IRWE: Inductive Random Walk for Joint Inference of Identity and Position Network Embedding

Anonymous authors
Paper under double-blind review

Abstract

Network embedding, which maps graphs to distributed representations, is a unified framework for various graph inference tasks. According to the topology properties (e.g., structural roles and community memberships of nodes) to be preserved, it can be categorized into the identity and position embedding. However, existing methods can only capture one type of property. Some approaches can support the inductive inference that generalizes the embedding model to new nodes or graphs but relies on the availability of attributes. Due to the complicated correlations between topology and attributes, it is unclear for some inductive methods which type of property they can capture. In this study, we explore a unified framework for the joint inductive inference of identity and position embeddings without attributes. An inductive random walk embedding (IRWE) method is proposed, which combines multiple attention units to handle the random walk on graph topology and simultaneously derives identity and position embeddings that are jointly optimized. In particular, we demonstrate that some random walk statistics can be informative features to characterize node identities and positions while supporting the inductive embedding inference. Experiments validate the superior performance of IRWE beyond various baselines for the transductive and inductive inference of identity and position embeddings.

1 Introduction

For various state-of-the-art graph inference techniques, network embedding (a.k.a. graph representation learning) is a commonly used framework. It maps each node of a graph to a low-dimensional vector representation (a.k.a. embedding) with some key properties preserved. The derived representations are further used to support several downstream graph inference tasks (e.g., node classification (Kipf & Welling, 2017; Veličković et al., 2018), node clustering (Ye et al., 2022; Qin et al., 2023a; Gao et al., 2023), and link prediction (Lei et al., 2018; 2019; Qin et al., 2023b; Qin & Yeung, 2023)).

According to the topology properties to be preserved, existing network embedding techniques can be categorized into the identity and position embedding (Rossi et al., 2020; Zhu et al., 2021). The identity embedding (a.k.a. structural embedding) aims to preserve the structural role that each node plays in graph topology, which is also defined as the node identity. In contrast, the position embedding (a.k.a. proximity-preserving embedding) captures the linkage similarity between nodes (e.g., community structures (Newman, 2006)), which is also defined as node position or proximity. In Fig. 1 (a), each color denotes a unique structural role. For instance, red and yellow may indicate the opinion leader and hole spanner in a social network (Yang et al., 2015). Moreover, there are two communities denoted by the two dotted circles in Fig. 1, where nodes in the same community have dense linkages and thus are more likely to have similar positions.

The identity and position embedding should respectively force nodes with similar identities (e.g., $\{v_1, v_8\}$) and positions (e.g., $\{v_1, v_2, v_6\}$) to have close low-dimensional representations. As a demonstration, we applied *struc2vec* (Ribeiro et al., 2017) and *node2vec* (Grover & Leskovec, 2016) (with embedding dimensionality $d = 2$), which are typical identity and position embedding methods, to the example graph in Fig. 1 (a) and vislized the derived embeddings. Note that two nodes may have the same identity even though they are far away from each other. In contrast, nodes with similar positions must be close to each other with dense

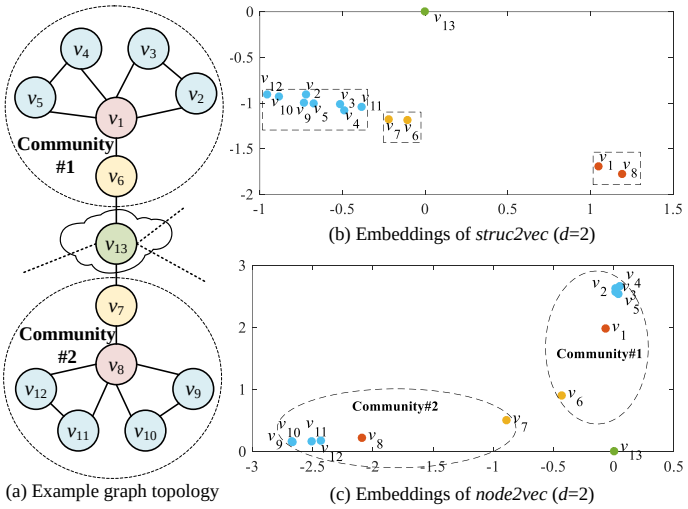


Figure 1: An example of identity and position embedding in terms of (b) *struc2vec* and (c) *node2vec*, respectively. In the (a) example graph, each color denotes a unique identity while nodes in the same community have similar positions.

linkage and short distances. Due to the contradiction, *it is challenging to simultaneously capture the two types of properties in a common embedding space*. For instance, v_1 and v_8 with the same identity have close identity embeddings in Fig. 1 (b). However, their position embeddings are far away from each other in Fig. 1 (c). Since the two types of embeddings may be appropriate for different downstream tasks (e.g., structural role classification and community detection), we expect a unified embedding model.

Most conventional embedding methods (Wu et al., 2020; Grover & Leskovec, 2016; Ribeiro et al., 2017; Donnat et al., 2018) follow the embedding lookup scheme and can only support the transductive embedding inference. In this scheme, the low-dimensional node representations are model parameters optimized only for the currently observed graph topology. When applying the model to new unseen nodes or graphs, one needs to optimize model parameters from scratch with high complexities. Compared with these transductive methods, some state-of-the-art embedding techniques (Hamilton et al., 2017; Velickovic et al., 2019) can support the advanced inductive inference, which aims to directly generalize the embedding model trained on observed topology to new unseen nodes or graphs without additional optimization.

Most existing inductive approaches (e.g., those based on graph neural networks (GNNs) (Wu et al., 2020)) rely on the availability of node attributes and derive inductive embeddings via attribute aggregation. However, previous studies (Qin et al., 2018; Li et al., 2019; Wang et al., 2020; Qin & Lei, 2021) have demonstrated some complicated correlations between graph topology and attributes. For instance, attributes may provide (i) complementary characteristics orthogonal to topology for better performance of downstream tasks or (ii) inconsistent noise resulting in unexpected quality degradation. Therefore, it is unclear for most inductive methods that their performance improvement is brought about by the incorporation of attributes or better exploration of topology. When attributes are unavailable, most inductive approaches need additional procedures to extract auxiliary attribute inputs from topology (e.g., one-hot encodings of node degrees). Our experiments demonstrate that some inductive baselines with naive auxiliary attribute extraction strategies may even fail to outperform conventional transductive methods on the inference of identity and position embeddings. Moreover, it is also hard to determine which type of topology properties (i.e., node identities or positions) that some inductive approaches can capture.

In this study, we focus on the unsupervised network embedding and explore the possibility of a unified framework for the joint inductive inference of identity and position embeddings. To clearly distinguish between the two types of embeddings, we consider the case where topology is the only available information source. This eliminates the unclear influence from graph attributes due to the complicated correlations between the two sources. Different from most existing inductive approaches relying on the availability of

node attributes, we propose an inductive random walk embedding (IRWE) method. It combines multiple attention units with different choices of key, query, and value to handle the random walk (RW) and induced statistics on graph topology.

RW is an effective technique to explore topology properties for network embedding. However, most RW-based methods (Grover & Leskovec, 2016; Ribeiro et al., 2017) follow the transductive embedding lookup scheme, failing to support the advanced inductive inference. In this study, we demonstrate that anonymous walk (AW) (Ivanov & Burnaev, 2018), the anonymization of RW, and its induced statistics can be informative features shared by all possible nodes and graphs and thus have the potential to support inductive inference.

Although the identity and position embedding encodes properties contradictory with one another, there remains a relation between node identities and positions that *nodes with different identities should have different contributions in forming the local community structures*. For the example in Fig. 1, v_1 and v_2 may correspond to an opinion leader and ordinary audience of a social network, where v_1 is expected to contribute more in forming community #1 than v_2. By incorporating this relation, IRWE simultaneously derives and jointly optimizes two sets of embeddings w.r.t. node identities and positions. In particular, we demonstrate that some AW statistics can characterize node identities to derive identity embeddings, which can be further used to generate position embeddings. It is also expected that the joint optimization of the two sets of embeddings can improve the quality of one another.

Our major contributions are summarized as follows.

- In contrast to most existing inductive embedding methods relying on the availability of node attributes, we propose an alternative IRWE approach, whose inductiveness is only supported by the RW on graph topology.
- To the best of our knowledge, we are the first to explore a unified framework for the joint inductive inference of identity and position embeddings using RW, AW, and induced statistics.
- Experiments on public datasets validate the superiority of IRWE beyond various baselines for the transductive and inductive inference of identity and position embeddings.

In the rest of this paper, we review some representative related work in Section 2. The problem statements and preliminaries of this study are given in Section 3. We further elaborate on the proposed IRWE method in Section 4, including the model architecture as well as the algorithms of optimization and inference. Experiment settings and results are described and analyzed in Section 5. Finally, Section 6 concludes this paper and indicates possible future directions.

2 Related Work

In the past several years, a series of network embedding techniques have been proposed. Rossi et al. (2020) gave a comprehensive overview of existing methods covering the identity and position embedding.

2.1 Identity & Position Embedding

To the best of our knowledge, existing embedding approaches can only capture one type of topology properties (i.e., node identities or positions). They cannot encode both types of properties in a unified framework and support the joint inference of the two types of embeddings.

Perozzi et al. (2014) proposed *DeepWalk* that applies skip-gram to learn node embeddings from RWs on graph topology. The ability of *DeepWalk* to capture node positions is further discussed and validated in (Pei et al., 2020; Rossi et al., 2020). Grover & Leskovec (2016) modified the RW in *DeepWalk* to a biased form and introduced *node2vec* that can derive richer position embeddings by adjusting the trade-off between breadth- and depth-first sampling. From a probabilistic view, Cao et al. (2015) reformulated the RW in *DeepWalk* to matrix factorization objectives w.r.t. all steps in RW. Moreover, Wang et al. (2017), Ye et al. (2022), and Chen et al. (2023) introduced community-preserving embedding methods based on nonnegative matrix factorization, hyperbolic embedding, and graph contrastive learning, respectively.

Ribeiro et al. (2017) proposed *struc2vec*, a typical identity embedding method, by applying RW to an auxiliary multilayer graph constructed via a hierarchical similarity measurement based on node degrees. From a graph signal processing view, Donnat et al. (2018) used spectral graph wavelets to develop *GraphWave* and theoretically proved its ability to capture node identities. To tackle the uncertainty of node embeddings, Pei et al. (2020) introduced *struc2gauss*, which encodes node identities in a space formulated by Gaussian distributions, and analyzed the effectiveness of different energy functions and structural similarity measures for Gaussian embeddings. Moreover, Guo et al. (2020) enhanced the capabilities of GNNs to preserve identity information by reconstructing several manually-designed statistics. Chen et al. (2022) enabled the graph transformer to capture node identities by incorporating the rooted subgraph of each node.

Some recent research discussed and analyzed the relation between the two types of embeddings. Zhu et al. (2021) proposed the *PhUSION* framework with three steps (i.e., computation of node proximities, non-linear filtering, and dimension reduction) and demonstrated which components (e.g., proximity measurements) can be used for the identity or position embedding. Although *PhUSION* can reveal the similarity and difference between the two types of embeddings, it can only derive one type of embedding under each unique setting. Rossi et al. (2020) proved that some typical techniques (e.g., RW and attribute aggregation) adopted by existing methods can only derive either identity or position embeddings, which validates our discussions in Section 1. Based on the invariant theory, Srinivasan & Ribeiro (2020) proved that the relation between identity and position embedding can be analogous to that of a probability distribution and its samples. They also demonstrated that transductive and inductive embedding learning is unrelated to which type of properties to be preserved. Although some conclusions in this research are consistent with our motivations regarding node identities and positions, it only considers the optimization of one type of embedding and the transformation to another type. Its inductive inference still relies on the availability of node attributes. In contrast, we focus on the joint optimization and inductive inference of the two types of embeddings without considering the effects of graph attributes.

2.2 Inductive Network Embedding

Most conventional embedding methods (Grover & Leskovec, 2016; Ribeiro et al., 2017; Donnat et al., 2018) are transductive. They optimize their model parameters for each single graph and can only support the downstream tasks on such a graph. Some state-of-the-art studies explore the inductive inference that directly derives embeddings for new unseen nodes or graphs by generalizing the model parameters optimized on known topology, without additional optimization.

Hamilton et al. (2017) introduced *GraphSAGE*, an inductive GNN framework, including the neighbor sampling and feature aggregation with different choices of aggregation functions (e.g., mean, max-pooling, and LSTM). *GAT* (Veličković et al., 2018) leverages self-attention into the feature aggregation of GNN, which automatically determines the aggregation weights for the neighbors of each node. Focusing on the unsupervised network embedding, Velickovic et al. (2019) proposed *DGI* that maximizes the mutual information between patch embeddings and high-level graph summaries derived from a specified GNN variant (e.g., GCN (Kipf & Welling, 2017)). Without using the feature aggregation of GNN, Nguyen et al. (2021) developed *SANNE* that applies self-attention to handle RWs sampled from graph topology. However, the inductiveness of the aforementioned methods relies on the availability of node attributes. It is unclear for some approaches which type of properties (i.e., node identities or positions) they can capture, due to the complicated correlations between graph topology and attributes as discussed in Section 1. When node attributes are unavailable, these inductive methods have to use additional procedures to extract auxiliary attribute inputs (e.g., one-hot encodings of node degrees).

Some recent research analyzed the ability of several new GNN structures to capture node identities or positions in specific cases regarding node attributes (e.g., all the nodes have the same scalar attribute input (Xu et al., 2019)). Wu et al. (2019) and You et al. (2021) proposed *DEMO-Net* and *ID-GNN* that can capture node identities using the degree-specific multi-task graph convolution and heterogeneous message passing on the rooted subgraph of each node, respectively. Jin et al. (2020) leveraged AW statistics into the feature aggregation to enhance the ability of GNN to preserve node identity information. Moreover, *P-GNN* (You et al., 2019) can derive position-aware embeddings based on a distance-weighted aggregation

Table 1: Definitions of Some Major Notations in This Study

Notations	Definitions
w, ω	an RW & an AW
$\psi(v)$	identity embedding of node v
$\gamma(v)$, $\bar{\gamma}(v)$	position & (auxiliary) context embeddings of node v
$\mathcal{W}^{(v)}$	set of sampled RWs starting from node v
Ω_l, $\tilde{\Omega}_l$	AW lookup table w.r.t. length l & its reduced version
η_l, $\tilde{\eta}_l$	number of AWs w.r.t. length l & its reduced value
$\rho(\omega)$	one-hot encoding of AW ω
$\varphi(\omega)$	AW embedding of AW ω
$\mathcal{G}_s(v, r)$	rooted subgraph of node v with distance r
$q(v, l)$	distribution of AWs w.r.t. RWs from node v with length l
$s(v)$, $\tilde{s}(v)$	AW statistics of node v & its reduced version
$\rho_d(v)$	bucket one-hot encoding w.r.t. the degree of node v
$\delta(v)$	high-order degree features of node v
$\mathcal{W}_I^{(v)}$	set of RWs used to infer the position embedding of node v
$r(v)$	RW statistics of node v
$\pi_l(j)$, $\pi_g(v)$	local & global position encodings of index j & node v
$\mathbf{C}$	contrastive statistics to optimize position embeddings $\{\gamma(v)\}$

scheme over the sets of sampled anchor nodes. However, these GNN structures can only capture either node identities or positions.

In contrast to all the aforementioned inductive methods, we explore a unified inductive framework for the joint inference of identity and position embeddings without relying on the availability and aggregation of graph attributes.

3 Problem Statements & Preliminaries

In this study, we consider the unsupervised network embedding. Table 1 summarizes some major notations used in this paper. In general, a graph can be represented as a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and $\mathcal{E} = \{(v_i, v_j)|v_i, v_j \in \mathcal{V}\}$ as the sets of nodes and edges, respectively. We also assume that graph topology is the only available information source and additional attributes are unavailable.

Definition 1 (*Network Embedding*). Given a graph $\mathcal{G}$, network embedding (a.k.a. graph representation learning) learns a function $f : \mathcal{V} \mapsto \mathbb{R}^d$ that maps each node v to a low-dimensional vector $f(v)$ (a.k.a. embedding), with some key properties of $\mathcal{G}$ preserved. In this study, we assume that f is learned via an unsupervised objective. The learned embeddings are adopted as the inputs of some downstream modules (e.g., logistic regression and KMeans) to support concrete inference tasks (e.g., node classification and clustering).

According to the topology properties to be preserved, network embedding can be categorized into the identity and position embedding.

Definition 2 (*Identity Embedding*). The identity embedding (a.k.a. structural or role-based embedding (Rossi et al., 2020; Zhu et al., 2021)), denoted as $f(v) := \psi(v)$, encodes the structural role (i.e., identity) that a node v plays in the graph topology. Namely, nodes with similar structural roles (e.g., nodes with the same color in Fig. 1) should have similar representations.

Definition 3 (*Position Embedding*). The position embedding (a.k.a. proximity-preserving embedding (Rossi et al., 2020; Zhu et al., 2021)), denoted as $f(v) := \gamma(v)$, encodes the high-order linkage similarities (i.e., proximity or position) between nodes. Namely, nodes with dense linkage (e.g., nodes within a community in Fig. 1) should have close representations.

The embedding inference includes the transductive and inductive settings. A transductive method focuses on the optimization of f on the currently observed topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and can only support inference tasks on $\mathcal{V}$. In contrast, model parameters of f for an inductive approach, which are first optimized on $(\mathcal{V}, \mathcal{E})$, can be generalized to new unseen nodes $\mathcal{V}'$ or a new graph $\mathcal{G}'' = (\mathcal{V}'', \mathcal{E}'')$. Hence, it can support inference tasks on $\mathcal{V}'$ or $\mathcal{V}''$ (i.e., the inductive inference for new nodes or across graphs) with model parameters shared by all

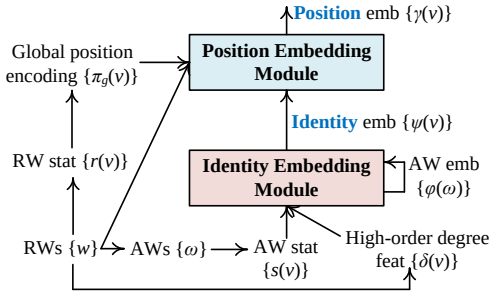


Figure 2: Model architecture of IRWE with an identity embedding module and a position embedding module.

possible nodes and graphs. A transductive method cannot support the inductive inference but an inductive approach can tackle both settings.

In this study, we focus on the joint inductive inference of identity and position embeddings. A novel IRWE method is proposed which combines multiple attention units with different designs to handle RWs and induced AWs.

Definition 4 (*Random Walk & Anonymous Walk, RW & AW*). An RW with length l can be described as a sequence $w = (w^{(0)}, w^{(1)}, \ldots, w^{(l)})$, where $w^{(j)} \in \mathcal{V}$ is the j-th node and $(w^{(j)}, w^{(j+1)}) \in \mathcal{E}$. We assume that the index j starts from 0. For an RW w, one can map it to an AW $\omega = (I_w(w^{(0)}), \ldots, I_w(w^{(l)}))$, where $I_w(w^{(j)})$ maps $w^{(j)}$ to its first occurrence index in w.

In Fig. 1, $(v_1, v_4, v_5, v_1, v_6)$ is a valid RW with $(0, 1, 2, 0, 3)$ as its AW. In particular, AW is the anonymization of RW, indicating that two RWs (e.g., (v_1, v_4, v_5, v_1) and (v_8, v_{10}, v_9, v_8)) can be mapped to a common AW (i.e., $(0, 1, 2, 0)$). In Section 4, we demonstrate that AW and its induced statistics can be features shared by all possible nodes and graphs. It can thus support the inductive embedding inference without relying on the availability and aggregation of attributes.

Definition 5 (*Attention*). A typical attention unit includes the inputs of key, query, and value, which can be described by matrices $\mathbf{K} \in \mathbb{R}^{m \times d}$, $\mathbf{Q} \in \mathbb{R}^{n \times d}$, and $\mathbf{V} \in \mathbb{R}^{m \times d}$. Assume that there are h attention heads. Let $\tilde{d} = d/h$. For the j-th head, a widely used design (Vaswani et al., 2017) first derives the linear mappings of inputs via $\tilde{\mathbf{K}}^{(j)} = \mathbf{K}\mathbf{W}_k^{(j)}$, $\tilde{\mathbf{Q}}^{(j)} = \mathbf{Q}\mathbf{W}_q^{(j)}$, and $\tilde{\mathbf{V}}^{(j)} = \mathbf{V}\mathbf{W}_v^{(j)}$, with $\{\mathbf{W}_k^{(j)} \in \mathbb{R}^{d \times \tilde{d}}, \mathbf{W}_q^{(j)} \in \mathbb{R}^{d \times \tilde{d}}, \mathbf{W}_v^{(j)} \in \mathbb{R}^{d \times \tilde{d}}\}$ as trainable parameters. The attention head further outputs another matrix $\mathbf{Z}^{(j)} \in \mathbb{R}^{n \times \tilde{d}}$ via

$$\mathbf{Z}^{(j)} = \mathrm{Att}_j(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := \mathrm{softmax}(\frac{\tilde{\mathbf{Q}}^{(j)}\tilde{\mathbf{K}}^{(j)T}}{\sqrt{\tilde{d}}})\tilde{\mathbf{V}}^{(j)}, \tag{1}$$

where each row of $\mathbf{Z}^{(j)}$ is the linear combination of rows in $\tilde{\mathbf{V}}^{(j)}$ with the combination weights determined by a row-wise softmax w.r.t. the inner product between $\tilde{\mathbf{Q}}^{(j)}$ and $\tilde{\mathbf{K}}^{(j)}$. To derive the final output, one can concatenate the outputs of all the heads via $\mathbf{Z} = \mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathbf{Z}^{(1)}|| \cdots ||\mathbf{Z}^{(h)}]$.

4 Methodology

In this section, we elaborate on the model architecture as well as the optimization and inference of IRWE. An overview of the model architecture is shown in Fig. 2. It includes two modules that derive identity embeddings $\{\psi(v)\}$ and position embeddings $\{\gamma(v)\}$ based on a series of statistics induced by sampled RWs. In particular, $\{\psi(v)\}$ from the first module is fed into the second module to generate $\{\gamma(v)\}$, where the relation between the two types of embeddings is incorporated. Both $\{\psi(v)\}$ and $\{\gamma(v)\}$ are also jointly optimized in IRWE. Based on the local topology of node v_1 in Fig. 1, Fig. 3 further gives running examples about the derivation of some RW-induced statistics, which are detailed in the rest of this section. Note that we set RW length $l = 3$ and number of sampled RWs $n_S = 5$ just for a simple demonstration. To fully explore the properties of real graphs, one may use larger values of these parameters (e.g., $l = 5$ and $n_S = 1,000$).

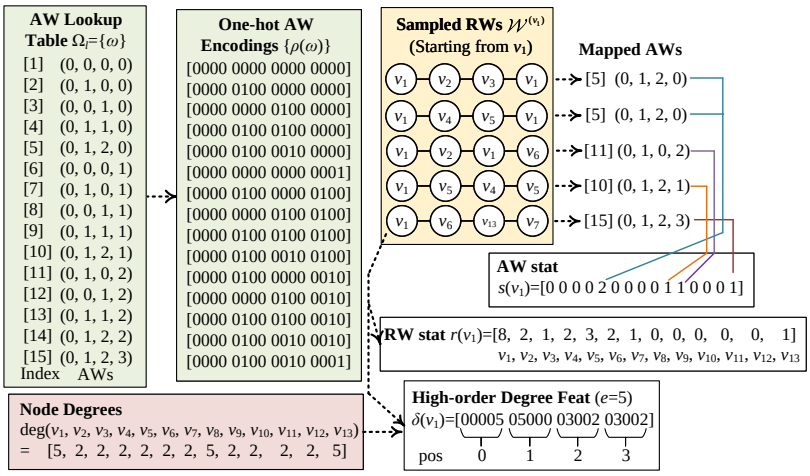


Figure 3: Running examples about the derivation of one-hot AW encodings $\{\rho(\omega)\}$, AW statistics $\{s(v)\}$, high-order degree features $\{\delta(v)\}$, and RW statistics $\{r(v)\}$ for node v_1 in the example topology of Fig. 1, with the RW length $l = 3$ and number of sampled RWs $n_S = 5$.

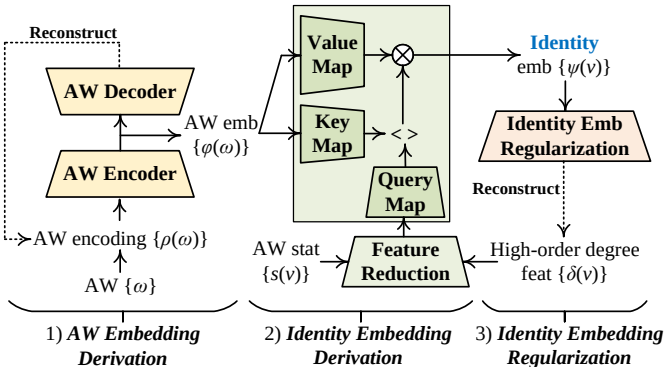


Figure 4: Overview of the identity embedding module.

4.1 Identity Embedding Module

Fig. 4 highlights details of the identity embedding module. It generates identity embeddings $\{\psi(v)\}$ based on the inputs of auxiliary AW embeddings $\{\varphi(\omega)\}$, AW statistics $\{s(v)\}$, and high-order degree features $\{\delta(v)\}$, with $\{s(v), \delta(v)\}$ extracted from the sampled RWs. To support the inductive inference of identity embeddings, we demonstrate that (i) $\{\varphi(\omega)\}$ *can be a special embedding lookup table shared by all possible topology* and (ii) $\{s(v), \delta(v)\}$ *are informative features to characterize node identities* in the rest of this subsection.

In summary, the optimization and inference of this module includes the (1) *AW embedding derivation*, (2) *identity embedding derivation*, and (3) *identity embedding regularization*.

4.1.1 AW Embedding Derivation

As defined in Section 3, AW is the anonymization of RWs. It is possible to map RWs with different sets of nodes to a common AW. For instance, $(0, 1, 2, 0)$ is the common AW of RWs (v_1, v_2, v_3, v_1) and (v_1, v_4, v_5, v_1) as illustrated in Fig. 3. Given a fixed length l, RWs on all possible topology structures can only be mapped to a finite set of AWs Ω_l. Namely, Ω_l *and its induced statistics can be shared by all possible nodes and graphs*, thus having the potential to support the inductive embedding inference.

Based on this intuition, IRWE maintains an auxiliary AW embedding $\varphi(\omega) \in \mathbb{R}^d$ (i.e., a d-dimensional vector) for each unique AW $\omega \in \Omega_l$. In this setting, $\{\varphi(\omega)\}$ can be used as a special embedding lookup table for the

derivation of inductive features regarding graph topology. The inductive inference of identity embeddings $\{\psi(v)\}$ is then supported by the combination of $\{\varphi(\omega)\}$ in the next *identity embedding derivation* step.

IRWE also applies an additional constraint on AW embeddings $\{\varphi(\omega)\}$. Concretely, two AWs with more common elements in corresponding positions are more likely to capture similar properties and thus should have closer representations. For instance, we expect that $(0, 1, 2, 1, 2)$ and $(0, 1, 0, 1, 2)$ should be closer in the AW embedding space than $(0, 1, 2, 1, 2)$ and $(0, 1, 0, 2, 3)$.

To apply this constraint, we transform each AW ω with length l to a one-hot encoding $\rho(\omega) \in \{0, 1\}^{(l+1)^2}$. It is a representation that can be handled by typical neural networks, where $\rho(\omega)_{jl:(j+1)l}$ (i.e., subsequence from the jl-th to the $(j + 1)l$-th positions) is the one-hot encoding of the j-th element in ω. For instance, we have $\rho(\omega) = [0000\ 0100\ 0010\ 0001]$ for $\omega = (0, 1, 2, 3)$ in Fig. 3. An auto-encoder is introduced to derive and regularize $\{\varphi(\omega)\}$, including an encoder and a decoder. For each AW ω, the encoder takes $\rho(\omega)$ as input and outputs AW embedding $\varphi(\omega)$. The decoder tries to reconstruct $\rho(\omega)$ with $\varphi(\omega)$ as its input. Given an AW ω, the encoder $\mathrm{Enc}_\varphi(\cdot)$ and decoder $\mathrm{Dec}_\varphi(\cdot)$ are defined as

$$\begin{aligned} \varphi(\omega) = \mathrm{Enc}_\varphi(\omega) &:= \mathrm{MLP}(\rho(\omega)), \\ \hat{\rho}(\omega) = \mathrm{Dec}_\varphi(\omega) &:= \mathrm{MLP}(\varphi(\omega)), \end{aligned} \tag{2}$$

which are both multi-layer perceptrons (MLPs). Since two similar AWs are expected to have similar one-hot encodings, similar AWs can have close AW embeddings by minimizing the reconstruction error between $\{\rho(\omega)\}$ and $\{\hat{\rho}(\omega)\}$.

4.1.2 Identity Embedding Derivation

IRWE generates identity embeddings $\{\psi(v)\}$ via the combination of AW embeddings $\{\varphi(\omega)\}$ based on the following **Theorem 1** (Micali & Zhu, 2016).

Theorem 1. *Let $\mathcal{G}_s(v, r)$ be the rooted subgraph induced by nodes $\{u\}$ with a distance less than r from v (i.e., $\mathrm{dist}(v, u) \leq r$). Let $q(v, l)$ denote the distribution of AWs w.r.t. RWs starting from v with length l. One can reconstruct $\mathcal{G}_s(v, r)$ using $(q(v, 1), q(v, 2), \cdots, q(v, l))$, where $l = 2(m+1)$; m is the number of edges in $\mathcal{G}_s(v, r)$.*

For a given length l, let η_l be the number of all possible AWs. $q(v, l)$ can be represented as an η_l-dimensional vector, with the j-th element as the occurrence probability of the j-th AW. Since AWs with length l include sequences of those with length less than l (e.g., $(0, 1, 2, 3)$ provides information about $(0, 1, 2)$), one can easily derive $q(v, k)$ $(k < l)$ based on $q(v, l)$. Therefore, $q(v, l)$ *can be used to characterize $\mathcal{G}_s(v, r)$ according to* ***Theorem 1***.

Furthermore, nodes with similar rooted subgraphs are usually expected to play similar structural roles, indicating that *a rooted subgraph $\mathcal{G}_s(v, r)$ can characterize the identity of node v*. For instance, in Fig. 1, $G(v_1, 1)$ and $G(v_8, 1)$ have the same topology structure, which is consistent with the same structural role (i.e., identity) they play, even though they are induced by $\{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $\{v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$, respectively. In summary, *the distribution $q(v, l)$ has the potential to characterize the structural identity of a node v*.

To estimate $q(v, l)$, we introduce the AW statistic $s(v)$ for each node v. As depicted in Fig. 3, we first sample RWs with length l starting from v using the standard unbiased strategy (Perozzi et al., 2014) (see Algorithm 6 in Appendix A). Let $\mathcal{W}^{(v)}$ be the set of sampled RWs starting from v. Each RW $w \in \mathcal{W}^{(v)}$ can be mapped to a corresponding AW. Let Ω_l be an AW lookup table including all the η_l AWs with length l, which is fixed and shared by all possible topology according to our previous discussions. We define the AW statistic as $s(v) := [c(\omega_1), \cdots, c(\omega_{\eta_l})]$, an η_l-dimensional vector with $c(\omega_j)$ as the frequency of the j-th AW in Ω_l w.r.t. $\mathcal{W}^{(v)}$. Algorithm 1 summarizes the procedure to derive $s(v)$ for a node v.

Although η_l grows exponentially with the increase of length l, $\{s(v)\}$ are usually sparse, which can be utilized to reduce the model complexity for a large l. Fig. 5 visualizes the example AW statistics $\{s(v)\}$ derived from RWs on the *Brazil* dataset (see Section 5.1 for its details) with $l = 4$ and $|\mathcal{W}^{(v)}| = 1,000$. The i-th row in Fig. 5 is the AW statistic $s(v_i)$ of node v_i. Dark blue indicates that the corresponding element is

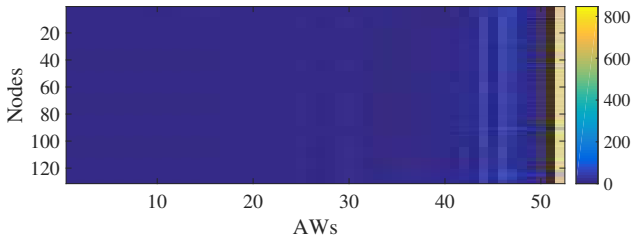


Figure 5: Visualization of example AW statistics $\{s(v)\}$ on the *Brazil* dataset in a matrix form, where the i-th row is the AW statistic $s(v_i)$ of node v_i; dark blue indicates that the corresponding element is 0.

Table 2: Variation of the Number of AWs and its Reduced Value w.r.t. RWs on *Brazil* as the Length Increases

l	4	5	6	7	8	9
η_l	52	203	877	4,140	21,147	115,975
$\tilde{\eta}_l$	15	52	195	610	1,540	3,173

0. There exist many AWs $\{\omega_j\}$ not observed during the RW sampling (i.e., $s(v)_j = 0$ for $\forall v \in \mathcal{V}$). We can remove terms w.r.t. these unobserved AWs in Ω_l and $\{s(v)\}$. Let $\tilde{\Omega}_l$, $\tilde{s}(v)$, and $\tilde{\eta}_l$ be the reduced Ω_l, $s(v)$, and η_l, respectively. Table 2 shows the variation of η_l and $\tilde{\eta}_l$ on *Brazil* as l increases from 4 to 9 (with $|\mathcal{W}^{(v)}| = 1,000$), where η_l can be significantly reduced (e.g., from more than 100K to about 3K for $l = 9$).

In addition to the reduced AW statistics $\{\tilde{s}(v)\}$, one can also characterize node identities from the view of node degrees (Ribeiro et al., 2017; Wu et al., 2019) based on the following **Hypothesis 1**.

Hypothesis 1. *Nodes with the same degree are expected to play the same structural role. This concept can be extended to the high-order neighbors of each node. Namely, nodes are expected to have similar identities if they have similar node degree statistics (e.g., frequencies or distribution over all possible degree values) w.r.t. their high-order neighbors.*

Consistent with **Hypothesis 1**, we introduce a high-order degree feature $\delta(v)$ for each node v. Algorithm 2 summarizes the procedure to derive $\delta(v)$. For each node u, we can obtain a bucket one-hot encoding $\rho_d(u) \in \{0,1\}^e$ w.r.t. its degree $\deg(u)$ (i.e., lines 5-7). Concretely, only the j-th element $\rho_d(u)_j$ is set to 1 with the remaining elements in $\rho_d(u)$ set to 0, where $j = \lfloor (\deg(u) - \deg_{\min})e/(\deg_{\max} - \deg_{\min}) \rfloor$; $\deg_{\min}$ and $\deg_{\max}$ are the minimum and maximum degrees among all the nodes. Since *high-order neighbors of a node v can be explored by RWs $\mathcal{W}^{(v)}$ starting from v*, we define $\delta(v) \in \mathbb{Z}^{(l+1)e}$ as an $(l+1)e$-dimensional vector with the subsequence $\delta(v)_{ie:(i+1)e}$ as the sum of bucket one-hot degree encodings w.r.t. nodes occurred at the i-th position of RWs in $\mathcal{W}^{(v)}$ (i.e., line 8). Fig. 3 gives a running example to derive $\delta(v_1)$ (with $e = 5$) for node v_1 in Fig. 1.

Following the aforementioned discussions regarding **Theorem 1** and **Hypothesis 1**, *IRWE derives identity embeddings $\{\psi(v)\}$ via the adaptive combination of AW embeddings $\{\varphi(\omega)\}$ according to the information encoded in AW statistics $\{\tilde{s}(v)\}$ and high-order degree features $\{\delta(v)\}$*. The multi-head attention is applied to automatically determine the contribution of each AW embedding $\varphi(\omega)$ in the combination, where we treat $\{\varphi(\omega)\}$ as the key and value; the concatenated feature $[\tilde{s}(v)||\delta(v)]$ is used as the query. Note that $[\tilde{s}(v)||\delta(v)]$ is an $(\tilde{\eta}_l + le)$-dimensional vector. Before feeding it to the multi-head attention, we introduce the feature reduction encoder $\mathrm{Red}_s(\cdot)$, an MLP, to reduce its dimensionality to d:

$$\bar{s}(v) = \mathrm{Red}_s(v) := \mathrm{MLP}([\tilde{s}(v)||\delta(v)]). \tag{3}$$

In summary, the multi-head attention that derives identity embeddings $\{\psi(v)\}$ can be described as

$$\mathbf{Z} = \mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Att}(\{\bar{s}(v)\}, \{\varphi(\omega)\}, \{\varphi(\omega)\}), \tag{4}$$

where $\mathrm{Att}(\cdot,\cdot,\cdot)$ has the same definition as that in (1) with $\mathbf{Q}_{i,:} = \bar{s}(v_i)$, $\mathbf{K}_{j,:} = \mathbf{V}_{j,:} = \varphi(\omega_j)$, and $\mathbf{Z}_{i,:} = \psi(v_i)$.

Algorithm 1: Derivation of AW Statistics

Input: target node v; RW length l; sampled RWs $\mathcal{W}^{(v)}$; AW lookup table Ω_l; number of AWs η_l
Output: AW statistic $s(v)$ w.r.t. v

$s(v) \leftarrow [0, 0, \cdots, 0]^{\eta_l}$ //Initialize $s(v)$
for each $w \in \mathcal{W}^{(v)}$ **do**
  map RW w to its corresponding AW ω
  get the index j of AW ω in lookup table Ω_l
  $s(v)_j \leftarrow s(v)_j + 1$ //Update $s(v)$

Algorithm 2: Derivation of Degree Features

Input: target node v; RW length l; one-hot degree encoding dimensionality e; sampled RWs $\mathcal{W}^{(v)}$; minimum degree $\deg_{\min}$; maximum degree $\deg_{\max}$
Output: high-order degree feature $\delta(v)$ w.r.t. v

$\delta(v) \leftarrow [0, 0, \cdots, 0]^{(l+1)e}$ //Initialize degree feature $\delta(v)$
for each $w \in \mathcal{W}^{(v)}$ **do**
  for i **from** 0 **to** l **do**
    $u \leftarrow w^{(i)}$ //i-th node in current RW w
    $\rho_d(u) \leftarrow [0, \cdots, 0] \in \mathbb{R}^e$ //Initialize degree encoding $\rho_d(u)$
    $j \leftarrow \lfloor (\deg(u) - \deg_{\min})e/(\deg_{\max} - \deg_{\min}) \rfloor$
    $\rho_d(u)_j \leftarrow 1$ //Update $\rho_d(u)$
    $\delta(v)_{ie:(i+1)e} \leftarrow \delta(v)_{ie:(i+1)e} + \rho_d(u)$//Update $\delta(v)$

4.1.3 Identity Embedding Regularization

In addition, statistics $\{[\tilde{s}(v)||\delta(v)]\}$ induced by the sampled RWs can also be used to regularize identity embeddings $\{\psi(v)\}$ during the model optimization. We introduce an identity embedding regularization unit $\mathrm{Reg}_\psi(\cdot)$, which takes the identity embedding $\psi(v)$ of each node v as input and uses an MLP to reconstruct the corresponding feature $[\tilde{s}(v)||\delta(v)]$:

$$\hat{g}(v) = \mathrm{Reg}_\psi(v) := \mathrm{MLP}(\psi(v)), \tag{5}$$

with $\hat{g}(v)$ as the reconstructed feature. By minimizing the reconstruction error between $\{\hat{g}(v)\}$ and $\{[\tilde{s}(v)||\delta(v)]\}$, it can force $\{\psi(v)\}$ to encode properties of node identities hidden in $\{[\tilde{s}(v)||\delta(v)]\}$. Note that we only apply $\mathrm{Reg}_\psi(\cdot)$ to optimize $\{\psi(v)\}$. We do not need this unit in the inference phase.

4.2 Position Embedding Module

An overview of the position embedding module is depicted in Fig. 6. It takes (i) $\{\psi(v)\}$ given by the previous identity embedding module and (ii) auxiliary position encodings $\{\pi_g(v), \pi_l(j)\}$ derived from the sampled RWs $\{\mathcal{W}^{(v)}\}$ as inputs and finally generates position embeddings $\{\gamma(v)\}$.

Instead of using the attribute aggregation on graph topology (e.g., message passing of GNNs), we convert the topology into a set of RWs, to which some neural network structures designed for sequential data (e.g., RNN and attention) can be applied. As a demonstration, we use the transformer encoder (Vaswani et al., 2017), a sophisticated attention-based structure, to handle RWs.

In addition to the sequential input (e.g., sampled RWs), transformer also includes the input of 'position' encoding that describes the position of each element in a sequence. However, graph topology is non-Euclidean, where (i) node indices are permutation-invariant and (ii) different nodes may have different numbers of neighbors. As described in **Definition 3**, the node position in graph topology has a different physical meaning from that in Euclidean sequences (e.g., sentences and RWs). To describe both the (i) Euclidean position in RWs and (ii) node position in graph topology, we introduce the *local* and *global* position encodings (denoted as $\pi_l(j)$ and $\pi_g(v)$) for a sequence position with index j and each node v, respectively.

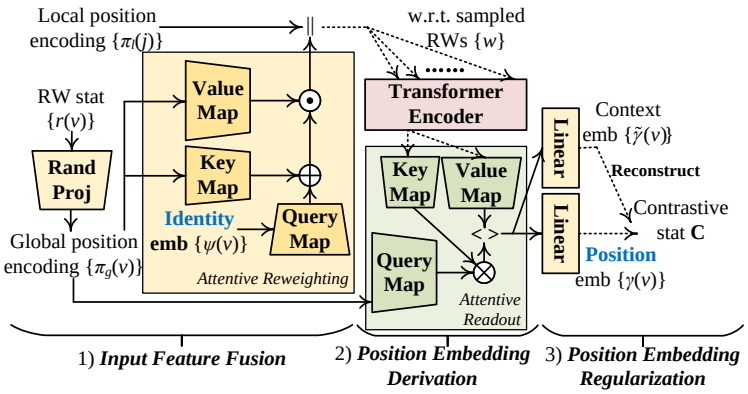


Figure 6: Overview of the position embedding module.

Algorithm 3: Derivation of Global Position Encoding

Input: target node v; sampled RWs $\mathcal{W}^{(v)}$; node set $\mathcal{V}$; random matrix $\mathbf{\Theta} \in \mathbb{R}^{|\mathcal{V}| \times d}$
Output: global position encoding $\pi_g(v)$ w.r.t. v

1 $r(v) \leftarrow [0, 0, \cdots, 0]^{|\mathcal{V}|}$ //Initialize RW statistic $r(v)$
2 **for each** $w \in \mathcal{W}^{(v)}$ **do**
3 **for each** $v_j \in w$ **do**
4 $r(v)_j \leftarrow r(v)_j + 1$ //Update $r(v)$
5 $\pi_g(v) \leftarrow r(v)\mathbf{\Theta}$ //Derive $\pi_g(v)$

In summary, the optimization and inference of this module includes the (1) *input feature fusion*, (2) *position embedding derivation*, and (3) *position embedding regularization*.

4.2.1 Input Feature Fusion

The first *input feature fusion* step extracts the local and global position encodings $\{\pi_l(j), \pi_g(v)\}$ and further derives inputs of the transformer encoder that incorporate identity embeddings $\{\psi(v)\}$.

Since the RW length l is usually not very large (e.g., less than 10 in our experiments), we define the local position encoding $\pi_l(j) \in \{0, 1\}^{l+1}$ as the standard one-hot encoding of index j, an $(l+1)$-dimensional vector where only the j-th element is set to 1 with the remaining elements set to 0.

Inspired by previous studies (Perozzi et al., 2014; Grover & Leskovec, 2016; Zhu et al., 2021) that validated the potential of RW for exploring node positions (e.g., community structures), we extract the global position encodings $\{\pi_g(v)\}$ based on auxiliary RW statistics $\{r(v)\}$.

Algorithm 3 summarizes the procedure to derive $\pi_g(v)$ for a node v. Concretely, we maintain a vector $r(v) \in \mathbb{Z}^{|\mathcal{V}|}$ with the j-th element $r(v)_j$ as the frequency that node v_j occurs in RWs $\mathcal{W}^{(v)}$ starting from v (i.e., lines 1-4), which is usually sparse. A running example to derive $r(v_1)$ w.r.t. node v_1 in Fig. 1 is depicted in Fig. 3. Since nodes in a community should be densely connected, *nodes within the same community are more likely to be reached via RWs compared with those in different communities.* Therefore, *nodes with similar positions (e.g., in the same community) are highly believed to have similar RW statistics* (e.g., in terms of $\{r(v)\}$). We then derive $\pi_g(v)$ by mapping $r(v)$ to a d-dimensional vector via the Gaussian random projection (i.e., line 5), an efficient dimension reduction technique that can preserve the relative distance between original features with a theoretical guarantee (Arriaga & Vempala, 2006). Concretely, we define

$$\pi_g(v) = r(v)\mathbf{\Theta} \text{ with } \mathbf{\Theta} \in \mathbb{R}^{|\mathcal{V}| \times d}, \mathbf{\Theta}_{ir} \sim \mathcal{N}(0, 1/d). \tag{6}$$

In this setting, the non-Euclidean positions between nodes in graph topology can be encoded in terms of the relative distance between $\{\pi_g(v)\}$. Therefore, $\pi_g(v)$ has the initial ability to encode the node position of v.

We further demonstrate that the utilization of transformer encoder, which integrates initial position encodings $\{\pi_g(v)\}$ and identity embeddings $\{\psi(v)\}$ w.r.t. the sampled RWs, can derive more informative position embeddings $\{\gamma(v)\}$. IRWE incorporates the intrinsic relation between node identities and positions based on the following **Hypothesis 2**.

Hypothesis 2. *In a (local) community (i.e., node cluster with dense linkages), nodes with different structural roles may have different contributions in forming the community structure.*

For instance, in a social network, an opinion leader (e.g., v_1 and v_8 in Fig. 1) is expected to have more contribution in forming the community it belongs to than an ordinary audience (e.g., v_2 and v_9 in Fig. 1). Based on this intuition, we use identity embeddings $\{\psi(v)\}$ to reweight global position encodings $\{\pi_g(v)\}$, with the reweighting contributions automatically determined by a modified attention unit. In this attentive reweighting unit, we set identity embeddings $\{\psi(v)\}$ as the query and let global position encodings $\{\pi_g(v)\}$ as the key and value (i.e., $\mathbf{Q}_{i,:} = \psi(v_i)$ and $\mathbf{K}_{i,:} = \mathbf{V}_{i,:} = \pi_g(v_i)$). Different from the standard attention unit described in (1), the modified attention unit is defined as

$$\begin{aligned} &\mathbf{Z} = \mathrm{ReAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) := (\mathrm{MLP}(\tilde{\mathbf{Q}}) + \mathrm{MLP}(\tilde{\mathbf{K}})) \odot \tilde{\mathbf{V}}, \\ &\tilde{\mathbf{Q}} := \mathrm{BN}(\mathbf{Q}), \tilde{\mathbf{K}} := \mathrm{BN}(\mathbf{K}), \tilde{\mathbf{V}} := \mathrm{BN}(\mathbf{V}), \end{aligned} \tag{7}$$

where $\mathrm{BN}(\cdot)$ and $\odot$ are the batch normalization and element-wise multiplication. Namely, the attentive reweighting unit first conducts the batch normalization on $\{\mathbf{Q}, \mathbf{K}, \mathbf{V}\}$. We then apply two MLPs to respectively derive nonlinear mappings of the normalized $\{\mathbf{Q}, \mathbf{K}\}$ and use their sum to support the *element-wise reweighting* of the normalized $\mathbf{V}$. We denote the reweighted vector w.r.t. a node v_i as $\bar{\pi}_g(v_i) = \mathbf{Z}_{i,:}$.

Given a sampled RW $w = (w^{(0)}, w^{(1)}, \cdots, w^{(l)})$, IRWE concatenates the reweighted vector $\bar{\pi}_g(w^{(j)})$ and local position encoding $\pi_l(j)$ for the j-th node and feeds its linear mapping to the transformer encoder, i.e.,

$$t(w^{(j)}) := [\bar{\pi}_g(w^{(j)}) || \pi_l(j)]\mathbf{W}_t, \tag{8}$$

where $\mathbf{W}_t \in \mathbb{R}^{(d+l+1)\times d}$ is trainable. In this setting, the global and local position encodings $\{\pi_g(v), \pi_l(j)\}$ as well as identity embeddings $\{\psi(v)\}$ can be adaptively integrated.

4.2.2 Position Embedding Derivation

In the *position embedding derivation* step, IRWE uses the transformer encoder to handle each sampled RW $w = (w^{(0)}, \cdots, w^{(l)})$, with the corresponding sequence of vectors $(t(w^{(0)}), \cdots, t(w^{(l)}))$ as input, which can be described as

$$(\bar{t}(w^{(0)}), \cdots) = \mathrm{TransEnc}(t(w^{(0)}), \cdots, t(w^{(l)})). \tag{9}$$

The transformer encoder $\mathrm{TransEnc}(\cdot)$ then outputs another sequence of vectors $(\bar{t}(w^{(0)}), \cdots, \bar{t}(w^{(l)}))$ that has the same dimensionality as the input. $\mathrm{TransEnc}(\cdot)$ follows a multi-layer structure, where each layer consists of the self-attention, skip connections, layer normalization, and feedforward mapping (Vaswani et al., 2017). Let $(\mathbf{u}_0^{<k-1>}, \cdots, \mathbf{u}_l^{<k-1>})$ and $(\mathbf{u}_0^{<k>}, \cdots, \mathbf{u}_l^{<k>})$ be the input and output of the k-th transformer encoder layer, where we have $(\mathbf{u}_0^{<0>}, \cdots, \mathbf{u}_l^{<0>}) = (t(w^{(j)}), \cdots, t(w^{(l)}))$. By reshaping the input to a matrix $\mathbf{U}^{<k-1>}$ with $\mathbf{U}_{j,:}^{<k-1>} = \mathbf{u}_j^{k-1}$, the k-th transformer encoder layer can be described as

$$\begin{aligned} &\bar{\mathbf{U}}^{<k-1>} = \mathrm{Att}(\mathbf{U}^{<k-1>}, \mathbf{U}^{<k-1>}, \mathbf{U}^{<k-1>}), \\ &\mathbf{Y}^{<k-1>} = \mathrm{LN}(\mathbf{U}^{<k-1>} + \bar{\mathbf{U}}^{<k-1>}), \\ &\mathbf{U}^{<k>} = \mathrm{LN}(\mathbf{Y}^{<k-1>} + \mathrm{FFN}(\mathbf{Y}^{<k-1>})), \end{aligned} \tag{10}$$

where the self-attention $\mathrm{Att}(\cdot, \cdot, \cdot)$ shares a definition with (1) and $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{U}^{<k-1>}$; $\mathrm{LN}(\cdot)$ is the layer normalization; $\mathrm{FFN}(\mathbf{X}) := \mathrm{ReLU}(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$ represents a 2-layer MLP with trainable parameters $\{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2\}$.

For a given RW w starting from each node v (i.e., $w^{(0)} = v$), the first vector $\bar{t}(w^{(0)}) = \bar{t}(v)$ in the output of transformer encoder can be used as a representation of v. Since we sample multiple RWs $\mathcal{W}^{(v)}$ starting from each node v, we can derive multiple such representations for v based on $\mathcal{W}^{(v)}$. However, we only need one unique representation $\gamma(v)$ for each node v to encode its position information. Let $\bar{t}^{(v)} := \{\bar{t}(w^{(0)}) | w \in \mathcal{W}^{(v)}\}$

(i.e., set of the multiple representations w.r.t. $\mathcal{W}^{(v)}$). A naive strategy to derive $\gamma(v)$ is to average the representations in $\bar{t}^{(v)}$. In contrast, we introduce an attentive readout function that computes the weighted mean of $\bar{t}^{(v)}$ with the weights automatically determined by another multi-head attention unit. The attentive readout operation to derive the unique representation w.r.t. a node v can be described as

$$\begin{aligned} &\mathbf{z} = \mathrm{ROut}(\bar{t}^{(v)}, \pi_g(v)) := \mathrm{Att}(\pi_g(v), \bar{t}^{(v)}, \bar{t}^{(v)}), \\ &\gamma(v) := \mathbf{z}\mathbf{W}_\gamma + \mathbf{b}_\gamma, \bar{\gamma}(v) := \mathbf{z}\mathbf{W}_{\bar{\gamma}} + \mathbf{b}_{\bar{\gamma}}, \end{aligned} \tag{11}$$

where $\mathrm{Att}(\cdot,\cdot,\cdot)$ has the same definition as (1); $\gamma(v)$ and $\bar{\gamma}(v)$ are the position embedding and auxiliary context embedding of node v derived via two linear mappings of the output vector $\mathbf{z} \in \mathbb{R}^d$; $\{\mathbf{W}_\gamma, \mathbf{b}_\gamma, \mathbf{W}_{\bar{\gamma}}, \mathbf{b}_{\bar{\gamma}}\}$ are trainable parameters. In particular, we let the global position encoding $\pi_g(v)$, which preserves initial node position information, be the query and set $\bar{t}^{(v)}$ to be the key and value (i.e., $\mathbf{Q} = \pi_g(v) \in \mathbb{R}^{1\times d}$ and $\mathbf{K}_{j,:} = \mathbf{V}_{j,:} = \bar{t}^{(v)}_j$).

4.2.3 Position Embedding Regularization

The *position embedding regularization* step optimizes the derived position embeddings $\{\gamma(v)\}$ together with auxiliary context embeddings $\{\bar{\gamma}(v)\}$. In general, some of existing embedding methods (Perozzi et al., 2014; Tang et al., 2015; Hamilton et al., 2017) are optimized via the following contrastive loss based on the approximated negative sampling:

$$\min \mathcal{L}_{\mathrm{cnr}} = -\sum_{(v_i,v_j)\in D} [p_{ij} \ln \sigma(\frac{\gamma(v_i)\tilde{\gamma}^T(v_j)}{\tau}) + Q n_j \ln \sigma(-\frac{\gamma(v_i)\tilde{\gamma}^T(v_j)}{\tau})], \tag{12}$$

where D denotes the training set including positive and negative samples in terms of node pairs $\{(v_i, v_j)\}$; p_{ij} is defined as the statistic of a positive node pair (v_i, v_j) (e.g., the frequency that (v_i, v_j) occurs in the l-step RW sampling or normalized edge weight of (v_i, v_j)); Q is the number of negative samples while n_j is usually set to be the probability that (v_i, v_j) is selected as a negative sample; $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function; τ is a temperature parameter to be specified. Note that different methods may have different definitions regarding the positive and negative samples associated with $\{D, p_{ij}, n_j\}$. We follow prior work (Tang et al., 2015) to let $p_{ij} := \mathbf{A}_{ij}/\deg(v_i)$ (i.e., the probability that there is an edge from v_i to v_j with $\mathbf{A} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$ as the adjacency matrix) and $n_j \propto (\sum_{i:(v_i,v_j)\in\mathcal{E}} p_{ij})^{0.75}$. In the next subsection, we demonstrate that the contrastive loss (12) can be converted to an equivalent reconstruction loss such that *the joint optimization of IRWE only includes the combination of several reconstruction objectives.*

4.3 Model Optimization & Inference

For a given RW length l, let $\tilde{\Omega}_l$ be the reduced AW lookup table w.r.t. the reduced AW statistics $\{\tilde{s}(v)\}$ in (3). According to our discussions in Section 4.1, the optimization objective of identity embeddings $\{\psi(v)\}$ can be described as

$$\min \mathcal{L}_\psi := \mathcal{L}_{\mathrm{reg}-\varphi} + \alpha \mathcal{L}_{\mathrm{reg}-\psi}, \tag{13}$$

$$\mathcal{L}_{\mathrm{reg}-\varphi} := \sum\nolimits_{\omega\in\tilde{\Omega}_l} |\rho(\omega) - \hat{\rho}(\omega)|_2^2, \tag{14}$$

$$\mathcal{L}_{\mathrm{reg}-\psi} := \sum\nolimits_{v\in\mathcal{V}} |[\tilde{s}(v)||\delta(v)]/|\mathcal{W}^{(v)}| - \hat{g}(v)|_2^2, \tag{15}$$

where $\mathcal{L}_{\mathrm{reg}-\varphi}$ regularizes auxiliary AW embeddings $\{\varphi(\omega)\}$ by reconstructing the one-hot AW encodings $\{\rho(\omega)\}$ via the auto-encoder defined in (2); $\mathcal{L}_{\mathrm{reg}-\psi}$ regularizes the derived identity embeddings $\{\psi(v)\}$ by minimizing the error between (i) features $\{[\tilde{s}(v)||\delta(v)]\}$ normalized by the number of sampled RWs $|\mathcal{W}^{(v)}|$ and (ii) reconstructed values $\{\hat{g}(v)\}$ given by (5); α is a tunable parameter.

As described in Section 4.2, one can optimize position embeddings $\{\gamma(v)\}$ via a contrastive loss (12). It can be converted to another reconstruction loss based on the following **Proposition 1**. In this setting, the optimization of $\{\psi(v)\}$ and $\{\gamma(v)\}$ only includes three simple reconstruction losses.

Proposition 1. *Let $\mathbf{\Gamma} \in \mathbb{R}^{|\mathcal{V}|\times d}$ and $\bar{\mathbf{\Gamma}} \in \mathbb{R}^{|\mathcal{V}|\times d}$ be the matrix forms of $\{\gamma(v_i)\}$ and $\{\bar{\gamma}(v_i)\}$ with the i-th rows denoting the corresponding embeddings of node v_i. We introduce the auxiliary contrastive statistics*

Algorithm 4: Model Optimization of IRWE

Input: topology $(\mathcal{V}, \mathcal{E})$; RW settings $\{l, n_S, n_I\}$; local position encodings $\{\pi_l(j)\}$; optimization settings $\{m, m_\psi, m_\gamma, \lambda_\psi, \lambda_\gamma\}$

Output: sampled RWs $\{\mathcal{W}^{(v)}, \mathcal{W}_I^{(v)}\}$; reduced AW lookup table $\tilde{\Omega}_l$ & induced statistics $\{\tilde{s}(v), \delta(v), \pi_g(v)\}$; optimized model parameters $\{\theta_\psi^*, \theta_\gamma^*\}$

1 get AW lookup table Ω_l w.r.t. length l
2 get min degree $\deg_{\min}$ & max degree $\deg_{\max}$ of $(\mathcal{V}, \mathcal{E})$
3 get contrastive statistic $\mathbf{C}$
4 **for each** *node* $v \in \mathcal{V}$ **do**
5 sample n_S RWs $\mathcal{W}^{(v)}$ staring from v via Algorithm 6
6 get AW statistic $s(v)$ w.r.t. $\mathcal{W}^{(v)}$ via Algorithm 1
7 get degree feature $\delta(v)$ w.r.t. $\{\mathcal{W}^{(v)}, \mathrm{d}_{\min}, \mathrm{d}_{\max}\}$ via Algorithm 2
8 get global position encoding $\pi_g(v)$ w.r.t. $\mathcal{W}^{(v)}$ via Algorithm 3
9 randomly select n_I RWs $\mathcal{W}_I^{(v)}$ from $\mathcal{W}^{(v)}$
10 get reduced AW statistic $\{\tilde{s}(v)\}$ by deleting unobserved AWs
11 get reduced AW lookup table $\tilde{\Omega}_l$ w.r.t. $\{\tilde{s}(v)\}$
12 initial model parameters $\{\theta_\psi, \theta_\gamma\}$
13 **for** *iter_count* **from** 1 **to** m **do**
14 **for** $count_\psi$ **from** 1 **to** m_ψ **do**
15 get $\{\hat{\rho}(\omega), \hat{g}(v)\}$ w.r.t. $\{\tilde{\Omega}_l, \tilde{s}(v), \delta(v)\}$
16 get training loss $\mathcal{L}_\psi$ via (13)
17 optimize *identity embeddings* $\{\psi(v)\}$ via $\mathrm{Opt}(\lambda_\psi, \theta_\psi, \mathcal{L}_\psi)$
18 **for** $count_\gamma$ **from** 1 **to** m_γ **do**
19 get identity embeddings $\{\psi(v)\}$ w.r.t. $\{\tilde{\Omega}_l, \tilde{s}(v), \delta(v)\}$
20 get position embeddings $\{\gamma(v)\}$ w.r.t. $\{\psi(v), \pi_g(v), \pi_l(j), \mathcal{W}_I^{(v)}\}$
21 get training loss $\mathcal{L}_\gamma$ via (16)
22 optimize *position embeddings* $\{\gamma(v)\}$ via $\mathrm{Opt}(\lambda_\gamma, \{\theta_\psi, \theta_\gamma\}, \mathcal{L}_\gamma)$
23 save model parameters $\{\theta_\psi, \theta_\gamma\}$

$\mathbf{C} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ *in terms of a sparse matrix where* $\mathbf{C}_{ij} = \ln p_{ij} - \ln(Qn_j)$ *if* $(v_i, v_j) \in \mathcal{E}$ *and* $\mathbf{C}_{ij} = 0$ *otherwise. The contrastive loss (12) is equivalent to the following reconstruction loss:*

$$\min \mathcal{L}_\gamma = \left\| \mathbf{\Gamma} \bar{\mathbf{\Gamma}}^T / \tau - \mathbf{C} \right\|_F^2 . \tag{16}$$

The basic idea to prove **Proposition 1** is to let the partial derivative $\partial \mathcal{L}_{cnr} / \partial [\gamma(v_i) \bar{\gamma}^T(v_j)/\tau]$ w.r.t. each edge (v_i, v_j) to 0. For convenience, we leave the detailed proof of **Proposition 1** in Appendix B.

Algorithm 4 summarizes the overall optimization procedure of IRWE. Before formally optimizing the model, we sampled n_S RWs $\mathcal{W}^{(v)}$ starting from each node v and derive statistics $\{\tilde{s}(v), \delta(v), \pi_g(v)\}$ induced by $\{\mathcal{W}^{(v)}\}$ (i.e., lines 4-10). In particular, we randomly select n_I RWs $\mathcal{W}_I^{(v)}$ from $\mathcal{W}^{(v)}$ ($n_I < n_S$) for each node (i.e., line 9), which are handled by the transformer encoder in the position embedding module for the inference of $\{\gamma(v)\}$. Namely, we use a small ratio of the sampled RWs to derive $\{\gamma(v)\}$ due to the high complexity of transformer encoder. Note that we only sample RWs and derive induced statistics once, which are shared by the following optimization iterations in lines 12-23.

To jointly optimize $\{\psi(v)\}$ and $\{\gamma(v)\}$, one can combine (13) and (16) to derive a single hybrid optimization objective. However, our pre-experiments show that better embedding quality can be achieved if we separately optimize the two types of embeddings. One possible reason is that the two modules in IRWE usually have unbalanced scales of parameters. Let θ_ψ and θ_γ be the sets of model parameters of the identity and position modules. The scale of θ_γ may be larger than θ_ψ due to the application of transformer encoder. As described in lines 14-17 and lines 19-22, we respectively update $\{\psi(v)\}$ and $\{\gamma(v)\}$ $m_\psi \geq 1$ and $m_\gamma \geq 1$ times based on (13) and (16) in each iteration, where we can effectively balance the optimization of $\{\psi(v)\}$ and $\{\gamma(v)\}$ by adjusting m_ψ and m_γ.

Algorithm 5: Inductive Inference within a Graph

Input: optimized model parameters $\{\theta_\psi^*, \theta_\gamma^*\}$; new topology $(\mathcal{V} \cup \mathcal{V}', \mathcal{E}')$; RW settings $\{l, n_S, n_I\}$; local position encodings $\{\pi_l(j)\}$; $\{\tilde{\Omega}_l, \deg_{\min}, \deg_{\max}, \tilde{s}(v), \delta(v), \pi_g(v)\}$ derived in model optimization on old topology $(\mathcal{V}, \mathcal{E})$

Output: *inductive* embeddings $\{\psi(v)\}$ & $\{\gamma(v)\}$ w.r.t. $\mathcal{V}'$

1 **for each** *node* $v \in \mathcal{V}'$ **do**
2 sample n_S RWs $\mathcal{W}^{(v)}$ from v w.r.t. $\mathcal{E}'$ via Algorithm 6
3 get AW statistic $\tilde{s}'(v)$ w.r.t. $\{\mathcal{W}^{(v)}, \tilde{\Omega}_l\}$ via Algorithm 8
4 get degree feature $\delta'(v)$ w.r.t. $\{\mathcal{W}^{(v)}, \mathrm{d}_{\min}, \mathrm{d}_{\max}\}$ via Algorithm 9
5 get global position encoding $\pi_g'(v)$ w.r.t. $\{\mathcal{W}^{(v)}, \mathcal{V}\}$ via Algorithm 10
6 randomly select n_I RWs $\mathcal{W}_I^{(v)}$ from $\mathcal{W}^{(v)}$
7 add $\tilde{s}'(v)$, $\delta'(v)$, & $\pi_g'(v)$ to $\{\tilde{s}(v)\}$, $\{\delta(v)\}$, & $\{\pi_g(v)\}$
8 get $\{\psi(v)\}$ based on $\{\tilde{\Omega}_l, \tilde{s}(v), \delta(v)\}$ w.r.t. $\mathcal{V} \cup \mathcal{V}'$
9 get $\{\gamma(v)\}$ based on $\{\psi(v), \pi_g(v), \pi_l(j), \mathcal{W}_I^{(v)}\}$ w.r.t. $\mathcal{V}'$

Note that $\{\psi(v)\}$ are inputs of the position embedding module, providing node identity information for the inference of $\{\gamma(v)\}$. The optimization of $\{\gamma(v)\}$ also includes the update of θ_ψ via the backpropagation of gradient descent, which may also affect the inference of $\{\psi(v)\}$. In this setting, *the two types of embeddings are jointly optimized although we adopt a separate updating strategy.* The Adam optimizer is used to update $\{\theta_\psi, \theta_\gamma\}$ during the optimization, with λ_ψ and λ_γ as the learning rates for $\{\psi(v)\}$ and $\{\gamma(v)\}$. Finally, we save model parameters after m iterations.

During the model optimization, we save the sampled RWs $\{\mathcal{W}^{(v)}, \mathcal{W}_I^{(v)}\}$, reduced AW lookup table $\tilde{\Omega}_l$, and induced statistics $\{\tilde{s}(v), \delta(v), \pi_g(v)\}$ (i.e., lines 4-11 in Algorithm 4) and use them as inputs of the *transductive inference* of $\{\psi(v)\}$ and $\{\gamma(v)\}$. Then, the *transductive inference* only includes one feedforward propagation through the model. For convenience, we summarize this simple inference procedure in Algorithm 7 (see Appendix A).

To support the *inductive inference for new nodes within a graph*, we adopt an incremental strategy to get the inductive statistics $\{\tilde{s}(v), \delta(v), \pi_g(v)\}$ via modified versions of Algorithms 1, 2, and 3 that utilize some intermediate results derived during the optimization on old topology $(\mathcal{V}, \mathcal{E})$. Algorithm 5 summarizes the *inductive inference within a graph*. Let $\mathcal{V}'$ and $\mathcal{E}'$ be the set of new nodes and edge set induced by $\mathcal{V} \cup \mathcal{V}'$. We sample RWs $\mathcal{W}^{(v)}$ for each new node $v \in \mathcal{V}'$ (i.e., line 2) and get the AW statistic $\tilde{s}(v)$ w.r.t. AWs in the lookup table $\tilde{\Omega}_l$ reduced on old topology $(\mathcal{V}, \mathcal{E})$ (i.e., line 3) rather than all AWs. $\delta(v)$ is derived based on the one-hot degree encoding truncated by the minimum and maximum degrees of $(\mathcal{V}, \mathcal{E})$ (i.e., line 4). In the derivation of $\pi_g(v)$, we compute truncated RW statistic $r(v)$ only w.r.t. previously observed nodes $\mathcal{V}$ (i.e., line 5). For convenience, we detail procedures to derive inductive $\{\tilde{s}(v), \delta(v), \pi_g(v)\}$ in Algorithms 8, 9, and 10 (see Appendix A). Similar to the *transductive inference*, given the derived $\{\tilde{s}(v), \delta(v), \pi_g(v)\}$, we can obtain the *inductive* $\{\psi(v)\}$ and $\{\gamma(v)\}$ via one feedforward propagation (i.e., lines 8-9).

For the *inductive inference across graphs*, we sample RWs $\{\mathcal{W}^{(v)}, \mathcal{W}_I^{(v)}\}$ on each new graph $(\mathcal{V}'', \mathcal{E}'')$. Since there are no shared nodes between the training and inference topology, we can only incrementally compute the reduced/truncated statistics $\{\tilde{s}(v), \delta(v)\}$ using the procedures of lines 3-4 in Algorithm 5. We have to derive global position encodings $\{\pi_g(v)\}$ from scratch via Algorithm 3. We summarize this *inductive inference* procedure in Algorithm 11 (see Appendix A).

4.4 Complexity Analysis

The complexity of the RW sampling starting from each node (i.e., Algorithm 6) is no more than $O(n_S l)$. The complexities to derive AW statistics $s(v)$ (i.e., Algorithm 1), high-order degree features $\delta(v)$ (i.e., Algorithm 2), and global position encoding $\pi_g(v)$ (i.e., Algorithm 3) w.r.t. a node v are $O(n_S)$, $O(n_S l)$, and $O(n_S l + k(v)d)$ (with $k(v)$ as the number of nodes observed in $\mathcal{W}^{(v)}$), respectively. Furthermore, the overall complexity to derive the RW-induced statistics (i.e., the feature inputs of IRWE) from a graph $(\mathcal{V}, \mathcal{E})$ is no

Table 3: Statistics of Datasets

Datasets	N	E	K	**Ground-Truth**
PPI	3,890	38,739	50	(Multi-Label) Node Position
Wiki	4,777	92,517	40	
BlogCatalog	10,312	333,983	39	
USA	1,190	13,599	4	(Multi-Class) Node Identity
Europe	399	5,993	4	
Brazil	131	1,003	4	
PPIs	1,021-3,480	4,554-26,688	10	N/A

more than $O(|\mathcal{V}|n_S l + |\mathcal{V}|n_S + |\mathcal{V}|n_S l + (|\mathcal{V}|n_S l + \bar{k}d)) = O(|\mathcal{V}|n_S l + \bar{k}d)$ (with $\bar{k} := \sum_{v\in\mathcal{V}} k(v)$), which can be significantly speeded up via parallel implementations.

As described in Algorithm 7, the *transductive inference* of IRWE only includes one feedforward propagation through the model. Its complexity is no more than $O(\tilde{\eta}_l l^2 d + |\mathcal{V}|(el + \tilde{\eta}_l)d + |\mathcal{V}|\tilde{\eta}_l dh + (|\mathcal{V}|d^2 + |\mathcal{V}|d) + |\mathcal{V}|(d + l)d + |\mathcal{V}|n_I l^2 dh + n_I d) = O(|\mathcal{V}|(\tilde{\eta}_l + n_I l^2)dh)$, where we assume that $el \approx d$, $l^2 \ll |\mathcal{V}|$, and $d \ll \tilde{\eta}_l$; h is the number of attention heads.

According to Algorithm 5, the complexity of *inductive inference for new nodes within a graph* is $O(|\mathcal{V}'|n_S l + \bar{k}'d + |\mathcal{V}\cup\mathcal{V}'|(\tilde{\eta}_l + n_I l^2)dh)$, with $\bar{k}' := \sum_{v\in\mathcal{V}'} k(v)$. In addition, the complexity of *inductive inference across graphs* (i.e., Algorithm 11) is $O(|\mathcal{V}''|n_S l + \bar{k}''d + |\mathcal{V}''|(\tilde{\eta}_l + n_I l^2)dh)$, with $\bar{k}'' := \sum_{v\in\mathcal{V}''} k(v)$.

5 Experiments

In this section, we elaborate on our experiments. We introduce experiment setups in Section 5.1, including datasets, downstream tasks, and baselines. Evaluation results for the transductive and inductive embedding inference are described and analyzed in Sections 5.2 and 5.3. Ablation study and parameter analysis are further introduced in Sections 5.4 and 5.5. Moreover, we also demonstrate the possible inconsistency between graph topology and attributes in Section 5.6, which verifies our motivation of not considering node attributes.

5.1 Experiment Setups

Datasets. We used seven datasets commonly used by related research to validate the effectiveness of IRWE. Statistics of these datasets are depicted in Table 3, where N, E, and K are the number of nodes, edges, and classes, respectively.

PPI, *Wiki*, *BlogCatalog* are the first type of datasets (Grover & Leskovec, 2016; Zhu et al., 2021) providing the ground-truth of node positions for multi-label classification, which are extracted from (i) social relationships of the BlogCatalog website, (ii) protein-protein interactions of a biology network, and (iii) word cooccurrence of a Wikipedia dump. *USA*, *Europe*, and *Brazil* are the second type of datasets (Ribeiro et al., 2017; Zhu et al., 2021) with node identity ground-truth for multi-class classification, which describe the commercial flights of three air-traffic networks in different places. In summary, *PPI*, *Wiki*, and *BlogCatalog* are widely used to evaluate the quality of *position embedding* while *USA*, *Europe*, and *Brazil* are well-known datasets for the evaluation of *identity embedding*.

Moreover, *PPIs* is a widely used dataset for the *inductive inference across graphs* (Hamilton et al., 2017; Veličković et al., 2018), which includes a set of protein-protein interaction graphs (in terms of connected components) with each graph corresponding to a human tissue. In addition to graph topology, the original *PPIs* dataset also provides node features and ground-truth for node classification. As stated in Section 3, we do not consider graph attributes due to the complicated correlations between topology and attributes. It is also unclear whether the classification ground-truth is dominated by topology or attributes. Therefore, we only used the graph topology of *PPIs* in our experiments.

Downstream Tasks. We followed the experiment settings of prior work (Grover & Leskovec, 2016; Ribeiro et al., 2017; Zhu et al., 2021) to adopt multi-label and multi-class node classification for the evaluation of position and identity embeddings on the first and second types of datasets, respectively. In particular, each

node may belong to multiple classes in multi-label classification while each node can only belong to one class in multi-class classification. We used Micro *F1-score* as the quality metric for the two classification tasks. In particular, to avoid the exception that some labels are not presented in all training examples, we removed classes with very few numbers of members (i.e., less than 8) when conducting node classification.

In addition to node classification, we also adopted unsupervised node clustering to evaluate the quality of identity and position embeddings. Inspired by spectral clustering (Von Luxburg, 2007) and **Hypothesis 1**, we can construct an auxiliary similarity graph $\mathcal{G}_D$ based on the high-order degree features $\{\delta'(v) \in \mathbb{R}^{(l+1)e}\}$ derived via a procedure similar to Algorithm 2. The only difference between $\{\delta'(v)\}$ (used for evaluation) and $\{\delta(v)\}$ (used in IRWE) is that $\delta'(v)$ is directly derived from the rooted subgraph $\mathcal{G}_s(v, l')$ but not the sampled RWs $\mathcal{W}^{(v)}$. We let $\mathcal{G}_D$ be the top-10 similarity graph based on the inner product $\delta'(v_i){\delta'}^T(v_j)/(|\delta'(v_i)||\delta'(v_j)|)$. To obtain $\{\delta'(v)\}$, we set $l' = 5$ (i.e., the order of neighbors) and $e = 500$ (i.e., the dimensionality of the one-hot degree encoding) for the first type of datasets while we let $l = 3$ and $e = 200$ for *PPIs*. Since the high-order degree features are expected to capture node identities, we expect that the node clustering evaluated on $\mathcal{G}_D$ can measure the quality of identity embeddings and define it as the *node identity clustering* task. In this task, we applied a clustering algorithm to embeddings learned on the original graph $\mathcal{G}$ but evaluated the clustering result on $\mathcal{G}_D$.

We also treated the node clustering evaluated on the original graph $\mathcal{G}$ as *community detection* (Newman, 2006), a typical task commonly used for the evaluation of position embeddings.

Normalized cut (*NCut*) (Von Luxburg, 2007) and *modularity* (Newman, 2006) can be used as quality metrics for node identity clustering and community detection. Given a clustering result $\mathcal{C} = \{\mathcal{C}_1, \cdots, \mathcal{C}_K\}$, we define *NCut* w.r.t. the auxiliary graph $\mathcal{G}_D$ as

$$\text{NCut}(\mathcal{C}; \mathcal{G}_D) = 0.5 \sum\nolimits_{r=1}^{K} [\text{cut}(\mathcal{C}_r, \bar{\mathcal{C}}_r)/\text{vol}(\mathcal{C}_r)], \tag{17}$$

where $\bar{\mathcal{C}}_r = \mathcal{V} - \mathcal{C}_r$, $\text{cut}(\mathcal{C}_r, \bar{\mathcal{C}}_r) = \sum_{v_i \in \mathcal{C}_r, v_j \in \bar{\mathcal{C}}_r} (\mathbf{A}_D)_{ij}$, and $\text{vol}(\mathcal{C}_r) = \sum_{v_i \in \mathcal{C}_r, v_j \in \mathcal{V}} (\mathbf{A}_D)_{ij}$, with $\mathbf{A}_D$ as the adjacency matrix of $\mathcal{G}_D$. For a clustering result $\mathcal{C} = \{\mathcal{C}_1, \cdots, \mathcal{C}_K\}$, *modularity* w.r.t. the original graph $\mathcal{G}$ is defined as

$$\text{Mod}(\mathcal{C}; \mathcal{G}) = \frac{1}{2e} \sum_{r=1}^{K} \sum_{v_i, v_j \in \mathcal{C}_r} [\mathbf{A}_{ij} - \frac{\deg(v_i)\deg(v_j)}{2e}], \tag{18}$$

where $e = \sum_i \deg(v_i)/2$ is the number of edges.

Logistic regression and KMeans were used as downstream algorithms for node classification and clustering, respectively. In general, larger F1-score and modularity as well as smaller NCut implies better performance of downstream tasks, thus indicating better embedding quality.

In summary, we adopted (i) *node identity clustering* and (ii) *multi-label node classification* as downstream tasks to respectively evaluate identity and position embeddings on the first type of datasets. For the second type of datasets, (i) *multi-label node classification* and (ii) *community detection* were used to measure the quality of identity and position embeddings. Moreover, we only applied the unsupervised (i) *node identity clustering* and (ii) *community detection* to evaluate the two types of embeddings for *PPIs*, since we did not consider its ground-truth.

Baselines. We compared IRWE with 16 unsupervised baselines, covering identity and position embedding as well as transductive and inductive approaches. Table 4 summarizes all the methods to be evaluated, where '-' denotes that it is unclear for a baseline which type of property (i.e., node identities or positions) it can capture. PhUSION has multiple variants using different proximities for different types of embeddings. We adopted two variants with (i) positive point-wise mutual information and (ii) heat kernel, which are recommended proximities for position and identity embedding, as two baselines denoted as PhN-PPMI and PhN-HK. Each variant of PhUSION can only derive one type of embedding.

For each transductive baseline, we can clearly distinguish that it captures node identities or positions. For inductive baselines, GraLSP, SPINE, and GAS are claimed to be identity embedding methods while P-GNN and CSGCL can preserve node positions. Similar to our method, GraLSP and SPINE use RWs and

Table 4: Details of Embedding Methods to be Evaluated

Methods	***Transductive***	***Inductive***	***Position***	***Identity***
node2vec (Grover & Leskovec, 2016)	√		√	
GraRep (Cao et al., 2015)	√		√	
struc2vec (Ribeiro et al., 2017)	√			√
struc2gauss (Pei et al., 2020)	√			√
PhUSION (Zhu et al., 2021)	√		Δ	Δ
GraphSAGE (Hamilton et al., 2017)		√	-	-
DGI (Velickovic et al., 2019)		√	-	-
GraphMAE (Hou et al., 2022)		√	-	-
GraphMAE2 (Hou et al., 2023)		√	-	-
P-GNN (You et al., 2019)		√	√	
CSGCL (Chen et al., 2023)		√	√	
GraLSP (Jin et al., 2020)		√		√
SPINE (Guo et al., 2019)		√		√
GAS (Guo et al., 2020)		√		√
SANNE (Nguyen et al., 2021)		√	-	-
UGFormer (Nguyen et al., 2022)		√	-	-
IRWE (ours)		√	√	√

induced statistics to enhance the embedding quality. SANNE applies the transformer encoder to handle RWs. However, all the inductive baselines rely on the availability of node attributes. We used the bucket one-hot encodings of node degrees as their attribute inputs, which is a widely-used strategy for inductive methods when attributes are unavailable. Moreover, the transductive methods learn their embeddings only based on graph topology.

All the aforementioned baselines can only generate one set of embeddings. Therefore, we have to use this unique set of embeddings to support two different tasks on each dataset. In contrast, our IRWE method can support the inductive inference of identity and position embeddings, simultaneously generating two sets of embeddings. Therefore, we used the two sets of embeddings to support different tasks, respectively.

To further validate the challenge of simultaneously capturing node identities and positions in one embedding space, we introduced an additional baseline (denoted as [n2v||s2v]) by concatenating node2vec and struc2vec.

Note that we consider the *unsupervised* network embedding as stated in Section 3. There exist several *supervised* inductive methods (e.g., GAT (Veličković et al., 2018), GIN (Xu et al., 2019), ID-GNN (You et al., 2021), DE-GNN (Li et al., 2020), DEMO-Net (Wu et al., 2019), and SAT (Chen et al., 2022)) that do not provide unsupervised training objectives in their original designs. To ensure the fairness of comparison, these supervised baselines are not included in our experiments.

All the experiments were conducted on a server with AMD EPYC 7742 64-Core CPU, 512GB main memory, and one NVIDIA A100 GPU (80GB memory). We used the official code or public implementations of all the baselines and tuned parameters to report their best performance. On each dataset, we set the same embedding dimensionality for all the methods. For simplicity, we leave details of layer configurations and parameter settings in Appendix C.

5.2 Evaluation of Transductive Embedding Inference

We first evaluated the transductive embedding inference of all the methods on the first and second types of datasets. For the two classification tasks (i.e., multi-label node position and multi-class node identity classification), we randomly sampled $T \in \{20\%, 40\%, 60\%, 80\%\}$ and 10% of the nodes to form the training and validation sets with the remaining nodes as the test set on each dataset. We repeated the data splitting for node classification 10 times and reported the mean quality metric w.r.t. the 10 splits for each method.

Evaluation results of the transductive embedding inference are shown in Tables 5 and 6, where metrics of IRWE are in **bold** or underlined if they perform the best or within top-3.

For transductive baselines, identity embedding approaches (i.e., struc2vec, struc2gauss, and PhN-HK) and position embedding methods (i.e., node2vec, GraRep, PhN-PPMI) are in groups with top clustering perfor-

Table 5: Transductive Embedding Inference w.r.t. Node Position Classification and Node Identity Clustering on the First Type of Datasets

	PPI					***Wiki***					***BlogCatalog***				
	F1-score↑ (%)				**Ncut↓**	**F1-score↑ (%)**				**Ncut↓**	**F1-score↑ (%)**				**Ncut↓**
	20%	40%	60%	80%		20%	40%	60%	80%		20%	40%	60%	80%	
node2vec	17.79	19.15	20.16	21.58	45.18	47.43	51.05	52.25	53.87	38.89	37.20	39.45	40.45	41.58	36.82
GraRep	17.94	20.54	22.00	23.49	39.92	49.87	53.33	54.18	55.09	37.12	30.83	33.58	34.71	35.68	34.32
PhN-PPMI	20.17	22.34	23.64	24.84	45.31	46.11	49.04	50.35	51.22	38.88	38.86	40.97	41.69	42.71	36.21
struc2vec	7.70	7.99	8.04	8.47	30.51	40.70	41.14	41.17	41.34	30.96	14.67	15.09	15.28	14.79	30.47
struc2gauss	10.59	11.40	11.91	12.59	38.01	41.09	41.06	40.86	41.13	27.66	17.16	17.21	17.28	16.95	34.41
PhN-HK	9.60	9.57	9.44	9.95	31.52	41.54	41.58	41.35	41.77	29.47	17.28	17.33	17.32	17.04	34.45
[n2v||s2v]	14.29	14.67	14.66	14.38	31.99	38.95	39.75	41.85	44.37	32.32	26.94	28.75	31.34	33.75	31.14
GraSAGE	6.59	6.29	7.12	6.88	36.00	41.14	41.06	40.82	40.89	30.71	16.79	16.77	16.70	16.56	34.28
DGI	10.98	12.37	13.36	14.24	45.35	42.63	43.44	43.91	44.33	36.85	19.24	20.81	21.92	22.22	33.35
GraMAE	11.58	12.76	13.76	14.00	37.72	42.01	42.52	42.87	43.32	25.14	19.29	20.38	20.57	21.02	28.35
GraMAE2	9.63	10.40	11.26	11.52	45.26	41.85	42.04	41.73	42.34	38.26	17.76	18.14	18.23	18.29	35.56
P-GNN	11.70	12.71	13.71	13.75	39.74	43.16	44.38	44.92	45.88	37.31	19.29	20.64	21.39	21.43	34.75
CSGCL	14.93	16.14	17.13	17.81	41.66	42.77	43.39	43.47	44.06	25.94	18.91	19.25	19.30	19.42	30.58
GraLSP	9.08	9.35	9.37	9.95	29.76	41.05	41.00	40.62	41.40	11.00	16.65	17.50	17.44	17.58	23.46
SPINE	8.36	9.07	9.97	10.41	44.49	40.92	40.87	40.59	40.50	38.89	16.25	16.51	16.50	16.39	37.47
GAS	9.25	9.88	10.59	11.15	39.47	41.29	41.40	41.44	42.24	34.59	18.07	18.47	18.76	18.94	34.11
SANNE	7.77	8.18	8.05	9.57	46.87	41.07	41.08	41.01	41.56	38.35	16.56	16.77	16.70	16.72	37.10
UGFormer	6.57	6.04	6.31	6.31	2.30	41.15	41.07	40.81	40.88	21.16	16.73	16.84	16.76	16.53	28.01
IRWE	19.63	**22.75**	**24.20**	**25.88**	**28.92**	**52.02**	**54.29**	**54.94**	**56.20**	**9.85**	**38.99**	**41.42**	**41.86**	**42.76**	24.58

Table 6: Transductive Embedding Inference w.r.t. Node Identity Classification and Community Detection on the Second Type of Datasets

	USA					***Europe***					***Brazil***				
	F1-score↑ (%)				**Mod↑**	**F1-score↑ (%)**				**Mod↑**	**F1-score↑ (%)**				**Mod↑**
	20%	40%	60%	80%	(%)	20%	40%	60%	80%	(%)	20%	40%	60%	80%	(%)
node2vec	47.02	50.42	53.16	53.36	25.88	36.19	39.65	41.98	41.46	7.43	32.50	32.12	39.75	37.14	11.76
GraRep	52.52	57.86	61.93	62.01	27.54	39.18	44.32	48.09	44.87	11.48	34.89	40.45	43.50	42.14	19.76
PhN-PPMI	50.28	54.31	57.45	57.05	25.03	36.58	40.54	44.21	43.17	7.26	32.60	36.51	39.00	40.00	9.12
struc2vec	56.85	58.97	59.91	62.52	0.38	51.85	53.93	57.27	57.31	-5.61	65.43	71.66	75.25	74.29	-1.43
struc2gauss	60.88	61.89	62.32	64.36	3.27	49.50	53.38	55.53	56.34	-6.49	68.69	72.72	75.50	73.57	-3.31
PhN-HK	58.64	60.97	62.43	63.19	13.14	50.32	52.13	54.79	56.09	-6.01	61.84	68.78	74.75	69.28	-5.19
[n2v||s2v]	54.02	55.69	58.79	57.05	2.91	48.25	52.23	54.79	52.43	-5.22	59.78	65.75	64.75	60.71	2.28
GraSAGE	45.49	50.06	54.70	55.37	1.55	34.23	46.31	45.70	46.82	-0.71	35.86	39.09	54.00	57.85	2.93
DGI	54.62	57.78	58.85	59.49	3.45	44.23	48.05	52.39	49.02	-4.78	36.19	41.36	48.25	47.85	9.18
GraMAE	58.86	62.33	64.62	64.11	5.86	45.19	49.10	52.72	49.26	1.70	44.56	55.00	63.00	66.42	3.18
GraMAE2	55.91	56.90	57.67	59.07	18.73	35.97	40.09	43.96	42.92	7.03	36.63	38.93	39.00	37.85	5.95
P-GNN	58.55	61.29	62.54	61.34	21.48	45.33	47.06	51.65	50.00	0.29	46.08	50.15	49.75	52.85	1.78
CSGCL	59.49	59.41	61.79	61.09	21.14	46.87	53.03	56.36	52.68	-8.61	38.91	44.39	48.50	52.14	13.04
GraLSP	57.89	58.87	60.58	61.84	2.72	42.59	47.66	45.70	51.70	0.65	43.15	52.12	61.25	64.28	0.32
SPINE	35.07	37.42	40.64	40.25	2.16	25.12	25.82	23.71	30.00	-0.08	23.36	21.51	19.25	23.57	0.05
GAS	60.46	62.97	64.48	64.45	22.45	51.56	52.18	55.12	58.04	5.20	67.06	69.09	72.75	74.28	1.51
SANNE	54.95	56.86	58.15	61.01	14.59	44.63	50.25	54.46	49.51	6.21	40.43	45.61	51.25	51.43	5.90
UGFormer	51.61	53.85	53.95	55.88	0.78	36.12	43.83	45.79	48.29	1.35	35.22	39.70	47.00	46.42	2.65
IRWE	58.88	**64.71**	**66.30**	**67.31**	**31.24**	**53.13**	**55.82**	**58.42**	**60.48**	**17.74**	**69.24**	71.67	**75.50**	**75.71**	**21.26**

mance (in terms of NCut and modularity) on the first and second types of datasets, respectively. Since prior studies have demonstrated the ability of these transductive baselines to capture node identities or positions, the evaluation results can *validate our motivation of using node identity clustering and community detection as the unsupervised tasks to evaluate the quality of identity and position embeddings.* Our node identity clustering results also validate **Hypothesis 1** that *the high-order degree features* $\{\delta(v)\}$ *can effectively encode node identity information.*

On each dataset, most baselines can only achieve relatively high performance for one task w.r.t. identity or position embedding. It indicates that *most existing embedding methods can only capture either node identities or positions.*

In most cases, [n2v||s2v] outperforms neither (i) node2vec and PhN-PPMI for tasks w.r.t. node positions nor (ii) struc2vec and PhN-HK for those w.r.t. node identities. It implies that *the simple integration of the two types of embeddings may even damage the quality of capturing node identities or positions.* Therefore, *it is challenging to preserve both properties in a common embedding space.*

For tasks w.r.t. each type of embedding, conventional transductive baselines can achieve much better performance than most of the advanced inductive baselines. One possible reason is that existing inductive

Table 7: Inductive Inference for New Nodes within a Graph and across Graphs

	PPI		*Wiki*		*BlogCatalog*		*USA*		*Europe*		*Brazil*		*PPIs*	
	F1↑ (%)	**Ncut↓**	**F1↑** (%)	**Ncut↓**	**F1↑** (%)	**Ncut↓**	**F1↑** (%)	**Mod↑** (%)	**F1↑** (%)	**Mod↑** (%)	**F1↑** (%)	**Mod↑** (%)	**Mod↑** (%)	**Ncut↓**
GraSAGE	7.35	36.13	40.71	28.86	16.30	26.50	57.81	0.88	52.68	0.02	71.42	2.18	3.90	6.69
DGI	14.64	45.18	44.16	36.89	22.76	33.71	65.54	3.90	52.19	0.69	58.57	3.65	3.52	8.31
GraMAE	14.54	38.58	43.78	24.73	20.94	27.78	66.72	1.62	54.15	1.11	64.29	3.42	2.82	7.40
GraMAE2	11.81	45.28	41.27	37.88	19.05	35.94	59.83	9.56	46.34	4.46	42.86	5.34	3.68	8.14
PGNN	14.29	42.91	43.74	37.57	22.06	35.33	59.49	16.15	51.70	1.36	61.42	3.93	7.83	7.95
CSGCL	16.13	41.46	43.96	25.54	19.30	31.14	63.36	18.17	56.59	-7.81	61.43	5.81	-0.26	5.88
GraLSP	6.39	47.24	40.62	31.78	16.51	37.39	25.21	-0.34	44.39	0.12	38.57	-0.80	0.88	8.48
SPINE	9.12	47.21	40.80	38.95	16.87	37.45	44.87	0.76	24.88	0.16	37.14	0.41	0.38	8.63
GAS	11.50	39.33	41.84	34.44	18.94	33.89	64.87	23.05	56.59	3.51	68.57	4.27	-2.10	7.15
SANNE	5.19	45.58	40.86	33.88	16.39	34.11	25.71	0.01	26.34	-0.01	25.13	-0.01	1.43	8.22
UGFormer	5.59	34.70	40.71	21.39	16.23	27.84	59.83	2.03	45.85	0.73	62.86	1.95	-0.83	5.43
IRWE	**18.29**	**32.95**	**47.32**	**15.93**	**0.28**	<u>27.72</u>	**68.40**	**25.80**	**59.02**	**11.65**	**74.29**	**12.83**	**11.41**	**4.62**

embedding approaches usually rely on the availability of node attributes. However, there are complicated correlations between graph topology and attributes as discussed in Section 1. Our results imply that *the embedding quality of some inductive baselines is largely affected by their attribute inputs. Some standard settings for the case without available attributes (e.g., using bucket one-hot degree encodings as attribute inputs) cannot help derive informative identity or position embeddings.*

Our IRWE method achieves the best quality for both identity and position embedding in most cases. It indicates that *IRWE can jointly derive informative identity and position embeddings in a unified framework.*

5.3 Evaluation of Inductive Embedding Inference

We further consider the inductive embedding inference (i) *for new unseen nodes within a graph* and (ii) *across graphs*, which were evaluated on the (i) first two types of datasets (i.e., *PPI*, *Wiki*, *BlogCatalog*, *USA*, *Europe*, and *Brazil*) and (ii) *PPIs*, respectively. Note that we could only evaluate the quality of inductive methods because transductive baselines cannot handle the inductive embedding inference.

For the *inductive inference within a graph*, we randomly selected 80%, 10%, and 10% of nodes on each single graph to form the training, validation, and test sets (denoted as $\mathcal{V}_{trn}$, $\mathcal{V}_{val}$, and $\mathcal{V}_{tst}$), where $\mathcal{V}_{val}$ and $\mathcal{V}_{tst}$ represent sets of new nodes not observed in $\mathcal{V}_{trn}$, with $\mathcal{V}_{trn} \cap \mathcal{V}_{val} = \emptyset$, $\mathcal{V}_{trn} \cap \mathcal{V}_{tst} = \emptyset$, and $\mathcal{V}_{val} \cap \mathcal{V}_{tst} = \emptyset$. The embedding model of each inductive method was optimized only on the topology induced by $\mathcal{V}_{trn}$. When validating and testing a method using the node classification task, embeddings w.r.t. $\mathcal{V}_{trn}$ and $\mathcal{V}_{trn} \cup \mathcal{V}_{val}$ were respectively used to train the downstream logistic regression. We repeated the data splitting 5 times, with the partitions of $\{\mathcal{V}_{trn}, \mathcal{V}_{val}, \mathcal{V}_{tst}\}$ shared by all the methods. Moreover, we independently conducted the model optimization and evaluation on each split and reported the mean quality metric w.r.t. the 5 splits for each method.

The *inductive inference across graphs* is usually conducted on graphs from the same domain with similar underlying properties (Hamilton et al., 2017; Qin et al., 2023a) (e.g., protein-protein interactions of different human tissues). We sampled 3 graphs from *PPIs* denoted as $\mathcal{G}_{trn}$, $\mathcal{G}_{val}$, and $\mathcal{G}_{tst}$, which were used for training, validation, and testing. For each method, we first optimized the embedding model on $\mathcal{G}_{trn}$. To validate or test the model, we derived inductive embeddings w.r.t. the new topology of $\mathcal{G}_{val}$ or $\mathcal{G}_{tst}$ and obtained clustering results for evaluation by applying KMeans. This procedure (i.e., graph sampling, model optimization, and evaluation) was repeated 5 times, where 15 graphs were sampled. Finally, the mean quality metric w.r.t. the 5 data splits was reported.

Evaluation results of the inductive embedding inference are depicted in Table 7, where metrics of IRWE are in **bold** or <u>underlined</u> if they perform the best or within top-3.

In Table 7, IRWE achieves the best quality for tasks w.r.t. both identity and position embedding in most cases. In particular, the quality metrics of IRWE are significantly better than other inductive baselines, whose inductiveness relies on the availability of node attributes. Our results further demonstrate that *IRWE can effectively support the inductive inference of identity and position embeddings, simulataneously generating two sets of informative embeddings without relying on the availability and aggregation of graph attributes.*

Table 8: Ablation Study w.r.t. Node Position Classification and Node Identity Clustering on *PPI* as well as Node Identity Classification and Community Detection on *USA*.

	PPI		***USA***	
	F1-Score↑ (%)	**Ncut**↓	**F1-Score**↑ (%)	**Mod**↑ (%)
IRWE	**25.88**	**28.94**	**67.31**	**31.24**
(1) w/o loss $\mathcal{L}_{\text{reg}-\varphi}$	25.43	30.14	66.55	30.82
(2) w/o input $\{\tilde{s}(v)\}$	24.76	29.68	65.21	29.31
(3) w/o input $\{\delta(v)\}$	25.14	30.61	67.07	31.08
(4) w/o loss $\mathcal{L}_{\text{reg}-\psi}$	25.65	36.02	45.79	30.11
(5) w/o input $\{\psi(v)\}$	24.95	29.28	65.79	30.44
(6) w/o input $\{\pi_g(v)\}$	25.08	29.62	65.79	29.45
(7) w/o ROut$(\cdot)$	13.39	29.39	66.47	-0.76
(8) w/o loss $\mathcal{L}_{\gamma}$	22.43	29.42	65.88	23.65
(9) base stat $\{\tilde{s}(v)\}$	–	46.05	56.63	–
(10) base stat $\{\delta(v)\}$	–	34.06	63.94	–
(11) base stat $\{\pi_g(v)\}$	17.52	–	–	21.85
(12) based stat $\mathbf{C}$ (SVD)	22.60	–	–	12.15

5.4 Ablation Study

In our ablation study, we respectively removed some key components from the IRWE model to explore their effectiveness for ensuring the high embedding quality of our method.

For the *identity embedding module*, we considered the (i) AW embedding regularization loss $\mathcal{L}_{\text{reg}-\varphi}$ (14), (ii) AW statistic inputs $\{\tilde{s}(v)\}$, (iii) high-order degree feature inputs $\{\delta(v)\}$, and (iv) identity embedding regularization loss $\mathcal{L}_{\text{reg}-\psi}$ (15). In cases (i) and (iv), identity embeddings were only optimized via one loss (i.e., $\mathcal{L}_{\text{reg}-\psi}$ or $\mathcal{L}_{\text{reg}-\varphi}$).

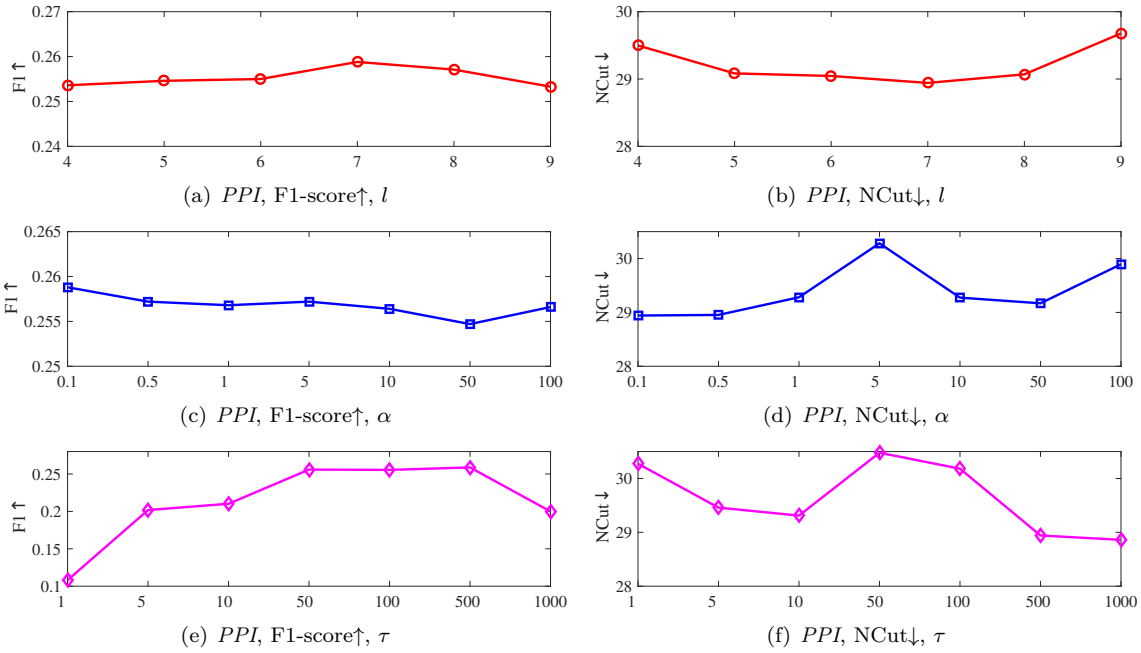
For the *position embedding module*, we checked the effectiveness of the (v) identity embedding inputs $\{\psi(v)\}$, (vi) global position encoding inputs $\{\pi_g(v)\}$, (vii) attentive readout function ROut$(\cdot)$ described in (11), and (viii) reconstruction loss $\mathcal{L}_{\gamma}$ (16) of position embeddings. In case (v), the two modules of IRWE were independently optimized. For case (vii), we used the average operation to replace the attention unit ROut$(\cdot)$ (i.e., simply averaging the representations in $\bar{t}^{(v)}$). For case (viii), we replaced the contrastive statistics $\mathbf{C}$ in (16) with adjacency matrix $\mathbf{A}$ (i.e., reconstructing $\mathbf{A}$ instead of $\mathbf{C}$ when optimizing position embeddings).

We also used some induced statistics as baselines by directly feeding them to the downstream tasks with logistic regression and KMeans for classification and clustering. Concretely, we evaluated the quality of (ix) AW statistics $\{\tilde{s}(v)\}$ and (x) degree features $\{\delta(v)\}$ to capture node identities. In contrast, we checked the quality of (xi) global position encodings $\{\pi_g(v)\}$ and (xii) contrastive statistics $\mathbf{C}$ for node positions. In case (xii), we derived representations with the same dimensionality as other embedding methods by applying SVD to $\mathbf{C}$.

As a demonstration, we report results of transductive embedding inference on *PPI* and *USA* (with 80% of nodes sampled as the training set for classification) in Table 8. According to our results, $\mathcal{L}_{\text{reg}-\psi}$ *is essential for identity embedding learning*, since there are significant quality declines for node identity clustering and classification in case (iv). ROut$(\cdot)$ *and* $\mathcal{L}_{\gamma}$ *are key components to capture node positions* due to the significant quality declines for node position classification and community detection in cases (vii) and (viii). In addition, *all the remaining components can further enhance the ability to capture node identities and positions. The joint optimization of identity and position embeddings can also improve the quality of one another.*

5.5 Parameter Analysis

We tested the effects of (i) RW length l, (ii) α in loss (13), and (iii) temperature parameter τ in loss (16). Concretely, we set $l \in \{4, 5, \cdots, 9\}$, $\alpha \in \{0.1, 0.5, 1, 5, 10, 50, 100\}$, and $\tau \in \{1, 5, 10, 50, 100, 500, 1000\}$. Example parameter analysis results of the transductive embedding inference on *PPI* and *USA* (with 80% of nodes sampled as the training set for classification) are illustrated in Fig. 7 and 8.

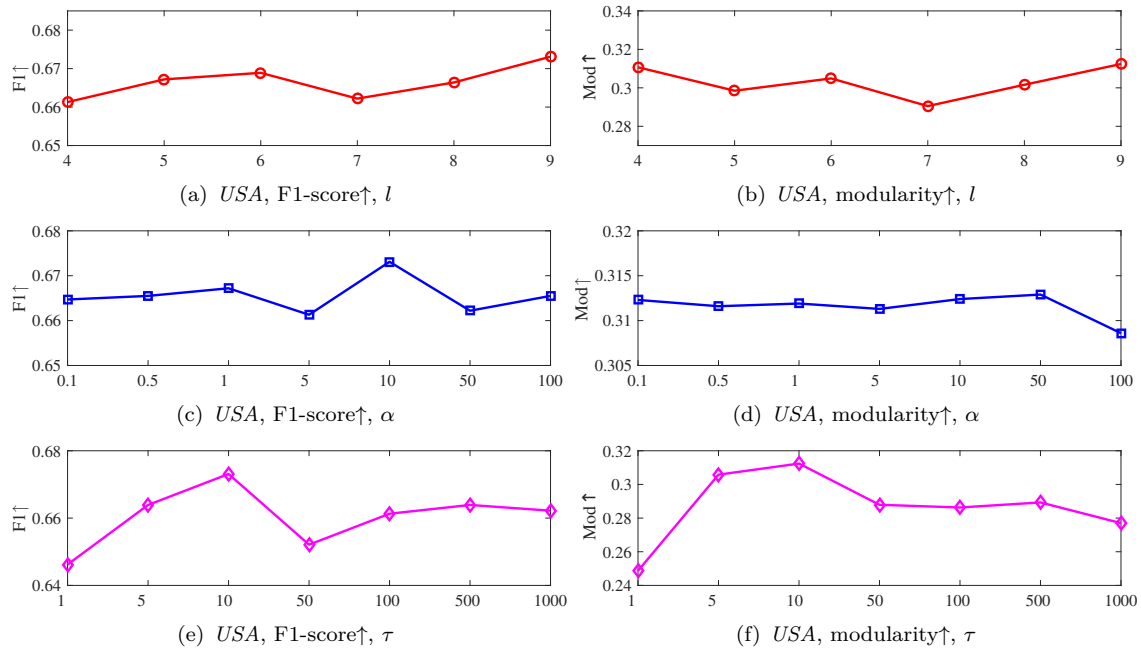


(a) *PPI*, F1-score↑, l (b) *PPI*, NCut↓, l

(c) *PPI*, F1-score↑, α (d) *PPI*, NCut↓, α

(e) *PPI*, F1-score↑, τ (f) *PPI*, NCut↓, τ

Figure 7: Parameter analysis w.r.t. l, α, and τ on *PPI* in terms of F1-score↑ (node position classification) and NCut↓ (node identity clustering).



(a) *USA*, F1-score↑, l (b) *USA*, modularity↑, l

(c) *USA*, F1-score↑, α (d) *USA*, modularity↑, α

(e) *USA*, F1-score↑, τ (f) *USA*, modularity↑, τ

Figure 8: Parameter analysis w.r.t. l, α, and τ on *USA* in terms of F1-score↑ (node identity classification) and modularity↑ (community detection).

According to the parameter analysis results, the quality of both types of embeddings is not sensitive to the settings of l. Compared with position embeddings, the quality of identity embeddings is more sensitive to α (e.g., in terms of F1-score of node classification on *USA* and NCut of node identity clustering on *PPI*).

Table 9: Evaluation Results on Attributed Graphs for the Validation of Inconsistency of Attributes.

	Cornell		**Texas**		**Washington**		**Wisconsin**	
	Mod↑ (%)	**NCut↓**	**Mod↑** (%)	**NCut↓**	**Mod↑** (%)	**NCut↓**	**Mod↑** (%)	**NCut↓**
node2vec	**56.93**	3.18	**45.99**	3.10	**44.94**	3.59	**54.73**	2.97
struc2vec	-9.50	**1.53**	-11.37	**1.17**	-9.33	**0.71**	-8.94	**1.34**
att-emb	-0.09	3.76	-0.01	3.75	-2.30	3.84	-3.54	3.75
[n2v||att]	50.80	3.38	35.26	3.12	44.05	3.50	54.02	3.13
[s2v||att]	-5.76	1.81	-12.77	1.33	-2.81	1.00	-9.58	1.35

Furthermore, the settings of τ would significantly affect the quality of the two types of embeddings. The recommended parameter settings of IRWE are given in the Appendix C.

5.6 Verification of the Inconsistency between Graph Topology and Attributes

To demonstrate the possible inconsistency of graph attributes for identity and position embedding discussed in Section 1, we conducted additional evaluation experiments on four public attributed graphs (i.e., *Cornell*, *Texas*, *Washington*, and *Wisconsin*) from the WebKB[1] dataset. For each graph, we first extracted the largest connected component from its topology. After the pre-processing, we have $(N, E, M, K) = (183, 227, 1703, 5)$, $(183, 279, 1703, 5)$, $(215, 365, 1703, 5)$, and $(251, 450, 1703, 5)$ for *Cornell*, *Texas*, *Washington*, and *Wisconsin*, where N, E, and K are numbers of nodes, edges, and clusters; M denotes the dimensionality of node attributes.

We then applied node2vec and struc2vec, which are typical position and identity embedding baselines as described in Table 4, to the extracted topology of each graph, where we set embedding dimensionality $d = 64$. Furthermore, we derived special attribute embeddings (denoted as att-emb) with the same dimensionality by applying SVD to node attributes. In this setting, we have three baseline methods (e.g., node2vec, struc2vec, and att-emb). To simulate the incorporation of attributes, we also concatenated att-emb with node2vec and struc2vec, forming another two baselines denoted as [n2v||att] and [s2v||att]. The unsupervised community detection and node identity clustering (with metrics of modularity and NCut) were adopted as the downstream tasks for position and identity embedding, respectively.

The evaluation results are depicted in Table 9, where att-emb outperforms neither (i) node2vec for community detection nor (ii) struc2vec for node identity clustering; the concatenation of att-emb cannot further improve the embedding quality of node2vec and struc2vec. The results imply that (i) *attributes may fail to capture both node positions and identities*; (ii) *the simple integration of attributes may even damage the quality of position and identity embeddings.*

6 Conclusion

In this paper, we considered unsupervised network embedding and explored the possibility of a unified framework for the joint optimization and inductive inference of identity and position embeddings without relying on the availability and aggregation of graph attributes. An IRWE method was proposed, which combines multiple attention units with different choices of key, query, and value to handle RWs on graph topology. In particular, we demonstrated that AW derived from RW and induced statistics can not only (i) be features shared by all possible nodes and graphs to support inductive inference but also (ii) characterize node identities to derive identity embeddings. Moreover, we also showed the intrinsic relation between the two types of embeddings. Based on this relation, the derived identity embeddings can be used for the inductive inference of position embeddings. Experiments on public datasets validated that IRWE can achieve superior quality compared with various baselines for the transductive and inductive inference of identity and position embeddings. We conclude this paper by discussing some future directions.

- In this study, we focused on network embedding where topology is the only available information source without attributes, due to the complicated correlations between the two sources (Qin et al.,

[1] https://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

2018; Li et al., 2019; Wang et al., 2020; Qin & Lei, 2021) discussed in Section 1. In our future work, we intend to explore the *adaptive incorporation of attributes*. Concretely, when attributes carry characteristics consistent with topology, one can fully utilize attribute information to enhance the embedding quality. In contrast, when there is inconsistent noise in attributes, we need to adaptively control the effect of attributes to avoid unexpected quality degradation.

- In addition to mapping each node to a low-dimensional representation (a.k.a. node-level embedding) as defined in Section 3, network embedding also includes the representation learning of each single graph (a.k.a. graph-level embedding). In our future research, we also plan to extend IRWE to the graph-level embedding and evaluate the embedding quality for some graph-level downstream tasks (e.g., graph classification). To analyze the relations of graph-level embeddings to identity and position embeddings is also our next focus.

- As described in Section 4.3, the optimization of IRWE adopts the standard full-batch setting, where we derive statistics or embeddings w.r.t. all the nodes $\mathcal{V}$ when computing the training losses. This setting may not be scalable to graphs with large numbers of nodes. Inspired by existing studies of scalable GNNs (Zhang et al., 2022; Liu et al., 2023), we intend to explore a scalable optimization strategy based on the mini-batch setting in our future research.

References

Rosa I Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine learning*, 63:161–182, 2006.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information & Knowledge Management*, pp. 891–900, 2015.

Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *Proceedings of the 2022 International Conference on Machine Learning*, pp. 3469–3489, 2022.

Han Chen, Ziwen Zhao, Yuhua Li, Yixiong Zou, Ruixuan Li, and Rui Zhang. Csgcl: Community-strength-enhanced graph contrastive learning. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pp. 2059–2067, 2023.

Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1320–1329, 2018.

Yu Gao, Meng Qin, Yibin Ding, Li Zeng, Chaorui Zhang, Weixi Zhang, Wei Han, Rongqian Zhao, and Bo Bai. Raftgp: Random fast graph partitioning. In *2023 IEEE High Performance Extreme Computing Conference*, pp. 1–7, 2023.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 855–864, 2016.

Junliang Guo, Linli Xu, and Jingchang Liu. Spine: Structural identity preserved inductive network embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2399–2405, 2019.

Xuan Guo, Wang Zhang, Wenjun Wang, Yang Yu, Yinghui Wang, and Pengfei Jiao. Role-oriented graph auto-encoder guided by structural information. In *Proceedings of the 25th International Conference on Database Systems for Advanced Applications*, pp. 466–481, 2020.

William Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 2017 Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 594–604, 2022.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the 2023 ACM Web Conference*, pp. 737–746, 2023.

Sergey Ivanov and Evgeny Burnaev. Anonymous walk embeddings. In *Proceedings of the 2018 International Conference on Machine Learning*, pp. 2186–2195, 2018.

Yilun Jin, Guojie Song, and Chuan Shi. Gralsp: Graph neural networks with local structural patterns. In *Proceedings of the 2020 AAAI Conference on Artificial Intelligence*, pp. 4361–4368, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Kai Lei, Meng Qin, Bo Bai, and Gong Zhang. Adaptive multiple non-negative matrix factorization for temporal link prediction in dynamic networks. In *Proceedings of the 2018 ACM SIGCOMM Workshop on Network Meets AI & ML*, pp. 28–34, 2018.

Kai Lei, Meng Qin, Bo Bai, Gong Zhang, and Min Yang. Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks. In *Proceedings of the 2019 IEEE Conference on Computer Communications*, pp. 388–396, 2019.

Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. *Proceedings of the 2020 Advances in Neural Information Processing Systems*, 33:4465–4478, 2020.

Wei Li, Meng Qin, and Kai Lei. Identifying interpretable link communities with user interactions and messages in social networks. In *Proceedings of the 2019 IEEE International Conference on Parallel & Distributed Processing with Applications*, pp. 271–278, 2019.

Zirui Liu, Chen Shengyuan, Kaixiong Zhou, Daochen Zha, Xiao Huang, and Xia Hu. Rsc: accelerate graph neural networks training via randomized sparse computations. In *International Conference on Machine Learning*, pp. 21951–21968, 2023.

Silvio Micali and Zeyuan Allen Zhu. Reconstructing markov processes from independent and anonymous experiments. *Discrete Applied Mathematics*, 200:108–122, 2016.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

Dai Quoc Nguyen, Tu Dinh Nguyen, and Dinh Phung. A self-attention network based node embedding model. In *Proceedings of the 2021 Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 364–377, 2021.

Dai Quoc Nguyen, Tu Dinh Nguyen, and Dinh Phung. Universal graph transformer self-attention networks. In *Companion Proceedings of the Web Conference 2022*, pp. 193–196, 2022.

Yulong Pei, Xin Du, Jianpeng Zhang, George Fletcher, and Mykola Pechenizkiy. struc2gauss: Structural role preserving network embedding via gaussian embedding. *Data Mining & Knowledge Discovery*, 34(4): 1072–1103, 2020.

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 & data Mining*, pp. 701–710, 2014.

Meng Qin and Kai Lei. Dual-channel hybrid community detection in attributed networks. *Information Sciences*, 551:146–167, 2021.

Meng Qin and Dit-Yan Yeung. Temporal link prediction: A unified framework, taxonomy, and review. *ACM Computing Surveys*, 56(4):1–40, 2023.

Meng Qin, Di Jin, Kai Lei, Bogdan Gabrys, and Katarzyna Musial. Adaptive community detection incorporating topology and content in social networks. *Knowledge-Based Systems*, 161:342–356, 2018.

Meng Qin, Chaorui Zhang, Bo Bai, Gong Zhang, and Dit-Yan Yeung. Towards a better trade-off between quality and efficiency of community detection: An inductive embedding method across graphs. *ACM Transactions on Knowledge Discovery from Data*, 2023a.

Meng Qin, Chaorui Zhang, Bo Bai, Gong Zhang, and Dit-Yan Yeung. High-quality temporal link prediction for weighted dynamic graphs via inductive embedding aggregation. *IEEE Transactions on Knowledge and Data Engineering*, 2023b.

Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 385–394, 2017.

Ryan A Rossi, Di Jin, Sungchul Kim, Nesreen K Ahmed, Danai Koutra, and John Boaz Lee. On proximity and structural role-based embeddings in networks: Misconceptions, techniques, and applications. *ACM Transactions on Knowledge Discovery from Data*, 14(5):1–37, 2020.

Balasubramaniam Srinivasan and Bruno Ribeiro. On the equivalence between positional node embeddings and structural graph representations. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 2017 Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics & Computing*, 17(4):395–416, 2007.

Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proceedings of the 2017 AAAI Conference on Artificial Iintelligence*, pp. 203–209, 2017.

Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. Am-gcn: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1243–1253, 2020.

Jun Wu, Jingrui He, and Jiejun Xu. Demo-net: Degree-specific graph neural networks for node and graph classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 406–415, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks & Learning Systems*, 32(1):4–24, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Algorithm 6: RW Sampling Starting from a Node

Input: topology $(\mathcal{V}, \mathcal{E})$; target node v; RW length l; number of samples n_S
Output: set of sampled RWs $\mathcal{W}^{(v)}$

```
1 W^(v) ← ∅ //Initialize W^(v)
2 for sample_count from 1 to n_S do
3     v_s ← v and w ← (v_s) //Initialize current RW w
4     while |w| ≤ (l + 1) do
5         randomly sample a node v_t from v_s's neighbors
6         append v_t to w
7         v_s ← v_t
8     add w to W^(v)
```

Algorithm 7: Transductive Inference

Input: RWs $\{\mathcal{W}^{(v)}, \mathcal{W}_I^{(v)}\}$, AW lookup table $\tilde{\Omega}_l$, & statistics $\{\tilde{s}(v), \delta(v), \pi_g(v)\}$ saved in model optimization; inference topology $(\mathcal{V}, \mathcal{E})$
Output: *transductive* embddings $\{\psi(v)\}$ & $\{\gamma(v)\}$ w.r.t. $\mathcal{V}$

```
1 get {ψ(v)} based on {Ω̃_l, s̃(v), δ(v)} w.r.t. V
2 get {γ(v)} based on {ψ(v), π_g(v), π_l(j), W_I^(v)} w.r.t. V
```

Yang Yang, Jie Tang, Cane Wing-ki Leung, Yizhou Sun, Qicong Chen, Juanzi Li, and Qiang Yang. RAIN: social role-aware information diffusion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 367–373, 2015.

Dongsheng Ye, Hao Jiang, Ying Jiang, Qiang Wang, and Yulin Hu. Community preserving mapping for network hyperbolic embedding. *Knowledge-Based Systems*, 246:108699, 2022.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *Proceedings of th 2019 International Conference on Machine Learning*, pp. 7134–7143, 2019.

Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proceedings of the 2012 AAAI Conference on Artificial Intelligence*, pp. 10737–10745, 2021.

Wentao Zhang, Yu Shen, Zheyu Lin, Yang Li, Xiaosen Li, Wen Ouyang, Yangyu Tao, Zhi Yang, and Bin Cui. Pasca: A graph neural architecture search system under the scalable paradigm. In *Proceedings of the 2022 ACM Web Conference*, pp. 1817–1828, 2022.

Jing Zhu, Xingyu Lu, Mark Heimann, and Danai Koutra. Node proximity is all you need: Unified structural and positional node and graph embedding. In *Proceedings of the 2021 SIAM International Conference on Data Mining*, pp. 163–171, 2021.

A Detailed Algorithms

The RW sampling procedure starting from a node is summarized in Algorithm 6, which uniformly sample the next node v_t from the neighbors of each source node v_s (i.e., line 5).

Algorithm 7 summarizes the *transductive inference procedure* of IRWE, where the RWs $\{\mathcal{W}^{(v)}, \mathcal{W}_I^{(v)}\}$, AW lookup table $\tilde{\Omega}_l$, and statistics $\{\tilde{s}(v), \delta(v), \pi_g(v)\}$ derived and saved during the model optimization are used as the inputs. The *transudcitve inference* of identity embeddings $\{\psi(v)\}$ and position embeddings $\{\gamma(v)\}$ only includes one feedforward propagation through the model (i.e., lines 1-2).

Procedures to get *inductive* AW statistics $\{s(v)\}$, high-order degree features $\{\delta(v)\}$, and global position encodings $\{\pi_g(v)\}$, which support the *inductive inference for new nodes within a graph* (i.e., Algorithm 5), are described in Algorithms 8, 9, and 10, respectively. When deriving $\{s(v)\}$, we only compute the frequency

Algorithm 8: Inductive Derivation of AW Statistics

Input: new target node $v \in \mathcal{V}'$; sampled RWs $\mathcal{W}^{(v)}$; AW lookup table $\tilde{\Omega}_l$ reduced on old topology $(\mathcal{V}, \mathcal{E})$
Output: *inductive* AW statistic $s(v)$ w.r.t. v

1 $\tilde{\eta}_l \leftarrow |\tilde{\Omega}_l|$ //Get size of reduced AW lookup table
2 $s(v) \leftarrow [0, 0, \cdots, 0]^{\tilde{\eta}_l}$ //Initialize $s(v)$
3 **for each** $w \in \mathcal{W}^{(v)}$ **do**
4 map RW w to its corresponding AW ω
5 **if** $\omega \in \tilde{\Omega}_l$ **then**
6 get the index j of AW ω in reduced lookup table $\tilde{\Omega}_l$
7 $s(v)_j \leftarrow s(v)_j + 1$ //Update $s(v)$

Algorithm 9: Inductive Derivation of Degree Feature

Input: new target node $v \in \mathcal{V}'$; RW length l; one-hot degree encoding dimensionality e; sampled RWs $\mathcal{W}^{(v)}$; $\deg_{\min}$ & $\deg_{\max}$ in old topology $(\mathcal{V}, \mathcal{E})$
Output: *inductive* degree feature $\delta(v)$ w.r.t. v

1 $\delta(v) \leftarrow [0, 0, \cdots, 0]^{(l+1)e}$ //Initialize degree feature $\delta(v)$
2 **for each** $w \in \mathcal{W}^{(v)}$ **do**
3 **for** i **from** 0 **to** l **do**
4 $u \leftarrow w^{(i)}$ //i-th node in current RW w
5 **if** $u \in \mathcal{V}$ **then**
6 $\rho_d(u) \leftarrow [0, \cdots, 0]^e$//Initialize one-hot degree encoding
7 **if** $\deg(u) < \deg_{\min}$ **then**
8 $j \leftarrow 0$
9 **else if** $\deg(u) > \deg_{\max}$ **then**
10 $j \leftarrow (e-1)$
11 **else**
12 $j \leftarrow \left\lfloor \frac{(\deg(u) - \deg_{\min})e}{(\deg_{\max} - \deg_{\min})} \right\rfloor$
13 $\rho_d(u)_j \leftarrow 1$ //Update $\rho_d(u)$
14 $\delta(v)_{ie:(i+1)e} \leftarrow \delta(v)_{ie:(i+1)e} + \rho_d(u)$

of AWs in the lookup table $\tilde{\Omega}_l$ reduced on $(\mathcal{V}, \mathcal{E})$ (i.e., lines 5-7 of Algorithm 8) rather than all AWs. Moreover, we get $\{\delta(v)\}$ based on the one-hot degree encoding truncated by the minimum and maximum degrees of the training topology $(\mathcal{V}, \mathcal{E})$ (i.e., lines 7-12 of Algorithm 9) but not those of the inference topology $(\mathcal{V} \cup \mathcal{V}', \mathcal{E}')$. For $\pi_g(v)$, we compute truncated RW statistic $r(v)$ only w.r.t. previously observed nodes $\mathcal{V}$ (i.e., lines 4-6 of Algorithm 10) rather than $\mathcal{V}' \cup \mathcal{V}$.

The *inductive inference across graphs* is summarized in Algorithm 11. Concretely, we sample RWs $\{\mathcal{W}^{(v)}, \mathcal{W}_I^{(v)}\}$ on each new graph $(\mathcal{V}'', \mathcal{E}'')$ (i.e., line 2). Since there are no shared nodes between the training topology $(\mathcal{V}, \mathcal{E})$ and inference topology $(\mathcal{V}'', \mathcal{E}'')$, we can only incrementally compute statistics $\{\tilde{s}(v), \delta(v)\}$ based on $\{\tilde{\Omega}_l, \deg_{\min}, \deg_{\max}\}$ derived from $(\mathcal{V}, \mathcal{E})$ (i.e., lines 3-4) but compute global position encodings $\{\pi_g(v)\}$ from scratch (i.e., line 5).

B Proof of Proposition 1

For simplicity, we let $z_{ij} := \gamma(v_i)\tilde{\gamma}^T(v_j)/\tau$. To minimize the contrastive loss $\mathcal{L}_{cnr}$, one can let its partial derivative $\partial \mathcal{L}_{cnr}/z_{ij}$ w.r.t. each edge $(v_i, v_j) \in \mathcal{E}$ to 0. Note that $\sigma(x) = 1/(1+e^{-x})$ and $d\sigma(x)/dx = \sigma(x)[1-\sigma(x)]$. Therefore, we have

$$0 = \partial \mathcal{L}_{cnr}/z_{ij} = -p_{ij}(1 - \sigma(z_{ij})) + Qn_j(1 - \sigma(-z_{ij})), \tag{19}$$

Algorithm 10: Inductive Derivation of Global Position Encoding

Input: new target node $v \in \mathcal{V}'$; sampled RWs $\mathcal{W}^{(v)}$; old training node set $\mathcal{V}$; random matrix $\mathbf{\Theta} \in \mathbb{R}^{|\mathcal{V}| \times d}$
Output: *inductive* global position encoding $\pi_g(v)$ w.r.t. v

$r(v) \leftarrow [0, 0, \cdots, 0]^{|\mathcal{V}|}$ //Initialize RW stat $r(v)$
for each $w \in \mathcal{W}^{(v)}$ **do**
  for each *node* $v \in w$ **do**
    if $v \in \mathcal{V}$ **then**
      get index j of v in the training node set $\mathcal{V}$
      $r(v)_j \leftarrow r(v)_j + 1$ //Update $r(v)$
$\pi_g(v) \leftarrow r(v)\mathbf{\Theta}$ //Derive $\pi_g(v)$

Algorithm 11: Inductive Inference across Graphs

Input: optimized model parameters $\{\theta_\psi^*, \theta_\gamma^*\}$; new topology $(\mathcal{V}'', \mathcal{E}'')$; RW settings $\{l, n_S, n_I\}$; local position encodings $\{\pi_l(j)\}$; $\{\tilde{\Omega}_l, \deg_{\min}, \deg_{\max}\}$ derived in model optimization on old topology $(\mathcal{V}, \mathcal{E})$
Output: *inductive* embeddings $\{\psi(v)\}$ & $\{\gamma(v)\}$ w.r.t. $\mathcal{V}''$

for each *node* $v \in \mathcal{V}''$ **do**
  sample n_S RWs $\mathcal{W}^{(v)}$ from v w.r.t. $\mathcal{E}''$ via Algorithm 6
  get AW statistics $\tilde{s}(v)$ w.r.t. $\{\mathcal{W}^{(v)}, \tilde{\Omega}_l\}$ via Algorithm 8
  get degree feature $\delta(v)$ w.r.t. $\{\mathcal{W}^{(v)}, \mathrm{d}_{\min}, \mathrm{d}_{\max}\}$ via Algorithm 9
  get global position encoding $\pi_g(v)$ w.r.t. $\mathcal{W}^{(v)}$ via Algorithm 3
  randomly select n_I RWs $\mathcal{W}_I^{(v)}$ from $\mathcal{W}^{(v)}$
get $\{\psi(v)\}$ based on $\{\tilde{\Omega}_l, \tilde{s}(v), \delta(v)\}$ w.r.t. $\mathcal{V}''$
get $\{\gamma(v)\}$ based on $\{\psi(v), \pi_g(v), \pi_l(j), \mathcal{W}_I^{(v)}\}$ w.r.t. $\mathcal{V}''$

which can be rearranged as

$$p_{ij}\sigma(z_{ij}) - Qn_j\sigma(-z_{ij}) = p_{ij} - Qn_j. \tag{20}$$

By applying $\sigma(-x) = e^{-x}\sigma(x)$, we have

$$\begin{aligned}
&p_{ij}\sigma(z_{ij}) - Qn_j \cdot \exp\{-z_{ij}\}\sigma(z_{ij}) = p_{ij} - Qn_j \\
&\Rightarrow \tfrac{p_{ij} - Qn_j \cdot \exp\{-z_{ij}\}}{1+\exp\{-z_{ij}\}} = p_{ij} - Qn_j \\
&\Rightarrow \tfrac{p_{ij} + Qn_j - Qn_j(1+\exp\{-z_{ij}\})}{1+\exp\{-z_{ij}\}} = p_{ij} - Qn_j \\
&\Rightarrow (p_{ij} + Qn_j)\sigma(z_{ij}) = p_{ij} \\
&\Rightarrow \sigma(z_{ij}) = p_{ij}/(p_{ij} + Qn_j) \\
&\Rightarrow 1 + \exp\{-z_{ij}\} = (p_{ij} + Qn_j)/p_{ij} \\
&\Rightarrow \exp\{-z_{ij}\} = Qn_j/p_{ij}
\end{aligned} . \tag{21}$$

By taking the logarithm of both sides, we further have

$$z_{ij} = \ln p_{ij} - \ln(Qn_j). \tag{22}$$

Let $\mathbf{C} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ be an auxiliary matrix with the same definition as that in **Proposition 1**. From the perspective of matrix factorization, we can rewrite the aforementioned equation to another matrix form $\mathbf{\Gamma}\tilde{\mathbf{\Gamma}}^T/\tau = \mathbf{C}$, which is equivalent to the reconstruction loss $\mathcal{L}_\gamma$.

C Detailed Experiment Settings

The parameter settings of IRWE for the *transductive* and *inductive* embedding inference are depicted in Tables 10 and 11, where d is the embedding dimensionality; e is the dimensionality of one-hot degree encoding tfor the degree features $\{\delta(v)\}$; n_S and n_I are defined as the number of sampled RWs (i.e., $n_S := |\mathcal{W}^{(v)}|$) and number of RWs used to infer position embeddings (i.e., $n_I := |\mathcal{W}_I^{(v)}|$) for each node v; λ_ψ and λ_γ are learning

Table 10: Parameter Settings for Transductive Inference.

	$(d,\ e,\ n_S,\ n_I)$	$(\lambda_\psi,\ \lambda_\gamma)$	$(m,\ m_\psi,\ m_\gamma)$	$(l,\ \alpha,\ \tau)$
PPI	(256, 100, 1e3, 10)	(5e-4,1e-3)	(2e3, 10, 1)	(7, 0.1, 5e2)
Wiki	(256, 100, 1e3, 10)	(1e-3,1e-3)	(1e3, 5, 1)	(7, 10, 1e3)
Blog	(256, 100, 1e3, 10)	(5e-4,5e-4)	(3e3, 1, 20)	(9, 10, 10)
USA	(128, 100, 1e3, 20)	(1e-3,5e-4)	(500, 10, 1)	(9, 10, 10)
Europe	(64, 100, 1e3, 20)	(5e-4,5e-4)	(200, 1, 1)	(9, 10, 10)
Brazil	(64, 32, 1e3, 20)	(5e-4,5e-4)	(200, 1, 1)	(9, 0.1, 1e2)

Table 11: Parameter Settings for Inductive Inference.

	$(d,\ e,\ n_S,\ n_I)$	$(\lambda_\psi,\ \lambda_\gamma)$	$(m,\ m_\psi,\ m_\gamma)$	$(l,\ \alpha,\ \tau)$
PPI	(256, 100, 1e3, 10)	(5e-4,1e-4)	(1e3, 20, 1)	(7, 10, 5e2)
Wiki	(256, 100, 1e3, 10)	(1e-3,5e-4)	(1e3, 1, 1)	(7, 10, 5e2)
Blog	(256, 100, 1e3, 10)	(5e-4,5e-4)	(1e3, 20, 5)	(5, 10, 5)
USA	(128, 100, 1e3, 10)	(5e-4,5e-4)	(500, 10, 1)	(9, 10, 10)
Europe	(64, 100, 1e3, 10)	(5e-4,5e-4)	(200, 1, 1)	(9, 10, 10)
Brazil	(64, 32, 1e3, 10)	(5e-4,5e-4)	(200, 1, 1)	(9, 0.1, 1e2)
PPIs	(256, 100, 1e3, 10)	(5e-4,5e-4)	(1000, 5, 1)	(9, 10, 50)

Table 12: Layer Configurations for Transductive Inference.

Datasets	**Identity Embedding Module**					**Position Embedding Module**		
	$\mathrm{Enc}_\varphi(\cdot)$	$\mathrm{Dec}_\varphi(\cdot)$	$\mathrm{Red}_s(\cdot)$	h_ψ	$\mathrm{Reg}_\psi(\cdot)$	MLP in ReAtt$(\cdot)$	$(L_{\mathrm{tran}},\ h_{\mathrm{tran}})$	h_{rout}
PPI	l^2,128,t,d,t	d,128,t,l^2,t	$\tilde{\eta}_l$+le,2048,r,1024,r,512,r,d,r	64	d,512,t,le,t	d,d,s,d,s,d,s,d,s	(4, 64)	64
Wiki	l^2,128,t,d,t	d,128,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,d,r	64	d,512,t,le,t	d,d,s,d,s,d,s,d,s	(4, 64)	64
Blog	l^2,128,t,d,t	d,128,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,d,r	64	d,512,t,le,t	d,d,s,d,s,d,s,d,s	(5, 64)	64
USA	l^2,100,t,d,t	d,100,t,l^2,t	$\tilde{\eta}_l$+le,4096,r,2048,r,512,r,d,r	32	d,le,t	d,d,s,d,s,d,s,d,s	(4, 32)	32
Europe	l^2,64,t,d,t	d,64,t,l^2,t	$\tilde{\eta}_l$+le,4096,r,1024,r,256,r,d,r	16	d,256,t,512,t,le,t	d,d,s,d,s	(4, 16)	16
Brazil	l^2,64,t,d,t	d,64,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,128,r,d,r	16	d,128,t,le,t	d,d,s,d,s	(4, 16)	16

Table 13: Layer Configurations for Inductive Inference.

Datasets	**Identity Embedding Module**					**Position Embedding Module**		
	$\mathrm{Enc}_\varphi(\cdot)$	$\mathrm{Dec}_\varphi(\cdot)$	$\mathrm{Red}_s(\cdot)$	h_ψ	$\mathrm{Reg}_\psi(\cdot)$	MLP in ReAtt$(\cdot)$	$(L_{\mathrm{tran}},\ h_{\mathrm{tran}})$	h_{rout}
PPI	l^2,128,t,d,t	d,128,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,d,r	64	d,le,t	d,d,s,d,s,d,s,d,s	(4, 64)	64
Wiki	l^2,128,t,d,t	d,128,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,d,r	64	d,512,t,le,t	d,d,s,d,s,d,s,d,s	(4, 64)	64
Blog	l^2,128,t,d,t	d,128,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,d,r	64	d,t,512,t,le,t	d,d,s,d,s,d,s,d,s	(5, 64)	64
USA	l^2,100,t,d,t	d,100,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,d,r	32	d,512,t,le,t	d,d,s,d,s,d,s,d,s	(4, 16)	16
Europe	l^2,64,t,d,t	d,64,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,d,r	16	d,256,t,512,t,le,t	d,d,s,d,s	(4, 16)	16
Brazil	l^2,64,t,d,t	d,64,t,l^2,t	$\tilde{\eta}_l$+le,512,r,128,r,d	16	d,256,t,le,t	d,d,s,d,s	(4, 16)	16
PPIs	l^2,128,t,d,t	d,128,t,l^2,t	$\tilde{\eta}_l$+le,1024,r,512,r,d,r	64	d,512,t,le,t	d,d,s,d,s,d,s,d,s	(6, 64)	64

rates to optimize identity and position embeddings; m is the number of iterations for model optimization; in each iteration, we update identity and position embeddings m_ψ and m_γ times; l is the RW length; α and τ are hyper-parameters in the training losses.

Furthermore, Tables 12 and 13 give layer configurations for the *transductive* and *inductive* embedding inference, where $\mathrm{Enc}_\varphi(\cdot)$ and $\mathrm{Dec}_\varphi(\cdot)$ denote the AW encoder and decoder described in (2); $\mathrm{Red}_s(\cdot)$ is the feature reduction encoder defined in (3); $\mathrm{Reg}_\psi(\cdot)$ represents the identity embedding regularization unit in (5); ReAtt$(\cdot)$ is the attentive reweighting unit in (7); $\tilde{\eta}_l$ is the reduced number of AWs (i.e., $\tilde{\eta}_l = |\tilde{\Omega}_l|$); h_ψ, h_{tran}, and h_{rout} represent the numbers of attention heads in (4), transformer encoder in (10), and attentive readout function in (11); L_{tran} is defined as the number of transformer encoder layers; 't', 's', and 'r' denote the activation functions of Tanh, Sigmoid, and ReLU, respectively. For the proposed IRWE method, we recommend setting $l \in \{4, 5, \cdots, 9\}$, $\alpha \in \{0.1, 0.5, 1, 5, 10\}$, $\tau \in \{1, 5, 10, 50, 100, 500, 1000\}$, and $m_\psi, m_\gamma \in \{1, 5, 10, 20\}$.