

Article

A Dynamic Heterogeneous Information Network Embedding Method Based on Meta-Path and Improved Rotate Model

Hualong Bu ¹, Jing Xia ², Qilin Wu ^{1,*} and Liping Chen ¹¹ School of Computer and Artificial Intelligence, Chaohu University, Hefei 238000, China² School of Mathematics and Big Data, Chaohu University, Hefei 238000, China

* Correspondence: lingqi@126.com

Abstract: Aiming at the current situation of network embedding research focusing on dynamic homogeneous network embedding and static heterogeneous information network embedding but lack of dynamic information utilization, this paper proposes a dynamic heterogeneous information network embedding method based on the meta-path and improved Rotate model; this method first uses meta-paths to model the semantic relationships involved in the heterogeneous information network, then uses GCNs to get local node embedding, and finally uses meta-path-level aggression mechanisms to aggregate local representations of nodes, which can solve the heterogeneous information utilization issues. In addition, a temporal processing component based on a time decay function is designed, which can effectively handle temporal information. The experimental results on two real datasets show that the method has good performance in networks with different characteristics. Compared to current mainstream methods, the accuracy of downstream clustering and node classification tasks can be improved by 0.5–41.8%, which significantly improves the quality of embedding, and it also has a shorter running time than most comparison algorithms.

Keywords: dynamic heterogeneous information network; meta-path; the improved rotate model



Citation: Bu, H.; Xia, J.; Wu, Q.; Chen, L. A Dynamic Heterogeneous Information Network Embedding Method Based on Meta-Path and Improved Rotate Model. *Appl. Sci.* **2022**, *12*, 10898. <https://doi.org/10.3390/app122110898>

Academic Editors: Yujin Lim and Hideyuki Takahashi

Received: 15 September 2022

Accepted: 25 October 2022

Published: 27 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Networks in real life are mostly heterogeneous information networks containing various types of nodes and edges/relationships, which integrate more information than homogeneous networks. Because heterogeneous information network (shorted as HIN) embedding has a strong expressive ability and effectively combines node attribute characteristics with the characteristics of structural information, it can not only solve the problems of network data such as data sparse [1] but also has achieved remarkable improvements in various downstream tasks, such as node classification [2], link prediction [3], node aggregation Class [4] and recommendation [5].

Most of the existing network embedding studies focus on static HIN embedding [1,6–9], mainly considering how to effectively utilize the network topology and semantic information in dynamic heterogeneous information networks (shorted as DHINs) without considering any temporal information, while the network in the real world will change rapidly over time. The DHIN is a network that changes with time and contains the semantic features implied by the types of time features, nodes, and edges [10]. Take the citation network as an example; every minute, there will be new cooperation between scholars to produce new papers, which will constantly change the co-authorship and collection of papers, as shown in Figure 1. The existing static embedding methods cannot capture the structure brought by temporal information and semantic changes. Existing research has proven that ignoring the heterogeneous nature will result in a decrease in embedding quality due to information loss [8].

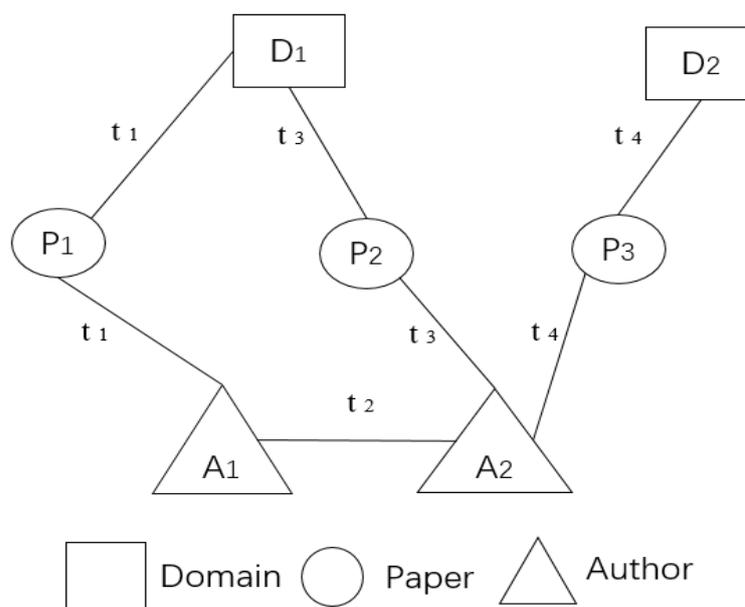


Figure 1. The Structure of citation DHIN. For example, if an author (A_1) published the paper (P_1), which belongs to the domain (D_1), timestamp t_1 is established between A_1 - P_1 and P_1 - D_1 .

Since DHINs have temporal information on the basis of HINs, the edges of networks are usually generated between nodes that are semantically closely related but belong to different categories, and the edges between heterogeneous nodes that are semantically related also often have the same timestamp (for example, A_1 - P_1 - D_1 , A_2 - P_2 - D_1 and A_2 - P_3 - D_2 are semantically closely related, but their categories are different). Different from the dynamic homogeneous network, affected by the semantic information of the DHIN, it is more likely that the semantically closely related nodes will have identical timestamps, rather than the occurrence time will have sequential and incremental timestamps (for example, the timestamps between A_1 - P_1 - D_1 are t_1 ; The timestamps between A_2 - P_2 - D_1 are t_2), which will bring new challenges to the DHIN embedding.

To sum up, the challenges of DHIN embedding mainly lie in how to make full use of the structural and semantic information in the network and how to make full use of the time information while considering the integrity of the network. In DHIN, the meta-paths contain rich semantic information, and a meta-path instance often corresponds to a specific event that occurs in the real world at a certain time, and it is of practical significance to explore the impact of meta path instances generated at different times on target nodes.

Therefore, this paper proposes an embedding method for DHINs (shorted as MRDE). Based on the meta-path, the structure and semantic information of DHIN can be learned first for GCN based local embedding, then a temporal information processing mechanism is designed to fully consider the influence of time information with the help of time delay function and the attention mechanism is used to enhance the time influence of different meta-path instances on target nodes, and a meta-path level aggression module is used to learn more comprehensive node embedding. Node classification and node clustering experiments on real datasets verified the effectiveness of MRDE. The contributions of this paper are as follows:

- (1) A DHIN embedding method based on meta-path and improved Rotate model (MRDE) is proposed, which can perform real-time and heterogeneous information learning;
- (2) Each meta-path can only reflect a specific structure and semantic information, and the node may be connected to other nodes through multiple meta-paths. An attention balance mechanism is proposed to learn more comprehensive node embedding;
- (3) Experiments on two real datasets show that MRDE can significantly improve the representation quality and achieve a performance improvement of 0.5~41.8% in downstream clustering and node classification tasks. Compared with other mainstream

algorithms, it was guaranteed by comparing the experimental results of two datasets with different characteristics, and it is proved to have good generality in different types of networks.

The remainder of this paper is organized as follows. Section 2 presents related works on DHIN embeddings, Section 3 introduces the basic notation in this paper and gives the problem definition, Section 4 details the proposed method based on meta-path and the improved Rotated model, Section 5 gives experimental details and discussions, and Section 6 summarizes the full text and gives the future improvement directions.

2. Related Works

2.1. Static HIN Embedding

Static information networks include static homogeneous information networks and static HINs. In traditional homogeneous information embedding methods [7], the nodes are represented as low-dimensional dense vectors. Representative models include DeepWalk [11], which is a combination of random walk, skip-gram, and LINE [12], which comprehensively considers the similarity of first-order and second-order neighbors. The emergence of Graph Neural Networks (GNN [13]) has accelerated the development of static homogeneous network embedding, typical representatives such as GAT [14], which uses an attention mechanism to perform weighted summation of adjacent node features; GraphSAGE [15] uses an inductive learning approach for training to adapt to large-scale network embeddings; SDNE [16] uses deep auto-encoders to extract network structures and nonlinear features. There are also many methods to improve node representation by using the content of nodes or text, images and labels, etc. Some review articles comprehensively summarize the work in this area [17,18].

The homogeneous information network often only extracts part of the information of the real network or does not distinguish the differences between objects and relationships in the real network, which can be regarded as a simplification of the real information network [19]. To model the real-world network more naturally, more and more researchers have focused on the analysis of HINs [8,9,20], the most classic of which is the Metapath2vec [8], which is based on meta-path-guided random walks. Metapath2vec utilizes domain knowledge, the meta-paths are pre-defined to guide random walk sampling to deal with the heterogeneity of the graph, and the representations of nodes are learned through Skip-Gram. HIN2Vec [9] realizes the performance of the prediction task while learning the representation vectors of the nodes and meta-paths. SHNE [20] jointly optimizes Skip-Gram and deep semantic coding to capture structural and non-structural semantic relations. GNNs have also made great progress in static HIN embedding [21]. The core idea of neural graph networks is to aggregate the feature information of neighbors through a neural network to learn node-independent information and corresponding structural information. Typical representatives include the relational graph convolutional network (RGCN) [22] and the heterogeneous graph attention network (HAN) [23].

2.2. DHIN Embedding

The actual networks are often dynamic. For example, new nodes, such as new users, new stores, and new products, are constantly generated on e-commerce platforms, and existing nodes can also generate new interactions. If the static HIN embedding method is used, a lot of repeated training is required. Therefore, it is necessary to study the embedding method of DHIN to realize dynamic updates. The DHIN can be divided into a snapshot network and a dynamic real-time network according to the dynamic representation. The snapshot network refers to a series of snapshots obtained at different time stamps, while the real-time dynamic network is a stream with time stamps.

Most of the current research focuses on dynamic homogeneous networks [24]. For example, the model Dynamic Triad [25] can preserve the structural information and evolution pattern of a given snapshot network; the heuristic algorithm (DNE) [26] can extend the Skip-Gram-based network representation method to dynamic environments; The in-

cremental Skip-Gram [27] model can well preserve the evolution information between snapshots by combining the walking method in Node2Vec [28]; Jiang et al. [29] encodes temporal information into the Knowledge Graph embedding for the first time. For fine-grained real-time HIN embedding, there are also many models that have achieved good performance, such as DANE [30], which well preserves the attribute labels and dynamic characteristics of the network but assumes all nodes are known in advance during the training process; DyHNE [31] incrementally captures changes by perturbing the meta-path augmented adjacency matrix to obtain an updated representation, but it lacks the ability to distinguish recent events from past events.

2.3. Knowledge-Graph-Based DHIN Embedding

In recent years, Knowledge Graphs have attracted a great deal of attention, and Knowledge Graphs are essentially heterogeneous networks that contain entities and relationships in networks [32]. According to modeling methods, they can be divided into Translation models, Bilinear models, Rotation models, Temporal Information Point Process models, Probability Distribution models, and Graph Neural Network models, etc.

The Translation model TransE [33] embeds the vector of the tail entity as the vector of the head entity plus the vector of the relationship, which can learn the rich semantics of entities and relationships. The Bilinear model RESCAL [34] models the relationships as a linear mapping matrix from the head entity to the tail entity. The Rotate model [35] solves the defect that the Translation model and the Bilinear model can only model part of the relationships and can model and infer all types of relationships in Knowledge Graphs. The Temporal Information Point Process model introduces temporal information and captures temporal information and dynamic properties in Knowledge Graphs. For example, the Know-Evolve temporal point model [36] can be used to model and predict future events, but it cannot model concurrent events that occur in the same time window. The Probability Distribution model RE-NET [37] solves the shortcomings of the Temporal Information Point Process methods and models events of the historical time series as a conditional probability distribution, but RE-NET cannot solve the long-term dependency problem, and the embedding of all time steps contributes the same to the current embedding. The GNN models, represented by RE-GCN [38], can capture the dependencies between entities under the same timestamp through a message-passing framework and capture sequential information patterns through the stacking of neural graph networks. However, the RE-GCN model may suffer from over-smoothing and overfitting problems.

At present, Knowledge Graph embedding and traditional HIN embedding are gradually integrated. For example, MPDRL [39] learns semantics through reinforcement learning on the basis of Knowledge Graphs; RHINE [40] draws on Knowledge Graph learning technology, divides relationships in heterogeneous networks into membership and interaction relationships, and learns them separately; Wang et al. [41] combine with neural networks to produce high-throughput, fast and real-time data, which can work for data centers, databases and other platforms as a service. Ni et al. [42] use the Adapter pre-trained by 'linking of GraphQL schemas and corresponding utterances' as an external knowledge introduction plug-in to better map between the logical language and natural language. Liu et al. [43] propose a real quadratic form-based graph pooling framework for GNN, which has stronger expressive power than the existing linear forms in Knowledge Graph. Although real-time Knowledge Graphs have achieved great success through inference means, how to further exploit the complex structure of the network itself is still an open topic [44].

To sum up, the possible routes for existing dynamic heterogeneous network embedding include HIN embedding, which can be repeatedly called at different time points, dynamic homogeneous network embedding, and dynamic Knowledge Graph embedding. In the experimental part, we will select representative algorithms from the above routes as experimental baselines.

3. Notations and Problem Definition

Here we first introduce some relevant definitions of DHINs and then give the formal descriptions and problem descriptions for DHINs.

Definition 1. *Heterogeneous information network.* To simplify the problem, a heterogeneous undirected network $G = (V, E)$ is given, where V is the set of nodes, E is the set of edges, and G contains maps of node types, the function $\varnothing : V \rightarrow \mathcal{A}$ and an edge type mapping function $\varphi : E \rightarrow \mathcal{R}$, satisfy $|\mathcal{A}| + |\mathcal{R}| > 2$. In addition, the node attribute is defined as the initial feature matrix $X \in \mathbb{R}^{|V| \times D}$.

Definition 2. *Dynamic heterogeneous information network.* A heterogeneous information network including time attributes $GT = (V, E; T)$, T represents the time attribute set. For each connecting u, v nodes, $e = (u, v, t)$, with timestamp $t \in T$.

Definition 3. *Meta-path [45].* Let V_t be the set of target class nodes. The meta-path between nodes v_{t1} and v_{tn} is defined as $v_{t1} \xrightarrow{R_1} v_{t2} \xrightarrow{R_2} \dots \xrightarrow{R_{n-1}} v_{tn}$, the meta-path can reflect the closeness between the target type nodes.

Definition 4. *Meta-path context.* Define the meta-path set as $\rho = \{\rho_1, \rho_2, \dots, \rho_p\}$, where ρ_i represents the i -th meta-path. Based on the predefined meta-path, the corresponding adjacency matrix set $A^\rho = \{A^{\rho_1}, A^{\rho_2}, \dots, A^{\rho_p}\}$ can be obtained, where $A^{\rho_i} \in \mathbb{R}^{|V_t| \times |V_t|}$.

Problem Description. Temporal information-based embedding of heterogeneous information networks, under the guidance of temporal information, will learn low-dimensional representations $H \in \mathbb{R}^{|V| \times d}$ of structural information and attribute information in G , where $d \ll D$.

4. Materials and Methods

4.1. The Whole Framework

This paper proposes a DHIN embedding method based on meta-path and an improved Rotate model, as shown in Figure 2. First, the different space of different types of node features are mapped into a unified space through the data preprocessing module; next, the representations of each meta-path instance under different meta-paths are learned by GCNs, and the temporal information processing component is used under the guidance of timestamps. Finally, the more comprehensive node embedding is obtained by aggregate the node embedding based on meta-path-level regression.

4.2. Data Preprocessing

The feature spaces of different types of nodes in heterogeneous information networks may be different, which brings challenges to subsequent processing. Hence we first map the different types of nodes into the same vector space. Given a node $v \in V$ of a certain type A , the mapping operations are as follows:

$$h'_v = P(h_v) = Q_A \cdot h_v + b \quad (1)$$

where $P(\cdot)$ is the mapping function, h_v is the feature vector of node v , h'_v represents the feature vector of node v after mapping, Q_A represents the mapping matrix of type A , and b is a bias vector.

After the mapping process, nodes can have the same feature space, so as to facilitate the processing of subsequent tasks.

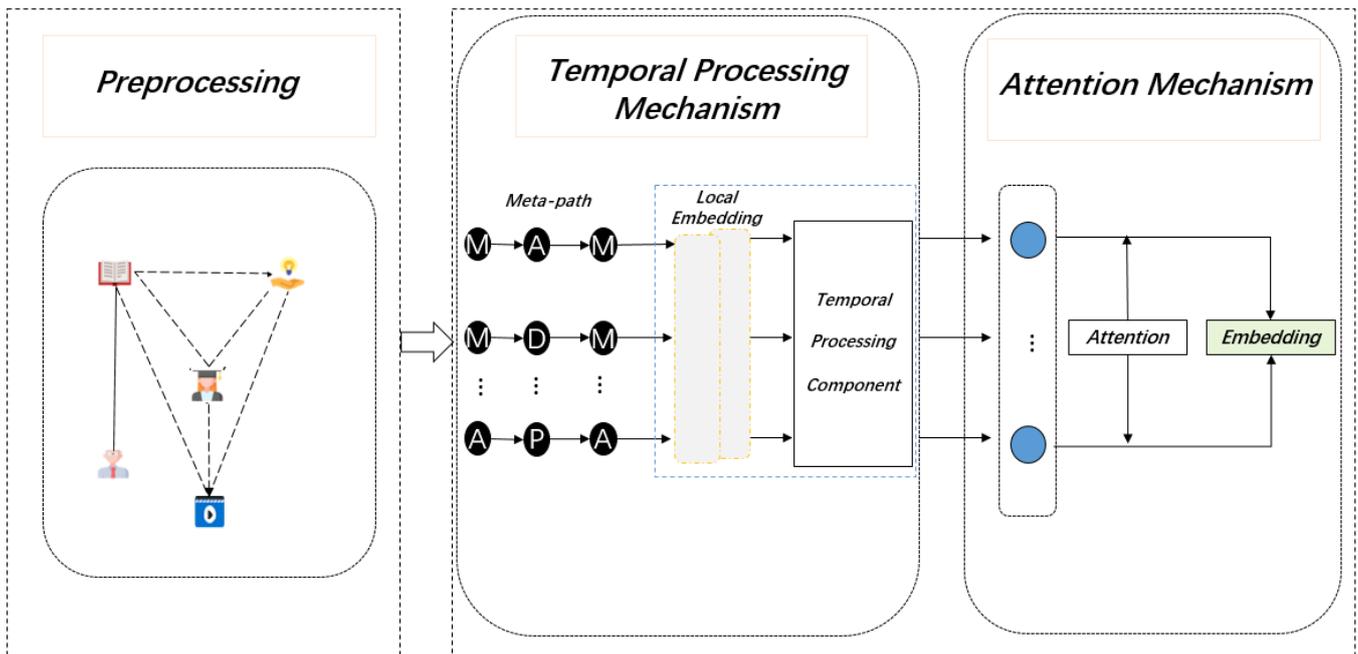


Figure 2. The Framework of the MRDE model.

4.3. DHIN Embedding Based on Meta-Path and Improved Rotate Model

4.3.1. Meta-Path-Based Node Local Representation

Each adjacency matrix can represent a homogeneous graph, and we first learn the encoder at the node level through local embedding to generate node embeddings containing the initial node feature X and A^{ρ_i} information:

$$H^{\rho_i} = f_{\rho_i}(X, A^{\rho_i}) \tag{2}$$

where $f_{\rho_i}(\cdot)$ represents the node-level encoder. Here, in order to effectively integrate the node-familiar features and local structural features, the GCN model is selected as the node-level encoder, and each node representation is obtained:

$$H^{\rho_i} = \left((D^{\rho_i})^{-\frac{1}{2}} \tilde{A}^{\rho_i} (D^{\rho_i})^{-\frac{1}{2}} \right) XW^{\rho_i} \tag{3}$$

where, $\tilde{A}^{\rho_i} = A^{\rho_i} + I$, D^{ρ_i} represents the degree diagonal matrix of A^{ρ_i} nodes, and $W^{\rho_i} \in \mathbb{R}^{d \times f}$ is the parameter matrix. After performing the convolution operation on each adjacency matrix, a node representation set $\{H^{\rho_i}\}_{i=1}^P$ is obtained, H^{ρ_i} represents the embedding of the target node V_i under meta path ρ_i .

4.3.2. Temporal Processing Component Based on Time Decay Function

In this work, an improved Rotate model is designed, which adds a temporal information processing component based on a time decay function to learn the effects of different meta-path instances over time.

Given a meta-path instance $\rho(v, u)$, $u \in \mathcal{N}_v^{\rho}$ represents the neighbors based on the meta-path, and the set of intermediate nodes in the meta-path is $m^{\rho(v,u)} = \rho(v, u) \setminus \{v, u\}$. Encode all nodes in this instance into a vector $h_{\rho(v,u)} = f(\rho(v, u)) = f\left(h'_v, h'_u, \left\{h'_i, \forall i \in m^{\rho(v,u)}\right\}\right)$, where $f(\cdot)$ represents the encoding function of $\rho(v, u)$.

There are many types of relationships in the Knowledge Graph, such as symmetric relationships, asymmetric relationships, semantically opposite relationships, compound relationships, etc. Traditional translation models can model semantically opposite relationships and compound relationships, while Bilinear models can model symmetric relationships, but no one model can model and infer all types of relationships. Inspired by

Euler’s formula ($e^{i\theta} = \cos\theta + i\sin\theta$), Sun et al. modeled the relationship as a rotation from the head entity to the tail entity and proposed the Rotate model [35]. Rotate is able to model all relation types and has achieved breakthroughs in problems such as link prediction. This work introduces Rotate model and proposes an improved Rotate model to make full use of complex structural and semantic information in DHIN.

Let $\rho(v, u) = (x_0, x_1, \dots, x_n)$, where $x_0 = v, x_n = u, r_i$ is the relation vector between relation nodes x_i and x_{i-1} . The improved Rotate process is as follows:

$$o_0 = h'_{x_0} = h'_v, \tag{4}$$

$$o_i = h'_{x_i} + o_i \odot r'_i, \tag{5}$$

$$h_{\rho(v,u)} = \frac{o_n}{n + 1}, \tag{6}$$

where \odot is the Hadamard product.

Given a node v , a series of meta-path instances in meta-path ρ , and their corresponding timestamps, i.e., $v \rightarrow \{(\rho(v, u)_1, t_1), (\rho(v, u)_2, t_2), \dots, (\rho(v, u)_n, t_n)\}$, the influence of each instance on the target node can be calculated as follows:

$$\lambda_m = \text{softmax}(\tilde{\lambda}_m) = \frac{\exp(\tilde{\lambda}_m)}{\sum_{l=1}^n \exp(\tilde{\lambda}_l)}, \tag{7}$$

$$\tilde{\lambda}_m = \alpha_i u_{v,\rho(v,u)_i} k(t - t_i), \tag{8}$$

$$k(t - t_i) = \exp(\delta_s(t - t_i)), \tag{9}$$

$$u_{v,\rho(v,u)_i} = f_\theta(v, \rho(v, u)_i) = -\|h'_v - h_{\rho(v,u)_i}\|^2 \tag{10}$$

It should be pointed out that, in order to evaluate the impact of different timestamps, this work builds a meta-path change model based on the time decay function with the following assumptions: when an instance just acts on the target node, it can be considered that it has the greatest impact on the target node at this moment, but if there is no continuous stimulus for a certain period time, the influence of the instance on the target node will continue to decay. Influenced by [46], we designed a time decay function $k(\cdot) = \exp(\delta_s(t - t_i))$, that is, according to the time sequence, the time factor and influence degree are connected by defining the attenuation function and attenuation strategy. Here t is the current time, when t_i is close to t , $\rho(v, u)_i$ has a greater impact on v ; δ_s represents the learnable decay rate; $u_{v,\rho(v,u)_i}$ measure the similarity between v and $\rho(v, u)_i$, where the Euclidean distance is used to measure the similarity.

In order to further enhance the expressiveness of the influence of meta path instances on target nodes over time, the attention mechanism is introduced in this work, and the attention coefficient is:

$$\alpha_m = \frac{\exp(\tilde{\alpha}_m)}{\sum_{l=1}^n \exp(\tilde{\alpha}_l)}, \tag{11}$$

$$\tilde{\alpha}_m = \sigma\left(k(t - t_i) a^T \left[Wh'_v \oplus Wh_{\rho(v,u)_i} \right]\right), \tag{12}$$

where a represents the attention vector, \oplus represents the splicing operation, W represents the weight matrix, and $\sigma(\cdot)$ represents the activation function. The representation of the target node v under the meta-path ρ is obtained by the weighted representation of the meta-path instance and its influence factor, namely.

$$h_v^{\rho} = \sigma\left(\sum_{i=0}^n \lambda_i h_{\rho(v,u)_i}\right). \tag{13}$$

To sum up, given the meta-path set $\{\rho_1, \rho_2, \dots, \rho_k\}$, after the temporal processing component, k groups of node representations with different semantics can be obtained, which can be expressed as $H^{\rho_i} = \{h_v^{\rho_1}, h_v^{\rho_2}, \dots, h_v^{\rho_k}\}$.

4.4. Meta-Path-Level Aggregation

Each meta-path contains a specific structure and semantic information, and the node may be connected to other nodes through multiple meta-paths, so we need to learn more comprehensive node embedding. In addition, traditional pooling strategies (average pooling or max pooling) cannot effectively measure the importance of different meta-paths. In this paper, a semantic-level attention mechanism is used to analyze the set $\{H^{\rho_i}\}_{i=1}^P$ performs overall aggregation to generate an overall representation containing multiple relational semantics. To this end, an intuitive idea is to measure the contribution of each meta-path to the final node representation and then use the respective contributions as weights to aggregate the individual node representations, so Here an attention layer L_{att} is added to learn the corresponding weights:

$$\{\beta^{\rho_1}, \beta^{\rho_2}, \dots, \beta^{\rho_P}\} = L_{att}(H^{\rho_1}, H^{\rho_2}, \dots, H^{\rho_P}) \quad (14)$$

First, calculate the importance of the single meta-path ρ_i .

$$e^{\rho_i} = \frac{1}{N} \sum_{n=1}^N \tanh(q^T [W_{sem} \cdot h^{\rho_i} + b]) \quad (15)$$

where, W_{sem} represents the parameter matrix, and q represents the semantic attention vector that needs to be learned.

Next, the generated set $\{e^{\rho_i}\}_{i=1}^P$ is regularized using the *softmax* function to obtain the weight β^{ρ_i} of the meta-path ρ_i :

$$\beta^{\rho_i} = softmax(e^{\rho_i}) = \frac{\exp(e^{\rho_i})}{\sum_{j=1}^P \exp(e^{\rho_j})} \quad (16)$$

Finally, the dynamic heterogeneous information network representation H will be obtained by the linear combination of the node representation set $\{H^{\rho_i}\}_{i=1}^P$:

$$H = \sum_{i=1}^P \beta^{\rho_i} \cdot H^{\rho_i} \quad (17)$$

It should be noted that the attention weight learned by the model is binary cross-entropy loss in the method of this paper, and the learned weight helps to measure the similarity of a node to its neighbors under different distributions, that is, the more similar the node features to its neighbors, the greater the assigned weight. At the same time, since the classification label is not involved, the weight will not be affected by the label.

5. Experiment and Discussion

5.1. Experimental Data

The data sets used in this paper are DBLP [47] and IMDB [48]. The DBLP data set collects research papers in the direction of the computer, including four node information of author, paper, conference, and keywords. The author node can also be divided into four sub-directions of the database, data mining, data information retrieval, and machine learning. This work takes the author as the target node, uses its research field as the label, and uses the bag-of-words model to generate initial features. The IMDB dataset collects movie knowledge graphs, which can be divided into three types: action, comedy, and drama. This paper selects the movie as the target node and uses the genre of the movie as

the label, and its features include color, title, language, keyword, country, rating, year, and TF-IDF encoding. Dataset summary information is shown in Table 1.

Table 1. Dataset summary information.

Dataset	Node	Node Number	Edge	Edge Number	Feature Dimension	Meta-Path
DBLP	Author (A)	4057	AP AC PT	19,645 14,328 88,420	334	APA APCPA APTPA
	Paper (P)	14,328				
	Conference (C)	20				
	Term (T)	8789				
IMDB	Movie (M)	4275	MA MD MK	12,838 4280 20,529	6344	MAM MDM MKM
	Actor (A)	5431				
	Director (D)	2082				
	Keyword (K)	7313				

5.2. Baselines

In order to fully verify the effectiveness of our method, we use seven representation learning methods as baselines, including:

1. Raw Feature, that is, the initial input features are used for node representation;
2. GCN [13], a homogeneous neural network model, introduces the neural network into the field of graph learning. Here we adapt standard experimental strategies and parameter settings and construct a meta-path-based adjacency matrix to facilitate its operation according to the requirements of homogeneous graphs;
3. GAT [14], a homogeneous neural network model, uses an attention mechanism. Here we adapt standard experimental strategies and parameter settings and construct a meta-path-based adjacency matrix to facilitate its operation according to the requirements of homogeneous graphs;
4. Metapath2vec (M2V) [8], a representation learning method for HINs, performs random one-walk based on meta-paths. Here we set the window size to 5, the walk length to 100, and the number of negative samples to 5;
5. JUST [49], a static HIN embedding method. Here, the initial stay probability parameter of the walk is set to $\alpha = 0.2$ so that the algorithm tends to explore new types of nodes through heterogeneous edges during the walk, and the remaining parameters are selected as the default settings in the original text;
6. CTDNE [10] is a real-time dynamic homogeneous network embedding method in which random walks are organized with edges as angles, and the next edge is selected according to the strict increasing order of timestamps. Here the minimum length of each walk sequence is 5. The size of the batch update is consistent with the sliding window step size to ensure that the required walking sequence can be generated in the network with a limited number of timestamps and that the sequence is not too short.
7. Change2vec (C2V) [50], a DHIN embedding method for snapshots, uses meta-path guidance. In order to ensure the quality of meta-path selection, the meta-paths used in each dataset are selected based on the semantics of the dataset. And considering the influence of selecting different meta-paths, multiple meta-paths are comprehensively considered for random walks in the experiment;

In this work, the Adam optimizer [51] is used, and the learning rate is set to 0.01, the weight decay is 0.001, the node dimension is 512, the attention dimension is 8, and the experimental platform is Pytorch. In the node classification task, the learning method adopts the Support Vector Machine, while the supervised method is an end-to-end model, so the classification results are directly output. 80% of the data set is used as the training set, 10% of the data is used as the validation set, and 10% of the data is used as the test set. In order to ensure the stability of the results, all experimental tasks were repeated 10 times.

5.3. Experiment Result and Discussion

5.3.1. Node Classification Task

First, we verified the node classification task, and the experimental results are shown in Table 2. We mark our results in bold and underline the best results (the same below). As it can be seen from Table 2, comparing the results of the MRDE method on DBLP and IMDB, it can be found that the classification effect of the MRDE method in this paper is improved on both datasets (the highest on DBLP is increased by 30%, and the lowest is increased by 0.7%. The highest increase is 41.8% on IMDB, and the lowest is 0.5%), which fully reflects the necessity of introducing local graph embedding and attention balance mechanism when considering multiple adjacency matrices with different structures at the same time.

Table 2. Experimental values of the node classification task.

Dataset	Training Set	Indicator	Raw	GCN	GAT	M2V	JUST	CTDNE	C2V	MRDE
DBLP	20%	Micro-F	0.715	0.699	0.819	0.593	0.824	0.899	0.906	<u>0.930</u>
		Macro-F	0.725	0.698	0.813	0.513	0.835	0.892	0.898	<u>0.904</u>
	80%	Micro-F	0.822	0.811	0.838	0.618	0.854	0.910	0.917	<u>0.947</u>
		Macro-F	0.825	0.803	0.839	0.521	0.868	0.906	0.910	<u>0.942</u>
IMDB	20%	Micro-F	0.502	0.582	0.633	0.435	0.698	0.608	0.681	<u>0.712</u>
		Macro-F	0.523	0.623	0.708	0.447	0.654	0.673	0.692	<u>0.710</u>
	80%	Micro-F	0.585	0.679	0.707	0.452	0.734	0.720	0.725	<u>0.738</u>
		Macro-F	0.592	0.687	0.736	0.487	0.725	0.739	0.737	<u>0.745</u>

In addition, for DBLP and IMDB, the JUST, CTDNE, and C2V methods, which are based on DHINs, are better than static methods (GCN, GAT, M2V), it shows the effectiveness of the DHIN methods, and also verifies the implicit assumption of mining semantic information and temporal information in HINs is beneficial to improve the quality of node representation. At the same time, we also compared the Raw method (input features are directly used as node representations), and we can find that the Raw method has a poor processing effect compared with each other method, thus excluding the possibility that the input features lead to better performance of the model.

Similarly, compared with M2V, which only uses the semantic relation network structure for representation learning, the HIN embedding methods, which effectively combine the input features and structural information, can usually obtain better node representation.

In addition, in the IMDB dataset, the effectiveness of MRDE is worse than that in DBLP. We analyze and find that the target nodes in the IMDB dataset are often weakly correlated; for example, the same director may direct different types of movies, while the same actor may also appear in different types of movies, thus introducing more noise.

5.3.2. Node Clustering Task

This paper uses the K-mean algorithm to cluster the generated node embeddings. The number of clusters is set as the number of category types of the target node. The clustering task is also repeated 10 times, and the average value of the Adjusted Rand index (ARI) is shown in Table 3.

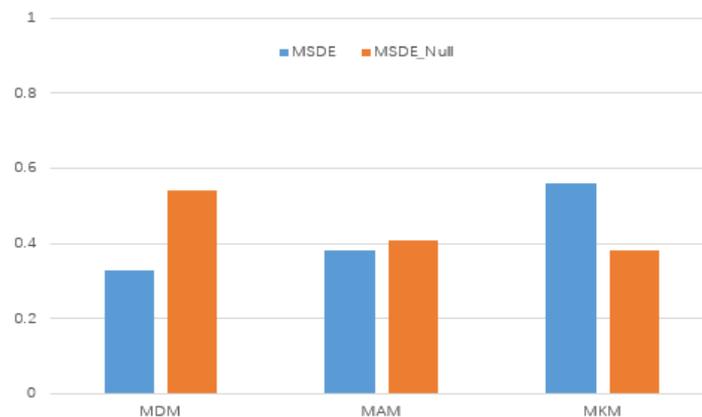
It is easy to find that MRDE is always better than other comparison methods in Table 3. Combined with the results of the node classification task, it is found that most comparison methods have different degrees of over-smoothing; that is, the node embeddings in the local structure become too similar. In other words, similar node representations are beneficial to the classifier to classify nodes to a certain extent. On the contrary, during node clustering, similar node representations will make nodes clustered together and become indistinguishable. By comprehensively considering the distribution of the nearest neighbors under multiple adjacency matrices and selectively extracting useful information from them, MRDE can effectively prevent the occurrence of the over-smoothing problem.

Table 3. Experimental results of node clustering task.

Method	DBLP	IMDB
	ARI	ARI
Raw	0.369	0.355
GCN	0.625	0.682
GAT	0.718	0.733
M2V	0.785	0.775
JUST	0.725	0.737
CTDNE	0.805	0.827
C2V	0.798	0.905
MRDE	0.815	0.929

5.3.3. The Effect of Attention Balance Parameter β

In order to further illustrate the role of the attention balance mechanism, the experiment verifies the effect with the attention balance mechanism (MRDE) and without the attention balance mechanism (MRDE_Null) on the IMDB dataset, and the results are shown in Figure 3.

**Figure 3.** The attention weight distribution of different meta-paths on the IMDB dataset.

As can be seen from Figure 3, MRDE get the best results on the MKM meta-path, which means it mainly focuses on the MKM relationship; on the contrary, MRDE_Null mainly focuses on the MDM and MAM relationship. It is not difficult to understand that MKM builds relationships through the same keywords between movies, with the commonality of most keywords, the movie nodes are more closely connected, and their representations become more similar, which makes MKM gain greater attention.

After adding the attention balance mechanism, MRDE can not only maintain a high attention weight for MDM and MAM but also assign a certain weight to MKM instead of ignoring it directly. In this way, MRDE can aggregate features of nodes that cannot be contracted in MDM and MAM.

5.3.4. Impact of Time Accumulation

In order to verify the impact of time accumulation, we record the DBLP data set as Y4, Y8, Y12, Y16, Y20, Y24, Y28, Y32, Y36, and Y40 according to the time sequence that every four years as a time stamp. For each time stamp, we use MRDE to learn the node embedding for the downstream node classification task, the results are shown in Figure 4. From the figure we can find that the accuracy of node classification is gradually improved with the continuous accumulation of time, which means that with the accumulation of time, the effective information on the network is gradually increasing, which is conducive for better node embedding. It also means that DHIN embedding method can capture the time consistency information in the network

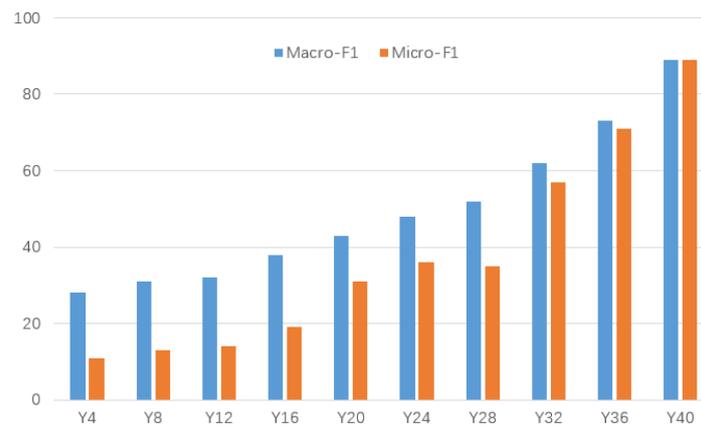


Figure 4. The effect of temporal variation on the classification task on the DBLP dataset.

5.3.5. Runtime Comparison

Because the network is constantly changing, the requirement of DHIN embedding will occur at any time and frequently. In order to meet the requirements of DHIN, the running time is one of the most important embedding method indicators under the premise of ensuring sufficient accuracy of downstream tasks. The lower the time cost of the algorithm, the better the comprehensive performance. In this section, we evaluate the running time of MRDE and other comparative methods. Table 4 shows the total running time of each method on the two datasets, where the underlined part is the best result, and the bold part is the MRDE result.

Table 4. Clustering task running time (in seconds).

Method	Dataset	
	DBLP	IMDB
Raw	1032.19	863.56
GCN	4563.25	2232.41
GAT	7563.43	4509.28
M2V	1589.68	1097.65
JUST	71,920.86	30,122.36
CTDNE	32,858.93	14,118.73
C2V	256.37	187.15
MRDE	815.06	327.92

As can be seen from Table 4, JUST and CTDNE spend more time on embedding, and the static method JUST obviously exceeds an order of magnitude. Whenever an embedding needs to be obtained, the static model must be run on the current full graph, and the model must be completely retrained. Therefore, static methods are difficult to adapt to the needs of real-time due to the long running time. The CTDNE will try to walk randomly many times whenever new edges arrive, but the DHINs allow parallel edges between the same node pair, this means the number of edges may be much greater than the number of nodes and the edge-based methods will take much longer running time than node-based methods on an average, especially on edge-dense networks.

5.3.6. Each Sub Module and Its Parameter Influence

In order to verify the effectiveness of each module and the influence of different parameters, we have conducted the following comparison experiments. The specific definitions of experiments are as follows:

1. MRDE, representing the method in this paper;
2. MRDE-NoF, indicating that node features are not used;
3. MDE-Bili, which means that the meta path instance is encoded with Bilinear [34];

4. MDE-Tran, which means that the meta path instance is encoded with TransE [33];
5. MRDE-Avg, the coding method of averaging node vectors;
6. MRDE-NoT means that the time attribute in the network is not considered, and only the attention mechanism is used to fuse the meta path instances;
7. MRDE-NoAtt, which means that only the time attribute in the network is considered, and no attention mechanism is used to fuse meta-path instances;
8. MRDE-NoC, which means that the central node of the meta path instance is not considered, and only the node pairs connected by the meta path are used.

The index values of each method in node classification tasks are shown in Figure 5, and the index values of node clustering tasks are shown in Figure 6.

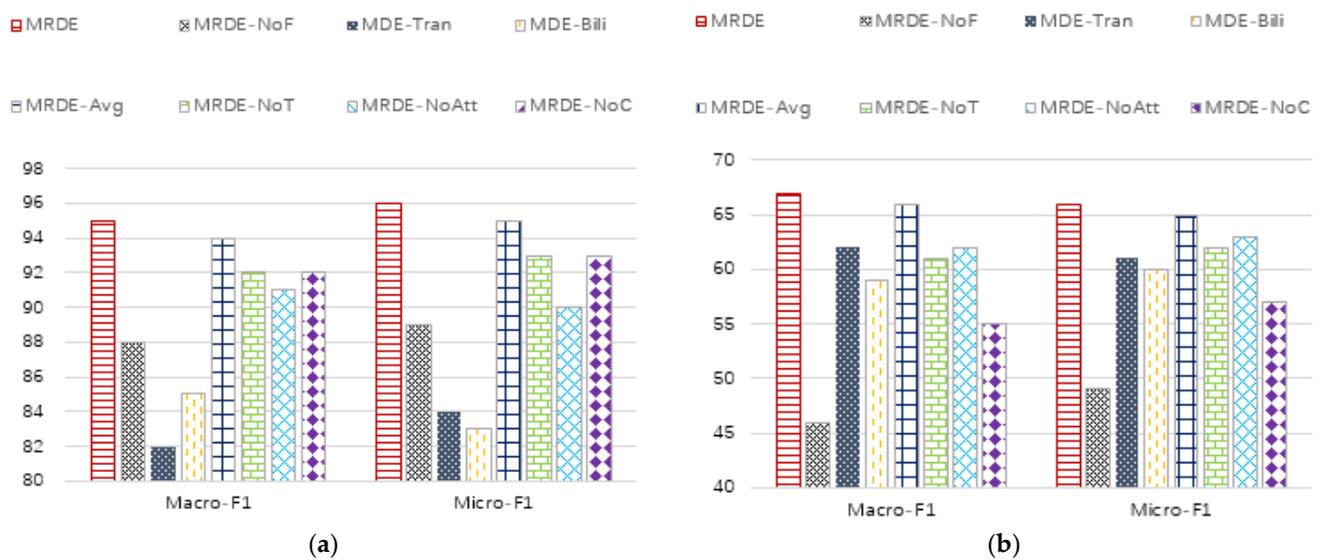


Figure 5. Node classification task. (a) DBLP. (b) IMDB.

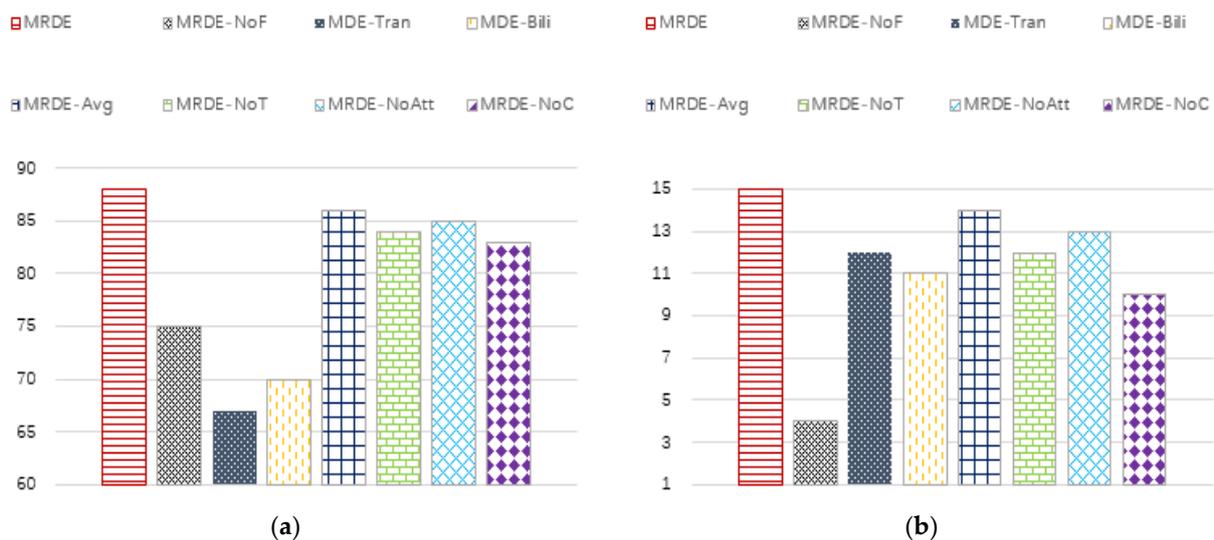


Figure 6. The ARI values of the node clustering task. (a) DBLP. (b) IMDB.

It can be seen from Figures 5 and 6 that:

1. The MRDE-NoF method is the worst or second worst for both classification tasks and clustering tasks, indicating that the attribute information of the node itself has a great impact on the performance of embedding. By observing (a) and (b), the degree of influence in the DBLP dataset is obviously lighter than that in the IMDB dataset. We

- judge that the core information in the DBLP dataset is structural information and time information, while the core information in the IMDB dataset contains both structural information, time information, and attribute information;
2. The performance of the MDE-Bili and MDE-Tran methods on DBLP and IMDB datasets is quite different, which indicates that translation models and bilinear models are more dependent on the matching degree between the data's internal patterns and models. When the matching degree is high, good results can be achieved. Otherwise, the effects may decline significantly;
 3. MRDE-NoT has different effects on different classification tasks and different data sets, but it is weaker than MRDE, indicating that the impact of time attribute information is unstable, but it will have a certain impact on the final embedding effect. MRDE-NoAtt, MRDE-Avg, and MRDE-NoC methods have similar characteristics;
 4. The MRDE model has achieved the best results in all data sets and tasks, indicating the rationality of the embedding framework we have adopted.

6. Conclusions and Future Works

In this work, we propose a representation learning method for DHINs based on meta-paths and an improved Rotate model. We use meta-paths to fully mine the complex structural information and rich semantic information in DHINs. We also consider the temporal properties and the influence of meta-path instances on target nodes at different time stamps with the help of proposed temporal processing component. The experimental results of node classification and clustering show that the MRDE method in this paper achieves better results than other methods, and it also has a shorter running time than most comparison baselines.

Although the effectiveness of this work have been verified, there are still some shortcomings, including the selection of meta-paths requires more domain knowledge, so it is necessary to study how to deeply mine the rich, heterogeneous information without predefined domain knowledge; There are also many more effective real-time processing methods, such as dynamic Knowledge Graph representation and dynamic GNNs methods, so one of the directions for future work is how to introduce these methods to achieve better results.

Author Contributions: H.B. has designed and implemented the scalability approach for our works and performed the evaluation and analysis; Q.W., L.C. and J.X. have been supervising the research during the whole process, providing insights, corrections, reviews, and proposing best practices; they were also in charge of funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Key Projects of Natural Sciences Research at Anhui Universities of China (KJ2016A502), the Key research projects of Chaohu University (XLZ-202106), and the Key Research and Development Plan of Anhui Province, China (201904a05020091), Anhui Province Teaching Demonstration Course Project (2020SJJXSK17), and Anhui Province Teaching Research Project (2019jyxm1187).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No declaration.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Cui, P.; Wang, X.; Pei, J.; Zhu, W. A survey on network embedding. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 833–852. [[CrossRef](#)]
2. Jacob, Y.; Denoyer, L.; Gallinari, P. Learning latent representations of nodes for classifying in heterogeneous social networks. Proceedings of the 7th ACM International Conference on Web Search and Data Mining, New York, NY, USA, 24–28 February 2014; pp. 373–382.

3. Liu, F.; Xia, S.T. Link prediction in aligned heterogeneous networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Cham, Switzerland, 2015; pp. 33–44.
4. Luo, C.; Guan, R.; Wang, Z.; Lin, C. Hetpathmine: A novel transductive classification algorithm on heterogeneous information networks. In *European Conference on Information Retrieval*; Springer: Cham, Switzerland, 2014; pp. 210–221.
5. Jamali, M.; Lakshmanan, L. Heteromf: Recommendation in heterogeneous information networks using context dependent factor models. In *Proceedings of the 22nd International Conference on World Wide Web*, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 643–654.
6. Shi, C.; Li, Y.; Zhang, J.; Sun, Y.; Philip, S.Y. A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.* **2016**, *29*, 17–37. [[CrossRef](#)]
7. Yang, C.; Xiao, Y.; Zhang, Y.; Sun, Y.; Han, J. Heterogeneous network representation learning: Survey, benchmark, evaluation, and beyond. *arXiv* **2020**, arXiv:2004.00216v1.
8. Dong, Y.; Chawla, N.V.; Swami, A. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, NS, Canada, 13–17 August 2017; pp. 135–144.
9. Fu, T.Y.; Lee, W.C.; Lei, Z. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, 6–10 November 2017; pp. 1797–1806.
10. Lee, J.B.; Nguyen, G.; Rossi, R.A.; Ahmed, N.K.; Koh, E.; Kim, S. Dynamic node embeddings from edge streams. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *5*, 931–946. [[CrossRef](#)]
11. 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.
12. 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.
13. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
14. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
15. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *arXiv* **2017**, arXiv:1706.02216.
16. 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.
17. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowl. Based Syst.* **2018**, *151*, 78–94. [[CrossRef](#)]
18. Hamilton, W.L.; Ying, R.; Leskovec, J. Representation learning on graphs: Methods and applications. *arXiv* **2017**, arXiv:1709.05584.
19. Shi, C.; Philip, S.Y. *Heterogeneous Information Network Analysis and Applications*; Springer International Publishing: Cham, Switzerland, 2017.
20. Zhang, C.; Swami, A.; Chawla, N.V. Shne: Representation learning for semantic-associated heterogeneous networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, Melbourne, Australia, 11–15 February 2019; pp. 690–698.
21. Lu, Y.; Shi, C.; Hu, L.; Liu, Z. Relation structure-aware heterogeneous information network embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4456–4463.
22. Chen, J.; Hou, H.; Gao, J.; Ji, Y.; Bai, T. RGCN: Recurrent graph convolutional networks for target-dependent sentiment analysis. In *International Conference on Knowledge Science, Engineering and Management*; Springer: Cham, Switzerland, 2019; pp. 667–675.
23. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous graph attention network. In *Proceedings of the World Wide Web Conference*, San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032.
24. Zhao, Z.; Li, C.; Zhang, X.; Chiclana, F.; Viedma, E.H. An incremental method to detect communities in dynamic evolving social networks. *Knowledge-Based Systems* **2019**, *163*, 404–415.
25. Zhou, L.; Yang, Y.; Ren, X.; Wu, F.; Zhuang, Y. Dynamic network embedding by modeling triadic closure process. In *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
26. Du, L.; Wang, Y.; Song, G.; Lu, Z.; Wang, J. Dynamic network embedding: An extended approach for skip-gram based network embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 13–19 July 2018; Volume 2018, pp. 2086–2092.
27. Peng, H.; Li, J.; Yan, H.; Gong, Q.; Wang, S.; Liu, L.; Wang, L.; Ren, X. Dynamic network embedding via incremental skip-gram with negative sampling. *Sci. China Inf. Sci.* **2020**, *63*, 202103. [[CrossRef](#)]
28. 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.
29. Jiang, T.; Liu, T.; Ge, T.; Sha, L.; Li, S.; Chang, B.; Sui, Z. Encoding temporal information for time-aware link prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, 1–5 November 2016; pp. 2350–2354.
30. Li, J.; Dani, H.; Hu, X.; Tang, J.; Chang, Y.; Liu, H. Attributed network embedding for learning in a dynamic environment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, 6–10 November 2017; pp. 387–396.

31. Wang, X.; Lu, Y.; Shi, C.; Wang, R.; Cui, P.; Mou, S. Dynamic heterogeneous information network embedding with meta-path based proximity. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 1117–1132. [[CrossRef](#)]
32. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
33. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 2, pp. 2787–2795.
34. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011.
35. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. Rotate: Knowledge Graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197.
36. Trivedi, R.; Farajtabar, M.; Wang, Y.; Dai, H.; Zha, H.; Song, L. Know-evolve: Deep reasoning in temporal knowledge graphs. *arXiv* **2017**, arXiv:1705.05742.
37. Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; Cheng, X. Temporal knowledge graph reasoning based on evolutionary representation learning. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 408–417.
38. Qu, M.; Tang, J.; Han, J. Curriculum learning for heterogeneous star network embedding via deep reinforcement learning. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 468–476.
39. Wan, G.; Du, B.; Pan, S.; Haffari, G. Reinforcement learning based meta-path discovery in large-scale heterogeneous information networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 6094–6101.
40. Jin, W.; Qu, M.; Jin, X.; Ren, X. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. *arXiv* **2019**, arXiv:1904.05530.
41. Wang, J.; Chang, V.; Yu, D.; Liu, C.; Ma, X.; Yu, D. Conformance-oriented predictive process monitoring in BPaaS based on combination of neural networks. *J. Grid Comput.* **2022**, *20*, 25. [[CrossRef](#)]
42. Ni, P.; Okhrati, R.; Guan, S.; Chang, V. Knowledge graph and deep learning-based text-to-GraphQL model for intelligent medical consultation chatbot. *Inf. Syst. Front.* **2022**. [[CrossRef](#)]
43. Liu, Y.; Chen, G. Real quadratic-form-based graph pooling for graph neural networks. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 580–590. [[CrossRef](#)]
44. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [[CrossRef](#)]
45. Sun, Y.; Han, J.; Yan, X.; Yu, P.S.; Wu, T. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proc. VLDB Endow.* **2011**, *4*, 992–1003. [[CrossRef](#)]
46. Hamilton, W.; Bajaj, P.; Zitnik, M.; Jurafsky, D.; Leskovec, J. Embedding logical queries on knowledge graphs. *arXiv* **2019**, arXiv:1806.01445.
47. Moreira, C.; Calado, P.; Martins, B. Learning to rank academic experts in the DBLP dataset. *Expert Syst.* **2015**, *32*, 477–493. [[CrossRef](#)]
48. Yenter, A.; Verma, A. Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis. In Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 19–21 October 2017; pp. 540–546.
49. Hussein, R.; Yang, D.; Cudré-Mauroux, P. Are meta-paths necessary? Revisiting heterogeneous graph embeddings. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 437–446.
50. Bian, R.; Koh, Y.S.; Dobbie, G.; Divoli, A. Network embedding and change modeling in dynamic heterogeneous networks. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 861–864.
51. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.