Received: 20 April 2021 | Revised: 11 August 2021 | Accepted: 26 August 2021

DOI: 10.1002/int.22664

RESEARCH ARTICLE

WILEY

AHNA: Adaptive representation learning for attributed heterogeneous networks

Lin Shu[1] | **Chuan Chen**[1] | **Xingxing Xing**[2] | **Xiangke Liao**[3] | **Zibin Zheng**[1]

[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China

[2]Netease Games UX Center, Guangzhou, China

[3]School of Computer Science, National University of Defense Technology, Changsha, China

Correspondence
Chuan Chen, School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China.
Email: chenchuan@mail.sysu.edu.cn

Funding information
Key-Area Research and Development Program of Guangdong Province, Grant/Award Number: 2020B010165003; National Natural Science Foundation of China, Grant/Award Number: 62176269, 11801595; Guangdong Basic and Applied Basic Research Foundation, Grant/Award Number: 2019A1515011043; UX Center, Netease Games

Abstract

Meta-path-based random walk strategy has attracted tremendous attention in heterogeneous network representation, which can capture network semantics with heterogeneous neighborhoods of nodes. Despite the success of meta-path-based random walk strategy in plain heterogeneous networks which contain no attributes, it remains unexplored how meta-path-based random walk strategy could be utilized on attributed heterogeneous networks to simultaneously capture structural heterogeneity and attribute proximity. Moreover, the importance of node attributes and structural relations generally varies across data sets, thus requiring careful considerations when they are incorporated into representations. To tackle these problems, we propose a novel method, *A*ttributed *H*eterogeneous *N*etwork embedding based on *A*ggregate-path (AHNA), which generates aggregate-path-based random walks on attributed heterogeneous networks and adaptively fuses topological structures and node attributes based on the learned importance. Specifically, AHNA first converts node attributes to additional links in the network to deal with the heterogeneity of structures and attributes, which is followed by an adaptive random walk strategy to strike the importance balance between node attributes and

Joint work with UX Center, Netease Games.

topological structures, thereby generating high-quality representations. Extensive experiments are conducted on three real-world data sets, where AHNA outperforms state-of-the-art approaches by up to 22.7%, 2.6%, and 2.3% on link prediction, community detection, and node classification, respectively. Moreover, our qualitative analysis indicates that AHNA can capture different balances of topological structures and node attributes on various data sets and thus boost the quality of node representations.

KEYWORDS

adaptive, attributed heterogeneous network, balance, network representation

1 | INTRODUCTION

In recent years, heterogeneous network representation/embedding has attracted tremendous attention and has a wide range of applications in domains varying from mobile edge computing,[1] social science[2,3] to biomedicine.[4,5] As a significant tool in heterogeneous network representation, meta-path-based random walk strategy preserves network semantics by exploring heterogeneous neighborhoods of nodes and has been demonstrated to be effective in various works.[6–8]

Though meta-path-based random walk strategy on plain heterogeneous networks has been extensively investigated, there exist numerous attributed heterogeneous networks where nodes are affiliated with rich attributes in real-world systems. For example, Figure 1 depicts an attributed heterogeneous academic network with three node types (i.e., *Author*, *Paper*, and *Conference*), where *Author* and *Paper* own attributes of *Affiliation* and *Keyword*,

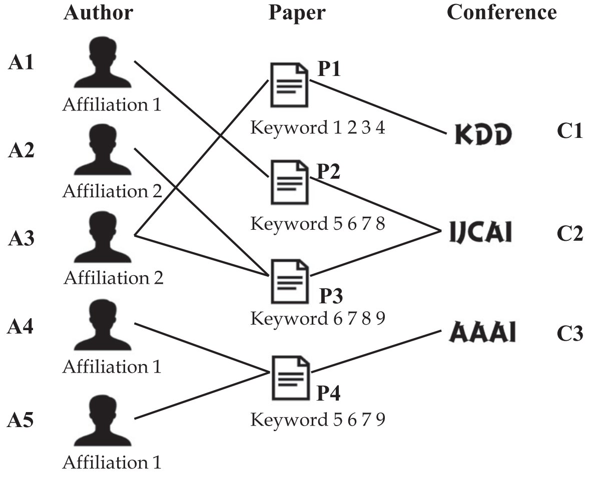

FIGURE 1 An illustrative example of attributed heterogeneous academic network

respectively. Since meta-path-based random walk strategy only leverages structural heterogeneity, we argue that introducing attribute proximity into walks, for example, aggregate-path-based random walks proposed in this paper, is a better way to capture node homophily. To prove the hypothesis, we extract meta-path-based random walks and aggregate-path-based random walks (detailed in Section 4.2) on Association for Computing Machinery (ACM) data set, respectively, where the difference is that aggregate-path-based random walks can leverage node attributes ignored by meta-path-based random walks. Table 1 shows the results of label consistency, which is the mean proportion of nodes that own the same label with the first node within each path. It can be observed that the label consistency of aggregate-path-based random walks is always higher than that of meta-path-based random walks when the path length increases from 5 to 10. The observations above verify that introducing attribute proximity into random walks is a more effective way to capture node homophily and thus generate higher-quality embeddings.

In addition, there exists different importance between node attributes and topological structures in different data sets. For instance, similarity preserving graph convolutional network (SimP-GCN)[9] demonstrates that disassortative data sets have unreliable structures, thus relying heavily on the topology rather than attributes of these data sets leads to poor performances on downstream tasks. Therefore, the balance of topological structures and node attributes should be taken into considerations when they are utilized together.

To sum up, how to introduce node attributes into random walks while in the meanwhile adaptively strike a balance between topological heterogeneity and attribute proximity remains unexplored. To resolve the aforementioned challenges, in this paper, we propose a novel method, namely, *A*ttributed *H*eterogeneous *N*etwork Embedding based on *A*ggregate-path (AHNA), to adaptively incorporate both attribute proximity and structural heterogeneity into node representations. Overall, the main contributions of this paper include:

- We convert node attributes to additional links to deal with the heterogeneity of structures and attributes in the network. Furthermore, we propose aggregate-path to introduce attribute proximity into random walks and present an aggregate-path-based random walk strategy, which can adaptively strike a balance between topological heterogeneity and node attributes based on their learned importance.
- On the basis of aggregate-path guided random walks, we utilize the skip-gram model and recurrent neural networks (RNNs) to generate node representations, respectively, where AHNA with skip-gram is suitable for downstream tasks relying more on topological structures and AHNA with RNNs is appropriate for tasks relying more on node attributes according to experimental results.

TABLE 1 The label consistency of meta-path-based random walks and aggregate-path-based random walks on ACM data set with the length of paths varying from 5 to 10

	Length of path					
Strategy	**5**	**6**	**7**	**8**	**9**	**10**
Meta-path	0.5559	0.5608	0.5420	0.5722	0.5970	0.5934
Aggregate-path	**0.5907**	**0.5993**	**0.6124**	**0.6137**	**0.6165**	**0.6155**

Note: The scheme of meta-path is *Author–Paper–Conference–Paper–Author*. The scheme of aggregate-path is [*Author*]–[*Paper*]–[*Conference*]–[*Paper*]–[*Author*].

Abbreviation: ACM, Association for Computing Machinery.

- We conduct extensive experiments on three real-world data sets and our results demonstrate the superior performance of AHNA over state-of-the-art baselines for various tasks, including link prediction, community detection, node classification, and relevance search.

The rest of this paper is organized as follows. We first introduce the related work in Section 2. Next, the problem of attributed heterogeneous network representation is formulated in Section 3, followed by the description of the proposed AHNA method in Section 4. In Section 5, the experimental results are presented to demonstrate the effectiveness of AHNA before concluding the paper in Section 6.

2 | RELATED WORK

In this section, we review related state-of-the-art methods for homogeneous network representation, plain heterogeneous network representation, and attributed heterogeneous network representation. Typical embedding methods for different network types are summarized in Table 2.

2.1 | Homogeneous network representation

In previous years, most researches focus on preserving structural information in the network. For example, DeepWalk[10] and Node2Vec[11] both utilize random walks and the skip-gram model to generate node embeddings. Large-scale information network embedding (LINE)[12] explores the first-order and second-order proximity to preserve structural information on large-scale networks. Structural deep network embedding (SDNE)[13] preserves highly nonlinear features to exploit both local and global proximity using deep autoencoders. Nevertheless, these methods are naturally designed for homogeneous networks, which are simplified representations of information networks in real-world scenarios, thus lacking the ability to handle the structural heterogeneity.

2.2 | Plain heterogeneous network representation

Recent years have witnessed a surge of interest in plain heterogeneous network representation, in which meta-path-based random walk strategy is a tool widely adopted. Representative researches such as Metapath2vec[6] which utilizes meta-path-based random walks and the skip-gram to preserve the structural heterogeneity. Heterogeneous embedding for recommendation (HERec)[8] leverages an extended Matrix Factorization model based on meta-path to obtain node embeddings on heterogeneous networks. Heterogeneous information network embedding (HINE)[7] investigates heterogeneous network representation by preserving meta-path-based proximities between nodes. Mg2vec[19] employs meta-graphs to learn embeddings for both meta-graphs and nodes jointly. CMG2Vec[20] proposes an extensible composite meta-graph to automatically select appropriate meta-path/meta-graph and then learn node representations from the extended heterogeneous autoencoder. However, many networks in real-world systems are heterogeneous and affiliated with rich attributes, while the aforementioned methods ignore node attributes and result in less informative embeddings.

TABLE 2 Typical embedding methods for different network types and their constraints

Network type	Method	Heterogeneity	Attribute	Constraint
Plain homogeneous network	DeepWalk[10] Node2Vec[11] LINE[12] SDNE[13]	No	No	W/o heterogeneity W/o attribute
Attributed homogeneous network	AANE[14] ANRL[15] DANE[16] SNE[17] NEC[18]	No	Yes	W/o heterogeneity
Plain heterogeneous network	Metapath2vec[6] HINE[7] HERec[8] Mg2vec[19] CMG2Vec[20]	Yes	No	W/o attribute
Attributed heterogeneous network	HAN[21] HetGNN[22] MAGNN[23] HetSANN[24] HGT[25]	Yes	Yes	Oversmoothing

Abbreviations: AANE, accelerated attributed network embedding; ANRL, attributed network representation learning; DANE, deep attributed network embedding; HAN, heterogeneous graph attention network; HERec, heterogeneous embedding for recommendation; HetGNN, heterogeneous graph neural network; HetSANN, heterogeneous graph structural attention neural network; HGT, heterogeneous graph transformer; HINE, heterogeneous information network embedding; LINE, large-scale information network embedding; MAGNN, meta-path aggregated graph neural network; NEC, network embedding for community; SDNE, structural deep network embedding; SNE, social network embedding.

2.3 | Attributed heterogeneous network representation

Most recently, there arise many researches that explore heterogeneous network representations with auxiliary information, among which graph neural networks[26,27] receive increased attention. For instance, heterogeneous graph attention network (HAN)[21] learns node-level and semantic-level attentions, which considers the importance of different nodes and meta-paths to obtain the ultimate node embeddings. Heterogeneous graph neural network (HetGNN)[22] aggregates neighbors' attributes according to node types and attribute types to generate representations. Meta-path aggregated graph neural network (MAGNN)[23] encapsulates input node attributes and incorporates intermediate semantic nodes in multiple meta-paths to boost the performance. Heterogeneous graph structural attention neural network (HetSANN)[24] leverages type-aware attention to learn representations without manually designing meta-path schemes. Heterogeneous graph transformer (HGT)[25] designs heterogeneous attention by node-type and edge-type dependent parameters to maintain heterogeneous structures. However, graph neural networks cannot extract high-order information since they suffer from the oversmoothing problem[28,29] when stacking too many layers, thus are not suitable for learning representations from random walks.

3 | PROBLEM DEFINITION

Definition 1 (Heterogeneous network). A heterogeneous network is denoted as $G = (V, E)$ consisting of various types of nodes V and links/edges E, where each link has a positive weight. Besides, each node v and link e is associated with a node-type mapping function $\phi(v) : V \to T$ and a link-type mapping function $\psi(e) : E \to R$, respectively, where T and R denote the sets of node types and links types and $|T| + |R| > 2$.

Definition 2 (Meta-path). A meta-path scheme is defined as a path in the form of $P : T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{L-1}} T_L$ (abbreviated as $T_1T_2 \cdots T_L$), where $T_i \in T, R_i \in R, L$ denotes the length of P and $R_1 \circ R_2 \circ \cdots \circ R_{L-1}$ describes a composite relation between T_1 and T_L.

Example 1. Given a meta-path scheme *Author–Paper–Conference–Paper–Author* (abbreviated as APCPA) of the heterogeneous network in Figure 1, we can obtain a node sequence $A1 \to P2 \to C2 \to P3 \to A3$, which reveals the semantics that authors $A1$ and $A3$ have both published papers on the conference $C2$.

Definition 3 (Attributed heterogeneous network). An attributed heterogeneous network can be defined as a heterogeneous network $G = (V, E, A)$, where $A_i \in \mathbb{R}^m$ denotes the attribute representation associated with node $v_i \in V$, m is the dimension of node attributes and varies with node types. For nodes with no attributes, A_i is represented by id embeddings.

Given an attributed heterogeneous network $G = (V, E, A)$, our aim is to learn a function $f : V \to \mathbb{R}^d$ for all nodes preserving both structural relations and attribute proximity, where d is the embedding size and $d \ll |V|$. Table 3 summarizes symbols used in this paper.

4 | FRAMEWORK

In this section, a new network is first constructed to deal with the heterogeneity of attributes and structures (Section 4.1). Then we formalize the definition of the proposed aggregate-path and explain the adaptive random walk strategy based on aggregate-path in detail (Section 4.2). Ultimately, two approaches are presented to generate final node representations (Section 4.3).

4.1 | Network construction

To address the heterogeneity of node attributes and topological structures, we convert attributes to additional links and construct a new network containing both structural relations and

TABLE 3 Notations and explanations

Notations	Explanations
G	Input network
V, E	Node/edge set of G
T, R	Node/edge-type set of G
A	Attribute set of G
$\phi(v)$	Node-type mapping function
$\psi(v)$	Edge-type mapping function
n	Number of nodes
m	Dimension of node attributes
d	Embedding size
L	Length of meta-path/aggregate-path
E_{stru}	Structural links
E_{attr}	Attribute links
G'	Constructed network
E'	Edge set of G'
$T_1 T_2 \cdots T_L$	Meta-path scheme
$[T_i]$	Node sequences aggregated by nodes of type T_i
$[T_1][T_2] \cdots [T_L]$	Aggregate-path scheme
T_i^j	jth node in $[T_i]$
h_{attr}	Attribute layer
h_{stru}	Structural layer
h	Attribute layer h_{attr} or structural layer h_{stru}
N_v^h	Nodes connected to v by attribute/structural links
E_v^h	Links between v and nodes in N_v^{attr} or N_v^{stru}
ρ_v^h	Importance of v's attribute/structural layer
w_{ij}	Edge weight between nodes i and j
$P_v(h)$	Probability of exploring in v's layer h next
$P_v^h(u)$	Probability of sampling node u from v's layer h
α	Hyperparameter indicating the importance of attributes
β	Hyperparameter indicating the importance of structures
X_v	Representation of node v generated by skip-gram
K	Number of negative samples
g_v	Representation of node v generated by RNN

Abbreviation: RNN, recurrent neural network.

attribute proximity. In the following sections, the links in the original network are referred to as structural links (denoted as E_{stru}), while the additional links constructed from attributes are attribute links (denoted as E_{attr}). Since the attribute type varies with the node type, attribute links are only constructed between two nodes with the same type. Specifically, given an attributed heterogeneous network $G = (V, E, A)$, for node type T_i, the cosine similarities between attribute representations of any two nodes with type T_i are computed. If the cosine similarity is relatively large, an attribute link is constructed and the weight of the attribute link is set to be the value of cosine similarity. Therefore we construct a new network $G' = (V, E')$, where $E' = E_{\text{stru}} \cup E_{\text{attr}}$.

Here, we take an example to illustrate the procedure of constructing networks. Specifically, the cosine similarity of one-hot representations of $A2$ and $A3$'s attributes is 1 in Figure 1, thus an attribute link is built by connecting $A2$ and $A3$. Similarly, $P2$ and $P4$ are also connected since the cosine similarity of their one-hot attribute representations is greater than 0. Ultimately, a new network G' is constructed as illustrated in Figure 2, where the black dotted lines are structural links and the black solid lines are attribute links.

4.2 | Aggregate-path-based random walk strategy

4.2.1 | Definitions of aggregate-path

Conventional meta-path-based random walk strategy leverages structural relations between diverse types of nodes but ignores attribute proximity. With regard to this, we propose aggregate-path-based random walk strategy to generate paths that integrate both structures and attributes. Specifically, after constructing a network containing both attribute links and structural links, nodes of the same type are aggregated into a group through attribute links and different groups are connected through structural links. Whether to explore in the same group or transfer to another group during random walks is determined by the learned importance of attributes and structures, which will be described later. Formally, the definition of aggregate-path is formulated as follows:

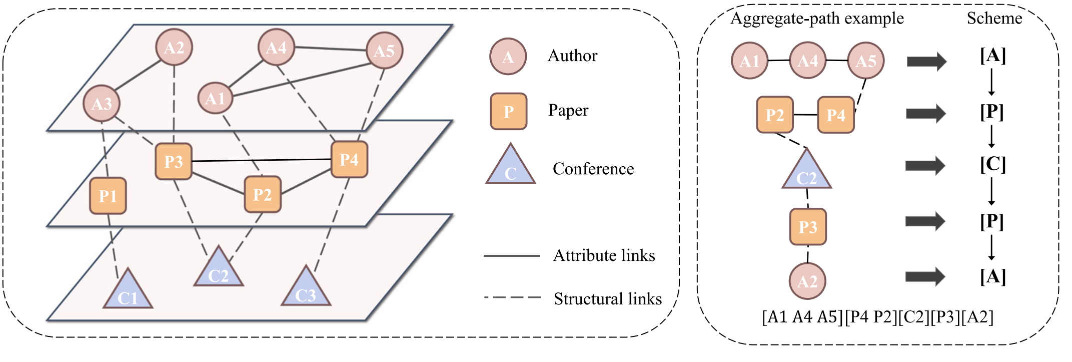

FIGURE 2 An illustrative example of network construction and aggregate-path-based random walks of an academic attributed heterogeneous network. The black dotted lines represent the structural links. The black solid lines represent the attribute links. Given an aggregate-path scheme $[A][P][C][P][A]$, we can obtain a node sequence of $[A1\ A4\ A5][P4\ P2][C2][P3][A2]$ as shown in the right part [Color figure can be viewed at wileyonlinelibrary.com]

Definition 4 (Aggregate-path). Given the network $G' = (V, E')$ constructed from an attributed heterogeneous network $G = (V, E, A)$, the aggregate-path scheme P on G' is defined in the form of $[T_1] \xrightarrow{R_1} [T_2] \xrightarrow{R_2} \cdots \xrightarrow{R_{L-1}} [T_L]$ (abbreviated as $[T_1][T_2] \cdots [T_L]$), where $R_i \in R$, $T_i \in T$. Specifically, we denote $[T_i] = [T_i^1 \longrightarrow T_i^2 \longrightarrow \cdots \longrightarrow T_i^{s_i}]$ (abbreviated as $[T_i^1 T_i^2 \cdots T_i^{s_i}]$) as the group or node sequence aggregated by nodes of type T_i, where s_i is the length of the group, T_i^j is the jth node in the group of type T_i and connects T_i^{j-1} through the attribute link. The last node in $[T_i]$ connects the first node in $[T_{i+1}]$ through the structural link.

Example 2. As an illustration, Figure 2 describes a network constructed from Figure 1, where the black solid lines represent attribute links and the black dotted lines represent structural links. Given the scheme $[A][P][C][P][A]$, the aggregate-path-based random walks can generate a path $[A1\,A4\,A5][P4\,P2][C2][P3][A2]$ shown in the right part of Figure 2.

4.2.2 | Adaptive aggregate-path-based random walk strategy

To begin with, we formalize definitions of the attribute layer and structural layer for each node. Given a node v, its attribute layer consists of N_v^{attr} and E_v^{attr}, where N_v^{attr} denotes nodes connected to v through attribute links and E_v^{attr} denotes attribute links between v and nodes in N_v^{attr}. Similarly, v's structural layer is composed of N_v^{stru} and E_v^{stru}, where N_v^{stru} denotes nodes connected to v through structural links which meanwhile belong to the following type group defined by the aggregate-path scheme P, E_v^{stru} denotes structural links between v and nodes in N_v^{stru}. For example, for node $P3$ in path $[A1\,A4\,A5][P4\,P2][C2][P3]$ of an academic network shown in Figure 2 with the aggregate-path scheme of $[A][P][C][P][A]$, nodes in $P3$'s attribute layer are $P2$ and $P4$ while nodes in its structural layer are $A2$ and $A3$. As a consequence, in aggregate-path-based random walks, the problem of whether to explore in the same group or transfer to another group is equivalent to the problem of whether to traverse in the attribute layer or in the structural layer.

Overall, considering an aggregate-path-based random walk that resides at node v, we sample through two phases: (1) Decide on whether to explore in v's attribute layer or in v's structural layer based on the learned importance of node attributes and topological structures. (2) Sample the next node in the selected v's attribute/structural layer.

Given a node v, it is intuitive that more attention should be paid on the layer where there exist more similar nodes connected to it.[30] To this end, we define the importance of v's attribute/structural layer in the following:

$$\rho_v^h = \frac{\text{mean}\left(\left\{w_{vu} \,\middle|\, u \in N_v^h\right\}\right)}{\text{mean}(\{w_{st} | s \in V_{\phi(v)}, t \in V_{\phi(u)}\})}, \tag{1}$$

where h can be either h_{attr} or h_{stru} that represents v's attribute or structural layer, w_{ij} is the edge weight between node i and j, N_v^h is the set of nodes in layer h connected to v, $V_{\phi(v)}$ and $V_{\phi(u)}$ denote the set of nodes whose type is the same with nodes v and u, respectively. $\rho_v^{(h)}$ is the importance of layer h, which is the normalized average edge weight of layer h. A larger $\rho_v^{(h)}$ indicates greater importance of layer h for node v, thus there exists higher probability to explore in layer h next. Overall, we can decide on whether to explore in v's attribute layer or in v's structural layer based on the following normalized probability distribution:

$$P_v(h) = \begin{cases} \dfrac{\rho_v^{(\text{attr})}}{\rho_v^{(\text{attr})} + \rho_v^{(\text{stru})}}, & h = \text{attr}, \\ \dfrac{\rho_v^{(\text{stru})}}{\rho_v^{(\text{attr})} + \rho_v^{(\text{stru})}}, & h = \text{stru}, \end{cases} \tag{2}$$

where $P_v(h)$ represents the probability of exploring in layer h next. Then, for an aggregate-path-based random walk that resides at node v, the next node u is sampled from the selected layer h by the weighted sampling:

$$P_v^h(u) = \frac{w_{vu}}{\sum_{t \in N_v^h} w_{vt}}, \tag{3}$$

where N_v^h represents nodes in layer h connected to node v and u is a node sampled from N_v^h.

Since the importance of node attributes and topological structures varies across networks, that is, some networks rely more on attributes while others rely more on structures, we design a more adaptive strategy of random walk to strike a balance between attributes and structures.

Intuitively, it is expected to traverse more nodes on the more important layer. Inspired by this idea, we adaptively fuse attributes and structures by introducing two hyperparameters α and β that influence the magnitude of $\rho_v^{(\text{attr})}$ and $\rho_v^{(\text{stru})}$, respectively. In the meanwhile, in an extreme case, the aggregate-path-based random walk strategy might only select the next node from the attribute layer, resulting in the overlook of nodes with other types. Therefore, to prevent repeatedly traversing in the same attribute layer, α should also play the role of attenuation factor, which decays as the path length of the current attribute layer increases. As a consequence, Equation (2) is converted to the following equation:

$$P_v(h) = \begin{cases} \dfrac{\alpha^{j-1} * \rho_v^{(h)}}{\rho_v^{(\text{attr})} + \rho_v^{(\text{stru})}}, & h = \text{attr}, \\ \dfrac{\beta * \rho_v^{(h)}}{\rho_v^{(\text{attr})} + \rho_v^{(\text{stru})}}, & h = \text{stru}, \end{cases} \tag{4}$$

where j is the path length of the current attribute layer (e.g., $j = 3$ for a walk AAPCPPP) and the range of α and β are both $(0, 1]$. A high α and a low β represent the higher importance of node attributes. On the contrary, a low α and a high β indicate the higher importance of topological structures. Therefore, the proposed strategy can strike a balance between attributes and structures by controlling the values of α and β.

Compared with traditional meta-path-based methods, the proposed aggregate-path-based random walk strategy introduces attribute proximity into random walks and adaptively balances the importance of node attributes and structural heterogeneity, which boosts the representation ability of node embeddings.

4.3 | Representation learning architecture

We now explore two approaches to generate effective node representations from the obtained aggregate-path-based random walk sequences.

4.3.1 | Skip-gram model

In this subsection, we propose to adopt the skip-gram model[31] to generate node representations. Specifically, the goal is to maximize the co-occurrence probability among nodes appearing within a window in all sequences, that is,

$$\underset{\theta}{\operatorname{argmax}} \prod_{v \in V} \prod_{c \in N(v)} p(c|v;\, \theta), \tag{5}$$

where $N(v)$ is the neighborhood of node v which appears within a window in a sequence, and $p(c|v;\, \theta)$ denotes the conditional probability of having a neighborhood c given a center node v.

To achieve efficient optimization, we further apply negative sampling technique[32,33] and minimize the following objective function:

$$O(X) = -\log \sigma(X_c \cdot X_v) - \sum_{k=1}^{K} \log \sigma(1 - X_{u_k} \cdot X_v), \tag{6}$$

where X is the node representations, u_k is the kth negative node sampled for node v, and K is the number of negative samples.

4.3.2 | Graph recurrent network

Apart from the skip-gram approach, there exists another solution to generate node representations on sequences, namely, RNN.[34] Figure 3 illustrates the procedure of obtaining the representation of a walk sequence with the RNN architecture.

Given a walk sequence of length L, we first initialize node representations with their attribute representations A_i. Then a fully connected layer is utilized to reduce and unify the dimension of diverse representations:

$$x_i = \sigma(A_i W_a + b_a), \tag{7}$$

where σ is the activation function, such as tanh, $W_a \in \mathbb{R}^{m \times d}$ and m is the dimension of node attribute which varies with the node type. Then a forward hidden state sequence (i.e., $\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_L$) and a backward hidden state sequence (i.e., $\overleftarrow{q}_1, \overleftarrow{q}_2, \ldots, \overleftarrow{q}_L$) are learned by employing the bidirectional RNN, such as bidirectional gated recurrent units (GRU).[35] Specifically, the forward hidden state sequence $\vec{q}_i$ is calculated as follows (the backward state sequence is calculated in the same way except that the input sequence is reversed):

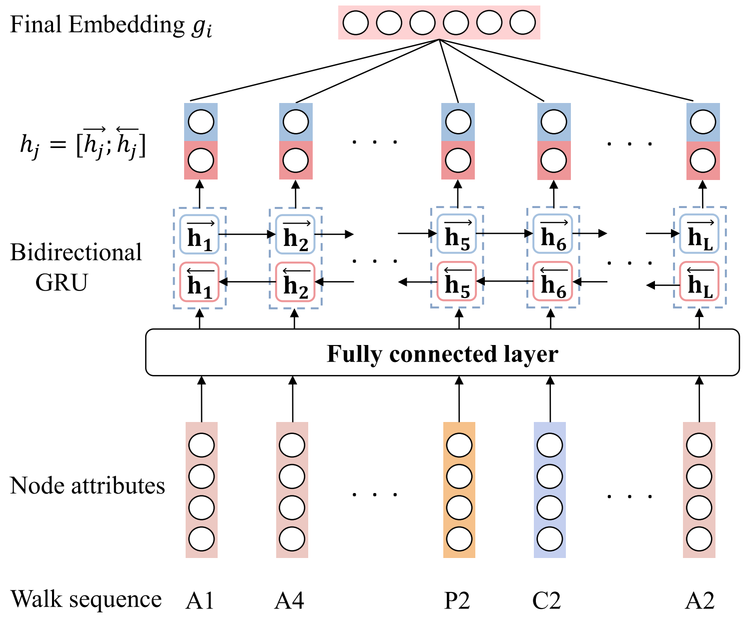

FIGURE 3 The procedure of generating the representation of a walk sequence with the RNN architecture. GRU, gated recurrent units; RNN, recurrent neural network [Color figure can be viewed at wileyonlinelibrary.com]

$$\begin{aligned} z_t &= \text{sigmoid}(W_z x_t + U_z \vec{q}_{t-1}), \\ r_t &= \text{sigmoid}(W_t x_t + U_t \vec{q}_{t-1}), \\ \widetilde{q_t} &= \tanh(W x_t + U(r_t \circ \vec{q}_{t-1})), \\ \vec{q}_t &= (1 - z_t) \circ \vec{q}_{t-1} + z_t \circ \widetilde{q_t}, \end{aligned} \tag{8}$$

where z_t and r_t are the update gate vector and reset gate vector, respectively, W and U are parameter matrices of the GRU. The output of the bidirectional GRU layer is the concatenation of forward and backward hidden state vectors, that is, $q_i = [\vec{q}_i, \overleftarrow{q}_i]$.

More generally, if we repeat random walks K times with length L starting from each node, there exist KL neighbors for each node, which correspond to KL hidden states in the bidirectional GRU layer. To obtain the final representations, we first exploit a pooling method (e.g., mean) to merge all the K sequences into $(\hat{g}_1, \hat{g}_2, ..., \hat{g}_L)$, followed by another pooling method to combine $(\hat{g}_1, \hat{g}_2, ..., \hat{g}_L)$ into g_i, which is the final embedding of node i. Similar to the loss function defined in Equation (6), this method leverages the negative sampling technique to keep neighbors in sequences to be close and others to be far apart.

The pseudocode of aggregate-path-based random walk strategy in AHNA is shown in Algorithm 1. Specifically, we first construct a network G' that contains both structural relations and attribute proximity. Then the Aggregate-path-based Random Walk strategy (APRandomWalk) is conducted on G', where α and β are used to adapt various networks with different importance between attributes and structures. After generating walk sequences, skip-gram or bidirectional GRU with negative sampling is performed to generate final representations.

Algorithm 1: Aggregate-path based random walks

Input: The constructed network $G^{'}=(V,E^{'},A)$, an aggregate-path scheme P, hyper-parameters α and β, dimensions d, walks per node w, walk length l, neighborhood size r,negative samples k.

Output: The latent node Embeddings $X \in \mathbb{R}^{|V| \times d}$

Initialize walks to Empty;
for $i = 1 \rightarrow w$ **do**
  for $v \in V$ **do**
    walk = APRandomWalk($G^{'}$,P,v,l);
    Append walk to walks;
return walks;
Function `APRandomWalk`($G^{'}$,P,v,l)**:**
  walk[1]= v;
  for $i = 1 \rightarrow l$ **do**
    calculate $\rho_v^{(h)}$ according to Equation 1;
    choose a layer S according to Equation 4;
    choose a node u according to Equation 3;
    walk[$i+1$]= u;
  return walk;

The time complexity of the proposed AHNA consists of two parts, that is, the complexity of generating aggregate-path-based random walks and the complexity of learning representations from the obtained walks. With the network containing both attribute and structural links constructed in advance, we utilize the alias method[36,37] to select the next node in each random walk, whose time complexity is $O(1)$. Therefore, the time complexity of generating a walk of length l is $O(l)$. Then, for each walk, the time complexity of skip-gram model is $O(lrkd)$ while one of the RNN models is $O(ld^2)$, where r, k, d is the value of window size, negative sample, and embedding dimension, respectively. Therefore, the whole time complexity of each walk is $O(lrkd)$ for AHNA with skip-gram and $O(ld^2)$ for AHNA with RNN.

5 | EXPERIMENT

In this section, three classic benchmark tasks (i.e., link prediction, community detection, and node classification) and a case study of relevance search are conducted on AHNA and six state-of-the-art methods to evaluate the effectiveness of the proposed method. The adaptability and parameter analysis of AHNA are evaluated at last.

5.1 | Data sets

In the experiments, we utilize three real-world data sets as follows:

- *ACM*[38]: We extract 7755 authors, 4132 papers, and 13 conferences from the ACM data set to construct the attributed heterogeneous network with two types of edges (i.e., edges between authors and papers, edges between papers and conferences). The attribute representations of authors and papers are one-hot vectors of affiliations to which authors belong and keywords that appear in more than 1% of papers after removing stop words from titles and abstracts, respectively. In this data set, conferences are labeled according to the subjects they belong to. The label of each paper is the same as the label of conference where it is published and the author's label is set to be the one that appears most frequently in the list of labels of papers that he/she published.

- *DoubanMovie*[39]: Consisting of 527 users, 1470 movies, and 30 types, three attributed heterogeneous networks are constructed on this data set (denoted as Douban1, Douban2, and Douban3, respectively, and collectively referred to as Douban), which have the same network structures, movie attributes but different user attributes. There exist two edge types in the network (i.e., edges between users and movies, edges between movies and types). Movies' attribute representations are one-hot vectors of directors, while users' attribute representations are one-hot vectors of social relationships, locations, and groups for Douban1, Douban2, and Douban3, respectively.
- *Movielens*[40]: This is a movie rating data set composed of 943 users, 1656 movies, and 18 genres with two edge types (i.e., edges between users and movies, edges between movies and genres). Given the rating matrix, the attribute representations of users/movies are one-hot vectors of their k nearest neighbors on users/movies, where Pearson's coefficient between nodes' rating vectors is utilized to measure similarities and k is set to be 50.

The statistics of these data sets are summarized in Table 4, where **A** represents author/user, **B** stands for paper/movie, and **C** denotes conference/type/genre corresponds to different data sets. Empirically, we choose [**A**][**B**][**C**][**B**][**A**] as the aggregate-path scheme on all data sets.

5.2 | Baselines

In the experiments, we abbreviate AHNA with skip-gram as *AHNA-Skip*, while AHNA with RNN is abbreviated as *AHNA-RNN*. We compare our AHNA with several methods in recent years, which can be categorized into three classes: (1) methods designed for plain networks, (2) methods designed for attributed homogeneous networks, and (3) methods designed for attributed heterogeneous networks.

- *DeepWalk*[10]: A network representation algorithm based on random walks and skip-gram architecture, which is suitable for homogeneous plain networks.

TABLE 4 Statistics of data sets

Data set	A	B	C	A–B	B–C	A–A	B–B
Douban1	527	1470	30	5819	3577	325	2815
Douban2	527	1470	30	5819	3577	7944	2815
Douban3	527	1470	30	5819	3577	1437	2815
Movielens	943	1656	18	99,963	2858	34,588	82,741
ACM	7755	4132	13	13,408	4132	22,228	7625

Note: **A** represents author/user, **B** stands for paper/movie, and **C** denotes conference/type/genre corresponds to different data sets. **A**–**A** and **B**–**B** represent the attribute links built by our method.

Abbreviation: ACM, Association for Computing Machinery.

- *Metapath2vec*[6]: A meta-path-based network representation method, which is designed for plain heterogeneous networks. This method obtains node sequences guided by meta-path, followed by a skip-gram model to obtain final node representations.
- *AANE*[14]: An attributed homogeneous network representation method based on matrix factorization. Specifically, this method decomposes the attribute affinity matrix of the given network into the inner product of the node representation matrix, and minimizes the decomposition error and representation differences between similar nodes simultaneously.
- *ANRL*[41]: It is an attributed homogeneous network representation method that utilizes a neighbor enhancement autoencoder and an attribute-aware skip-gram model jointly to model node attributes.
- *DIME-SH*[15]: This is an important component of framework DIME proposed by this paper, which can be applied to attributed heterogeneous networks. This method converts node attributes to nodes in the network and trains a series of autoencoders with different meta-path proximity matrices as input to learn embeddings in the latent feature space.
- *HetGNN*[22]: A heterogeneous graph neural network model that considers both topological structures and node attributes. This model encodes diverse attributes as the initial node representations and aggregates heterogeneous neighbors according to different types of nodes and attributes to generate node representations.

5.3 | Parameter settings

To achieve a fair comparison, we follow the authors' suggested hyperparameter settings and set the embedding dimension to be 128 (same as AHNA) for all baselines. For Metapath2vec, we employ *ABCBA* as the predefined meta-path scheme. For DIME-SH, we utilize seven meta-paths, that is, *AA*, *ABA*, *ABCBA*, *BB*, *BCB*, *BAB*, and *CBABC*, where *A*, *B*, and *C* are the same meanings as those in Table 4.

We implement our proposed method on the basis of Pytorch and conduct grid search to choose the value of walk length l from $\{10, 20, 40, 80, 100\}$, window size r from $\{10, 15, 20\}$, α and β from $\{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$. The value of walks per node defaults to be 20 for AHNA-Skip and 80 for AHNA-RNN. The parameter settings for different data sets in the experiments are detailed as follows:

For AHNA-Skip, we employ Adam optimizer and the learning rate is 0.05 for Douban and Movielens, 0.01 for ACM. The batch size, walks per node, walk length, and window size are 128, 20, 80, and 10 for Douban, Movielens, classification, and case study of relevance search on ACM and 1000, 20, 20, and 20 for community detection on ACM. The number of negative sample is set to be 10 for Douban, Movielens, and community detection on ACM, 5 for classification and case study of relevance search on ACM.

For AHNA-RNN, Adam optimizer is utilized and the learning rate is 0.00001 for all data sets. The values of walks per node, walk length are set to be 80, 20 for Douban1 and ACM, 80, 10 for Movielens, Douban2, and Douban3. The values of negative samples and batch size are set to be 5 and 64, respectively, for all data sets.

The embedding dimension is 128 for all data sets. As for α and β, they are set to be 0.3, 1 for Douban1, Douban3, ACM, 0.5, 0.5 for Douban2 and 1, 0.3 for Movielens. For other methods, we keep the parameters following the suggestions in their papers or source codes. All the reported experimental data are the average results by 10 runs.

5.4 | Link prediction

To validate the effectiveness of the proposed method, we first conduct link prediction on Douban and Movielens, which measures the ability of retaining topological structures of networks. To be more specific, we randomly pick up 30% structural links of each data set as positive samples and generate an equal number of negative samples that are nonexisting edges in the network. After learning on the training set containing the rest 70% structural links, the cosine similarity between representations of two nodes on each edge is employed to get a score that measures whether the test edge exists. Ultimately, the area under curve (AUC) score is reported to measure the performance.

Table 5 shows the results of performance comparison on link prediction task. Notably, attributed heterogeneous methods (i.e., DIME-SH and HetGNN) generally perform superior to methods of attributed homogeneous networks (i.e., AANE and ANRL), demonstrating the usefulness of structural heterogeneity. Besides, our proposed method AHNA-Skip can always achieve the highest score on all data sets, indicating the benefits of adaptively fusing node attributes and heterogeneous structures. In addition, it can be observed that AHNA-RNN performs worse than AHNA-Skip. It is reasonable because, unlike AHNA-Skip which optimizes each neighbor as an independent positive sample and captures structural links directly, AHNA-RNN pools the whole walk sequences of each node into a representation, which explores structural links indirectly and thus weakens the utilization of structural links.

5.5 | Community detection and node classification

We conduct community detection and node classification by utilizing k-means algorithm[42] and logistic regression classifier[43] with varying the training ratio from 20% to 80%, respectively, on the ACM data set. Normalized Mutual Information (NMI)[44] is adopted as the evaluation metric of community detection while Micro-F1 and Macro-F1 are adopted for node classification.

Table 8 depicts the performance comparison of community detection, Tables 6 and 7 demonstrate the results of node classification. It is worth noting that heterogeneous structural information plays a more significant role than node attributes on the ACM data set (detailed in Section 5.7). Owing to this fact, methods designed for attributed homogeneous networks (i.e., AANE and ANRL) achieve low scores and Metapath2vec outperforms them due to the ability of leveraging the structural heterogeneity. What is more, AHNA-RNN can always obtain the best performance, which reflects the benefits of adaptively incorporating node attributes and topological structures. In addition, there is a phenomenon that AHNA-Skip does not perform as

TABLE 5 AUC scores of link prediction on Douban and Movielens

Data sets	DeepWalk	Metapath2vec	ANRL	AANE	DIME-SH	HetGNN	AHNA-Skip	AHNA-RNN
Douban1	0.5064	0.5146	0.5232	0.5172	0.6143	0.7014	**0.7052**	0.6313
Douban2	0.5064	0.5146	0.5171	0.5082	0.5602	0.6998	**0.7149**	0.6413
Douban3	0.5064	0.5146	0.5220	0.5137	0.5681	0.6956	**0.7142**	0.6407
Movielens	0.5017	0.5037	0.6059	0.5546	0.6824	0.7189	**0.8819**	0.7943

Abbreviations: AANE, accelerated attributed network embedding; AHNA, Attributed Heterogeneous Network embedding based on Aggregate-path; ANRL, attributed network representation learning; AUC, area under curve; HetGNN, heterogeneous graph neural network; RNN, recurrent neural network.

TABLE 6 Node classification results of authors on ACM data set

Metric	Ratio (%)	DeepWalk	Metapath2vec	ANRL	AANE	DIME-SH	HetGNN	AHNA-Skip	AHNA-RNN
Macro-F1	20	0.4954	0.6701	0.5603	0.5443	0.7105	0.8756	0.8582	**0.8810**
	40	0.5345	0.6971	0.5611	0.5506	0.7114	0.8795	0.8694	**0.8856**
	60	0.5455	0.7025	0.5556	0.5498	0.7085	0.8763	0.8686	**0.8856**
	80	0.5522	0.7141	0.5581	0.5521	0.7198	0.8800	0.8756	**0.8895**
Micro-F1	20	0.3127	0.5618	0.2394	0.2693	0.5945	0.8216	0.8052	**0.8294**
	40	0.2816	0.5728	0.2399	0.2610	0.5904	0.8279	0.8204	**0.8394**
	60	0.2603	0.5775	0.2385	0.2528	0.5884	0.8239	0.8198	**0.8421**
	80	0.2488	0.5875	0.2397	0.2508	0.6021	0.8252	0.8258	**0.8439**

Abbreviations: AANE, accelerated attributed network embedding; ACM, Association for Computing Machinery; AHNA, Attributed Heterogeneous Network embedding based on Aggregate-path; ANRL, attributed network representation learning; HetGNN, heterogeneous graph neural network; RNN, recurrent neural network.

TABLE 7 Node classification results of papers on ACM data set

Metric	Ratio (%)	DeepWalk	Metapath2vec	ANRL	AANE	DIME-SH	HetGNN	AHNA-Skip	AHNA-RNN
Micro-F1	20	0.5015	0.7223	0.6227	0.5891	0.8625	0.8855	0.8811	**0.8880**
	40	0.5483	0.7677	0.6301	0.5890	0.8640	0.8844	0.8809	**0.8896**
	60	0.5658	0.7831	0.6208	0.5847	0.8667	0.8869	0.8845	**0.8915**
	80	0.5993	0.7983	0.6368	0.6094	0.8730	0.8918	0.8947	**0.9012**
Macro-F1	20	0.3264	0.6308	0.3558	0.2790	0.7975	0.8313	0.8322	**0.8347**
	40	0.2993	0.6647	0.3707	0.2792	0.7941	0.8271	0.8280	**0.8325**
	60	0.2786	0.6787	0.3727	0.2815	0.7976	0.8281	0.8306	**0.8336**
	80	0.2775	0.6875	0.3862	0.2993	0.7966	0.8287	0.8386	**0.8392**

Abbreviations: AANE, accelerated attributed network embedding; ACM, Association for Computing Machinery; AHNA, Attributed Heterogeneous Network embedding based on Aggregate-path; ANRL, attributed network representation learning; HetGNN, heterogeneous graph neural network; RNN, recurrent neural network.

TABLE 8 NMI results of community detection on ACM data set

Method	Author	Paper	Overall
DeepWalk	0.0002	0.0005	0.0007
Metapath2vec	0.0082	0.2419	0.2501
ANRL	0.0041	0.0469	0.0500
AANE	0.0009	0.0177	0.0186
DIME-SH	0.0156	0.3183	0.3339
HetGNN	0.4455	0.4531	0.8986
AHNA-Skip	0.4249	0.4645	0.8894
AHNA-RNN	**0.4513**	**0.4649**	**0.9102**

Abbreviations: AANE, accelerated attributed network embedding; ACM, Association for Computing Machinery; AHNA, Attributed Heterogeneous Network embedding based on Aggregate-path; ANRL, attributed network representation learning; HetGNN, heterogeneous graph neural network; NMI, Normalized Mutual Information; RNN, recurrent neural network.

well as AHNA-RNN. We argue that AHNA-RNN initializes node representations with their attributes, which strengthens the utilization of node attributes, therefore performs better on these tasks requiring a good use of attributes.

5.6 | Relevance search: Superiority proof of AHNA

In this section, we present a case study of relevance search to validate AHNA's superior ability of incorporating attributes and structures compared with existing attributed heterogeneous methods. For the query paper "Effective keyword-based selection of relational database", Table 9 presents top-5 papers returned by DIME-SH, HetGNN and AHNA, whose rankings are sorted based on cosine similarities between representations.

From this table: (1) All papers returned by DIME-SH, HetGNN, and AHNA belong to the SIGMOD conference (i.e., the P–C structural links connected to the query paper), yet AHNA-Skip simultaneously returns more papers having coauthor relationships with the query paper (i.e., the A–P structural links connected to the query paper), implying that AHNA-Skip has the superior ability of capitalizing on topological structures. Besides, as analyzed in link prediction, AHNA-Skip is more suitable for directly capturing structural links than AHNA-RNN, thus returning more papers with coauthor relationship with the query one; (2) both AHNA-Skip and AHNA-RNN list top-5 papers with higher similarities of attributes (i.e., the number of same keywords between the returned paper and the query one) than DIME-SH and HetGNN, which verifies that proposed methods can take better advantages of node attributes. (3) In terms of the integrated ability of capturing both attributes and structures, AHNA incorporates them better thanks to the adaptive random walk strategy that strikes a balance based on the learned importance.

5.7 | Adaptability analyses and ablation study of AHNA

In this section, we investigate the adaptability of AHNA on data sets containing different importance of attributes and structures. As claimed in the method, α and β control the importance of attributes and structures, respectively. Therefore, we vary α and β from 0.1 to 1 to

TABLE 9 Case study of relevant paper search to reveal AHNA's superior ability of incorporating attributes and structures

Query: Effective keyword-based selection of relational databases (SIGMOD 2007) (Bei Yu, Guoliang Li, Karen Sollins, and Anthony K. H. Tung)					
			Coauthor	**Same conference**	**Keywords**
Model	**Rank**	**Returned paper**	**A–P**	**P–C**	**Same attribute**
DIME-SH	1	Continuous monitoring of top-*k* queries over sliding windows (SIGMOD 2006) (Kyriakos MOURATIDIS, Spiridon BAKIRAS, Dimitris Papadias)	No	Yes	14
	2	Approximating multidimensional aggregate range queries over real attributes (SIGMOD 2000) (Dimitrios Gunopulos, George Kollios, Vassilis J. Tsotras, Carlotta Domeniconi)	No	Yes	10
	3	Efficient type-ahead search on relational data: a TASTIER approach (SIGMOD 2009) (Guoliang Li, Shengyue Ji, Chen Li, Jianhua Feng)	Yes	Yes	19
	4	Time-parameterized queries in spatiotemporal databases (SIGMOD 2002) (Yufei Tao, Dimitris Papadias)	No	Yes	14
	5	Dictionary-based order-preserving string compression for main memory column stores (SIGMOD 2009) (Carsten Binnig, Stefan Hildenbrand, Franz Färber)	No	Yes	8
HetGNN	1	EASE: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data			

TABLE 9 (Continued)

Query: Effective keyword-based selection of relational databases (SIGMOD 2007) (Bei Yu, Guoliang Li, Karen Sollins, and Anthony K. H. Tung)					
Model	**Rank**	**Returned paper**	**Coauthor A–P**	**Same conference P–C**	**Keywords Same attribute**
		(SIGMOD 2008)			
		(Guoliang Li, Beng Chin Ooi, Jianhua Feng, Jianyong Wang, Lizhu Zhou)			
		Efficient type-ahead search on relational data: a TASTIER approach	Yes	Yes	19
		(SIGMOD 2009)			
		(Guoliang Li, Shengyue Ji, Chen Li, Jianhua Feng)			
	3	Searching trajectories by locations: an efficiency study	No	Yes	13
		(SIGMOD 2010)			
		(Zaiben Chen, Heng Tao Shen, Xiaofang Zhou, Yu Zheng, Xing Xie)			
	4	An adaptive peer-to-peer network for distributed caching of OLAP results	No	Yes	9
		(SIGMOD 2002)			
		(Panos Kalnis, Wee Siong Ng, Beng Chin Ooi, Dimitris Papadias, Kian-Lee Tan)			
	5	Private queries in location-based services: anonymizers are not necessary	No	Yes	8
		(SIGMOD 2008)			
		(Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, Kian-Lee Tan)			
AHNA-Skip	1	Efficient type-ahead search on relational data: a TASTIER approach	Yes	Yes	**19**
		(SIGMOD 2009)			
		(Guoliang Li, Shengyue Ji, Chen Li, Jianhua Feng)			
	2	EASE: an effective 3-in-1 keyword search method for unstructured,	Yes	Yes	**15**

(Continues)

TABLE 9 (Continued)

Query: Effective keyword-based selection of relational databases (SIGMOD 2007) (Bei Yu, Guoliang Li, Karen Sollins, and Anthony K. H. Tung)					
Model	Rank	Returned paper	Coauthor A–P	Same conference P–C	Keywords Same attribute
		semi-structured and structured data (SIGMOD 2008) (Guoliang Li, Beng Chin Ooi, Jianhua Feng, Jianyong Wang, Lizhu Zhou)			
	3	A graph method for keyword-based selection of the top-*k* databases (SIGMOD 2008) (Quang Hieu Vu, Beng Chin Ooi, Dimitris Papadias, Anthony K. H. Tung)	Yes	Yes	**25**
	4	Combining keyword search and forms for ad hoc querying of databases (SIGMOD 2009) (Eric Chu, Akanksha Baid, Xiaoyong Chai, AnHai Doan, Jeffrey Naughton)	No	Yes	**14**
	5	Spark: top-*k* keyword query in relational databases (SIGMOD 2007) (Yi Luo, Xuemin Lin, Wei Wang, Xiaofang Zhou)	No	Yes	**24**
AHNA-RNN	1	Estimating alphanumeric selectivity in the presence of wildcards (SIGMOD 1996) (P. Krishnan, Jeffrey Scott Vitter, Bala lyer)	No	Yes	**17**
	2	A graph method for keyword-based selection of the top-*k* databases (SIGMOD 2008) (Quang Hieu Vu, Beng Chin Ooi, Dimitris Papadias, Anthony K. H. Tung)	Yes	Yes	**25**

TABLE 9 (Continued)

Query: Effective keyword-based selection of relational databases					
(SIGMOD 2007) (Bei Yu, Guoliang Li, Karen Sollins, and Anthony K. H. Tung)					
Model	**Rank**	**Returned paper**	**Coauthor A–P**	**Same conference P–C**	**Keywords Same attribute**
	3	Query Optimization In Compressed Database Systems (SIGMOD 2001) (Zhiyuan Chen, Johannes Gehrke, Flip Korn)	No	Yes	**18**
	4	A rule-based object/task modeling approach (SIGMOD 1986) (Qiming Chen)	No	Yes	**9**
	5	Just-in-time query retrieval over partially indexed data on structured P2P overlays (SIGMOD 2008) (Sai Wu, Jianzhong Li, Beng Chin Ooi, Kian-Lee Tan)	No	Yes	**13**

Note: The coauthors of the query paper in the returned papers are marked with the same color.

observe the influences on the experimental results. Since AHNA-RNN treats node attributes as initial node representations, which enhances the importance of attributes in advance, we choose AHNA-Skip for verification in this section to reveal the relationship of importance between attributes and structures more purely. The experimental results are illustrated in Figure 4 in the form of heat maps.

As depicted in Figure 4: (1) On Douban1, Douban3, and ACM, the experimental results become better with the increase of the importance of structural information, and on the contrary become worse when increasing the attribute importance, which reflects that the structural information plays a more significant role than attributes on these data sets. (2) On Movielens, the proposed method achieves the highest score with a low β and a high α, indicating the higher importance of attributes than structures on Movielens. (3) On Douban2, it seems that attributes and structures are relatively balanced because the experimental results become better when α and β are similar.

In Figure 4, when $\alpha = \beta$, the model does not consider the different importance of attributes and structures, which can be regarded as the ablation study. It can be observed that our method which adaptively balances the importance of attributes and structures achieves greater improvements, verifying the advantage of considering the importance balance between structural relations and attribute proximity.

Therefore, we can conclude from the observations that, by controlling values of α and β, AHNA can always achieve an adaptive fusion of topological structures and node attributes for data sets with diverse attribute and structure importance, which verifies the adaptability and effectiveness of our method.

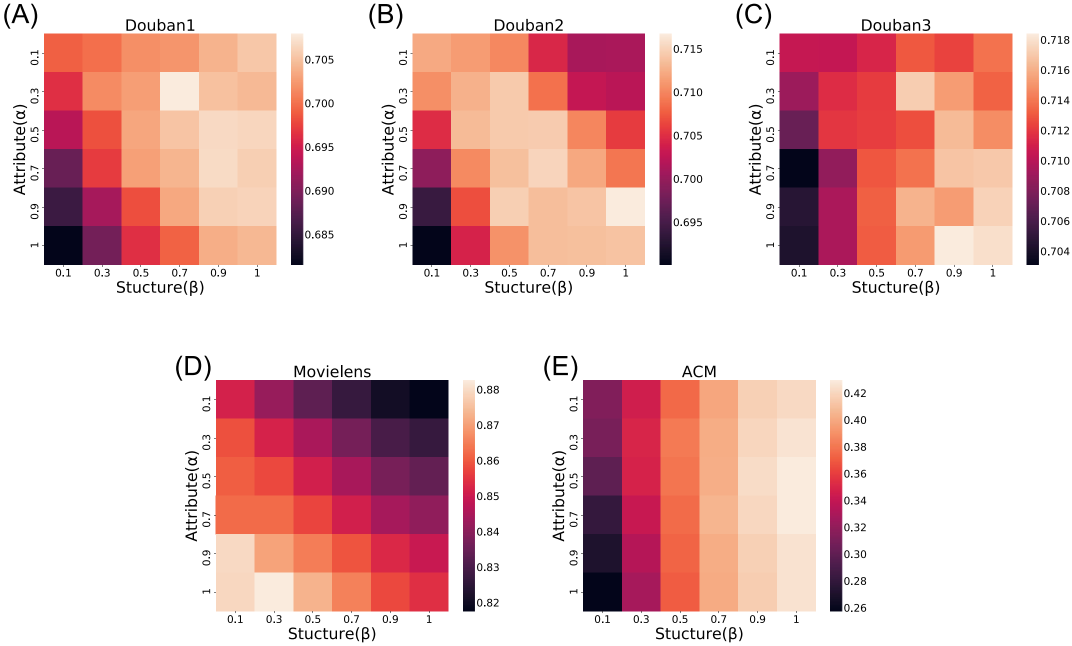

FIGURE 4 Heat maps of experimental results on Douban, Movielens, and ACM when varying α and β from 0.1 to 1. (A–D) illustrate the results of link prediction on Douban1, Douban2, Douban3, and Movielens while (E) depicts the community detection results of authors on ACM. ACM, Association for Computing Machinery [Color figure can be viewed at wileyonlinelibrary.com]

5.8 | Parameter sensitivity analysis

In this section, we investigate the sensitivity of parameters, including the embedding dimension d, window size r, walks per node w, and walk length l on AHNA-Skip, embedding dimension d, walks per node w, walk length l on AHNA-RNN, following the parameter settings introduced in Section 5.2. Experiments are conducted on link prediction on Douban, Movielens, and community detection of papers on ACM by fixing other parameters when evaluating each of them. The detailed performances are shown in the form of boxplots in Figures 5 and 6, and we summarize observations as follows.

For AHNA-Skip, as presented in Figure 5A, the performance rises with the increase of d on Douban, yet drops on the other data sets when d is too large, which might owe to the overfitting phenomenon. Besides, AHNA-Skip is relatively stable within a large range in terms of r, w, and l on Douban and ACM, and drops slightly when the parameters are either too small or too large, as observed from Figure 5B–5D. With regard to the Movielens data set, the performance

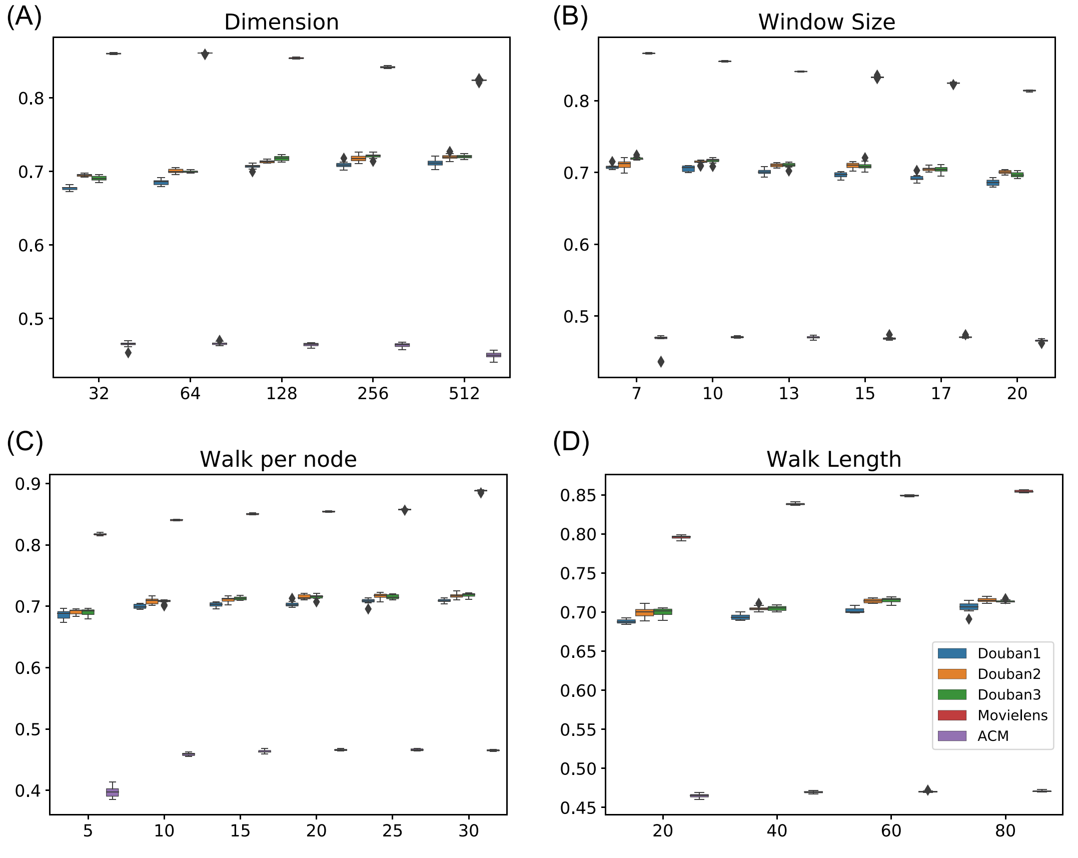

FIGURE 5 Parameter sensitivity of AHNA-Skip of link prediction on Douban, Movielens, and community detection of papers on ACM in the form of boxplots. (A) Dimension, (B) window size, (C) walk per node, and (D) walk length. Since the experimental results are quite stable, all boxes in the figure are small, where boxes at the bottom represent the results of ACM, those at the top are the results of Movielens and others in the middle are the results of Douban. ACM, Association for Computing Machinery; AHNA, Attributed Heterogeneous Network embedding based on Aggregate-path [Color figure can be viewed at wileyonlinelibrary.com]

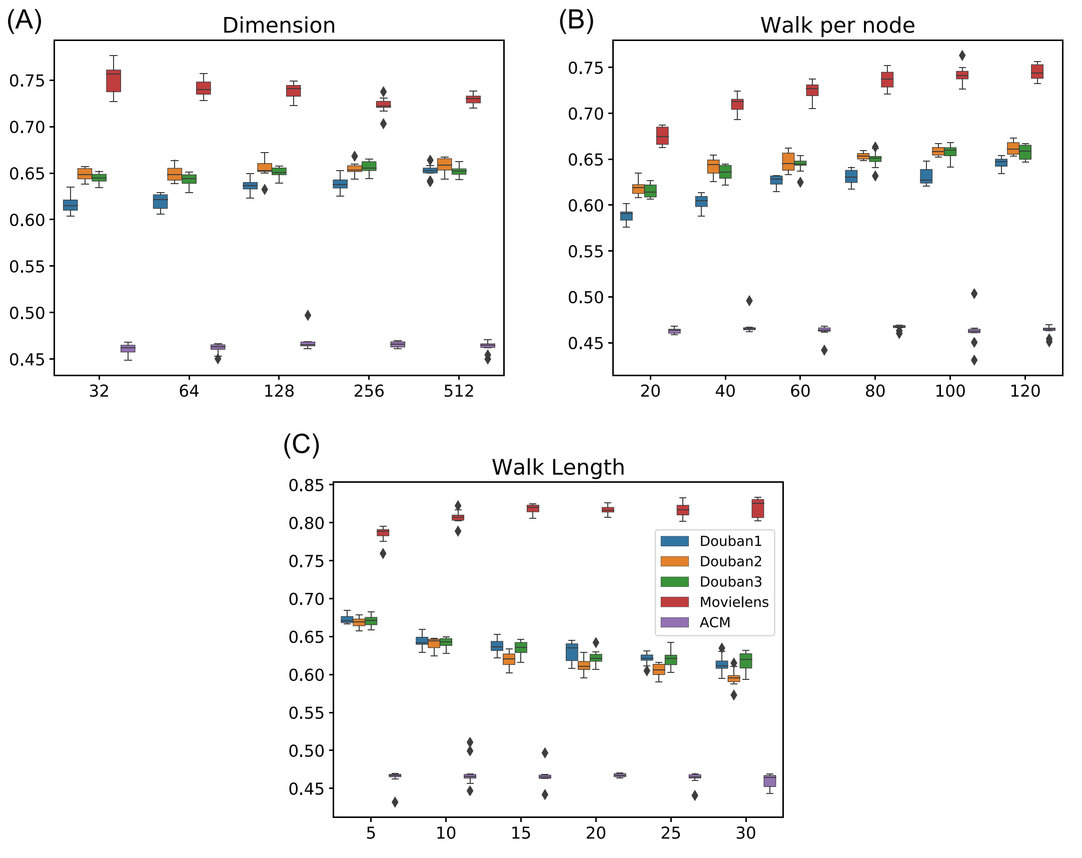

FIGURE 6 Parameter sensitivity of AHNA-RNN of link prediction on Douban, Movielens, and community detection of papers on ACM in the form of boxplots. (A) Dimension, (B) walk per node, and (C) walk length. ACM, Association for Computing Machinery; AHNA, Attributed Heterogeneous Network embedding based on Aggregate-path; RNN, recurrent neural network [Color figure can be viewed at wileyonlinelibrary.com]

shows a descending trend with the increase of r and presents a positive effect when increasing w and l.

For AHNA-RNN, the results of altering d and w as depicted in Figure 6A,B are similar to the AHNA-Skip as analyzed above except that the performance on ACM is more stable in AHNA-RNN. As for the parameter l in Figure 6C, the performance on ACM is relatively stable while Douban and Movielens appear opposite trends, which implies that Douban requires a smaller length to produce high-quality embeddings yet Movielens requires a larger one.

6 | CONCLUSION

In this paper, we cope with the attributed heterogeneous network representation problem and propose a novel model, AHNA, which incorporates structural relations and attribute proximity in a unified model. An adaptive random walk strategy based on aggregate-path is further designed to strike a balance between node attributes and topological structures. Extensive experiments are conducted on multiple tasks on three real-world data sets and experimental results validate the superiority of the proposed AHNA method against state-of-the-art methods. In particular, AHNA achieves better performance than existing attributed heterogeneous methods, especially on the case study of relevance search, verifying that the adaptive aggregate-path guided random walk strategy is beneficial for generating higher-quality node representations.

ACKNOWLEDGMENT

The research is supported by the Key-Area Research and Development Program of Guangdong Province (2020B010165003), the National Natural Science Foundation of China (62176269 and 11801595), and the Guangdong Basic and Applied Basic Research Foundation (2019A1515011043). This study is also supported by the UX Center, Netease Games. Lin Shu and Chuan Chen conceived of the presented idea. Lin Shu carried out the experiment and wrote the manuscript with support from Chuan Chen, Xingxing Xing, Xiangke Liao, and Zibin Zheng. Chuan Chen helped supervise the project and contributed to the final version of the manuscript. All authors provided critical feedback and helped shape the research, analysis, and manuscript.

ORCID

Lin Shu https://orcid.org/0000-0001-7468-8766
Chuan Chen https://orcid.org/0000-0002-7048-3445
Xingxing Xing https://orcid.org/0000-0003-3305-5158
Zibin Zheng https://orcid.org/0000-0002-7878-4330

REFERENCES

1. Zhang K, Mao Y, Leng S, et al. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. *IEEE Access*. 2016;4:5896-5907.
2. Wang Y, Feng C, Chen L, Yin H, Guo C, Chu Y. User identity linkage across social networks via linked heterogeneous network embedding. *World Wide Web*. 2019;22(6):2611-2632.
3. Rizi FS, Granitzer M. Signed heterogeneous network embedding in social media. In: Hung C-C, Černý T, Shin D, Bechini A, eds. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. ACM; 2020:1877-1880.
4. Lu Z, Wang Y, Zeng M, Li M. HNEDTI: prediction of drug–target interaction based on heterogeneous network embedding. In: Yoo I, Bi J, Hu X, eds. *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE; 2019:211-214.
5. Xiong Y, Ruan L, Guo M, et al. Predicting disease-related associations by heterogeneous network embedding. In: Zheng HJ, Callejas Z, Griol D, Wang H, Hu Xiaohua, Schmidt HHHW, Baumbach J, Dickerson J, Zhang L, eds. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE Computer Society; 2018:548-555.
6. Dong Y, Chawla NV, Swami A. metapath2vec: scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM; 2017:135-144.
7. Huang Z, Mamoulis N. Heterogeneous information network embedding for meta path based proximity. *CoRR*. 2017. arXiv preprint arXiv:1701.05291.
8. Shi C, Hu B, Zhao WX, Philip SY. Heterogeneous information network embedding for recommendation. *IEEE Trans Knowl Data Eng*. 2018;31(2):357-370.
9. Jin W, Derr T, Wang Y, Ma Y, Liu Z, Tang J. Node similarity preserving graph convolutional networks. In: Lewin-Eytan L, Carmel D, Yom-Tov E, Agichtein E, Gabrilovich E, eds. *WSDM '21, The Fourteenth {ACM} International Conference on Web Search and Data Mining, Virtual Event*. ACM;2021:148-156.
10. Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: Macskassy SA, Perlich C, Leskovec J, Wang W, Ghani R, eds. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM; 2014:701-710.

11. Grover A, Leskovec J. Node2Vec: scalable feature learning for networks. In: Krishnapuram B, Shah M, Smola AJ, Aggarwal CC, Shen D, Rastogi R, eds. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM; 2016:855-864.
12. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. LINE: large-scale information network embedding. In: Gangemi A, Leonardi S, Panconesi A, eds. *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. ACM; 2015:1067-1077.
13. Wang D, Cui P, Zhu W. Structural deep network embedding. In: Krishnapuram B, Shah M, Smola AJ, Aggarwal CC, Shen D, Rastogi R, eds. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM; 2016:1225-1234.
14. Huang X, Li J, Hu X. Accelerated attributed network embedding. In: Chawla NV, Wang W, eds. *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM; 2017:633-641.
15. Zhang J, Xia C, Zhang C, Cui L, Fu Y, Philip SY. BL-MNE: emerging heterogeneous social network embedding through broad learning with aligned autoencoder. In: Raghavan V, Aluru S, Karypis G, Miele L, Wu X, eds. *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE Computer Society; 2017:605-614.
16. Gao H, Huang H. Deep attributed network embedding. In: Lang J, ed. *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org; 2018:3364-3370.
17. Liao L, He X, Zhang H, Chua T-S. Attributed social network embedding. *IEEE Trans Knowl Data Eng*. 2018; 30(12):2257-2270.
18. Sun H, He F, Huang J, et al. Network embedding for community detection in attributed networks. *ACM Trans Knowl Discovery Data (TKDD)*. 2020;14(3):1-25.
19. Zhang W, Fang Y, Liu Z, Wu M, Zhang X. Mg2vec: learning relationship-preserving heterogeneous graph representations via metagraph embedding. IEEE Trans Knowl Data Eng. 2020.
20. Zhang Z, Huang J, Tan Q, Sun H, Zhou Y. CMG2Vec: a composite meta-graph based heterogeneous information network embedding approach. *Knowl-Based Syst*. 2021;216:106661.
21. Wang X, Ji H, Shi C, et al. Heterogeneous graph attention network. In: Liu L, White RW, Mantrach A, Silvestri F, McAuley JJ, Baeza-Yates R, Zia L, eds. *The World Wide Web Conference*. ACM; 2019:2022-2032.
22. Zhang C, Song D, Huang C, Swami A, Chawla NV. Heterogeneous graph neural network. In: Teredesai A, Kumar V, Li Y, Rosales R, Terzi E, Karypis G, eds. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM; 2019:793-803.
23. Fu X, Zhang J, Meng Z, King I. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In: Huang Y, King I, Liu T-Y, Steen MV, eds. *Proceedings of The Web Conference 2020*. ACM/IW3C2; 2020:2331-2341.
24. Hong H, Guo H, Lin Y, Yang X, Li Z, Ye J. An attention-based graph neural network for heterogeneous structural learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol 34, no. 04. AAAI Press; 2020:4132-4139.
25. Hu Z, Dong Y, Wang K, Sun Y. Heterogeneous graph transformer. In: Huang Y, King I, Liu T-Y, Steen MV, eds. *Proceedings of the Web Conference 2020*. ACM/IW3C2; 2020:2704-2710.
26. Zhou J, Cui G, Zhang Z, et al. Graph neural networks: a review of methods and applications. *AI Open*. 2020;1:57-81. https://doi.org/10.1016/j.aiopen.2021.01.001
27. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY. A comprehensive survey on graph neural networks. *IEEE Trans Neural Networks Learn Syst*. 2021;32(1):4-24. https://doi.org/10.1109/TNNLS.2020.2978386
28. Liu M, Gao H, Ji S. Towards deeper graph neural networks. In: Gupta R, Liu Y, Tang J, Aditya Prakash B, eds. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM; 2020:338-348.
29. Chen D, Lin Y, Li W, Li P, Zhou J, Sun X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol 34, no. 04. AAAI Press; 2020:3438-3445.
30. Chen Z, Cai T, Chen C, Zheng Z, Ling G. SINE: side information network embedding. In: Li G, Yang J, Gama J, Natwichai J, Tong Y, eds. *International Conference on Database Systems for Advanced Applications*. Springer; 2019:692-708.
31. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: Bengio Y, LeCun Y, eds. *1st International Conference on Learning Representations, {ICLR} Workshop Track Proceedings*. 2013. arXiv preprint arXiv:1301.3781.

32. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. *Adv Neural Inf Process Syst*. 2013;26:3111-3119.
33. Goldberg Y, Levy O. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *CoRR*. 2014. arXiv preprint arXiv:1402.3722.
34. Graves A. Generating sequences with recurrent neural networks. *CoRR*. 2013. arXiv preprint arXiv:1308.0850.
35. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*. 2014. arXiv preprint arXiv:1412.3555.
36. Ahrens JH, Dieter U. An alias method for sampling from the normal distribution. *Computing*. 1989;42(2): 159-170.
37. Kronmal RA, Peterson AV Jr. On the alias method for generating random variables from a discrete distribution. *Am Stat*. 1979;33(4):214-218.
38. Shi C, Kong X, Huang Y, Philip SY, Wu B. HeteSim: a general framework for relevance measure in heterogeneous networks. *IEEE Trans Knowl Data Eng*. 2014;26(10):2479-2492.
39. Zheng J, Liu J, Shi C, Zhuang F, Li J, Wu B. Dual similarity regularization for recommendation. In: Bailey J, Khan L, Washio T, Dobbie G, Huang JZ, Wang R, eds. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer; 2016:542-554.
40. Hu B, Shi C, Zhao WX, Yang T. Local and global information fusion for top-n recommendation in heterogeneous information network. In: Cuzzocrea A, Allan J, Paton NW, Srivastava D, Agrawal R, Broder AZ, Candan KS, Labrinidis A, Schuster A, Wang H, eds. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM; 2018:1683-1686.
41. Zhang Z, Yang H, Bu J, et al. ANRL: attributed network representation learning via deep neural networks. In: *IJCAI*. Vol 18; 2018:3155-3161.
42. Likas A, Vlassis N, Verbeek JJ. The global k-means clustering algorithm. *Pattern Recognition*. 2003;36(2): 451-461.
43. Wright RE. Logistic regression. American Psychological Association; 1995.
44. Wu F, Huberman BA. Finding communities in linear time: a physics approach. *Eur Phys J B*. 2004;38(2): 331-338.

How to cite this article: Shu L, Chen C, Xing X, Liao X, Zheng Z. AHNA: Adaptive representation learning for attributed heterogeneous networks. *Int J Intell Syst*. 2022;37: 1157-1185. https://doi.org/10.1002/int.22664