T} = (\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m)$ for each $u_i \in N_{\text{source}}$ and $v_i \in N_{\text{target}}$. Figure 2 illustrates the model architecture of GAP.

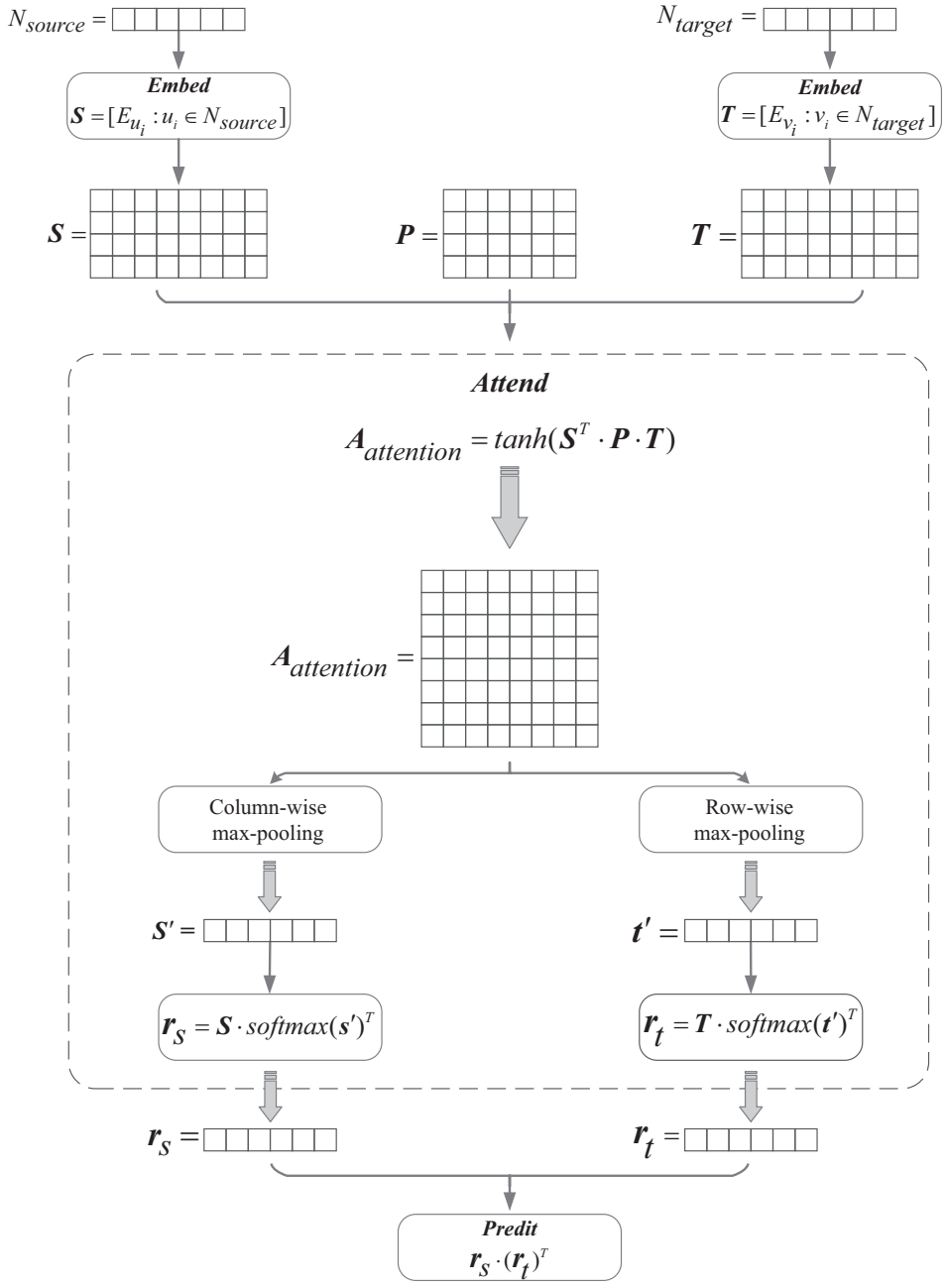

Figure 2. GAP.

The processing of GAP can be described as follows:

1. Utilize the sampling function $f : \mathcal{V} \to 2^{\mathcal{V}}$ to perform first-order neighbor sampling and employ graph embedding to obtain representations for the source node and target node, denoted as $\boldsymbol{S} = (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m)$ and $\boldsymbol{T} = (\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m)$.
2. Employ a trainable parameter matrix $\boldsymbol{P} \in \mathbb{R}^{d \times d}$ to compute the attention matrix $\boldsymbol{A}_{attention}$ between node pairs (u_s, v_t) . The attention matrix is calculated as $\boldsymbol{A}_{attention} = \tanh\left(\boldsymbol{S}^T \cdot \boldsymbol{P} \cdot \boldsymbol{T}\right)$.
3. Perform pooling operations on $\boldsymbol{A}_{attention}$ separately to obtain the unnormalized attention vectors: $\boldsymbol{s}'_i = \max(\boldsymbol{A}_{attention}(i,:))$, $\mathbf{t}'_j = \max(\boldsymbol{A}_{attention}(:,j))$.
4. Use the $softmax(\cdot)$ to normalize the attention vector and then apply $\boldsymbol{r}_s = \boldsymbol{S} \cdot \text{softmax} \cdot (\boldsymbol{s}')^T$ and $\boldsymbol{r}_t = \boldsymbol{T} \cdot \text{softmax} \cdot (\boldsymbol{t}')^T$ to obtain the node representations for the source node u_s and the target node v_t.
5. Compute $\boldsymbol{r}_s \cdot (\boldsymbol{r}_t)^T$ to obtain the score of the node pair (u_s, v_t).

Applying GAP reveals that the mutual attention mechanism between nodes solely focuses on their common neighbors. This method could overlook crucial information essential for node representations, such as implicit topological and structural information.

4. Graph Embedding with Similarity Metric Learning

Figure 3 illustrates the model architecture of the GESML. It comprises three modules: the graph structure learning module, the graph representation learning module, and the prediction module. The graph structure learning module aims to sample and obtain embedded representations of the source and target nodes from the graph and generate a similarity matrix using an attention-based symmetric similarity metric function. The graph representation learning module encodes the source and target nodes separately to generate embedded representations for node pairs. The prediction module utilizes the embedded representations of the source and target nodes to compute soft alignment scores.

4.1. The Graph Structure Learning Module

The primary objective of the graph structure learning module is to sample and obtain embedded representations of the source and target nodes from the graph and then generate a similarity matrix using an attention-based symmetric similarity metric function. Similar to GAP [13], the model initially defines a neighbor-sampling function $f : \mathcal{V} \to 2^{\mathcal{V}}$ that maps each node $u \in \mathcal{V}$ in the graph to a set of nodes $f_u \subseteq \mathcal{V}$. One simple method to implement f_u is by sampling the first-order neighbors of node u ($f_u = [v : (u,v) \in \mathcal{E} \vee (v,u) \in \mathcal{E}]$). Secondly, the neighbor set for the node pair $(u_s, v_t) \in \mathcal{E}$ is obtained by applying the first-order neighbor-sampling function f, resulting in $N_{\text{source}} = (u_1, \ldots, u_m)$ and $N_{\text{target}} = (v_1, \ldots, v_m)$, where m denotes the number of sampled neighbor nodes. Furthermore, for each $u_i \in N_{source}$ and $v_i \in N_{target}$, we can generate the embedding vectors $\boldsymbol{S} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)$ and $\boldsymbol{T} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$. Finally, a trainable parameter matrix $\boldsymbol{P} \in \mathbb{R}^{d \times d}$ and an attention-based symmetric similarity metric function are introduced to calculate the similarity matrix $\boldsymbol{A}_{similarity}$ between node pairs (u_s, v_t) using $\boldsymbol{A}_{similarity} = \text{softmax}\big(\text{ReLU}(\boldsymbol{P} \cdot \boldsymbol{S})^T \, \text{ReLU}(\boldsymbol{P} \cdot \boldsymbol{T})\big)$. In this case, $\text{ReLU}(x) = \max(0, x)$ refers to a rectified linear unit function employed to enforce sparsity in the output similarity matrix, while $\text{softmax}(\cdot)$ is applied to obtain a row-normalized similarity matrix.

The similarity matrix $\boldsymbol{A}_{\text{similarity}} \in \mathbb{R}^{m \times m}$ represents the degree of association between node pairs. The i-th row vector $\overline{\boldsymbol{A}}_{i:} \in \mathbb{R}^m$ in $\boldsymbol{A}_{similarity}$ is associated with the source node $u_i \in N_{source}$. The element $\overline{\boldsymbol{A}}_{ij}$ of $\overline{\boldsymbol{A}}_{i:}$ encodes the similarity between the global embedding $\boldsymbol{S}^T_{u_i} = E_{u_i}$ of source node $u_i \in N_{source}$ and the neighbors ($\left[E_{v_j} : v_j \in N_{target}\right]$) of the target node $v_j \in N_{target}$. Similarly, The j-th column vector $\overline{\boldsymbol{A}}_{:j} \in \mathbb{R}^m$ in $\boldsymbol{A}_{similarity}$ is associated with the target node $v_j \in N_{target}$. The element $\overline{\boldsymbol{A}}_{ij}$ of $\overline{\boldsymbol{A}}_{:j}$ encodes the similarity between the global embedding $\boldsymbol{T}_{v_j} = E_{v_j}$ of target node $v_j \in N_{target}$ and the neighbors ($[E_{u_i} : u_i \in N_{source}]$) of the source node $u_i \in N_{source}$. Thus, $\boldsymbol{A}_{similarity}$ can be interpreted as a weighted adjacency matrix among the nodes. Additionally, using $\text{ReLU}(\cdot)$ to enforce sparsity in the output

similarity matrix is beneficial due to the high computational costs and potential noise introduced by fully connected graphs, such as irrelevant edges. Enhancing the sparsity of the learned graph structure in this manner is advantageous. Furthermore, the application of $softmax(\cdot)$ for row normalization of $\boldsymbol{A}_{similarity}$ satisfies the input requirement of the asymmetric encoder in the graph representation learning module.

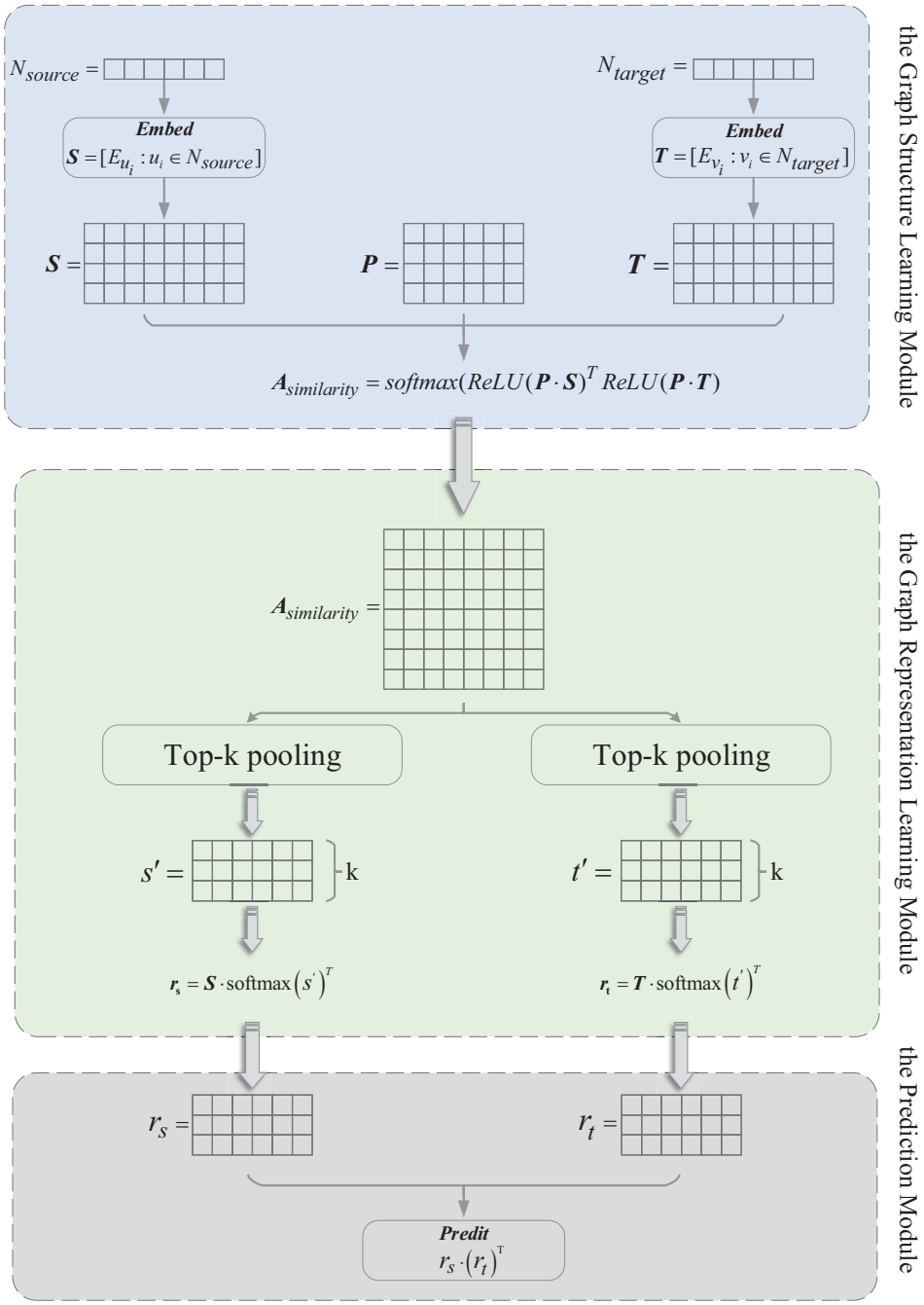

Figure 3. GESML.

4.2. The Graph Representation Learning Module

The goal of the graph representation learning module is to independently encode the source and target nodes and generate embedded representations for node pairs. Firstly, for each row (or column) of the similarity matrix $\boldsymbol{A}_{similarity} \in \mathbb{R}^{m \times m}$, we compute the unnormalized attention matrices $\boldsymbol{s}'_i = \mathrm{pool}_k(\boldsymbol{A}_{i,:})$ and $\boldsymbol{t}'_j = \mathrm{pool}_k(\boldsymbol{A}_{:,j})$ using top-k pooling

operations. These operations identify the top-k target nodes $vinN_{target}$ with the highest similarity scores captured by each neighbor $u \in N_{source}$ of the source node s. The same method is also applied to the target nodes $v \in N_{target}$. This method allows the neighborhood sequences of the source and target nodes to influence each other mutually, facilitating the learning of context-sensitive representations for both. Following, the $softmax(\cdot)$ function is applied to obtain $\boldsymbol{s} = \text{softmax}(\boldsymbol{s}')$ and $\boldsymbol{t} = \text{softmax}(\boldsymbol{t}')$. Finally, the context-sensitive representations for the source and target nodes are computed as $\boldsymbol{r}_s = \boldsymbol{S} \cdot \boldsymbol{s}^T$ and $\boldsymbol{r}_t = \boldsymbol{T} \cdot \boldsymbol{t}^T$.

4.3. The Prediction Module

The model aims to predict edge similarity by leveraging the dot product of the representations for the source node and the target node, denoted as $\boldsymbol{r}_s \cdot \boldsymbol{r}_t$, to assess the probability of a connection. The model utilizes the edge margin loss function:

$$\mathcal{L}(s,t) = \max\left(0, 1 - \text{mean}\left(\boldsymbol{r}_s \cdot \boldsymbol{r}_t - \boldsymbol{r}_s \cdot \boldsymbol{r}_t^-\right)\right)$$

where $\boldsymbol{r}_t^-$ denotes the representation of the target node acquired through negative sampling ($(s, t^-) \notin E$), and mean$(\cdot)$ computes the average of the results. The model strives to learn context-sensitive node embeddings in an unsupervised manner, where the ranking of positive edges $(s,t) \in E$ surpasses negative edges $(s,t^-) \notin E$. Finally, the computational complexity of GESML is proportional to the number of edges in the graph since we consider each edge as a pair of inputs.

5. Experiments

We will conduct extensive experiments on three datasets to evaluate the model performance of GESML. Additionally, we will employ visualizations to supplement the explanation of experimental results.

5.1. Datasets and Benchmarking Methods

Following a similar method to the work by Kefato et al. [13], GESML will be evaluated using three datasets: Cora [23], Zhihu [11], and Email [24]. Cora is a citation network dataset that papers relationships through citations. Nodes represent papers, and labels represent topics. Node features correspond to the words in the papers, and edges between nodes represent citation relationships. Zhihu, China's largest online question-and-answer community, focuses on modeling relationships among registered users, with user identification based on their posts. Email represents the communication network system of a prominent European research institution, primarily focused on modeling communication relationships among users within the institution. Table 1 presents the primary statistical information for these three benchmark datasets.

Table 1. The basic statistical information for three datasets.

Dataset	Nodes	Edges	Feature
Cora	2277	5214	Abstract of Paper
Zhihu	10,000	43,894	Posts of the User
Email	1005	25,571	

We aim to facilitate method comparison by initially categorizing all benchmark methods. Table 2 illustrates the division of 16 benchmark methods into four categories based on two criteria: context sensitivity and the utilization of structural information. Moreover, the models' performance evaluation primarily involves two types of experiments: link prediction and node clustering. Lastly, all results represent the average of 20 model runs to ensure experimental fairness.

Table 2. Classification results of the benchmark methods.

Context-Free		Context-Sensitive	
Structure	**Structure and Feature**	**Structure**	**Structure and Feature**
DeepWalk [7]	TRIDNR [9]	SPLITTER [12]	CANE [10]
Node2Vec [15]	TADW [18]	GAP [13]	DMTE [11]
WalkLets [25]	CENE [26]	PPR [20]	VHE [22]
Attentive Walk [27]		GOAT [24]	ACNE [11]
Line [16]			

The following is a brief introduction to the benchmark methods:

- DeepWalk [7] captures the neighborhood structure by performing random walks and maximizes the probability of neighboring nodes appearing in the walk sequences using the Skip-Gram approach.
- Node2Vec [15] employs second-order random walk strategies to sample neighboring nodes in the graph, smoothly interpolating between breadth-first and depth-first sampling. Nodes are treated as words, and the Word2Vec algorithm from NLP is used to train the nodes.
- WalkLets [25] is a multi-scale representation learning method for network nodes, which generates multi-scale relationships by performing secondary sampling of short random hashes on the nodes in the graph.
- Attentive Walk [27] learns the network parameters by treating the network parameters as the probability distribution of neighbors in a random walk.
- Line [16] embeds large-scale information networks into low-dimensional vector spaces and designs a method that preserves both local (first-order similarity) and global (second-order similarity) network structures.
- TRIDNR [9] proposes a coupled neural network algorithm that utilizes the relationships between nodes, the relevance of node content, and label content to obtain the optimal representation for each node.
- TADW [18] demonstrates the equivalence of deep walk to matrix factorization and proposes a network learning method that combines text information.
- CENE [26] treats text content as a specific node type and performs node embedding using both node links and node content links.
- SPLITTER [12] introduces Splitter, a novel technique that learns multiple embeddings for individual nodes, enabling a better description of networks with overlapping communities.
- GAP [13] presents a novel context-aware graph-embedding algorithm that utilizes attention pooling networks to focus on different parts of the node neighborhood.
- PPR [20] introduces personalized PageRank methods to obtain larger receptive fields and higher quality text embedding representations.
- GOAT [24] proposes GOAT, a context-aware algorithm inspired by node information propagation and mutual attention mechanisms, which can obtain high-quality context-aware node representations relying solely on graph structure.
- CANE [10] introduces CANE, a mutual attention mechanism that combines node structural information and text information, and the representation of nodes considers contextual information and different interaction relationships.
- DMTE [11] learns low-dimensional vector representations for nodes that contain rich text information relevant to the network.
- VHE [22] considers VHE an embedding graph form that explicitly focuses on text and models text information and network topology.
- ACNE [11] proposes ACNE, an adversarial mechanism that utilizes text-embedding discriminators and structure-embedding generators to learn effective representations.

5.2. Link Prediction

This section presents an evaluation of the performance of GESML on three datasets for link-prediction experiments. Link prediction is a fundamental task in graph embedding that aims to predict missing or future links between pairs of nodes. We use part of the edges for training the model parameters and the rest of the edges to test the effect of the model. The training set encompasses edge proportions ranging from 0.15 to 0.95, with gaps of 0.2. A fixed number of neighbor samples is employed across the three datasets. Each node is represented by a 200-dimensional vector, and Table 3 provides comprehensive information on other experimental parameters. Notably, the results of this experiment are quantified with the AUC score, which indicates the probability of randomly selected node pairs $(u_i, v_j) \in \mathcal{E}$ obtaining higher similarity scores compared to node pairs $(u_i, v_k) \notin \mathcal{E}$. In addition, numbers marked in bold indicate state-of-the-art results.

Table 3. Experimental parameters statistics.

Dataset	Neighbors	Dropout	Learning Rate
Cora	100	0.50	0.0001
Zhihu	250	0.65	0.0001
Email	100	0.80	0.0001

From Tables 4–6, we can conclude the following observations:

- In general, GESML outperforms the baseline methods on the three datasets.
- A notable improvement in link prediction results is observed when the proportion of edges in the training set is relatively small on the three datasets. This phenomenon means the attention-based similarity measurement function can learn more effective graph structures.
- GESML exhibits a significant variation in its impact on link prediction results on three datasets. The performance improvement is relatively modest on the Cora. This result can reason by the varying nodes and edges among the three datasets (Cora: 2.29; Zhihu: 4.39; Email: 25.44). The Cora dataset contains relatively less structural information.

Table 4. Linking prediction task scores on the Cora dataset (%).

Methods	The Percentage of the Training Set				
	15%	**35%**	**55%**	**75%**	**95%**
DeepWalk [7]	56.0	70.2	80.1	85.3	90.3
Node2Vec [15]	55.9	66.1	78.7	85.9	88.2
Line [16]	55.0	66.4	77.6	85.6	89.3
Attentive Walk [27]	64.2	81.0	87.1	91.4	93.0
WalkLets [25]	69.8	82.8	86.6	90.9	93.3
CENE [26]	72.1	84.6	89.4	93.9	95.9
TRIDNR [9]	85.9	90.5	91.3	93.0	93.7
TADW [18]	86.6	90.2	90.0	91.0	92.7
CANE [10]	86.8	92.2	94.6	95.6	97.7
DMTE [11]	91.3	93.7	96.0	97.4	98.8
SPLITTER [12]	65.4	73.7	80.1	83.9	87.2
GOAT [24]	96.7	97.1	97.6	97.8	98.8
GAP [13]	95.8	97.1	97.6	97.8	98.2
PPR [20]	86.0	91.3	92.4	95.8	98.1
VHE [22]	94.4	97.6	98.3	99.0	99.4
ACNE [11]	95.8	97.6	98.5	99.0	**99.5**
GESML	**96.7**	**98.1**	**98.7**	**99.1**	99.4

Table 5. Linking prediction task scores on the Zhihu dataset (%).

Methods	The Percentage of the Training Set				
	15%	**35%**	**55%**	**75%**	**95%**
DeepWalk [7]	56.6	60.1	61.8	63.3	67.8
Node2Vec [15]	52.3	59.9	64.3	67.7	71.1
Line [16]	54.2	57.3	58.7	66.2	68.5
Attentive Walk [27]	50.7	52.6	55.5	57.9	58.1
WalkLets [25]	69.4	74.0	76.4	74.7	66.8
CENE [26]	52.3	55.6	60.8	65.2	69.0
TRIDNR [9]	53.8	57.9	63.0	66.0	70.3
TADW [18]	56.2	60.3	66.3	70.2	73.8
CANE [10]	56.8	62.9	68.9	71.4	75.4
DMTE [11]	58.4	67.5	74.0	78.7	82.2
SPLITTER [12]	59.8	61.8	62.1	61.0	58.6
GOAT [24]	**82.2**	82.3	85.1	84.5	83.7
GAP [13]	72.6	81.2	81.4	82.0	86.3
PPR [20]	66.3	76.4	78.7	83.9	87.2
VHE [22]	66.8	74.1	81.6	84.7	86.4
ACNE [11]	73.4	82.4	88.6	**91.1**	**93.2**
GESML	77.8	**84.5**	**89.0**	**91.1**	92.9

Table 6. Linking prediction task scores on the Email dataset (%).

Methods	The Percentage of the Training Set				
	15%	**35%**	**55%**	**75%**	**95%**
DeepWalk [7]	69.2	74.1	76.6	78.7	79.0
Line [16]	65.6	73.8	76.7	78.5	78.8
Node2Vec [15]	66.4	71.2	72.7	74.5	76.1
WalkLets [25]	70.3	75.2	78.2	78.9	78.5
Attentive Walk [27]	68.8	73.5	74.1	73.0	68.6
SPLITTER [12]	69.2	69.1	70.6	73.3	75.2
GOAT [24]	78.9	81.2	81.7	82.3	83.1
GAP [13]	77.6	81.9	83.1	84.5	84.8
GESML	**82.7**	**83.7**	**85.4**	**87.1**	**87.7**

5.3. Node Clustering

Node clustering is the process of categorizing samples into predefined classes based on their similarities. This section evaluates the model's performance on the Email dataset, specifically comparing it with baseline methods that leverage structural information. Like the link prediction experiments, we train the model parameters using a subset of edges and test them on the remaining edges. The training set encompasses edge proportions ranging from 0.35 to 0.95 with an interval of 0.2 while keeping other settings constant. To evaluate the experimental results, we measure the consistency between the given actual labels y and the predicted labels $\hat{y}$ using the $NMI(y, \hat{y})$ and $AMI(y, \hat{y})$ metrics. NMI represents the normalized mutual information ($I(y, \hat{y})$), and AMI indicates the adjusted mutual information [28].

Table 7 reveals that GESML outperforms the baseline methods. Thus, the results reinforce the idea that the model initially learns optimal graph structures through an attention-based similarity measurement function. It then leverages top-k pools to simultaneously capture the topological information of nodes and their neighborhoods. Consequently, this method enables encoding decadent topological information from node neighborhoods, leading to enhanced performance.

Table 7. Node clustering results on the Email dataset (%).

Methods	The Percentage of the Training Set							
	35%		55%		75%		95%	
	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI
DeepWalk [7]	41.3	28.6	53.6	44.8	50.6	42.4	57.6	49.9
Line [16]	44.0	30.3	49.9	38.2	53.3	42.6	56.3	46.5
Node2Vec [15]	46.6	35.3	45.9	35.3	47.8	38.5	53.8	45.5
WalkLets [25]	47.5	39.9	55.3	47.4	54.0	45.4	50.1	41.6
Attentive Walk [27]	42.9	30.0	45.7	36.5	44.3	35.7	47.4	38.5
SPLITTER [12]	38.9	23.8	43.2	30.3	45.2	33.6	48.4	37.6
GOAT [24]	66.5	57.2	65.6	56.7	66.4	57.9	65.5	57.0
GAP [13]	67.8	58.8	64.7	55.7	65.6	57.6	65.4	58.7
GESML	**73.9**	**64.2**	**70.9**	**59.8**	**73.5**	**65.2**	**70.8**	**64.1**

5.4. Hyperparameter Analysis

In this section, we first analyze the influence of the number of sampled neighbors on the performance of GESML. We conduct link prediction experiments on three datasets to evaluate how the number of sampled neighbors affects the model's performance. The experimental results are illustrated in Figures 4–6.

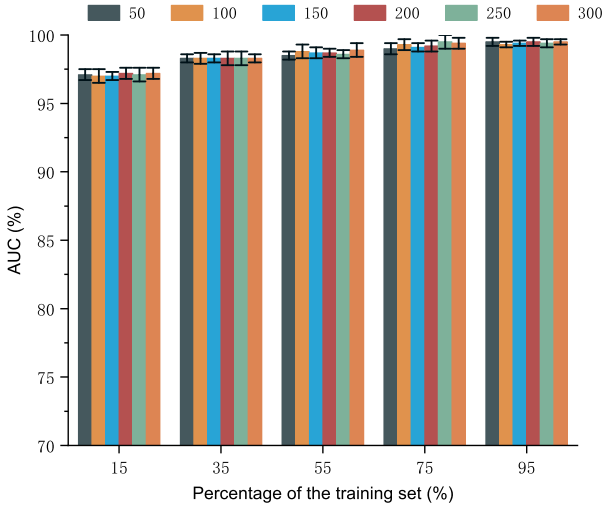

Figure 4. Effect of the number of neighbors sampled on the Cora dataset.

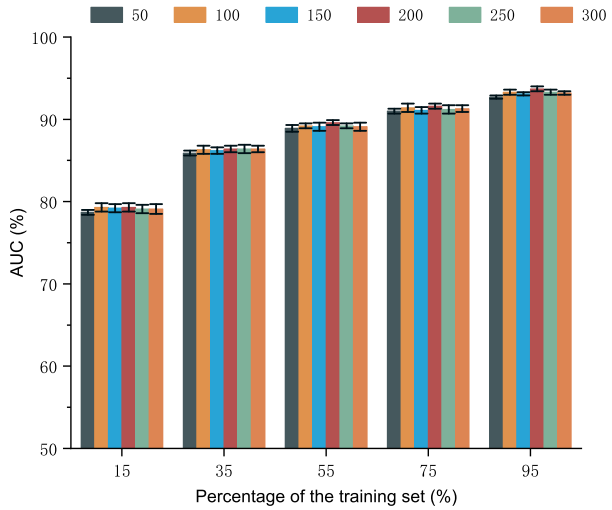

Figure 5. Effect of the number of neighbors sampled on the Email dataset.

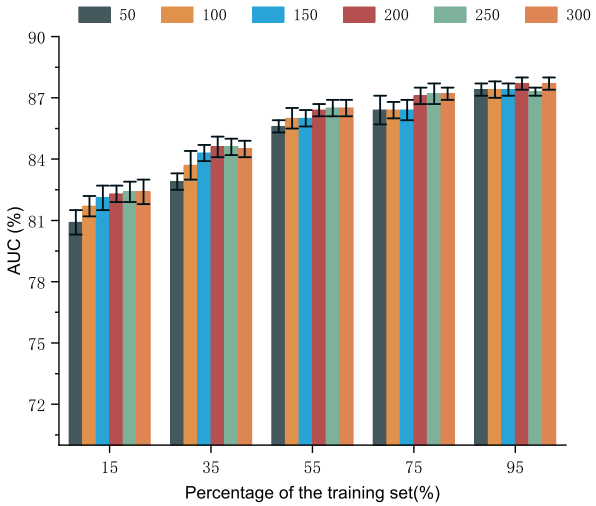

Figure 6. Effect of the number of neighbors sampled on the Zhihu dataset.

Observing Figures 4–6 reveals that the model's performance remains unaffected mainly by the variation in the number of sampled neighbors, irrespective of changes in the percentage of edges in training set across the three datasets.

This section conducts clustering task experiments on the Email dataset to ensure experimental comprehensiveness. These experiments aim to investigate the effect of the number of sampled neighbors on the model's performance. Figure 7 illustrates the relationship between the number of sampled neighbors and the model's performance, with a fixed proportion of 55% of edges in the training set.

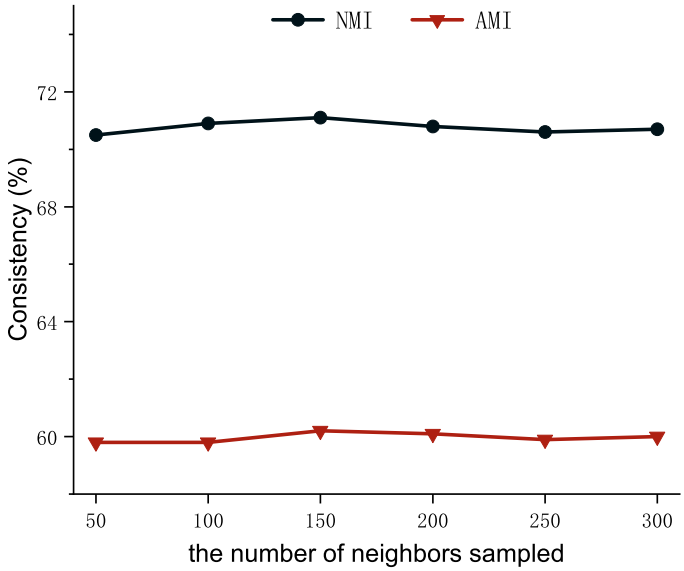

Figure 7. The effect of the number of sampled neighbors on the model's performance.

The analysis of Figure 7 indicates that variations in the number of sampled neighbors have a negligible impact on the NMI and AMI consistency metrics in the node clustering task. Based on the results of the link prediction and node clustering experiments regarding the influence of the number of sampled neighbors on model performance, we can infer that the model exhibits insensitivity to this hyperparameter.

Furthermore, we explored the influence of different values of k in top-k pooling. Specifically, we utilized the Cora dataset, with 15% of the edges used for training, to examine various values of k.

Observing Figure 8 reveals that:

- The model's performance improves as k increases within the range of $k \in [1,5]$. This suggests that encoding topological structural information can enhance the model's performance.
- The model shows insensitivity to k when $k > 5$.

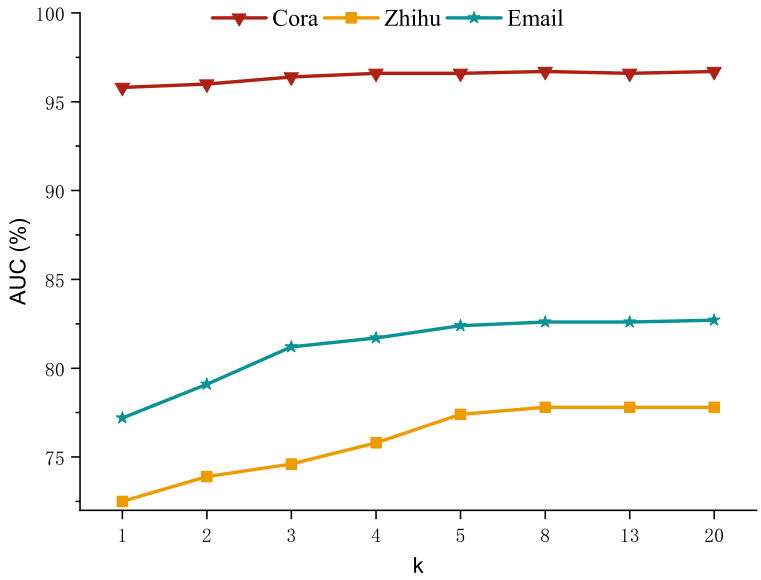

Figure 8. The effect of various k on the model's performance.

5.5. Model Limitations Analysis

This section aims to identify several limitations inherent in the current model.

Global Structure of Nodes: The current model primarily emphasizes the first-order neighborhood structure of nodes while neglecting the broader global structure. By incorporating the global network, the model's receptive field would expand, enabling a more comprehensive acquisition of structural information.

Selection of Metric Learning Methods: The current model incorporates an attention-based similarity measurement function that promotes mutual attention among nodes. However, this approach should align optimally with the model's primary objective of maximizing the preservation of structural information. We can use an alternative metric learning approach that integrates structural awareness to address this concern. This alternative technique would facilitate the extraction of implicit graph structures from the data, leading to more effective capture of intricate structural patterns.

6. Conclusions

Achieving context-sensitive graph embeddings has been the subject of numerous attempts, yet they have limitations. Some methods need supplementary information, while others use community algorithms to capture multiple contexts. Moreover, some methods require capturing sufficient structural information effectively. This paper presents GESML, a novel graph embedding with a similarity metric learning model incorporating a metric learning method that relieves these limitations. GESML leverages an attention-based symmetric similarity metric function and a TOP-K pooling operation to acquire high-quality node representations, integrating node and structural information from the graph neighborhood as input features. Finally, we assess the effectiveness of GESML through extensive experimentation on three datasets, focusing on its performance in link prediction and node-clustering tasks. The experimental results conclusively demonstrate that GESML outperforms baseline methods in both tasks, firmly establishing its superiority. In the future, our research endeavors will explore cross-fusion among neighboring nodes. These will enhance the capabilities of GESML further and unlock even more promising results.

Author Contributions: T.T. designed the framework and realized and verified the proposed scheme. Q.W., Y.R. and X.L. were responsible for the organizational structure and edited the manuscript. X.W. reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Program of the Natural Science Foundation of the Educational Commission of Anhui Province of China (Grant No. 2022AH050319, 2022AH052740), and the Natural Science Foundation Project of Anhui Province of China (Grant No. 2008085QF305).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no potential conflicts of interest with respect to the research, authorship, or publication of this article.

References

1. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 478–487.
2. Haghani, S.; Keyvanpour, M. R. A systemic analysis of link prediction in social network. *Artif. Intell. Rev.* **2019**, *52*, 1961–1995. [CrossRef]
3. Chen, L.; Guan, Z.; Xu, Q.; Zhang, Q.; Sun, H.; Lu, G.; Cai, D. Question-driven purchasing propensity analysis for recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 35–42. [CrossRef]
4. Kipf, T. N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
5. Qin, J.; Zeng, X.; Wu, S.; Tang, E. E-GCN: Graph convolution with estimated labels. *Appl. Intell.* **2021**, *51*, 5007–5015. [CrossRef]
6. Qin, J.; Zeng, X.; Wu, S.; Zou, Y. Multi-Semantic Alignment Graph Convolutional Network. *Connect. Sci.* **2022**, *34*, 2313–2331. [CrossRef]
7. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710. [CrossRef]
8. Sheikh, N.; Kefato, Z.; Montresor, A. gat2vec: Representation learning for attributed graphs. *Computing* **2019**, *101*, 187–209. [CrossRef]
9. Pan, S.; Wu, J.; Zhu, X.; Zhang, C.; Wang, Y. Tri-party deep network representation. *Network* **2016**, *11*, 12. [CrossRef]
10. Tu, C.; Liu, H.; Liu, Z.; Sun, M. Cane: Context-aware network embedding for relation modeling. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1; Long Papers; pp. 1722–1731. [CrossRef]
11. Gracious, T.; Dukkipati, A. Adversarial context aware network embeddings for textual networks. *arXiv* **2020**, arXiv:2011.02665.
12. Epasto, A.; Perozzi, B. Is a single embedding enough? learning node representations that capture multiple social contexts. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 394–404. [CrossRef]
13. Kefato, Z. T.; Girdzijauskas, S. Graph neighborhood attentive pooling. *arXiv* **2020**, arXiv:2001.10394.
14. Qin, J.; Zeng, X.; Wu, S.; Zou, Y. Context-sensitive graph representation learning. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 2193–2203. [CrossRef]
15. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864. [CrossRef]
16. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077. [CrossRef]
17. Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234. [CrossRef]
18. Yang, C.; Liu, Z.; Zhao, D.; Sun, M.; Chang, E. Y. Network representation learning with rich text information. In Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 2111–2117. [CrossRef]
19. Huang, X.; Li, J.; Hu, X. Accelerated attributed network embedding. In Proceedings of the 2017 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, Houston, TX, USA, 27–29 April 2017; pp. 633–641. [CrossRef]
20. Li, T.; Dou, Y. Representation learning on textual network with personalized PageRank. *Sci. China Inf. Sci.* **2021**, *64*, 212102. [CrossRef]
21. Zhang, X.; Li, Y.; Shen, D.; Carin, L. Diffusion maps for textual network embedding. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7587–7597.
22. Wang, W.; Tao, C.; Gan, Z.; Wang, G.; Chen, L.; Zhang, X.; Zhang, R.; Yang, Q.; Henao, R.; Carin, L. Improving textual network learning with variational homophilic embeddings. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 2074–2085.
23. McCallum, A. K.; Nigam, K.; Rennie, J.; Seymore, K. Automating the construction of internet portals with machine learning. *Inf. Retr.* **2000**, *3*, 127–163. [CrossRef]
24. Kefato, Z.; Girdzijauskas, S. Gossip and attend: Context-sensitive graph representation learning. In Proceedings of the 14th International AAAI Conference on Web and Social Media, Atlanta, GA, USA, 8–11 June 2020; Volume 14, pp. 351–359. [CrossRef]

25. Perozzi, B.; Kulkarni, V.; Skiena, S. Walklets: Multiscale graph embeddings for interpretable network classification. *arXiv* **2016**, arXiv:1605.02115.
26. Sun, X.; Guo, J.; Ding, X.; Liu, T. A general framework for content-enhanced network representation learning. *arXiv* **2016**, arXiv:1610.02906.
27. Abu-El-Haija, S.; Perozzi, B.; Al-Rfou, R.; Alemi, A. A. Watch your step: Learning node embeddings via graph attention. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 9180–9190.
28. Vinh, N. X.; Epps, J.; Bailey, J. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 1073–1080. [CrossRef]