

# Biasing second-order random walk sampling for heterogeneous graph embedding\*

1<sup>st</sup> Mauricio Soto-Gomez  
*Dipartimento di Informatica*  
*Università degli Studi di Milano*  
Milan, Italy  
0000-0001-5977-9467

2<sup>nd</sup> Carlos Cano  
*Department of Computer Science and*  
*Artificial Intelligence,*  
*University of Granada*  
Granada, Spain  
0000-0002-0181-2444

3<sup>rd</sup> Justin Reese  
*Lawrence Berkeley National Laboratory*  
Berkeley, USA  
0000-0002-2170-2250

4<sup>th</sup> Peter N. Robinson  
*Berlin Institute of Health - Charité,*  
*Universitätsmedizin*  
Berlin, Germany  
0000-0002-0736-9199

5<sup>th</sup> Giorgio Valentini  
*Dipartimento di Informatica*  
*Università degli Studi di Milano*  
Milan, Italy  
*ELLIS - European Laboratory for*  
*Learning and Intelligent Systems*  
Milan unit, Italy  
0009-0001-6278-1207

6<sup>th</sup> Elena Casiraghi  
*Dipartimento di Informatica*  
*Università degli Studi di Milano*  
Milan, Italy  
*Lawrence Berkeley National Laboratory*  
Berkeley, USA  
*ELLIS - European Laboratory for*  
*Learning and Intelligent Systems*  
Milan unit, Italy  
elena.casiraghi@di.unimi.it 0000-0003-2024-7572

**Abstract**—We present *heterogeneous-node2vec*, a novel method that leverages the well-known *node2vec* algorithm to enable the generation of random-walk samples in a heterogeneous context. Specifically, we propose a strategy to bias the random walk, enabling type-aware transitions between different node and edge types. We evaluate the proposed technique on node-label prediction tasks, applied to various real-world, complex networks. A comparison with state-of-the-art techniques for heterogeneous graph embedding demonstrates that our strategy achieves competitive results for node-label prediction. This evidences that graph representation methods based on heterogeneous random-walk sampling can attain strong performance on standard supervised tasks when the sampling procedure incorporates the semantic information defined by the type heterogeneity of entities within the graph. This approach provides an effective and scalable solution for representing and learning from complex heterogeneous graphs.

**Index Terms**—machine learning, graphs and networks, algorithms for data and knowledge management

## I. INTRODUCTION

The use of heterogeneous graphs has become a ubiquitous model for representing interactions between typed entities in complex networks. A natural problem associated with the analysis of knowledge encoded in these typically massive structures is their condensed and efficient representation. To this aim, Network Representation Learning (NRL) seeks to embed complex networks into low-dimensional vector spaces while preserving their structural properties. The representation

of nodes and edges as vectors enables the application of machine learning models for various supervised and unsupervised tasks, including node classification, link prediction, visualization, and clustering.

Most of the studies in the literature have focused on the analysis of homogeneous graphs. However, recent efforts have shifted toward the representation of more general heterogeneous networks, that is, graphs whose nodes and edges are characterized by different types. Research in this area, commonly referred to as Heterogeneous Graph Representation Learning (HGRL) or Multi-relational Learning, aims to produce low-dimensional representations that capture both the structural properties of the network and the semantic relationships induced by the heterogeneous components of the graph.

Among the main HGRL strategies in the literature, four primary research directions stand out.

*Factorization methods* work by approximating the graph's structural information (usually encoded by the adjacency or Laplacian matrix) into a low-dimensional vector space. The idea is to factorize a large, sparse matrix that represents relationships between nodes (or edges) into smaller matrices, where each row or column corresponds to a low-dimensional embedding of a graph entity. These methods aim to preserve the most important structural or semantic information of the graph. Despite being effective, these techniques are generally not scalable to large graphs.

*Graph Neural Networks* (GNNs) compute vectorial representations of graph nodes by leveraging deep neural network

\* This work was supported by National Center for Gene Therapy and Drugs Based on RNA Technology—MUR (Project no. CN 00000041) funded by NextGeneration EU program

encoders, which recursively aggregate information from the neighborhoods of nodes [9], [11], [17], [26], [29], [30]. While highly effective, these methods face scalability challenges, especially when applied to large-scale graphs. Moreover, the embeddings produced by GNNs are typically tailored to a specific predictive task, limiting their generalizability across different prediction tasks.

*Relational-learning* approaches use contrastive learning to embed entities and relationships into low-dimensional space by representing relations as operators that combine the representations of their extreme entities [1], [23], [25], [27]. While effective, these models may not fully capture the graph’s structural nuances, such as higher-order connections or intricate dependencies between nodes. Additionally, since these models are often tailored to a specific predictive task (e.g., link prediction), they may not generalize well across multiple tasks or datasets without additional fine-tuning.

*Random-Walk* (RW) based methods represent entities and relations by sampling the neighborhood of nodes using a biased path generated according to a stochastic process [5], [10], [12], [15], [31]. Sampled random walks are then input into shallow neural networks to obtain vector representations of the graph components [14]. Inspired by word embedding strategies used in the natural language processing field, this representation technique aims to preserve the neighborhoods of nodes and edges as defined by the samples generated through the random walks.

In the context of homogeneous networks, DeepWalk [16] is a seminal approach that generates simple random walks starting from each node in the graph. This method is extended in node2vec [10], where random walks are generated based on second-order random walks, which can be parameterized to create biased node neighborhoods that mimic local (BFS) or global (DFS) visit patterns.

In this work, we propose a novel heterogeneous-node2vec method for learning node embeddings in complex heterogeneous networks. Heterogeneous-node2vec extends the node2vec algorithm to handle heterogeneous graphs by incorporating the semantic information induced by node/edge types into the random walk generation process. The method is defined in an efficient and flexible manner, allowing for the definition of diverse sampling strategies that exploit the structural and semantic properties of the target graphs.

Our work makes the following contributions:

- We introduce *heterogeneous-node2vec*, a type-aware random walk-based sampling strategy that incorporates node/edge type information in the generation of vectorial graph representations.
- We propose a simple and flexible sampling strategy implementation that allows for customizable representations, depending on both the input network and the predictive task. For instance, it can focus on a specific node/edge type or manage under-represented node/edge types effectively.
- We provide an efficient implementation of *heterogeneous-node2vec* that preserves the scalability of homogeneous

methods.

- We conduct node-label prediction experiments on real-world graphs, demonstrating how heterogeneous vectorial representations can enhance the performance of standard predictive tasks.

## II. METHODS

Let  $G = (V, E)$  be a heterogeneous multigraph where  $\phi : V \rightarrow \Sigma_\phi$  denotes the function defining the type of a node, and  $\psi : E \rightarrow \Sigma_\psi$  denotes the function defining the type of an edge.

### A. heterogeneous-node2vec Random Walk Generation

Along the generation process of a random walk, let  $X_t$  denote the node visited at step  $t$ , and  $E_{t+1}$  the edge traversed from the node  $X_t$  to the node  $X_{t+1}$ . Transition probabilities from a node visited by a random walk will be defined according to a second-order random walk. Namely, consider a random walk currently residing at node  $X_t = v$ , coming from node  $X_{t-1} = r$  through the edge  $E_t = e_{rv}$ . If  $x$  is a neighbor of  $v$ , *heterogeneous-node2vec* computes the transition probability of stepping to  $X_{t+1} = x$  through an edge  $E_{t+1} = e_{vx}$  in a way that is proportional to the function  $\hat{\pi}_{rvx, e_{rv}e_{vx}}$  as follows:

$$\hat{\pi}_{rvx, e_{rv}e_{vx}} = \Phi_{sc} \cdot \alpha_{pq} \cdot w_{e_{vx}} \quad (1)$$

$$P(X_{t+1} = x, E_{t+1} = e_{vx} | X_t = v, X_{t-1} = r, E_t = e_{rv}) = \hat{\pi}_{rvx, e_{rv}e_{vx}} / C, \quad (2)$$

where  $\hat{\pi}_{rvx, e_{rv}e_{vx}}$  denotes the unnormalized transition probability and  $C$  is a normalization constant. The function  $\hat{\pi}_{rvx, e_{rv}e_{vx}}$  (equation 1) is the product of three terms: the weight  $w_{e_{vx}}$  over the edge  $e_{vx}$  connecting  $v$  and  $x$ ; the function  $\alpha_{pq}$ , defined as in node2vec, and accounting for the structural properties of the network; and the function  $\Phi_{sc}$ , which accounts for the semantic properties of the network and depends on both the type of the nodes and the type of the edges involved in the transition.

More precisely, function  $\alpha_{p,q}$  depends only on the hop-distance  $d_{rx}$  between nodes  $r$  and  $x$  and is defined as:

$$\alpha_{p,q}(r, x) = \begin{cases} \frac{1}{p} & \text{if } d_{rx} = 0 \\ 1 & \text{if } d_{rx} = 1, \\ \frac{1}{q} & \text{if } d_{rx} = 2 \end{cases} \quad (3)$$

The hyperparameters  $p$  (return/inward) and  $q$  (in-out/explore) control the random walk’s behavior biasing the walk towards a depth-first search (DFS)-like ( $p \gg q$ ) exploration or towards a breadth-first search (BFS)-like exploration ( $p \ll q$ ).

On the other hand, the function  $\Phi_{sc}$  is the product of two parametric functions  $\beta_s$  and  $\gamma_c$ , which depend respectively on the type of the nodes and the type of the edges involved in the transition:

$$\Phi_{sc}(v, x, e_{rv}, e_{vx}) = \beta_s(v, x) \cdot \gamma_c(e_{rv}, e_{vx}), \quad (4)$$

The multiplicative structure of the function  $\Phi_{sc}$  decouples the contribution to the transition probability from changes in the node types ( $\beta_s$ ) and the edge types ( $\gamma_c$ ) along the second-order random walk. More precisely,  $\beta_s$  and  $\gamma_c$  bias the walk according to the following user-defined parameters:  $s$ , the *node-type switching* weight, biasing the second-order random walk according to node-type transitions; and  $c$ , the *edge-type switching* weight, biasing the walk according to the node and edge types.

Note that different definitions of  $\beta_s$  and  $\gamma_c$  allow the construction of customized switching strategies that can bias the walk to favor specific node/edge type transitions.

In particular, we defined the two strategies detailed in the following.

1) *Special Switching Strategy*: In several real-world applications, predictive tasks focus on a specific subset of node types. To handle this scenario, we can partition the set of node types into two subsets: special node-types (those of interest) and non-special ones. Based on this partition, a special node-type switching strategy defines transition probabilities that either promote or demote switching between the special and non-special nodes. For instance, by defining the function  $\beta_s$  as:

$$\beta_s(v, x) = \begin{cases} \frac{1}{s} & \text{if } \phi(x) \text{ is special} \\ 1 & \text{otherwise,} \end{cases} \quad (5)$$

transitions into nodes with special types are either promoted ( $s < 1$ ) or demoted ( $s > 1$ ). It is important to note that various strategies of this type can be constructed following this schema. Similarly, a special edge-type switching strategy can be defined, where transitions are promoted or demoted based on the presence of edges with special types.

2) *Generic Switching Strategy*: This strategy biases transitions in such a way that random walks either maintain or swap node and edge types during the traversal in a generic way, i.e. without considering any special node. That is, we have a “switch” any time we move from a node or edge type to any other node or edge type. This behavior can be achieved by defining:

$$\beta_s(v, x) = \begin{cases} \frac{1}{s} & \text{if } \phi(v) \neq \phi(x) \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

where the parameter  $s$  can be adjusted to either preserve node types ( $s > 1$ ) or switch them ( $s < 1$ ) along the random walk. Similarly, the function  $\gamma_c$  can be defined to control the preservation ( $c > 1$ ) or switching ( $c < 1$ ) of edge types along the random walk.

Both special and generic switching strategies can be especially useful in predictive tasks involving node and edge types that are underrepresented in the heterogeneous network. These strategies can bias the random walk to favor or limit transitions

to specific node and edge types, even when these types are less frequent compared to others. We outline that node2vec is a particular case of *heterogeneous-node2vec*, when  $s = 1$ , i.e., when we have no type-aware switching.

### B. Node Vector Representation

The random walks generated according to the strategies described in Section II-A are used as input to a neural network architecture that produces vector representations for the network nodes. To achieve this, we use the Skipgram model [14], a widely adopted technique that has demonstrated its promise in learning node representations. In particular, early studies [10], [16] showed that Skipgram, especially when combined with negative sampling, effectively captures network structure, even for infrequent nodes; this is a key advantage over approaches like CBOW. Additionally, Mikolov et al. [14] provided a comprehensive comparison between Skipgram and CBOW for word embeddings, where Skipgram consistently performed better in capturing fine-grained contextual relationships—a finding that has been successfully translated to the graph domain.

In our graph-based context, the Skipgram architecture consists of a single hidden layer trained to predict the neighborhood of a target node across a random walk sample in a given graph. The neighborhood of a target node  $v$ , also referred to as its context, comprises the nodes that appear within a fixed-size window around  $v$  in the generated random walk path. The objective of the Skipgram model is to maximize the probability of predicting the context nodes, given a target node. This probability is computed using the softmax function applied to the product of the vector representations of the target and context nodes.

### C. heterogeneous-node2vec Implementation

*heterogeneous-node2vec* is implemented using the efficient random walk generation provided by the GRAPE Python library [3], which supports both first-order and second-order random walk generation. It achieves high efficiency by utilizing compact data structures and an optimized Rust implementation, with Python bindings for ease of use.

The sampling process in *heterogeneous-node2vec* can be implemented optimally when the functions associated with the switching strategy depend solely on the types of the nodes involved in the transition, and not on the direction of the transition. This is the case for both the generic switching strategy and the special switching strategies defined in 5 and 6, respectively.

According to these strategies, the function  $\beta_s$  can be precomputed and incorporated into the edge weights as a multiplicative factor. This precomputation ensures that using *heterogeneous-node2vec* with either the generic or special switching strategies does not introduce additional space or time complexity compared to the original node2vec algorithm.

## III. DATA AND EXPERIMENTAL SET-UP

To empirically evaluate the quality of the vector representations obtained with *heterogeneous-node2vec*, we utilized the

benchmark framework proposed in [28], which enables the comparison of various Heterogeneous Graph Representation Learning (HGRL) algorithms.

Specifically, we employed the four heterogeneous graphs provided by the authors (subsection III-A) and performed a node-label prediction task. The framework supplies a labeled node set for each benchmark graph along with an evaluation pipeline, ensuring an objective and fair comparison of node-label prediction performance across methods.

The evaluation pipeline for node-label prediction involves five stratified holdouts (provided by the authors), with an 80:20 train:test split. For each holdout, a linear support vector machine (SVM) [6] is trained on the embeddings of the training nodes. The micro-F1 and macro-F1 scores are then computed on the test nodes.

Finally, the results across the five holdouts are averaged to obtain the overall micro-F1 and macro-F1 scores, providing a robust measure of performance.

Using the evaluation pipeline and the train-test splits provided by the authors, we conducted two sets of experiments.

The first experiment (subsection IV-A) assesses the impact of incorporating semantic information into the second-order random walk sampling process. This evaluation focuses on understanding how the random walk bias induced by using the node-type information influences the quality of the generated embeddings.

The second experiment (subsection IV-B) evaluates *heterogeneous-node2vec* by comparing its performance against several state-of-the-art HGRL methods. This comparison provides insights into the competitive advantages of *heterogeneous-node2vec* in heterogeneous graph representation learning.

#### A. Datasets

1) **Freebase Network:** The Freebase network is derived from the collaborative knowledge base Freebase<sup>1</sup>, which contains relations across domains such as books, films, music, sports, people, locations, organizations, and businesses. Each node is associated with a unique type but does not have attributes. Edge types are determined by the types of their endpoint nodes.

Freebase is the largest graph used in the experiments in terms of node cardinality, comprising 12,164,758 nodes and 62,982,566 edges. It also exhibits the highest diversity in types, featuring eight distinct node types connected by 36 edge types. Both node and edge types are distributed unevenly, with two specific types dominating significantly over the others.

2) **DBLP Network:** The DBLP network is derived from the well-known DBLP dataset<sup>2</sup>, which collects bibliographical information on computer science publications. It ranks second in terms of node cardinality (1,989,077 nodes) but contains the largest number of edges (258,850,593 edges), resulting in the highest mean node degree among the networks.

This dataset, constructed by [28], is an attributed multi-graph where node types include authors, phrases, venues, and years, connected by six distinct edge types. Nodes are attributed with 300-dimensional feature vectors. Each phrase node is linked to the authors, venues, and year nodes associated with the paper from which the phrase originated.

Attributes for `phrase` and `paper` nodes were generated by aggregating the word2vec representations of their constituent words. For `author`, `venue`, and `year` nodes, attributes were obtained by aggregating the feature vectors of their related papers (e.g., papers authored by an individual, published in a specific venue, or within a given year). Additionally, a small subset of authors has been categorized into 12 research groups spanning four research areas through a web mining process, which is used for the node-label prediction task.

3) **Yelp Network:** The Yelp network represents relationships among reviews (`phrases`), businesses, locations, and star ratings, extracted from the Yelp dataset<sup>3</sup>. The network contains 82,465 nodes and 30,542,675 edges, with four node types and four edge types.

A notable characteristic of this network is the over-representation of one node type. Specifically, `phrase` nodes dominate the graph, accounting for approximately 91% of all nodes. Similarly, edge types are largely dominated by the `phrase-context-phrase` type, which constitutes 91% of the total edges.

4) **PubMed Network:** The PubMed network is constructed from the PubMed database<sup>4</sup>, comprising four node types: genes, diseases, chemicals, and species. All nodes are extracted from PubMed articles using the AutoPhrase algorithm [18], typed using bioNER [19], and further filtered by human experts. Each node in the network is attributed.

Among the benchmark networks, PubMed is the smallest, containing 63,109 nodes and 244,986 edges. Compared to other datasets, PubMed exhibits a more balanced distribution of node types, edge types, and (disease) labels.

### IV. RESULTS AND DISCUSSIONS

#### A. Node Label Prediction: Switching Parameter Sensitivity

In this section, we empirically evaluate the relationship between the switching parameter in *heterogeneous-node2vec* and the performance of the node-label prediction task. To avoid confounding effects, we focus solely on node-type heterogeneity. We conduct experiments by fixing the return parameter  $1/p = 0.25$  and the outward parameter  $1/q = 4$ , while varying the value of the (generic or special) node-type switching parameter  $\beta_s$  in the range  $[10^{-1}, 10^2]$ . This analysis aims to assess whether the semantic information induced by node types impacts the quality of the graph representation.

Figure 1 illustrates the effect of the switching parameter  $s$  on the Macro/Micro-F1 values obtained in the node-label prediction task across all benchmark networks, comparing the generic (left column) and special (right column) switching





















<sup>1</sup><http://www.freebase.com>

<sup>2</sup><https://dblp.org>

<sup>3</sup><https://www.yelp.com/dataset/challenge>

<sup>4</sup><https://www.ncbi.nlm.nih.gov/pubmed/>

TABLE I  
NODE TYPE DISTRIBUTION AND BASIC DEGREE STATISTICS IN THE  
BENCHMARK HETEROGENEOUS NETWORK.

Graph	Node type distribution		Degree		
			mean	min	max
Freebase	music	0.466 	14.52	1	1,086,802
	<b>book</b>	0.228 	3.73	1	131,957
	people	0.130 	5.17	1	130,116
	location	0.060 	9.35	1	684,726
	film	0.051 	11.36	1	100,825
	business	0.041 	11.15	1	445,716
	organization	0.013 	7.85	1	174,646
	sports	0.011 	19.289	1	1,170,520
DBLP	<b>author</b>	0.8880 	188.68	1	71,861
	phrase	0.1094 	739.55	1	294,453
	venue	0.0026 	981.97	1	54,396
	year	0.00004 	58742.77	1	385,014
Yelp	phrase	0.9088 	761.57	1	121,333
	<b>business</b>	0.0906 	357.14	78	3,625
	location	0.0005 	191.64	1	2,443
	stars	0.0001 	830.44	4	2,239
Pubmed	chemical	0.420 	8.04	1	7,272
	<b>disease</b>	0.319 	7.48	1	18,714
	gene	0.215 	6.83	1	2,474
	species	0.045 	5.69	1	1,008

strategies. Special nodes are those highlighted in bold in Table I.

Using the generic switching strategy (Figure 1, left column), we observe a decline in performance for Freebase and DBLP as the switching probability  $\beta_s = 1/s$  increases. Conversely, Yelp and PubMed show an opposite trend, with performance improving as  $\beta_s$  increases. This behavior can be attributed to the degree distribution of labeled nodes within the networks. In Freebase and DBLP, labeled nodes (e.g., **book** in Freebase and **author** in DBLP) exhibit relatively lower mean degrees compared to other node types (see Table I). As a result, promoting heterogeneity through generic switching may lead to random walks that fail to adequately capture the local topological neighborhoods of these labeled nodes.

In contrast, for networks where the labeled nodes have higher mean degrees and are more interconnected (e.g., Yelp and PubMed), incorporating node-type heterogeneity through the generic switching strategy proves beneficial, as it enriches the representation by exploring diverse neighborhoods.

Under the special switching strategy (Figure 1, right column), the behavior differs. Higher biases toward special-type nodes ( $1/s > 1$ ) enhance performance in Freebase and Yelp but degrade it in DBLP and PubMed. This variation likely relates to the proportion of special nodes within the network. When special nodes represent a small fraction of the overall network (e.g., Freebase and Yelp), emphasizing these nodes in the sampling process yields a more informative representation, as it ensures the inclusion of relevant nodes in the random walk.

However, when the special nodes are over-represented (e.g., **author** nodes in DBLP), biasing the walks toward these

nodes leads to an over-concentration within a subset of the network. This confinement restricts the walk’s ability to explore diverse contexts, reducing the accuracy of the representation for these nodes. Similarly, in PubMed, where node types are more balanced, excessive focus on special nodes disrupts the network’s overall structural representation, leading to diminished performance.

We emphasize that, in both cases, leveraging semantic information significantly enhances the quality of the final representation. Consequently, a careful selection of this parameter can markedly improve the prediction performance of traditional homogeneous random walk (RW)-based embedding techniques. Note that by setting  $1/s = 10^0$  in Fig. 1, *heterogeneous-node2vec* boiled down to the classical node2vec algorithm.

#### B. Node Label Prediction. Comparison with state-of-the-art HGRL methods

To demonstrate the effectiveness of *heterogeneous-node2vec* compared to state-of-the-art techniques, we adopted the experimental setup published in [28] and described in Section III, ensuring a fair comparison with the following methods for node-label prediction in heterogeneous graphs:

- Random walk-based embedding methods: metapath2vec [5], PTE [22], Aspem [20], HIN2Vec [7], and HEER [21];
- GCN-based embedding methods: R-GCN [17], HAN [24], HGT [13], and MAGNN [8];
- Relational learning neural methods: TransE [2], DistMult [27], ConvE [4], and ComplEx [23].

The results for the node-label prediction task are summarized in Table II, where *heterogeneous-node2vec-special* refers to the special node-type switching strategy, where the special node-type is the one targeted by the node-label prediction task. In the table, for both the *heterogeneous-node2vec* settings, we report only the values obtained by the two extreme values of the node-type switching parameter  $1/s = 0.1, 100$ . To avoid confounding effects, the edge-type switching parameter is set to  $1/c = 1$ .

Results highlight the superiority of *heterogeneous-node2vec* across all graphs except DBLP. These findings indicate that the proposed heterogeneous random walk (RW) approach effectively captures both the structural and semantic information embedded in the graphs.

Among the two node-type switching strategies, the special node-type switching strategy achieved the best overall results. However, the generic type switching strategy also outperformed competing heterogeneous graph representation learning (HGRL) methods on average. By carefully tuning the  $1/s$  parameter, both strategies yield results that are either superior to or on par with state-of-the-art methods.

## V. CONCLUSIONS

In this paper, we introduced *heterogeneous-node2vec*, a biased second-order random walk strategy that leverages the well-known node2vec algorithm for heterogeneous graph embedding.

TABLE II  
PERFORMANCE METRIC FOR THE NODE-LABEL PREDICTION TASK ON THE BENCHMARK GRAPHS<sup>A</sup>.

Model	Node label prediction (Macro-F1 & Micro-F1)							
	DBLP		Yelp		Freebase		Pubmed	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
<i>heterogeneous-node2vec</i> generic								
1/s = 0.1	30.09	37.53	5.07	23.85	<u>36.46</u>	54.70	11.89	14.10
1/s = 100	28.23	38.03	5.52	<b>24.73</b>	31.06	51.93	<u>13.87</u>	16.10
1/c = 0.1	26.91	31.71	5.09	23.96	35.62	54.18	9.68	13.22
1/c = 100	27.01	34.14	5.20	24.09	27.79	50.47	11.53	15.85
<i>heterogeneous-node2vec</i> special								
1/s = 0.1	31.67	41.26	5.09	23.96	33.34	53.25	<b>18.84</b>	<b>21.58</b>
1/s = 100	26.55	33.82	<b>6.11</b>	<b>24.73</b>	<b>38.93</b>	<b>55.45</b>	8.77	11.67
1/c = 0.1	30.96	38.18	5.08	23.91	29.08	51.71	13.82	17.40
1/c = 100	25.83	33.65	<u>6.08</u>	<b>24.73</b>	35.70	<u>54.77</u>	9.90	12.55
node2vec	29.04	37.22	5.09	23.96	36.04	54.71	11.93	15.85
metapath2vec	<b>43.85</b>	<b>55.07</b>	5.16	23.32	20.55	46.43	12.90	15.51
PTE	<u>43.34</u>	<u>54.53</u>	5.10	23.24	10.25	39.87	09.74	12.27
HIN2Vec	12.17	25.88	5.12	23.25	17.40	41.92	10.93	15.31
AspEm	33.07	43.85	5.40	23.82	23.26	45.42	11.19	14.44
HEER	09.72	27.72	5.03	22.92	12.96	37.51	11.73	15.29
R-GCN	09.38	13.39	5.10	23.24	06.89	38.02	10.75	12.73
HAN	07.91	16.98	5.10	23.24	06.90	38.01	09.54	12.18
MAGNN	06.74	10.35	5.10	23.24	06.89	38.02	10.30	12.60
HGT	15.17	32.05	5.07	23.12	23.06	46.51	11.24	18.72
TransE	22.76	37.18	5.05	23.03	31.83	52.04	11.40	15.16
DistMult	11.42	25.07	5.04	23.00	23.82	45.50	11.27	15.79
ComplEx	20.48	37.34	5.05	23.03	35.26	52.03	09.84	<u>18.51</u>
ConvE	12.42	26.42	5.09	23.02	24.57	47.61	13.00	14.49

<sup>A</sup> In each graph, the highest-performing result is emphasized using bold font, while the second-best performance is indicated with underlined text.

Empirical results—reported in subsection IV-A—highlight the advantages of incorporating node and edge heterogeneity into graph embedding techniques. Traditional methods like node2vec and DeepWalk, originally designed for homogeneous graphs, can be applied to heterogeneous networks but often fail to capture the diversity of node types and relationships, resulting in suboptimal performance in practical scenarios. In contrast, many state-of-the-art methods for heterogeneous graphs are highly dependent on specific graph characteristics, rely on large and non-scalable neural network architectures, or produce embeddings tailored to specific predictive tasks. *heterogeneous-node2vec* addresses these limitations by introducing type-aware second-order random walks, providing a scalable and flexible alternative that captures both the topological structure and semantic diversity of graphs.

The embeddings generated by *heterogeneous-node2vec* are task-agnostic, enabling application to various downstream tasks. Comparison with state-of-the-art methods across multiple benchmark datasets—reported in subsection IV-B—demonstrates that *heterogeneous-node2vec* offers effective strategies for balancing the exploration of graph heterogeneity, based on the graph’s topological characteristics and the relative distribution of node types.

In particular, we show that increasing heterogeneity in random walks enhances representation by emphasizing relationships between diverse node types in more evenly distributed networks, such as Freebase and PubMed, thereby enriching the semantic understanding of the graph. In datasets

with less evenly distributed nodes, such as Yelp, promoting homogeneity in the walks improves performance by focusing on specific target node types. This is especially beneficial when addressing underrepresented nodes or edges in the graph.

However, excessive focus on specific node types can result in overly homogeneous embeddings that fail to capture the broader network structure, thus limiting representation capabilities. Therefore, carefully tuning the switching parameters ( $s$  and  $c$ ) is critical for optimizing performance, as shown by our results, which are competitive with respect to state-of-the-art methods for heterogeneous graphs. Additionally, *heterogeneous-node2vec* exhibits time complexity comparable to its homogeneous counterpart, node2vec, making it scalable for large networks.

While *heterogeneous-node2vec* demonstrates strong performance, its full potential remains underexplored. Future work will focus on analyzing the interplay between node/edge type distributions and model parametrization, incorporating techniques to automatically learn and dynamically update the switching strategies for even more robust and adaptive representations; furthermore, we plan to also consider edge prediction tasks, and to extend the *heterogeneous-node2vec* strategy to dynamic heterogeneous graphs.

#### CODE AND DATA AVAILABILITY

*heterogeneous-node2vec* is implemented using the GRAPE library [3] and available at [https://github.com/AnacletoLAB/hetnode2vec\\_ensmallen](https://github.com/AnacletoLAB/hetnode2vec_ensmallen).

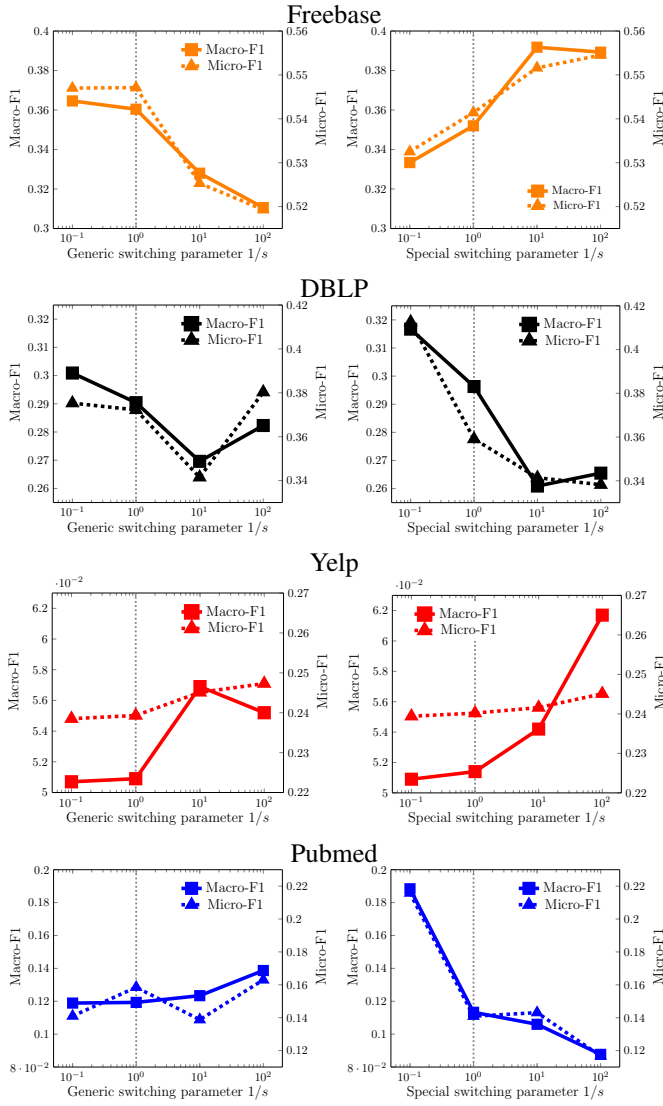


Fig. 1. Node-label prediction: macro-F1 (left axis - continuous line) and micro-F1 (right-axis - dashed line) values obtained for varying values of the  $\beta_s$  parameter. Plots in the first column depict the variation of the performance metrics when the generic switching strategy is used; the second column shows the results obtained when using the special switching strategy. In each plot, different scales are used for the left (macro-F1) and right (micro-F1) axis. The plots referring to the same graph use the same macro-F1 and micro-F1 scales to allow a comparison between the generic and the special node-type switching strategy.

The datasets and the benchmark pipeline used in the experiments follow the experimental set-up proposed in [28] and are available from <https://github.com/yangji9181/HNE>.

#### ACKNOWLEDGMENT

This work was supported by National Center for Gene Therapy and Drugs Based on RNA Technology—MUR (Project no. CN 00000041) funded by NextGeneration EU program. JR was funded by the Director, Office of Science, Office of Basic Energy Sciences of the U.S. Department of Energy Contract No. DE-AC02-05CH11231.

#### REFERENCES

- [1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [2] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the AAAI conference on artificial intelligence*, volume 25, pages 301–306, 2011.
- [3] Luca Cappelletti, Tommaso Fontana, Elena Casiraghi, Vida Ravanmehr, Tiffany J. Callahan, Carlos Cano, Marcin P. Joachimiak, Christopher J. Mungall, Peter N. Robinson, Justin Reese, and Giorgio Valentini. Grape for fast and scalable graph processing and random-walk-based embedding. *Nature Computational Science*, 3(6):552–568, June 2023.
- [4] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [5] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17*, page 135–144, New York, NY, USA, 2017. Association for Computing Machinery.
- [6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *the Journal of machine Learning research*, 9:1871–1874, 2008.
- [7] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1797–1806, 2017.
- [8] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Maggn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, pages 2331–2341, 2020.
- [9] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, page 1263–1272. JMLR.org, 2017.
- [10] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 855–864, New York, NY, USA, 2016. Association for Computing Machinery.
- [11] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [12] Yu He, Yangqiu Song, Jianxin Li, Cheng Ji, Jian Peng, and Hao Peng. Hetspacewalk: A heterogeneous spacey random walk for heterogeneous information network embedding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19*, page 639–648, New York, NY, USA, 2019. Association for Computing Machinery.
- [13] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020, WWW '20*, New York, NY, USA, 2020. Association for Computing Machinery.
- [14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.
- [15] Chanyoung Park, Donghyun Kim, Qi Zhu, Jiawei Han, and Hwanjo Yu. Task-guided pair embedding in heterogeneous network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19*, page 489–498, New York, NY, USA, 2019. Association for Computing Machinery.
- [16] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery.
- [17] M. Schlichtkrull, T.N. Kipf, P. Bloem, vandenBerg R., I. Titov, and M. Welling. Modeling relational data with graph convolutional networks.

In *The Semantic Web. ESWC 2018*, volume 10843 of *Lecture Notes in Computer Science*. Springer, 2018.

- [18] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837, 2018.
- [19] Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium, October 2018. Association for Computational Linguistics.
- [20] Yu Shi, Huan Gui, Qi Zhu, Lance Kaplan, and Jiawei Han. Aspem: Embedding learning by aspects in heterogeneous information networks. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 144–152. SIAM, 2018.
- [21] Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. Easing embedding learning by comprehensive transcription of heterogeneous information networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2190–2199, 2018.
- [22] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1165–1174, New York, NY, USA, 2015. Association for Computing Machinery.
- [23] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.
- [24] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference, WWW '19*, page 2022–2032, New York, NY, USA, 2019. Association for Computing Machinery.
- [25] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.
- [26] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2020.
- [27] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*, 2014.
- [28] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4854–4873, 2020.
- [29] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, page 793–803, New York, NY, USA, 2019. Association for Computing Machinery.
- [30] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 06 2018.
- [31] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 07 2017.