SchemaWalk: Schema Aware Random Walks for Heterogeneous Graph Embedding

Ahmed E. Samy
aesy@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

Lodovico Giaretta
lodovico@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

Zekarias T. Kefato
zekarias@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

Šarūnas Girdzijauskas
sarunasg@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

ABSTRACT

Heterogeneous Information Network (HIN) embedding has been a prevalent approach to learn representations off semantically-rich heterogeneous networks. Most HIN embedding methods exploit meta-paths to retain high-order structures, yet, their performance is conditioned on the quality of the (generated/manually-defined) meta-paths and their suitability for the specific label set. Whereas other methods adjust random walks to harness or skip certain heterogeneous structures (e.g. node type(s)), in doing so, the adjusted random walker may casually omit other node/edge types. Our key insight is with no domain knowledge, the random walker should hold no assumptions about heterogeneous structure (i.e. edge types). Thus, aiming for a flexible and general method, we utilize network schema as a unique blueprint of HIN, and propose `SchemaWalk`, a random walk to uniformly sample all edge types within the network schema. Moreover, we identify the *starvation* phenomenon which induces random walkers on HINs to under- or over-sample certain edge types. Accordingly, we design `SchemaWalkHO` to skip local deficient connectivity to preserve uniform sampling distribution. Finally, we carry out node classification experiments on four real-world HINs, and provide in-depth qualitative analysis. The results highlight the robustness of our method regardless to the graph structure in contrast with the state-of-the-art baselines.

CCS CONCEPTS

• **Computing methodologies** → *Knowledge representation and reasoning*; Unsupervised learning; • **Information systems** → Social networks;

KEYWORDS

Heterogeneous Information Network; Representation Learning; Random Walk; Network Embeddings

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9130-6/22/04.
https://doi.org/10.1145/3487553.3524728

ACM Reference Format:
Ahmed E. Samy, Lodovico Giaretta, Zekarias T. Kefato, and Šarūnas Girdzijauskas. 2022. SchemaWalk: Schema Aware Random Walks for Heterogeneous Graph Embedding. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3487553.3524728

1 INTRODUCTION

Most of today's data is highly connected. It is structured in the shape of networks — from social networks to genome and protein networks. Network embedding has become a ubiquitous approach for projecting the nodes into a dense space that captures the underlying structure of the network [5]. The embeddings can be useful in automating prediction and downstream tasks such as node classification and personalized recommendation.

Random walks are widely adopted to explore nodes within close proximity in the network [5, 17, 21]. To learn node embeddings, the SkipGram model [12] is typically trained to infer co-occurring nodes within a sliding context window over random walks. Biased random walks were also proposed to explore complex phenomena such as community and role equivalences [4]. These approaches can learn useful representations on homogeneous networks. Yet, they are less suitable for heterogeneous networks, such as that in Figure 1(a), in which multiple node and edge types are expected, as they are oblivious to these heterogeneous structures.

For Heterogeneous Information Networks (HINs), a widespread embedding approach is to guide random walks via meta-paths [2, 18]; a meta-path is a composite set of relations with a distinguishable semantic meaning. For example, in Figure 1(b), the A-P-A meta-path represents collaboration between two authors, while A-P-V-P-A connects two authors publishing in the same venue and possibly sharing research interests. Selecting the optimal meta-paths remains an open challenge in terms of time and quality. The choice of meta-paths typically requires domain knowledge and can be task-specific [9]. Some existing methods promise strategies to automatically extract all meta-paths shorter than a fixed length [3, 18]. However, the number of meta-paths can grow exponentially as their length or the number of node types increases. In either case, the results can be heavily conditioned on the chosen length [8] or the quality of the pre-defined meta-paths. Alternatively, different strategies of adjusted random walks have been proposed to equally sample all nodes types [9]. Yet we observe that, in doing so, their random walks may overlook some key semantic structures such as the collaboration interaction A-P-A in Figure 1(b). Also, because their sampling is oriented towards node types, in certain scenarios the random walker may choose not to sample a specific node type—accidentally discarding other node and edge types neighbors.

We observe that HIN embedding methods are concerned with efficiently capturing high-order semantics (e.g. meta-paths). Our core insight is that, in the absence of knowledge about the importance of edge and node types, the random walker should hold no assumptions when exploring heterogeneous structures. Ergo, we choose network schema as a meta-template of HINs [24]. As depicted in Figure 1(c), the network schema for a HIN is the smallest meta-graph with all node and edge types.

To that end, we first propose `SchemaWalk`, a flexible notion of random walk for HINs. The idea is to tweak the random walk to alleviate the bias in exploring the network schema. Precisely, we aim for a uniform sampling distribution among the *edge types*. To realize a desired distribution, the walker makes a probabilistic choice of which edge type should be sampled next. By tuning an exponential decay function, we can control how much uniformity is achieved when exploring the network schema; hence, `SchemaWalk` can be conceptualized as a general approach to explore HINs.

Second, we identify the phenomenon of *starvation*, which can affect random walkers on heterogeneous networks. Starvation may take place when certain edge types are infrequent or unevenly distributed in the graph. As a result, `SchemaWalk` and similar approaches may under- or over-sample them in each local context, potentially leading to poor learned representations. Thus, we present `SchemaWalkHO`, in which the random walker may skip direct neighbours in order to reach undersampled edge types.

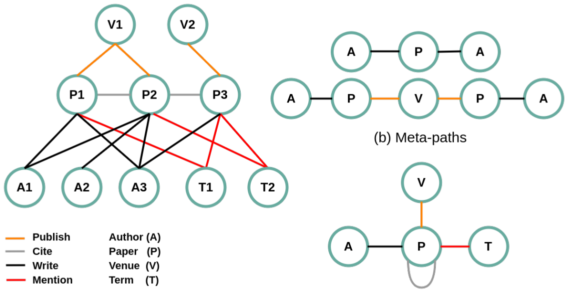
Our main contributions are highlighted as follows:

- We propose `SchemaWalk`, a flexible notion of random walk for HINs. Based on the network schema, `SchemaWalk` is a principled *general* embedding algorithm that captures composite interactions, and mitigates the dilemma of selecting meta-paths.
- We highlight the drawbacks of node type-based sampling, and suggest the edge type-based sampling as a flexible and more fine-grained approach to explore heterogeneous networks.
- We identify the *starvation* issue and propose `SchemaWalkHO` to mitigate it. By jumping further than the immediate neighbours, `SchemaWalkHO` can skip local deficient connectivity patterns and preserve the desired sampling distribution.
- We evaluate the proposed methods for multi-label classification on several real-world datasets, and further provide detailed qualitative analysis. The results show the robustness of `SchemaWalk` that achieves the best to second best performance regardless of the graph structure, in contrast with the baselines.

2 PROBLEM DEFINITION

Here, we provide key concepts to formulate the heterogeneous network embedding problem, in line with previous studies [19].

Definition 2.1. ***Heterogeneous Information Network (HIN).*** *is a graph $G = (V, E)$ where V and E are the vertex set and edge set, respectively. Given a node type mapping $\phi : V \longrightarrow: \mathcal{A}$, and an edge type mapping $\psi : E \longrightarrow: \mathcal{R}$ where $\mathcal{A}$ and $\mathcal{R}$ are the node type set, and edge type set respectively, G is a HIN when $|\mathcal{A}| + |\mathcal{R}| > 2$. Otherwise, it is called a homogeneous information network.*



(b) Meta-paths

(a) Heterogeneous Network

(c) Network schema

Figure 1: An academic Heterogeneous Information Network (HIN).

Definition 2.2. ***Network Schema.*** *Given a HIN $G = (V, E)$, a network schema is a graph $\mathcal{T}_G = (\mathcal{A}, \mathcal{R})$ that consists of all the node types $\mathcal{A}$ and the edge types $\mathcal{R}$ from graph G, as its nodes and edges respectively.*

Definition 2.3. ***Heterogeneous Network Embedding.*** *Given a HIN $G = (V, E, \mathcal{A}, \mathcal{R})$, for each node $v \in V$, the objective is to learn d-dimensional continuous embedding $f : V \longrightarrow \mathbb{R}^d$, where $d \ll V$, that represents the structure and semantics of node v in the network.*

3 SCHEMAWALK FOR HETEROGENEOUS NETWORK EMBEDDING

3.1 Sampling with SchemaWalk

Inspired by the path ranking algorithm (PRA) [11], we define the following transition probabilities:

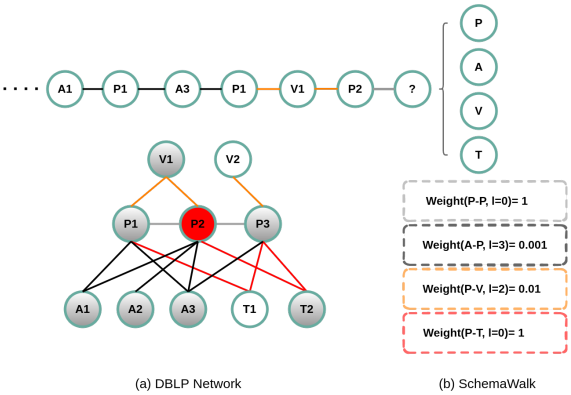
$$P_{A_l,A_{l+1}} = D^{-1}_{A_l,A_{l+1}} M_{A_l,A_{l+1}}, \tag{1}$$

where $M_{A_l,A_{l+1}}$ is the adjacency matrix for edge type $R = (A_l, A_{l+1})$ between nodes of type A_l and nodes of type A_{l+1}. $D_{A_l,A_{l+1}}$ is the degree matrix defined as $deg_{A_l,A_{l+1}}(v_i) = \sum_j M_{A_l,A_{l+1}}(v_i, v_j)$. Thus, $P_{A_l,A_{l+1}}(v_i, v_j)$ is the probability that node v_j of type A_{l+1} will be chosen next by a random walker visiting node v_i of type A_l. To guide the behavior of the `SchemaWalk`-based random walk, we first define the following probability to choose the next edge type $\Psi(i + 1)$.

$$Pr(\Psi(i+1) = R|v_i) \triangleq \begin{cases} z_{i+1}(R), & R = (\phi(v_i), A), A \in N_{\mathcal{T}_G}(v_i) \\ 0, & otherwise, \end{cases} \tag{2}$$

where $N_{\mathcal{T}_G}(v_i)$ is the neighboring node types for node v_i, and A is the type of the next node. After selecting the next edge type to be R, the next node is probabilistically chosen as follows.

$$Pr(v_{i+1} = v_j|v_i, \Psi(i{+}1) = R) \triangleq \begin{cases} P_{A_l,A_{l+1}}(v_i, v_j), & \psi(v_i, v_j) = R \\ 0, & otherwise. \end{cases} \tag{3}$$



(a) DBLP Network (b) SchemaWalk

Figure 2: SchemaWalk example on academic graph, α is set to be 0.1. The walker is at node $P2$ and about to choose a next edge type from {PP, PA, PV, PT}. The grey nodes are the candidates for the next transition, depending on the chosen edge type.

The next node v_j can be chosen from node v_i with a transition probability defined as $Eq.$1, only if they are connected through the edge type R. z_i is the normalized "visit" probabilities vector over the edge types at step i.

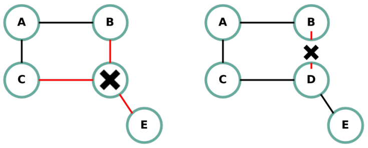
$$z_i(R) \triangleq \frac{\alpha^{l_{R_i}}}{\sum_{\hat{R}} \alpha^{l_{\hat{R}_i}}}. \tag{4}$$

Where l_{R_i} is the number of visits for the edge type R till step i, and $\alpha \in [0, 1]$.

The above equations define the basic principles of `SchemaWalk` where the walker dynamically determines the next edge type and node to visit. The key motivation is to present no assumptions over the heterogeneous structure (i.e. the edge types in the network schema). To achieve that, `SchemaWalk` targets a *global* uniform sampling distribution over edge types. In detail, the behavior of the walker is defined by the exponential decay function α^l; the more an edge type is visited, the exponentially lower its sampling weight becomes. An example of a walker on DBLP graph is in Figure 2. Moreover, the value of α decides the rate of the decay over time as the walk unfolds. Tuning α, controls the level of uniformity among the edge types within a *local* context window; Very small α results in a near-uniform *local* distributions of edge types. As α approaches 1, individual local contexts are allowed to be skewed, however the *global* sampling of edge types remains uniform. Finally, when the nodes of one type are considerably more than others (e.g. authors » venues), `SchemaWalk` may not able to sample all of them as contexts. This is a design property of `SchemaWalk`; with a high number of authors compared to venues, sampling all authors should not provide more significant contextual information.

3.2 Node vs. Edge Type-based Sampling

`SchemaWalk` differs from other HIN embedding approaches based on adjusted random walks, such as *JUST* [9] or *HeteSpaceyWalk* [6], as they seek uniform sampling distributions over the node types. We argue that uniform sampling over the edge types provides a more flexible exploration strategy. First, it can generalize to graphs in which multiple, semantically-distinct edge types connect the same pair of node types. Second, it better handles oversampling scenarios. Figure 3 shows a theoretical example of a network schema that highlights the latter difference, where the edge type $B - D$ is assumed to be oversampled. Particularly, we can observe in Figure 3(a) a node type-based approach that penalizes node D, potentially causing undersampling of other node and edge types, such as node type E and edge type $C - D$. However, this can be avoided by penalizing on an individual edge type; as in Figure 3(b) where the surrounding edge and node types remain reachable, reducing the risk of undersampling.



(a) Discarding node type "D" (b) Discarding edge type "B-D"

Figure 3: Example of a difference between penalizing on node type vs. edge type when the edge type "B-D" is overly sampled.

On the other hand, *JUST* [9] follows a rather rigid approach of node-type based sampling. Specifically, the random walker has a zero chance of revisiting the same node type for a minimum number of steps after choosing a different next node type. For example, a palindrome meta-path such as A-P-A in Figure 1 can not be sampled. Therefore, *JUST* may miss key semantic structures.

3.3 Avoiding Starvation with SchemaWalkHO

Unfortunately, on certain types of datasets, `SchemaWalk` may suffer from a *starvation* issue. For example, in the academic graph in Figure 4, P-V edges are very rare and thus a random walker may perform a large number of steps without any chance to visit any of them. The same is true for P-P edges, which are relatively frequent but concentrated mostly in a small subset of the overall graph. In this situation, based on $Eq.$4, the weights $z(R)$ of the visited edge types decrease exponentially, while those of the edge types that are lacking do not change for long periods of time.

Because of this imbalance, when the random walker finally encounters an underrepresented edge type, it has an overwhelming chance to only visit that edge type for several consecutive hops, till the $z(R)$ scores are somewhat balanced. As such, other edge types around it are ignored, leading to skewed edge visit distributions and potentially poorer node representations.

To overcome the above limitations we propose `SchemaWalkHO`, a variant of `SchemaWalk` that performs the random walks on a higher-order graph, in which all suitable edge types are present at each node (e.g. in the example of Figure 4, all papers are associated to at least one venue and at least one author).

In particular, we consider the possibility of building a weighted, fully-connected graph from the original sparse, binary adjacency matrix, using techniques that assign higher weights to direct neighbours and closer nodes. This overcomes the lack of edges in the original graph while still ensuring that most random walk transitions preserve locality. Several well-known techniques exist that can be used to build a weighted, fully-connected graph with these

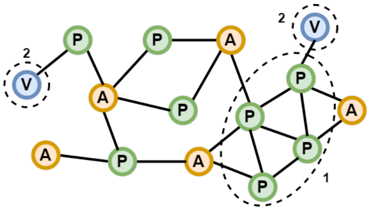


Figure 4: Example of academic graph inducing starvation in (1) P-P edges, which are relatively frequent, but concentrated mostly in a small subgraph and (2) P-V edges, which are very scarce and sparse.

properties, such as Personalized PageRank (PPR) scores [16] or Katz similarities [14]. In this work we employ the latter. From a practical perspective, we replace the sparse binary adjacency matrix M in $Eq.1$ with the dense weights matrix

$$\hat{M} = \sum_{i}^{H} \beta^i M^i \tag{5}$$

The weights in $\hat{M}$ aggregate all possible paths of length up to H between each node pair, with the influence of each path decreasing exponentially based on hyperparameter $\beta \in [0, 1]$. To ensure $\hat{M}$ is dense, assuming the original graph is connected, we set H to the diameter of the original graph.

When dealing with large graphs, to which a dense adjacency approach would not scale, it is possible to sparsify $\hat{M}$, during or after computation, while still retaining the key property that each node has access to all relevant edge types.

3.4 Random Walk based Embedding

To build node embeddings from the random walks, we adopt the SkipGram model [12]. Specifically, the model learns to maximize the joint probability of nodes appearing in the same context window k across the generated random walks corpus $\mathcal{P}$. The objective function is to minimize the following:

$$\underset{\theta}{argmin} - \sum_{w \in \mathcal{P}} \sum_{v_i \in w} \sum_{v_j \in C_{v_i, w, k}} \log(Pr(v_j | v_i, \theta)), \tag{6}$$

where w is a random walk and $C_{v_i, w, k}$ is the set of context nodes that are no further than k steps from node v_i in walk w. Typically, the joint probability is the softmax function:

$$Pr(v_j | v_i, \theta) = \frac{exp(\vec{v}_i . \vec{v}_j)}{\sum_{v_k} exp(\vec{v}_i . \vec{v}_k)}, \tag{7}$$

Given the number of the nodes is typically large, we approximate the softmax function with negative sampling, similar to [13]. Ergo, the log-probability in Eq. 6 can be formulated as follows:

$$\log(Pr(v_j | v_i, \theta)) = \log \sigma(\vec{v}_i . \vec{v}_j) + \sum_{1}^{|Neg|} \mathbb{E}_{N \sim P(v)}[\log \sigma(-\vec{v}_i . \vec{v}_N)]. \tag{8}$$

$|Neg|$ is the number of negative samples, and $P(v)$ is the sampling distribution. The above equations defines how SkipGram learns node representations in HINs based on random walks.

Table 1: The statistics of the experimental heterogeneous networks.

Dataset	Edge Types and Number of Edges				
Foursquare	U-U	U-C	C-T	C-P	-
(29771 nodes)	5695	25904	25904	25904	-
DBLP	P-P	P-A	P-V	P-T	-
(15649 nodes)	6984	13589	4258	26532	-
ACM	A-P	P-S	-	-	-
(11246 nodes)	13407	4019	-	-	-
Movie	M-A	M-C	M-D	A-A	M-M
(20784 nodes)	23223	4001	5630	43395	6194

4 EXPERIMENTS

4.1 Experimental Setup

Dataset. We evaluate on four real-life heterogeneous networks: Foursquare [23], DBLP [8], ACM [24], and Movie [9]. The statistics of the datasets are detailed in Table 1.

- **Foursquare**[23] is a graph based on users' check-in history in New York city. The graph has four types of nodes: 2,449 users (U), and 25,904 check-ins (C), 1,250 points of interest (P) and 168 timestamps (T). The edge types are U-U, U-C, C-T, C-P. Each point of interest (P) is assigned a label based on its category, e.g. "bar".
- **DBLP**[8] is an academic network with 5237 papers (P), 5915 authors (A), 18 venues (V), and 4479 topics (T). The edge types are P-P, P-A, P-V, P-T. Authors are labeled based on their research interests with one of the following four areas: "data mining", "information retrieval", "database", and "machine learning".
- **ACM**[24] is another academic network that consists of 4019 papers (P), 7167 authors (A), and 60 subjects (S). The edge types are A-P, P-S. Each paper has one of three research categories: "databases", "wireless communications", and "data mining".
- **Movie** is a movies graph [8], augmented by [9]. We experiment on the biggest connected component with 6517 movies (M), 10350 actors (A), 1335 composers (C), and 2582 directors (D). The original edge types are M-A, M-C, M-D. [9] augments the graph with edge types M-M and A-A, indicating respectively that two movies are produced by the same producer and that two actors follow each other on Twitter. Table 1 shows a huge imbalance among the edge types. With further investigation, we observe that the augmented edges of type A-A are mostly present in the network as a very small and dense region encompassing around 18% of the nodes of type A. Such deficient connectivity patterns may present a challenge to `SchemaWalk`, as explained in section 3.2. Each movie is multi-labeled with a combination of the genres "action", "horror", "adventure", "scifi" and "crime".

Baselines. To evaluate the embeddings' quality, we compare our methods with state-of-the-art random walk approaches for homogeneous and heterogeneous graphs, as follows:

- **DeepWalk**[17] learns the latent representations of the nodes by running a set of uniform random walks to explore the graph, and learning embeddings via the SkipGram [12] model. DeepWalk was originally designed for homogeneous graphs.

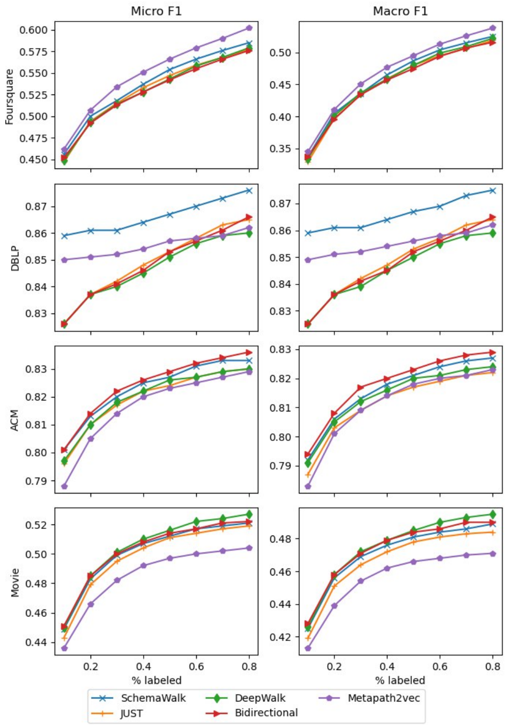


Figure 5: Results of node classification over training percentages. Note that the scale of y-axis is not the same for every dataset (e.g. the scale for DBLP is higher than that of Foursquare).

- **Metapath2Vec**[2] is a heterogeneous graph embedding method that binds the random walk to follow a set of manually defined meta-paths. Similarly, embeddings are learned via SkipGram. The meta-paths U-C-P-C-U, P-C-T-C-P (Foursquare), A-M-D-M-A and A-M-C-M-A (Movie) are chosen similar to [9]. For DBLP, the meta-paths are P-A-P and A-P-V-P-A, as in [8]. Finally, P-A-P and P-S-P are chosen as meta-paths for ACM, following [24].
- **JUST**[9] is a heterogeneous graph embedding method that biases the random walk to first balance between selecting homogeneous edges (such as P-P in DBLP), or heterogeneous edges (e.g. A-P). Second, they sample node types uniformly when choosing heterogeneous edges. SkipGram is used to learn node representations.
- **Bidirectional Random Walks** were introduced in [10] to overcome the intrinsic bias of traditional random walks that causes low-degree node to appear mostly at the very beginning of a walk, where only a smaller context window is available. By starting two independent random walks from each node and joining them as one, the starting nodes can also enjoy a full context window and thus better embeddings. While [10] combines these walks with a heterogeneous variant of SkipGram, we evaluate them with the original SkipGram algorithm.

Note there are other heterogeneous graph embedding methods that replace SkipGram with a different learning component, such as HIN2Vec [3], or that propose heterogeneous variants of it, such as Metapath2Vec++ [2] and MARU [10]. However, we devote this work to the analysis of different random walks as a sampling technique, with the belief that the most expressive of them may then be combined with different learning approaches. Therefore, a comparison with the aforementioned methods that involve more sophisticated learning components is out of the scope of this work.

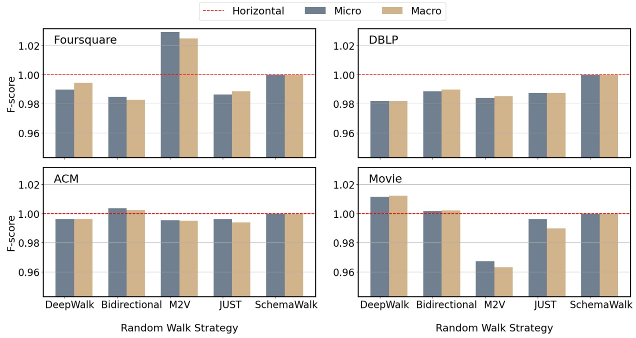


Figure 6: Normalized node classification results w.r.t. SchemaWalk.

Implementation Details. For `SchemaWalk`, we tune the decay hyper-parameter α using node classification on the Movie dataset with a search over $\alpha \in [0.1, 0.9]$ and a step of 0.1. Accordingly, α is fixed as 0.8 for all datasets. We adopt the same hyper-parameter values as DeepWalk [17] for our methods and the baselines. Specifically, we set walk length $l = 40$, number of walks $\lambda = 80$, window size $w = 5$, and dimension size $d = 128$. For bidirectional walks, we treat λ as the number of walk pairs and l as the combined length of a walk pair, in order to preserve the same amount of training data across all methods. We report the average results for 10 random data splits. For each split, the hyper-parameters are tuned with 5−fold cross validation and evaluated on 20% held-out testing set. The code and datasets are available on GitHub[1].

4.2 Node Classification Task

The task is a multi-label node classification, where every node is associated with one or more labels in the label set $\mathbb{L}$. To evaluate the quality of the resulting embeddings, we train a one-vs-all logistic regression with the same scoring function in [17]. For training, 10 training data splits/sets, each divided up at 8 different percentages [0.1, 0.8], are used to show the learning behavior and robustness of each model. For evaluation, the remaining 20% of the data is held out as a testing set. Micro-F1 and Macro-F1 are selected as the evaluation metrics for multi-label classification.

Figure 5 shows the performance of five random walk techniques on four heterogeneous networks. The results show generally comparable performances of the random walks, with `SchemaWalk` achieving the best to second best results across the datasets.

First, we observe DeepWalk and Bidirectional walks perform competitively on Movie and ACM. That suggests the homogeneous structure (i.e. regardless of node and edge types) of these networks is more relevant to classify nodes correctly. While the heterogeneous knowledge seems more crucial in Foursquare and DBLP that `SchemaWalk` and Metapath2Vec yield noticeably better performance. Thereby, we conclude that *for scenarios where homogeneous exploration of the structure is enough, approaches such as DeepWalk*

[1] https://github.com/AhmedESamy/SchemaWalk

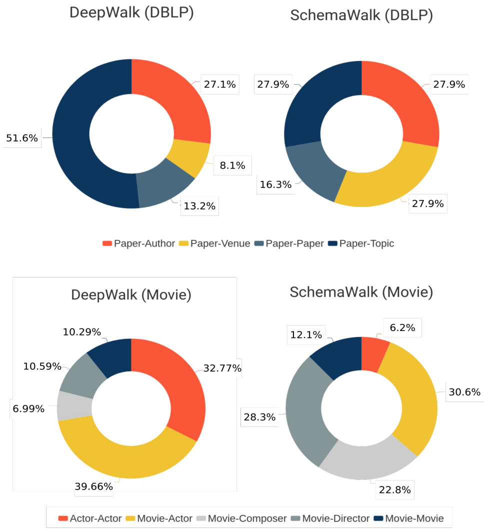


Figure 7: Doughnut charts of the edge types distribution over the generated random walks.

perform well. As the heterogeneous knowledge becomes imperative, heterogeneous network embedding approaches can be essential.

Second, SchemaWalk renders better results on DBLP than the heterogeneous network embedding baselines — as well performs consistently well on the other datasets. Particularly, it is notable in Figure 5 that SchemaWalk reaches high accuracy scores on DBLP with 10% training data, and maintains highest Micro-F1 and Macro-F1 scores across the training percentages. Moreover, viewing Figure 6, SchemaWalk is consistently high over all datasets. For instance, Metapath2Vec achieves the best result on Foursquare, but the worst on Movie. That’s because Metapath2Vec’s performance is conditioned on the quality of the chosen meta-paths and their suitability for the specific label set. Similarly, DeepWalk achieves the best performance on Movie, but the worst on DBLP. On the contrary, SchemaWalk exhibits stable performance on all graph structures, compared to both the homogeneous and heterogeneous network embedding baselines.

Finally, SchemaWalk outperforms *JUST* in all experiments. This observation validates the intuition that edge type-based sampling is a more flexible and more promising embedding approach for metapath-less heterogeneous network embeddings.

4.3 Qualitative Analysis of SchemaWalk

In this section, we provide a qualitative analysis and comparison of SchemaWalk with respect to DeepWalk. We do this through the following visualizations.

Sampling Distribution over Edge Types. Figure 7 shows distributions of the edge types across random walks generated by DeepWalk and SchemaWalk. The edge types distributions are visualized as doughnut charts based on DBLP and Movie datasets. Examining the DBLP-based charts, it is noticeable the key difference between each walk type. As SchemaWalk aims for exploring the network schema, the chart shows a near-uniform distribution over the edge types; except for the edge type "Paper-Paper" that is slightly missing in the DBLP graph. Whereas, the distribution of the edge types is skewed in the case of DeepWalk. DeepWalk biases the sampling towards the high degree nodes (or the frequent edge types) in the graph. For example, nodes of type "Venue", and their associated edge type "Paper-Venue" are considerably less frequent, thereby they are less sampled by DeepWalk.

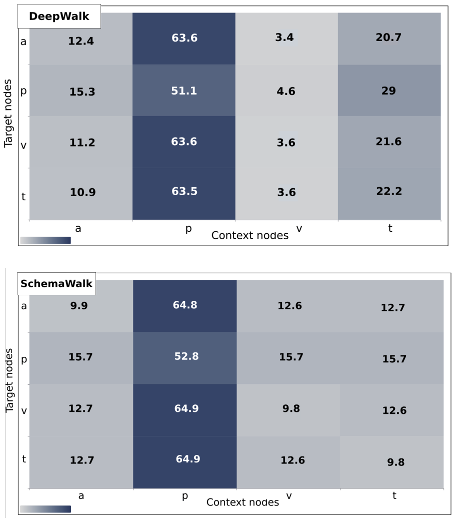


Figure 8: Heatmap charts of the correlations between the target and context node types, within SkipGram’s context window, on DBLP.

As for the Movie dataset, the graph presents structural issues, such as missing edge types (e.g. Movie-Movie) or highly-skewed distribution of edge types (e.g. Actor-Actor). Thus, sampling over the Movie graph is particularly challenging for SchemaWalk to achieve uniform distribution. For example, the starvation phenomenon is clear for the edge type "Actor-Actor"; While DeepWalk gives highest probability to the "Actor-Actor" being the most frequent edge type, SchemaWalk struggles to sample the edge type adequately. That’s caused by the highly unbalanced distribution of "Actor-Actor" with only 18% of the actors. Hence, SchemaWalk samples this edge type the least, in contrast to DeepWalk.

Correlations between Node Types. To further understand the difference between DeepWalk and SchemaWalk (as homogeneous versus heterogeneous network sampling approaches), we draw heatmaps for the co-occurrences of the node types within the SkipGram’s context windows. Figure 8 shows the heatmaps for both approaches on DBLP. Examining the figure, both approaches show high correlations for all node types with node-type P ("Paper") where P appears as a context node. That’s an expected behavior given that node-type P is a hub node in the DBLP’s network schema (Figure 1(b)). However, we can see a difference when viewing node-type P as a target node. SchemaWalk shows the same correlation between the target node-type P and each of other edge types, which highlights the property of uniform-distribution over the edge types where P is involved. In contrast, DeepWalk shows a high variance and favors P-A and P-T over P-V. That’s justified by the property of DeepWalk to favor high frequency.

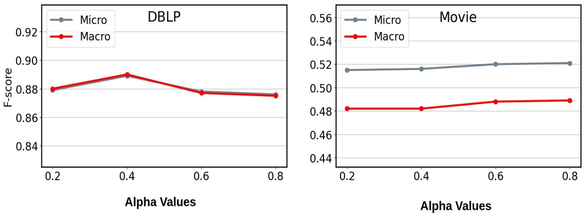


Figure 9: Classification results of SchemaWalk over different α.

Moreover, observing the columns where node-types V and T are context nodes in both charts (Figure 8), we can see that with DeepWalk, node-type V has a low chance to be chosen as a context node, while node-types A and T have high chances. Whereas, the three node types have very similar chances to be chosen in the case of `SchemaWalk`.

These observations consolidate our intuition and motivation; that `SchemaWalk` aims for exploring the network schema, while DeepWalk targets the homogeneous structure of the graph, typically favoring high degree nodes (or frequent edge types).

4.4 Impact of the Decay Parameter α

Figure 9 illustrates the impact of parameter α on the quality of the resulting node embeddings on the DBLP and Movie networks. As the random walk unfolds, the value of α controls the decay rate of the sampling probability of each edge type. Particularly, higher decay rates lead to a more uniform sampling of the edge-types within each single context window, as sampling the same edge type multiple times in rapid sequence is discouraged. As α approaches 1, this kind of local skew is allowed, while $Eq.$3 guarantees that the global sampling distribution is still uniform. During the experiments, the decay parameter α was tuned on the Movie dataset via a search over $\alpha \in$[0.1, 0.9] with a step of 0.1. In Figure 9, we report the testing results on the task of node classification.

Investigating the results, the performance on DBLP reaches its peak when $\alpha = 0.4$, then falls afterwards as the value approaches 1. While on Movie, the performance is rather steady, and improves slightly when $\alpha > 0.6$. Except for Movie dataset, we observe in the experiments that $\alpha \in [0.4, 0.6]$ usually gives the best results, assuming the other hyper-parameters are set to their default values. That suggests achieving a modest uniform sampling of the edge types within SkipGram's context window gives best performance for most of networks. Reducing the value of α can still be helpful to compensate when some nodes lack certain edge types within their direct neighborhoods.

The analysis is, yet, different for the Movie dataset; reducing α seems to worsen the starvation issue described in Section 3.3. As the edge types A-A and M-M are severely missing, the urge for `SchemaWalk` to compensate exponentially increases with smaller α, therefore leading to oversampling the missing edge type. Notably, the performance gain is still not significant regardless of the value of α; this is because node classification on Movie seems to benefit more from exploring the graph in a homogeneous manner, as discussed earlier. For evaluation, α was set to 0.8 on all datasets.

Table 2: Node classification for SchemaWalkHO VS. SchemaWalk

	DBLP		Movie	
	Micro	Macro	Micro	Macro
`SchemaWalkHO` ($\beta = 0.1$)	84.5	84.5	35.5	28.4
`SchemaWalkHO` ($\beta = 0.01$)	87.5	87.4	51.8	48.4
`SchemaWalk`	**87.6**	**87.5**	**52.1**	**48.9**

4.5 Performance of SchemaWalkHO

Table 2 compares the performance of `SchemaWalkHO` versus `SchemaWalk` in the node classification task on the DBLP and Movie datasets. It is notable that walking on the higher-order graph rather than the original negatively impacts performance, in some cases dramatically. Furthermore, when the hyperparameter β is reduced, bringing the higher-order graph closer in structure to the original, the classification performance climbs back towards the level of the original `SchemaWalk`.

The theoretical advantage of `SchemaWalkHO` is its ability to maintain the uniform sampling distribution of edge types even when some of these present skewed distributions in the original graph. However, these results indicate that this advantage is offset by harmful side-effects.

One side-effect may be the appearance of nodes that are more than k hops away from the target node within the size k SkipGram window — therefore diluting the notion of proximity that is key to the use of SkipGram for node embeddings. When the value of β is low, this phenomenon should happen infrequently on graphs with well-distributed edge types. Indeed, when $\beta = 0.01$, the performance on DBLP climbs back to near-optimal scores. On the other hand, the phenomenon still continues, despite the small value of β, when a node is lacking certain edge types from the network schema, as `SchemaWalkHO` tries to overcome this edge type deficit and avoid starvation. This is the case with M-M and A-A edges in the Movie dataset. The results show a wider performance gap on this dataset even with low β.

This indicates that "completing" deficient neighbourhoods based on structural patterns (such as number of connecting paths in our implementation) may represent a bad approximation of the underlying phenomena, at least on the Movie dataset. The lack of certain edge types may in fact be caused not by the incompleteness of the dataset, but may rather be itself part of the input signal, providing valuable information about the role of certain nodes in the graph. It is therefore important for future work to examine ways to capture these differences and deficiencies in connectivity, while still retaining the advantage that uniform sampling of edge types provides in certain circumstances.

4.6 Limitations of SkipGram in Heterogeneous Graph Embedding

The main focus of this work is to propose and compare novel random walk techniques, rather than alternative learning components to SkipGram. Nonetheless, we provide a comparison with HIN2Vec [3], that was performed as part of this research.

As shown in Table 3, HIN2Vec [3] exhibits better results for classification than `SchemaWalk` over almost all datasets. HIN2Vec

Table 3: Node classification results for SchemaWalk VS. HIN2Vec

		SchemaWalk (SkipGram)	HIN2Vec
DBLP	Micro	**0.876**	**0.876**
	Macro	0.875	**0.876**
ACM	Micro	0.833	**0.903**
	Macro	0.827	**0.901**
Movie	Micro	0.521	**0.54**
	Macro	0.489	**0.51**

does not employ the same SkipGram model as SchemaWalk and DeepWalk [17]. Rather, it proposes a different learning model that explicitly represents the heterogeneous knowledge via meta-paths and learns by predicting the right meta-path/relation for any two pair of input nodes. Therefore, the model has a bigger learning capacity with a direct access to the heterogeneous knowledge, while SkipGram is oblivious to the node types of the target-context pairs and to the edge types that connect them. We also observe that HIN2Vec does not impose strong assumptions on the random walk when exploring the graph. Therefore, we argue that much of their performance superiority results from the learning component. The same has been also observed with other state-of-the-art approaches. For example, although Metapath2Vec [2] employs meta-path guided random walks, considerable improvements are achieved only when the authors combine them with an heterogeneous variant of SkipGram. Similarly, though MARU [10] proposes a bidirectional random walk for a comprehensive exploration of graphs, its advantages are only observed when combined with heterogeneous SkipGram. Viewing the last observation as well as Figure 6 and Table 3, we hereby conclude that *adequate understanding/sampling of the network structure combined with an explicit learning/modeling of the complex knowledge in HINs is a highly recommended research direction for heterogeneous network embedding.*

5 RELATED WORK

Research on learning representations in heterogeneous networks has been on a huge rise. Many embedding methods address the task as a stochastic optimization problem [2, 3, 5, 15]. For instance, some earlier work have tried to predict binary relations between two types of nodes in heterogeneous graphs [1, 20]. TransE [1] learns entities and relation vectors where the relation vectors translate between entity types based on their co-occurrences, thus starting a research trend on knowledge graphs [15]. While PTE [20] extends LINE [21] to heterogeneous networks by extracting bipartite networks based on the edge types. To learn node embeddings, they then capture the one-hop neighborhood. However, in targeting binary relationships, these approaches overlook complex semantics of relationships between nodes. More recently, several methods explicitly harness the heterogeneous structure i.e. node types by using meta-paths [3, 6, 8]. Metapath2vec [2] extends DeepWalk [17] by restricting the random walk to follow a pre-defined set of meta-paths. While *HeteSpaceyWalk* [6] proposes a spacey random walk to approximate the stationary distribution of the meta-path based random walks. The performance of these methods, although generally satisfactory, is nevertheless conditioned on the quality of the selected meta-paths — that are typically hand-crafted by domain experts. Approaches as HIN2Vec [3] and HINE [8] avoid the latter issue by defining meta-paths under specific criteria such as maximum length. Longer length yet leads to an exponential computational increase [3] while the choice of the length may still affect the final performance [8]. To avert using meta-paths altogether, *JUST* [9] biases the random walk so that all node types are selected in a fair equal distribution. However, their notion of random walk is rather aggressive; for example a palindrome semantic sequence such as Author-Paper-Author is not possible to sample. Thereby, *JUST*may overlook key heterogeneous structures. More akin to us, *HeteSpaceyWalk* [6] leverages the network schema to guide their spacey random walk. They nevertheless aim for a balance between maintaining uniform distribution and favoring the most previously sampled, for the *node types*. Distinctly, SchemaWalk balances the *edge type* choices in the network schema and flexibly compensate for missing edges under a chosen edge type.

Finally, there are other methods such as HeGAN [7], which applies adversarial learning to HIN embeddings, Metapath2Vec++ [2], MARU [10], and HIN2Vec [3]. These methods address the learning component. This work, however, furthers the sampling process, namely, the random walk, and learns via the SkipGram model [12].

6 CONCLUSIONS

In this paper, we propose SchemaWalk, a flexible random walk for heterogeneous network embedding. The core insight of our work is that with the absence of knowledge about the importance of each node and edge type in heterogeneous graphs, the random walker should aim for a fair sampling with no assumptions over the node/edge types. Additionally, we argue that exploration based on the edge types can be more flexible and granular as opposed to sampling based on the node types, which may miss vital semantics in heterogeneous networks. Thus, we exploit the network schema to realize uniform sampling distribution over the edge types. Finally, we identify the phenomenon of *starvation* in heterogeneous networks and propose SchemaWalkHO to tackle this issue. Evaluation on multi-label node classification demonstrates the robust performance of SchemaWalk in real-life heterogeneous networks, while also hinting at the unsuitability of SkipGram for embedding heterogeneous networks. Also, we provide a detailed qualitative analysis on SchemaWalk versus DeepWalk. The final insights and conclusions can be summarized as follows: (1) heterogeneous networks present rich structures that homogeneous embedding methods may be inadequate at capturing; (2) unbiased exploration of the edge types can offer a more fine-grained and general approach to heterogeneous network embedding compared to node type-based alternatives; (3) the homogeneous nature of SkipGram makes it less suitable for heterogeneous network embedding.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813162: RAIS – Real-time Analytics for the Internet of Sports (http://rais-itn.eu).

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[2] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.

[3] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1797–1806.

[4] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[5] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[6] Yu He, Yangqiu Song, Jianxin Li, Cheng Ji, Jian Peng, and Hao Peng. 2019. Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 639–648.

[7] Binbin Hu, Yuan Fang, and Chuan Shi. 2019. Adversarial learning on heterogeneous information networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 120–129.

[8] Zhipeng Huang and Nikos Mamoulis. 2017. Heterogeneous information network embedding for meta path based proximity. *arXiv preprint arXiv:1701.05291* (2017).

[9] Rana Hussein, Dingqi Yang, and Philippe Cudré-Mauroux. 2018. Are meta-paths necessary? Revisiting heterogeneous graph embeddings. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 437–446.

[10] Jyun-Yu Jiang, Zeyu Li, Chelsea J-T Ju, and Wei Wang. 2020. Maru: Meta-context aware random walks for heterogeneous network representation learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 575–584.

[11] Ni Lao and William W Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine learning* 81, 1 (2010), 53–67.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[14] Mark E. J. Newman. 2010. *Networks: An Introduction.* Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199206650.001.0001

[15] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104, 1 (2015), 11–33.

[16] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[17] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[18] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2018), 357–370.

[19] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159.

[20] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. 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*. 1165–1174.

[21] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[22] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[23] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 3 (2016), 1–23.

[24] Jianan Zhao, Xiao Wang, Chuan Shi, Zekuan Liu, and Yanfang Ye. 2020. Network Schema Preserved Heterogeneous Information Network Embedding. In *29th International Joint Conference on Artificial Intelligence (IJCAI)*.

A APPENDIX

Embeddings Visualization. Based on TSNE [22], Figure 10 depicts 2-dimensional node representations for DeepWalk (on the left) and SchemaWalk (on the right). The embeddings are computed on the DBLP and ACM datasets, and colored according to their ground-truth labels. As shown earlier in section 4.2, `SchemaWalk` surpasses DeepWalk on both datasets. However, DeepWalk still shows competitive performance on ACM. Viewing Figure 10, we observe that `SchemaWalk`'s embeddings yield a more well-defined clustering that aligns with the label set, in comparison with DeepWalk. As both approaches perform comparably on ACM, the clusterings appear similar. These observations support our earlier findings on node classification; Learning embeddings in DBLP appears to benefit more from the heterogeneous knowledge that's provided by `SchemaWalk`. While on the ACM graph, there is less emphasis on such knowledge that sampling in homogeneous way should be sufficient.

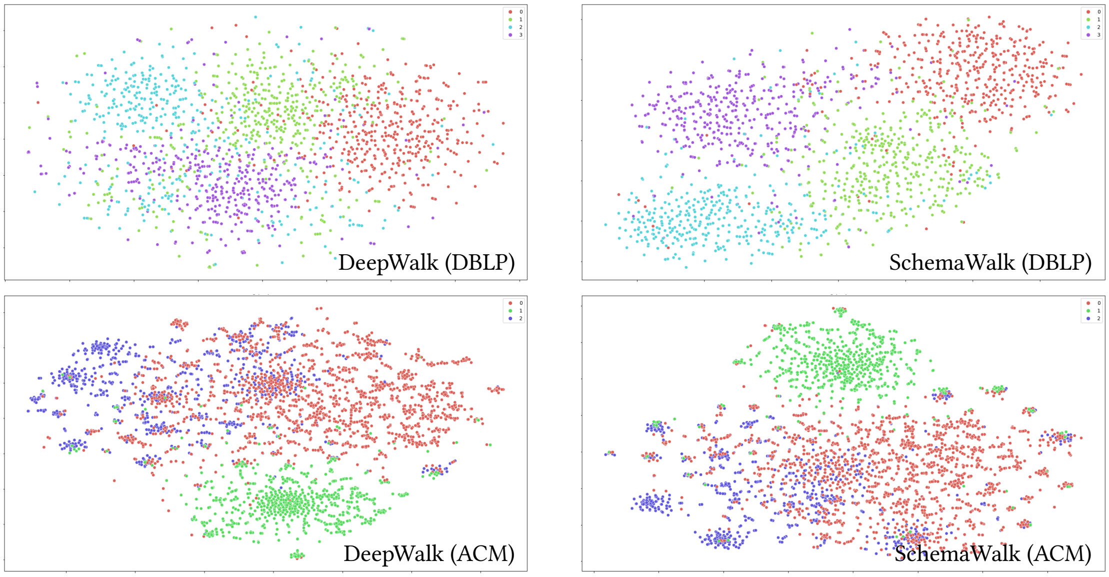


Figure 10: 2D visualization of the node embeddings for DeepWalk (on the left) and SchemaWalk (on the right) where the first and second rows are the DBLP and ACM datasets, respectively.