N-STaRW: Normalized Spatiotemporal Random Walks on Dynamic Graphs

L.F Steve Ataucuri Cruz[1], **Alneu de Andrade Lopes**[1]

[1]Institute of Mathematics and Computer Sciences
University of Sao Paulo, Sao Carlos, Brazil

steveac@usb.br, alneu@icmc.usp.br

Abstract. *Graph representation learning for dynamic graphs has become an essential tool for modeling evolving interactions in real-world systems. Random walk-based methods are widely used to learn node embeddings, but many existing approaches focus primarily on either structural patterns or temporal dynamics without effectively balancing both. As a result, temporal walk strategies often prioritize recent interactions while failing to balance exploration between high- and low-degree nodes. In this work, we propose N-STaRW* [1]*, a spatiotemporal random walk strategy that combines temporal constraints with a degree-normalized structural component. The main idea is to guide the sampling process by preserving temporal ordering while reducing the dominance of high-degree nodes during exploration. We evaluate the method on several real-world datasets for temporal link prediction. The results show that the proposed strategy is competitive with strong baseline methods, particularly in transductive settings. Overall, our findings suggest that balancing temporal and structural information can improve representation quality in dynamic graphs.*

1. Introduction

Learning latent representations of nodes in graphs has been a fundamental problem in Graph Representation Learning (GRL). The main idea behind this process is to encode a node's topological (i.e., spatial/structural) information into a low-dimensional vector representation while preserving important features, such as structural, temporal, or semantic features [Chen et al. 2020]. This low-dimensional representation encodes the topological properties of a node and its neighborhood. Once node representations are learned, they can be used as input features for machine learning algorithms in downstream tasks such as link prediction, node classification, and clustering [Wang et al. 2016].

A common structural issue in graphs is the dominance of high-degree nodes (hubs). These nodes tend to be visited disproportionately often during random walks, which leads to biased training samples. As a result, node co-occurrence pairs appear repeatedly in similar contexts, causing the learned embeddings to reflect hub proximity rather than more balanced structural relationships [Tomčić et al. 2025]. This effect can also distort the representation of low-degree nodes, which are often pulled toward hubs in the embedding space. Moreover, similar behavior is observed in Graph Neural Networks (GNNs), where the repeated aggregation of messages from high-degree nodes can lead to over-smoothing, reducing the discriminative power of node representations [Li et al. 2018], [Oono and Suzuki 2019].

[1]Source code: https://github.com/stonescenter/n-starw

Early approaches to learning low-dimensional embeddings include encoder-decoder frameworks, matrix factorization-based strategies, random path-based approaches, neural network architectures, or combinations of the latter two [Celikkanat 2021]. Traditional graph representation learning methods such as DeepWalk and Node2Vec rely on random walks to explore graph topology and learn node representations, inspired by the Skip-Gram model [Mikolov et al. 2013]. However, these methods assume static graph data, fail to capture the temporal dynamics of interactions, and may perform poorly when processing large graphs.

In this work, we propose a spatiotemporal normalized random walk sampling strategy for representation learning in dynamic graphs. The main idea is to guide the exploration process by considering topological information and temporal properties. By introducing these two components into the transition process, the model can better capture structural patterns and temporal dependencies, enabling richer information extraction from the graph, which can then serve as input to machine learning models.

The objective of this proposal is to investigate whether biased exploration can effectively facilitate knowledge acquisition from dynamic networks, revealing patterns that cannot be discovered using purely structural sampling strategies or only temporal constraints. We primarily address two questions: (1) Do spatiotemporal normalized random walks improve representation quality compared to static and temporal only walks? and (2) How do different random walk biases affect node similarity consistency? Our main contributions can be summarized as follows:

- We propose a novel spatiotemporal random walk strategy that combines temporal decay with degree-normalized structural exploration.
- A bias balancing mechanism with degree normalization and temporal decay.
- We provide an empirical evaluation on seven dynamic graph datasets.
- We show that the proposed sampling strategy is a strong and simple transductive baseline with competitive performance against static and dynamic baselines.

We evaluate our approach on the temporal link prediction task using seven benchmark datasets from different domains. The results show that the proposed random walk is competitive with strong baselines, particularly in a transductive context.

2. Related Work

Graph representation learning has attracted significant attention in recent years. A line of research for learning an embedding relies on random walk-based approaches to explore a graph using hundreds of paths, starting the walks from random nodes. The simplest algorithms use the randomness of the walks or sampling strategies to explore the graph and capture local and global structural information by traversing neighboring vertices. These random walk-based methods typically leverage probabilistic language models, such as bag-of-words and Skip-gram [Mikolov et al. 2013], to learn node embeddings from sequences. Representative methods include DeepWalk [Perozzi et al. 2014] and Node2vec [Grover and Leskovec 2016] for static graphs, as well as SGNE [Du et al. 2018] and DyNode2Vec [Mahdavi et al. 2018] for dynamic settings. However, existing methods tend to emphasize either structural exploration or temporal consistency without balancing both aspects. In particular, they overlook the bias introduced by high-degree nodes, motivating the need for more balanced sampling strategies.

Since the early 2010s, the rapid advancement of deep learning, driven by the success of recurrent neural networks (RNNs) and convolutional neural networks (CNNs), has motivated its extension to graph-structured data. Therefore, a wide range of neural network-based methods for graph representation learning has emerged. These approaches generally employ deep neural architectures to directly model the entire graph structure or its adjacency matrix and map nodes into a low-dimensional embedding space. Notable methods in this category include SDNE [Wang et al. 2016], DNGR [Cao et al. 2016], ChebNet [Defferrard et al. 2016], GCN [Kipf and Welling 2017], and GraphSAGE [Hamilton et al. 2017], attention-based models like GAT [Veličković et al. 2018], which enhance the ability to model complex dependencies in graph-data.

3. Theoretical Background

Dynamic graphs. The defining characteristic of this type of graph is its evolution over time. In contrast to static graphs, dynamic graphs provide a more faithful representation of real-world systems and networks since they capture structural and attribute information that evolves over time. Dynamic graphs can be classified into two types according to their nature. The literature classifies them as Discrete-time dynamic graphs (DTDG), and Continuous-time dynamic graphs (CTDG) [Souza et al. 2022, Rossi et al. 2020].

A CTDG can be modeled as $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T, \mathcal{T})$ if node features are not used or $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T, \mathcal{T}, \mathcal{X})$, where $\mathcal{T}$ is the time domain and $\mathcal{X}$ is the set of node features. We denote $\mathcal{E}_T$ as the edge temporal set, which contains triplets representing interactions between nodes u and v at time t. The set $\mathcal{E}_T = \{(u_1, v_1, t_1), (u_2, v_2, t_2), ..., (u_i, v_i, t_i)\}$ represents an undirected edge between nodes u_i and v_i occurring at timestamp t_i, where $(u_i, v_i) \in \mathcal{V}$. The timestamps are real-valued and strictly ordered, such that $t_i < t_{i+1}$, where $t_i \in \mathbb{R}$.

Random Walk is a mathematical process that describes a trajectory consisting of successive random steps. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and a starting node $u \in \mathcal{V}$, the walk evolves by repeatedly selecting a neighboring node according to a transition probability function. This function can incorporate different bias mechanisms to guide the exploration. At each step, we randomly select a node from the neighbor set $\Gamma(u)$ and move it. Repeating this process generates a sequence of nodes that defines a random walk over the graph [Silva and Zhao 2016]. Formally, a temporal random walk is defined as a sequence of node-time pairs that respects chronological order $W = \{(v_0, t_0), (v_1, t_1), ..., (v_l, t_{\mathcal{L}})\}$. Each $v_i \in \mathcal{V}$ is the node visited at time t_i. Nodes are selected by a sampling strategy, and timestamps can be ordered by $t_l < t_{l+1}$ or $t_l > t_{l+1}$.

Several random walk strategies have been proposed, but not all of them effectively capture the temporal and dynamic nature of graphs. Key aspects such as time-respecting transitions, causality, and node influence are often neglected. We describe the principal sampling strategies considered in this work.

1. **True Self-Avoiding Random Walk (TSAW)** [Kim et al. 2016] – This strategy defines a random walk where the probability of moving to the next node decreases exponentially with its visit frequency. Compared to standard random walk, it discourages revisiting nodes and promotes exploration of less visited areas of the graph by adjusting transition probabilities based on past visits. As a result, the walk explores the graph more diversely, but it does not capture temporal patterns

Equation 1 defines this probability distribution:

$$P(v \mid u) = \frac{e^{-\lambda f(v)}}{\sum_{k \in \Gamma(u)} e^{-\lambda f(k)}} \tag{1}$$

Let $\Gamma(u)$ denote the set of neighbors of node u, $f(k)$ is the number of visits to node k, and λ is the parameter controlling the strength of self-avoidance. Small values of λ promote exploration of new nodes, whereas large values strongly discourage revisiting nodes. When $\lambda = 0$, the walk reduces to a standard unbiased random walk.

2. **Degree-biased Random Walk (DRW)** – A centrality- or degree-biased random walk favors transitions toward more important nodes in the graph. Node importance can be measured using centrality metrics such as degree, closeness, betweenness, and PageRank [Noh and Rieger 2004]. High degree nodes are often referred to as hubs, and the simplest measure of node importance as defined in Equation 2.

$$P(v \mid u) = \frac{d_u}{\sum_{k \in \Gamma(u)} d_k} \tag{2}$$

d_k is the degree of node k. However, this strategy introduces a strong bias towards hubs. As a result, hub nodes tend to dominate the walks and reduce the structural diversity of the sampled paths. Figure 1 (a) illustrates nodes a, b, and v with different degrees. Node v has the highest degree and is therefore more likely to be selected than nodes a and b, receiving more attention during the exploration because it connects to multiple neighbors (e.g., v_1, v_2, v_3).

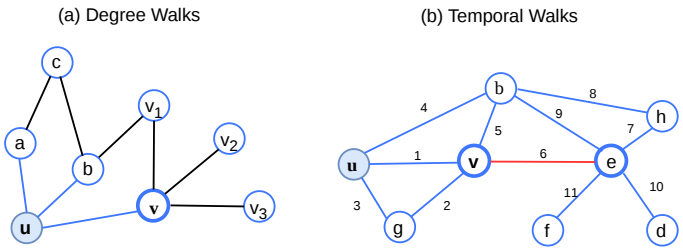


Figure 1. (a) Static graph where nodes v and b have the highest degree. (b) Dynamic graph with time-stamped edges, where nodes v and e may have similar degrees over time, node e becomes more important than node v as time evolves.

4. The proposed Method

Our proposal consists of three main steps: sampling generation, embedding generation, and link prediction.

4.1. Sampling Strategy

Our proposed sampling strategy is based on two key principles for defining transition probabilities in random walks in order to capture both temporal and structural patterns in the graph. First, nodes with higher importance should receive a higher probability of being sampled while avoiding the dominance of high-degree nodes and promoting a diverse exploration of the graph. Second, more recent edges should have a higher probability of being selected than older ones, reflecting the temporal evolution of the network. To incorporate these aspects, we combine structural and temporal patterns into a unified sampling strategy.

Given a non directed graph $\mathcal{G}_T = (\mathcal{V}, \mathcal{E})$, with N nodes $v_i \in \mathcal{V}$. We define an unnormalized Laplacian matrix as $\mathcal{L} = D - A$ over $\mathcal{G}_T$, where D is the degree matrix and A is the adjacency matrix. Let $d_u = \sum_v^N A_{uv}$ as degree; a simple random walk has a transition matrix $P = D^{-1}A$ that moves from u to v with probability $P_{uv} = \frac{A_{uv}}{d_u}$

$$A(u,v) = \begin{cases} d_u, & \text{if } u = v, \\ -1, & \text{if } (u,v) \in \mathcal{E}, \\ 0, & \text{otherwise} \end{cases} \quad \mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v \text{ and } d_v \neq 0, \\ -\dfrac{1}{\sqrt{d_u d_v}}, & \text{if } (u,v) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases}$$

We define the symmetric Laplacian and rewrite it in Equation 3:

$$\mathcal{L}_{sym} = D^{-1/2}\mathcal{L}D^{-1/2} = I - D^{-1/2}AD^{-1/2} \tag{3}$$

The term $D^{-1/2}$ is the element-wise inverse square root of D, I is the identity matrix. Our structural component is defined as follows in Equation. 4.

$$P_S(v \mid u) = \frac{(1/\sqrt{d_u d_v})^\beta}{\sum_{k \in \Gamma(u)} (1/\sqrt{d_u d_k})^\beta} \tag{4}$$

The term $1/\sqrt{d_u d_v}$ is the symmetric degree normalization applied to edges (u, v); this reduces the dominance of high-degree nodes in the graph. The parameter $\{0 < \beta \leq 1\}$ controls the power of the symmetric normalization.

The temporal component is controlled by Equation 5. Previous sampling strategies are characterized by their lack of awareness of edge temporality, generating random sequences in static graphs. The general idea is to select the next node according to temporal order, either forward or backward in time. A common formulation defines the transition probability using exponential decay over the elapsed time difference $\Delta t_{uv} = (t_{uv} - t)$, where t is the current time and t_{uv} is the timestamp of the candidate edge with $t_{uv} > t$ [Sheng et al. 2025]:

$$P_T(v \mid u, t) = \frac{e^{-\lambda(t_{uv} - t)}}{\sum_{k \in \Gamma(u)} e^{-\lambda(t_{uk} - t)}} \tag{5}$$

The term $e^{-\lambda \Delta t}$ assigns larger probabilities to neighbors whose timestamps are closer to the current time, and the parameter λ controls the decay rate of this recency bias: small values produce a flatter distribution, whereas larger values assign higher probabilities to more recent interactions. For example, as illustrated in Fig. 1(b), a time-respecting walk $u \rightarrow v \rightarrow g \rightarrow u \rightarrow b \rightarrow v \rightarrow e \rightarrow d$ follows the restriction $t_{i+1} > t$, ensuring chronological order and time dependency. We observe that nodes v and e initially exhibit high importance; however, node e becomes more relevant than node v at later time steps. This observation is very important to consider it in our modeling.

In summary, we combined two sampling strategies to capture both structural and temporal patterns. Given a node u, the transition probability of selecting node v is defined as the average of the degree-normalized structural term (Eq. 4) and the temporal decay term (Eq. 5), as shown in Equation 6:

$$P(v \mid u, t) = \text{AVG}(P_S(v \mid u) + P_T(v \mid u, t)) \tag{6}$$

4.2. Node Embedding

The main objective of this step is to transform each node into a feature vector, preserving neighborhood information and node similarity in the embedding space. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the function $\mathbf{f} : v_i \rightarrow \mathbb{R}^d$ maps each node $v_i \in \mathcal{V}$ to a d-dimensional vector representation $\mathbf{z_v} \in \mathbb{R}^d$, where $d < |\mathcal{V}|$. After generating temporal walk sequences in the previous step, we learn time-respecting embeddings for CTDGs by generalizing the *Skip-Gram* architecture. This algorithm maximizes the average logarithmic probability of all pairs of nodes in a context window for node v_i, $\Gamma(v_i) = \{v_{i-\omega}, ..., v_{i-1}, v_{i+1}, ..., v_{i+\omega}\}$, as follows in Equation 7:

$$\max \sum_{i=1}^{|L|} \sum_{u \in \Gamma(v_i)} \log P(u \mid v_i) \tag{7}$$

The ω is the size of the context; $|L| = |\Gamma(v_i)|$ is the length of the sequence.

4.3. Link Prediction

Link prediction aims to identify potential or missing connections between pairs of nodes [Lü and Zhou 2011]. In this context, the task is defined on a fixed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where all nodes are observed during training, and the goal is to predict whether an edge will exist between two nodes at a later time. This is done by distinguishing observed (positive) edges from non-existent (negative) ones.

We follow the methodology proposed in [Grover and Leskovec 2016]. We first learn node embeddings $\mathbf{z}_u = \mathbf{f}(u)$ and $\mathbf{z}_v = \mathbf{f}(v) \in \mathbb{R}^d$, and then combine them to form an edge representation $e_{uv} = g(u, v)$. We use the Weighted-L2 operator, defined as $e_{uv} = |\mathbf{z}_u - \mathbf{z}_v|^2$, which measures the squared distance between node embeddings and is used as input to the classifier. We choose this operator because it achieved the best experimental results.

5. Experiment Setting

- **Dataset**. We conducted experiments on seven real-world dynamic datasets, covering diverse domains, including social networks, transportation, email communication, student forums, and interaction networks. All datasets contain only the edge

creation timestamps and do not include any information about node attributes. Summary statistics for all datasets are reported in Table 1.

Table 1. Dataset Statistics. [Rossi and Ahmed 2015]

Datasets	Domain	Nodes	Edges	Avg. Degree	Max. Degree	Granularity
Wikipedia	Social	9227	157474	34	1937	1 M
Reddit	Social	10984	672447	122	58727	1 M
Enron	Social	184	125235	1361	14573	3 Y
UCI	Social	1899	59834	63	1546	196 D
MOOC	Interac.	7144	411749	115	19474	17 M
ia-enron	Social	151	50572	670	4815	1137 D
ia-contact	Comm.	274	28244	206	2092	3.97 D
fb-forum	Comm.	899	33719	75	1841	164 D

We used these datasets because they are widely adopted benchmarks for dynamic graphs and contain explicitly recorded interaction timestamps.

- **Baseline**. We compared our proposed method with the most representative state-of-the-art methods for both static and dynamic graphs. Our strong baselines for static graphs are DeepWalk [Perozzi et al. 2014], Node2Vec [Grover and Leskovec 2016], GCN [Kipf and Welling 2017], GraphSAGE [Hamilton et al. 2017], and GAT [Veličković et al. 2018] while the algorithms for dynamic graphs are CTDNE [Nguyen et al. 2018], TGAT [Xu et al. 2020], and TGN [Rossi et al. 2020].
- **Evaluation Tasks**. We evaluated temporal link prediction, which estimates the probability that a future edge exists between a pair of nodes. We consider two evaluation settings in our experiments: *transductive* and *inductive*.
- **Data Processing and Sampling**. Data preparation follows three steps. First, for transductive link prediction, edges are sorted by timestamp to preserve chronological order. Then, 70% of the edges are used to learn node embeddings, and the remaining 30% are treated as positive edges, split again into training 70%, testing 20%, and validation 10% sets. Finally, to train the classifier, we created a negative set. These samples are created by keeping the source nodes and timestamps of positive edges while randomly choosing destination nodes outside the positive set. We reported the best sampling configuration in our experiments, we used a maximum walk length of $L = 60$, a minimum length of $\{5, 10\}$, 32 walks per node, embedding size of 128, $\beta = 0.80$, and $\lambda = 0.85$.
- **Evaluation Protocol/training details**. We evaluate both transductive and inductive settings using AUC-ROC and Average Precision (AP). Our pipeline first generates walks, then learns low-dimensional node representations, and finally uses these learned embeddings as input for classification. We use logistic regression (LR) as a classifier for transductive models. LR allows us to evaluate the quality of the embeddings independently of more complex classifiers, making it easier to attribute performance gains to our proposal.

5.1. Results and Discussion

Table 2 reports the AUC-ROC performance of different methods in transductive and inductive settings. For each method, we present the score corresponding to the best valida-

tion set. In the transductive results, N-STaRW achieved the highest average performance among temporal methods, outperforming its closest competitor, CTDNE by 13.58% on average across all datasets. It also achieved competitive average gains of 5.62%, 7.66%, and 5.25% over TSAW, STaRW(without degree normalization), and Node2Vec, respectively. These improvements correspond to the mean relative AUC-ROC gain computed across all datasets [2]

However, N-STaRW's performance is closer to that of DeepWalk, which performs particularly well on the Wikipedia and Reddit datasets. This suggests that, in some transductive cases, structural patterns may dominate, while temporal information can be redundant or even introduce noise. Since DeepWalk relies on a uniform random walk, its unrestricted exploration may better capture connectivity patterns when temporal and hub-aware constraints are less beneficial.

Our proposal achieves a competitive average gain of 6.87% in all transductive methods and datasets[3]. For example, N-STaRW achieved the best or second-best results on Wikipedia, Enron, UCI, and MOOC, with AUC scores of 91%, 95.05%, 77.26%, and 96.76%, respectively. These results indicate that combining normalized structural bias with temporal information can effectively capture important connectivity patterns. Meanwhile, the STaRW variant ranks first on Enron (95.30%) and UCI (79.62%), suggesting that using structural and temporal information alone can already be highly effective, even without degree normalization in some datasets. The main difference between STaRW and N-STaRW is that STaRW uses Equation 2, whereas N-STaRW applies Equation 4.

Table 2. AUC-ROC scores for temporal edge prediction in transductive and inductive settings. The best, second-best, and third-best results in each column are highlighted in bold, underline, and *italic*, respectively. Static methods*. The % Gain column reports the mean relative AUC-ROC improvement of our method over each baseline across all datasets.

Task	Method	Wikipedia	Enron	UCI	MOOC	Reddit	% Gain
Transductive							
	DeepWalk*	84.93	*93.82*	*77.11*	93.21	**96.53**	1.95%
	CTDNE	65.97	89.36	69.15	86.23	94.26	13.58%
	TSAW	77.62	91.34	73.52	95.57	93.48	5.62%
	N2V$^*_{p=1.5,q=0.5}$	80.38	93.38	72.55	*95.32*	90.98	5.25%
	STaRW	*78.72*	**95.30**	**79.62**	84.43	84.35	7.66%
	N-STaRW	**91.00**	95.05	77.26	**96.76**	*94.19*	-
Inductive							
	GCN*	*71.06*	63.58	81.79	**99.50**	*89.79*	14.83%
	GraphSAGE*	61.10	57.80	79.60	74.40	*78.40*	32.13%
	GAT*	*78.70*	53.50	**84.20**	81.00	54.00	35.79%
	TGN	**97.81**	**68.47**	78.59	89.04	**97.55**	7.08%
	TGAT	93.99	63.97	79.54	85.50	96.62	10.64%

In the inductive setting, N-STaRW shows clear gains over static GNN baselines such as GCN, GraphSAGE, and GAT, indicating that including temporal information into

[2]Gain $= \frac{1}{D}\sum_{i=1}^{D}\left(\frac{AUC_{\text{our}}-AUC_i}{AUC_i}\right)\times 100$, where D is the number of datasets.

[3]We applied the average of all gains for transductive methods.

the representation-learning process can be advantageous over purely structural neighborhood aggregation in dynamic graphs. However, these baselines do not explicitly model temporal dependencies and therefore serve mainly as reference baselines. A more direct comparison is with temporal models such as TGN and TGAT, against which N-STaRW remains competitive. Overall, these results suggest that carefully constructed transductive embeddings can still generalize well in inductive scenarios.

Compared to dynamic methods, N-STaRW achieves moderate performance, comparable to TGN and TGAT, with an average gain of approximately 8.86%[4]. Generally, TGN and TGAT perform very well in inductive scenarios due to their ability to model temporal dependencies via learnable neighborhood aggregation functions, achieving good results on Wikipedia, UCI, and Reddit. Overall, the results indicate that N-STaRW remains competitive with dynamic inductive methods. These findings suggest that carefully balanced transductive embeddings can still provide strong performance in inductive scenarios.

5.2. Ablation Study

We conduct an ablation study on two datasets to assess the contribution of each N-STaRW component, as shown in Table 3. We executed 5 runs with different seeds in this experiment. Our $t_c = 2.776$, $\alpha = 0.05$ and $p < \alpha$.

Table 3. Ablation study of N-STaRW components. We report transductive link prediction metrics in percentage. The best, second-best are highlighted in bold, and underline.

N.	**Dataset**	**ia-enron**		**ia-contact**	
	Metric	AUC-ROC	AP	AUC-ROC	AP
	N-STaRW(full)	$\mathbf{96.64 \pm 0.10}$	$\mathbf{96.15 \pm 0.07}$	$\underline{96.99} \pm \underline{0.29}$	95.09 ± 0.67
1	w/o struct.	92.91 ± 0.32	92.63 ± 0.56	96.62 ± 0.11	95.07 ± 0.33
2	w/o temp.	95.79 ± 0.13	94.87 ± 0.30	96.48 ± 0.19	95.11 ± 0.41
3	w struct. norm.	96.34 ± 0.16	96.07 ± 0.30	96.10 ± 0.19	90.58 ± 0.86
4	w struct.+temp.	96.19 ± 0.17	95.98 ± 0.24	$\mathbf{97.20 \pm 0.23}$	$\mathbf{96.33 \pm 0.36}$

Ablations 1 and 2 remove the structural and temporal components defined in Eqs. 2 and 5, respectively. Removing the structural component causes the largest performance drop, especially on the *ia-enron* dataset, where AUC-ROC falls from 96.64 to 92.91 and AP from 96.15 to 92.63 ($p < 0.05$), highlighting the importance of structural information in datasets with strong topological patterns. In contrast, removing the temporal component with ablation 2 leads to a significant decrease(both $p < 0.05$), suggesting that temporal and structural components contribute to performance on *ia-enron*. When only structural normalization is used with ablation 3 (Eq. 4), the performance remains close to the full model on *ia-enron* (AUC-ROC 96.34 vs. 96.64), suggesting that normalization is particularly beneficial in graphs with stronger hub-dominance patterns (e.g., in this case, *ia-enron* has an average node degree 670, which is much higher than in *ia-contact* with 206 on average degree).

[4]We applied the average of all gains for inductive dynamic methods.

Additionally, when combining structural and temporal components without normalization in Ablation 4 results in a slight decrease on *ia-enron*, approximately -0.45 AUC when compared with the full version 96.64, but in the case of *ia-contact*, the best performance on AUC-ROC is 97.20 compared to 96.99 for the full model. This indicates that normalization may not always be beneficial in certain datasets. This behavior is reflected in AP. On *ia-contact*, AP decreases substantially from 95.09 in the full model to 90.58 when structural normalization is used alone. At the same time, the best AP 96.33 is obtained when structural and temporal components are combined without normalization. Overall, this suggests that the effectiveness of normalization depends on graph topology, hub concentration, and the temporal distribution of interactions.

5.3. Hyperparameter Sensitivity

We analyze the sensitivity of N-STaRW on three representative datasets by varying λ, β, walk length, and the number of walks per node. Figure 2(a) shows that larger values of λ generally improve performance on *ia-enron*, *ia-contact*, and *fb-forum*, with values around 0.85 yielding the best overall results. This suggests that a stronger recency bias is beneficial in these datasets. We also study the structural intensity controlled by β; values around $0.5-0.85$ consistently provide strong performance across datasets. Extremely low or extremely high values do not produce substantial improvements; therefore, the model is not highly sensitive to β, indicating robustness. Finally, Figures 2(c) and (d) indicate that 32 walks per node are generally sufficient, while walk lengths between 20 and 60 are adequate for denser graphs, and up to 100 steps may be needed for sparser ones.

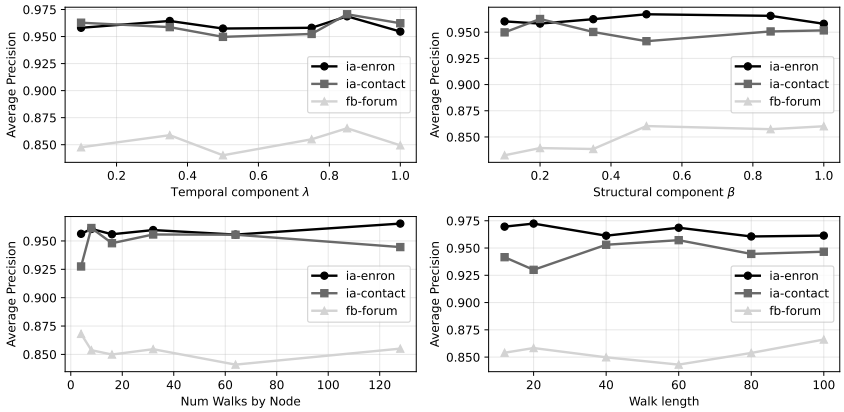


Figure 2. Experimental study of the setting of N-STaRW.

6. Conclusions and Future Work

In this paper, we addressed the problem of learning node representations in dynamic graphs by focusing on the limitations of existing random walk-based approaches, which often fail to balance structural and temporal information. We proposed a normalized spatiotemporal random walk strategy for learning node representation. First, it combines temporal constraints with a degree-normalized structural component to reduce the dominance of high-degree nodes while preserving time-aware exploration. Second, we presented experimental results on multiple real-world datasets that show our proposal is simple and competitive with both static and dynamic methods, particularly in the transductive

context with 6.87% relative mean AUC. Finally, structural normalization should be interpreted as a dataset-dependent bias control mechanism rather than a universally beneficial component. Our sampling strategy downweights high-degree neighbors and favors low-degree neighbors compared with other biased random walks. As future work, we plan to investigate how to learn temporal and structural parameters from data using a learning mechanism to avoid the need for dataset-specific parameters and heuristics because datasets have different characteristics, especially with respect to temporal sparsity.

7. Acknowledgments

This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil [grant number: $88887.822245/2023\text{-}00$, $88887.239502/2025\text{-}00$].

References

Cao, S., Lu, W., and Xu, Q. (2016). Deep neural networks for learning graph representations. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1145–1151.

Celikkanat, A. (2021). *Graph Representation Learning with Random Walk Diffusions*. PhD thesis. Thèse de doctorat dirigée par Paragios, Nikos et Malliaros, Fragkiskos Informatique université Paris-Saclay 2021.

Chen, F., Wang, Y.-C., Wang, B., and Kuo, C.-C. J. (2020). Graph representation learning: a survey. *Transactions on Signal and Information Processing (APSIPA)*.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3844–3852.

Du, L., Wang, Y., Song, G., Lu, Z., and Wang, J. (2018). Dynamic network embedding : An extended approach for skip-gram based network embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. KDD '16, page 855–864. Association for Computing Machinery.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS.

Kim, Y., Park, S., and Yook, S.-H. (2016). Network exploration using true self-avoiding walks. *Phys. Rev. E*, 94:042309.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Lü, L. and Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170.

Mahdavi, S., Khoshraftar, S., and An, A. (2018). dynnode2vec: Scalable dynamic network embedding. In *2018 IEEE international conference on big data (Big Data)*, pages 3762–3765. IEEE.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Nguyen, G. H., Lee, J. B., Rossi, R. A., Ahmed, N. K., Koh, E., and Kim, S. (2018). Continuous-time dynamic network embeddings. International World Wide Web Conferences Steering Committee.

Noh, J. D. and Rieger, H. (2004). Random walks on complex networks. *Physical Review Letters*, 92(11):118701.

Oono, K. and Suzuki, T. (2019). Graph neural networks exponentially lose expressive power for node classification. In *8th International Conference on Learning Representations (ICLR)*.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery & data mining*, KDD '14. ACM.

Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., and Bronstein, M. (2020). Temporal graph networks for deep learning on dynamic graphs.

Rossi, R. and Ahmed, N. (2015). The network data repository with interactive graph analytics and visualization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29.

Sheng, J., Zhang, Y., and Wang, B. (2025). Continuous-time dynamic graph learning based on spatio-temporal random walks. *The Journal of Supercomputing*, 81(2):389.

Silva, T. and Zhao, L. (2016). *Machine Learning in Complex Networks*. Springer International Publishing.

Souza, A. H., Mesquita, D., Kaski, S., and Garg, V. (2022). Provably expressive temporal graph networks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Tomčić, A., Savić, M., Simić, D., and Radovanović, M. (2025). Dynamic graph embedding through hub-aware random walks. In *International Conference on Similarity Search and Applications*, pages 315–329. Springer.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Wang, D., Cui, P., and Zhu, W. (2016). Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1225–1234.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., and Achan, K. (2020). Inductive representation learning on temporal graphs. In *International Conference on Learning Representations (ICLR)*.