Link Membership Inference Attacks against Unsupervised Graph Representation Learning

Xiuling Wang
Stevens Institute of Technology
Hoboken, NJ, USA
xwang193@stevens.edu

Wendy Hui Wang
Stevens Institute of Technology
Hoboken, NJ, USA
hwang4@stevens.edu

ABSTRACT

Significant advancements have been made in recent years in the field of unsupervised graph representation learning (UGRL) approaches. UGRL involves representing large graphs as low-dimensional vectors, commonly referred to as embeddings. These embeddings can be publicly released or shared with third parties for downstream analytics. However, adversaries can deduce sensitive structural information from the target graph through its embedding using various types of privacy inference attacks. This paper investigates the privacy vulnerabilities of UGRL models through the lens of *link membership inference attack* (LMIA). Specifically, an LMIA adversary aims to infer whether any two nodes are connected in the target graph from the node embeddings generated by a UGRL model. To achieve this, we propose two LMIA attacks that leverage the properties of node embeddings and various forms of adversary knowledge for inference. By conducting experiments on four state-of-the-art UGRL models using five real-world graph datasets, we demonstrate the effectiveness of the two LMIA attacks against these UGRL models. Furthermore, we conduct a comprehensive analysis to examine how varying degrees of preserved structural information in the embeddings impact the performance of LMIA. To enhance the security of UGRL models against LMIA, we design a family of defense mechanisms that perturb the least significant dimensions of embeddings. Our experimental results show that our defense mechanism achieves a favorable balance between defense effectiveness and embedding quality.

CCS CONCEPTS

• **Security and privacy**;

KEYWORDS

Unsupervised graph representation learning, membership inference attack, machine learning privacy, graph learning, trustworthy machine learning.

ACM Reference Format:
Xiuling Wang and Wendy Hui Wang. 2023. Link Membership Inference Attacks against Unsupervised Graph Representation Learning. In *Annual Computer Security Applications Conference (ACSAC '23), December 4–8, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3627106.3627115

ACSAC '23, December 4–8, 2023, Austin, TX, USA

ACM ISBN 979-8-4007-0886-2/23/12…$15.00
https://doi.org/10.1145/3627106.3627115

1 INTRODUCTION

Many complex systems, such as social networks, biological networks, and information networks, are represented as graphs. The scale of these graphs can vary from hundreds to millions, or even billions, of vertices [51]. Analytics of such graphs heavily relies on the representation methods employed.

In many practical graph learning tasks, labeled data is expensive [65], difficult to obtain [64], and potentially biased [10, 42]. For these scenarios, unsupervised graph representation learning (UGRL) offers an alternative paradigm that not only enables the use of larger (and unlabeled) datasets but has also been shown to improve model robustness [21]. Due to the empirical success of UGRL, it has found applications in a wide range of fields such as natural language processing [31], computer vision [9], and healthcare [57]. Essentially, UGRL aims to learn latent, low-dimensional vector representations of graphs in an unsupervised manner, while preserving crucial information such as the topology structure, vertex content, and other related side information. These learned representations typically take the form of node embeddings, which are subsequently utilized in various downstream analytical tasks like link prediction and node classification, facilitating their execution with ease.

One advantage of UGRL is its independence from downstream learning tasks [60]. This allows data owners to share their data with other parties, such as machine learning (ML) service providers, by providing them with embeddings suitable for various types of learning tasks [34, 38]. For instance, data owners can use Google's Embedding Projector service[1] to interactively visualize the structure of their graph. They can achieve this by uploading their locally-trained embeddings through the service's Web API.

In various applications, the input graph may contain sensitive information, such as social relations [25] or mobility traces [3]. It is occasionally presumed that sharing embeddings is a "safer" alternative to sharing raw data, given their lower-dimensional nature [48]. Nevertheless, it is crucial to recognize that node embeddings inherently encapsulate the structure and characteristics of the original graph, thereby enabling adversaries to extract sensitive private information from them.

Recent studies [11, 63] have present the vulnerability of graph representations generated by supervised learning models like Graph Neural Networks (GNNs) to a specific type of privacy attack known as *Link Membership Inference Attack* (LMIA) [19, 56]. LMIA is designed to deduce the presence of specific links (edges) within the

[1] https://projector.tensorflow.org/

	Target model		Adversary knowledge
	Supervised	Unsupervised	
[63]	GNN		Graph embedding
[11]	GNN		Node embedding + Auxiliary subgraph (mandatory)
Ours		DeepWalk, node2vec, LINE, GAE	Node embedding + Shadow graph (optional)

Table 1: Comparison between our work and the existing LMIAs against graph representation.

training graph of the target GNN model, using either access to the entire graph embedding [63] or the node embeddings produced by the target model itself [11]. However, prior research has not delved into the vulnerabilities of the node embeddings generated by *Unsupervised Graph Representation Learning* (UGRL) methods to LMIAs. Notably, none of the existing methods for LMIAs against graph representations [11, 19] can be readily adapted to UGRL models. These studies either presume the availability of an auxiliary subgraph for the attack [11] or employ graph embeddings but not node embeddings for the attack [63]. In contrast to these approaches, we investigate a more realistic scenario where the adversary only has access to node embeddings and is not equipped with any prior knowledge about the target graph. Table 1 summarizes the key distinctions between our approach and existing studies that primarily focus on LMIA against graph representations.

Our contributions. In this paper, we conduct an investigation into the privacy implications of UGRL models through the lens of link membership inference attacks (LMIAs). More specifically, we delve into the scenario where an LMIA adversary seeks to determine whether a specific pair of nodes in the target graph are connected, relying on access to the embeddings of these nodes generated by a UGRL model. We make the following contributions[2].

Design of attacks and evaluation. We consider two distinct types of adversaries: those with access solely to the node embeddings and those equipped with knowledge of a "shadow graph", which may originate from external sources such as public data repositories. It is important to note that the structure, node attributes, and distribution of the shadow graph may differ significantly from those of the target graph. To address these scenarios, we devise two link membership inference attacks (referred to as Attacks 1 and 2) tailored to each type of adversary. Through an extensive series of experiments, we demonstrate the effectiveness of both attacks against four cutting-edge UGRL models: *node2vec* [15], *DeepWalk* [18], *LINE* [51], and *GAE* [27]. These experiments are conducted on five real-world graphs. Our findings reveal that both attacks exhibit remarkable success rates against all four UGRL models, achieving accuracy levels as high as 0.95. Notably, our attacks surpass the approach proposed in a previous study [11], resulting in accuracy gains of up to 0.24.

[2]Our code and datasets are available online: https://gitfront.io/r/user-8658281/Mpx6p294FzeZ/LMIA/

Factor analysis. To gain a more profound comprehension of the privacy vulnerabilities inherent in UGRL models, we systematically investigate how varying degrees of preserved structural information in the embeddings impact the performance of LMIA. Our analysis yields the following crucial insights: (i) Embeddings that retain a lower level of proximity information are more susceptible to LMIA; and (ii) Embeddings characterized by higher dimensions are more prone to LMIA. These findings illuminate the intricate relationship between the extent of preserved structural information in the embeddings and the effectiveness of LMIA, offering valuable insights into the privacy vulnerabilities associated with UGRL models.

Design of defense mechanisms and evaluation. We propose a family of defense mechanisms designed to alleviate privacy vulnerabilities in UGRL models. These defense strategies revolve around the concept of perturbing the embeddings by introducing Laplace noise. To minimize any adverse effects on embedding accuracy, we judiciously apply noise exclusively to dimensions deemed less critical. To determine the importance of embedding dimensions, we develop three distinct estimation methods. Through comprehensive empirical assessments, we substantiate the efficacy of our defense mechanisms. Our findings clearly indicate that our proposed defenses strike a more favorable balance between defense effectiveness and embedding quality when compared to baseline methods.

2 UNSUPERVISED GRAPH REPRESENTATION LEARNING

Consider a graph $G(V, E)$, where V represents a set of nodes and $E \subseteq (V \times V)$ represents a set of edges (links) connecting the nodes. In the context of graph representation learning (GRL), the objective is to learn a mapping function $\Phi : v \rightarrow \vec{z_v} \in \mathbb{R}^d$ for each node $v \in V$. Here, $\vec{z_v}$ denotes the learned vector representation or embedding of node v, while d represents the dimensionality of the learned representation. The fundamental aim of the mapping function Φ is to retain the original network structure, ensuring that nodes that exhibit similarity in G are also represented as similar vectors in the learned vector space.

The node embeddings in graph representation learning preserve various types of local structure information within the graph, including first-order, second-order, and high-order proximity. The first-order proximity captures the similarity between directly connected vertices, while the second-order and high-order proximity capture similarities between vertices sharing common neighbors [60].

Broadly, the existing GRL approaches can be classified into two categories: *supervised* and *unsupervised* approaches [60]. Unsupervised approaches are oriented towards preserving the inherent structure of the graph and generating graph representations without relying on labeled nodes or graphs. In contrast, supervised methods involve the learning of graph representations using labeled samples. In this paper, we primarily focus on UGRL models. It is worth noting that we exclude GNNs [16, 39, 52] as they fall within the realm of supervised learning methods.

Existing UGRL models can be categorized into five algorithmic perspectives: *matrix factorization*, *random walk*, *edge modeling*, *deep*

learning, and *hybrid methods* [60]. For the purpose of this paper, we primarily consider random walk-based, edge modeling-based, and deep learning-based methods. We make this choice because they are recognized as the most efficient and practical among the five categories.

Random walk based methods. In random walk-based methods, the use of random walks allows for the efficient capture of structural relationships between vertices. By performing truncated random walks, the original graph is transformed into a collection of paths. Two representative random walk-based UGRL methods considered in this paper are DeepWalk [41] and node2vec [15]. DeepWalk employs a uniform random walk approach, where neighborhood nodes have an equal probability of being selected for the next step of the walk. On the other hand, node2vec follows a biased random walk, determined by parameters p and q. Parameter p prioritizes a breadth-first-search (BFS) procedure, while q prioritizes a depth-first-search (DFS) procedure. Both methods use SkipGram [36] on the random walk sequences to maximize the probability of observing a node's neighborhood given its embedding.

Edge modeling based methods. In contrast to approaches that utilize random walks to capture network structure, edge modeling-based methods directly learn node embeddings from the connections between nodes. In this paper, we focus on a representative edge modeling-based method called LINE [51]. LINE models both the joint probability distribution and conditional probability distribution of connected nodes to learn the node embeddings.

Deep learning based methods. Some UGRL models utilize deep learning techniques to capture the non-linearity of complex graphs. In this paper, we examine a representative UGRL model called GAE [27]. GAE employs an encoder-decoder scheme to learn node embeddings. The encoder consists of a graph convolutional network [28], which takes node features and the adjacency matrix of the graph as input. It aggregates feature information from neighboring nodes. The decoder utilizes an inner product operation to project pairwise node embeddings onto a similarity value. The similarity values of all node pairs are then passed through sigmoid functions to compute the probability of entries in the adjacency matrix, resulting in a predicted adjacency matrix. The loss is calculated using Binary Cross Entropy (BCE) loss between the predicted adjacency matrix and the ground-truth adjacency matrix.

Table 2 summarizes the key differences in methodology and preserved proximity between DeepWalk, node2vec, LINE, and GAE. These models have various user-defined parameters, such as the length and number of random walks (for DeepWalk and node2vec), the dimension of the target vector space, and the neighborhood size. Different parameter choices lead to distinct node embeddings, affecting their vulnerability to LMIA. In Section 5.3, we will discuss how the different parameters of UGRL models influence the preservation of structural information in node embeddings and their vulnerability to privacy attacks.

Algorithm	Methodology	Preserved proximity
DeepWalk	Uniform random walk	1^{st}, 2^{nd} and high-order
node2vec	Biased random walk	1^{st}, 2^{nd} and high-order
LINE	Edge modeling	1^{st}
GAE	Deep learning based	# of layers in encoder

Table 2: Comparison of Four UGRL models.

3 PROBLEM FORMULATION

In this section, we define the scope and goal of our problem.

Learning setting. Similar to the supervised GRL models, there are two different UGRL paradigms: *transductive* and *inductive* settings. In the transductive setting, all nodes in the graph are available at training time [15, 51, 53]. In contrast, the inductive setting includes only a subset of nodes in the training graph [4, 16]. Since most of the existing UGRL models, including the four models considered in this paper, are transductive, we solely focus on transductive UGRL models in this study.

Threat model. ML models are susceptible to privacy attacks [22, 43]. These attacks can be categorized into two groups based on the sensitive assets in ML models that the adversary seeks to obtain [43]: (1) *Privacy attacks on training data*: The adversary aims to infer sensitive information within the training dataset. (2) *Privacy attacks on ML models*: The adversary considers ML models themselves as sensitive, such as valuable company assets, and attempts to extract information about the model's structure and parameters.

In this paper, our focus is primarily on privacy attacks on training data, specifically the *membership inference attack (MIA)*. We should note that in the transductive UGRL setting (in particular for learning of node embeddings), determining node-level membership (i.e., whether a node is included in the training graph) is straightforward since all nodes are necessarily included. Hence, we focus on the *link membership inference*, which involves determining whether two specific nodes u and v are connected in the target graph of a given UGRL model by employing the node embeddings of u and v generated by the target model.

We assume that the adversary possesses the capability to access the embeddings of nodes generated by the target UGRL model and connect these embeddings to individual users. This feasibility arises due to the prevalent practice in many third-party social applications, such as Flickr and Twitter, where users are allowed to link their accounts to a common system, often facilitated by services like Google accounts. Despite the existence of these common accounts, users can maintain distinct social relationships within these applications, and the confidentiality of these relationships is crucial [35]. In scenarios where data owners of these social applications collaborate to derive a "global" graph representation, they can concurrently train a "local" graph representation of their specific data and subsequently share it with other data owners [35]. As the local embeddings can be easily linked to individual users by their accounts (e.g., Google accounts) which are shared across the different systems, an honest-but-curious data owner may attempt to infer sensitive structural information from other data owners' local data using the shared node embeddings, thereby posing privacy risks to individual users.

In addition to node embeddings, the adversary may possess additional background knowledge K_{Aug}, which can be categorized along three dimensions:

- *Node embeddings Z*: The adversary may have access to the embedding of a set of nodes $\{v|v \in G_{target}, v \neq v_i, v_j\}$ generated by the UGRL model Φ on the target graph G_{target}.

These embeddings can be either complete (i.e., encompassing all nodes) or partial (i.e., representing a subset of nodes).
- *Shadow graph* G_{shadow}: The adversary possesses a shadow graph G_{Shadow} with its own structure and node attributes. The shadow graph can originate from a different domain and exhibit different data distributions compared to G_{target}.
- *Target UGRL model* Φ: The adversary may have access to the target UGRL model Φ either in a white-box or black-box fashion. White-box access to the model reveals its architecture, parameters, and loss function, while black-box access allows the adversary to only access the node embeddings.

Given that white-box attacks may be theoretically optimal but inefficient in practice [44, 49], we restrict our focus to black-box attacks in this paper.

Problem definition. Formally, given a target node pair (v_i, v_j) where $v_i, v_j \in G_{target}$ and their embeddings $\vec{z_i}$ and $\vec{z_j}$, a UGRL model Φ, and the adversary's auxiliary knowledge K_{Aug} that provides black-box access to Φ, the *link membership inference attack* (LMIA) is a mapping function

$$f : \vec{z_i}, \vec{z_j}, \Phi, K_{Aug} \rightarrow \{0, 1\}, \tag{1}$$

where 0 (or 1) indicates that the edge e does not (or does, respectively) belong to G_{target}.

LMIA can be formally described using the hypothetical *inference game* between a challenger and an adversary [6, 44, 58, 59]. The game is formalized below:

Definition 1 (Link Membership inference game). *Let $\mathcal{G}$ be a set of graphs drawn from the distribution π. Let $\vec{V}$ be the embedding space. A UGRL algorithm Φ is a function that maps an instance $G(V, E) \in \mathcal{G}$ to a set of embeddings in $\vec{V}$.*

- *A fresh dataset is sampled from the distribution π, and Φ is trained on this dataset.*
- *A coin is flipped to determine $b \in \{0, 1\}$ uniformly at random.*
- *If $b = 0$, a disconnected node pair (v_i, v_j) is randomly selected from π (non-member). Otherwise, a connected node pair (v_i, v_j) is randomly chosen from G (member).*
- *The adversary $\mathcal{A}$ uses the additional knowledge K_{Aug} and the embeddings $\vec{z_i}$ and $\vec{z_j}$ to compute $\hat{b} = f(\vec{v_i}, \vec{v_j}, \Phi, K_{Aug})$ (Equation 1).*
- *The adversary succeeds (output 1) if $\hat{b} = b$, and fails (output 0) otherwise.*

The performance of the LMIA attack is evaluated by averaging its success rate over multiple repetitions of the inference game.

4 DETAILS OF OUR ATTACKS

Recent studies have revealed that the nodes which are connected in the target graph (i.e., member edges) have higher similarity in their classification posterior probabilities by the target models (e.g., GNNs) than those which are not (i.e., non-member edges) [19, 56]. However, as UGRL models do not produce such posterior probabilities but node embeddings instead, we first investigate the properties of node embeddings that can be exploited by LMIA.

We conducted experiments using four prominent UGRL algorithms: node2vec, DeepWalk, LINE, and GAE. In our analysis, we measured three types of embedding similarity: *Dot product similarity*, *Cosine similarity*, and *Euclidean distance*. By comparing the embedding similarity between member and non-member links, we aimed to identify distinguishable patterns. Our findings revealed a consistent trend across all four UGRL models: *node pairs belonging to member links exhibited higher average embedding similarity compared to node pairs in non-member links*. More details of the similarity property of node embeddings can be found in Appendix A.

Intuitively, leveraging the property of embedding similarity, the attacker can effectively distinguish between member and non-member links. Building upon this intuition, we propose two black-box attacks for adversaries with different levels of knowledge regarding G_{shadow}: **Attack 1**, when G_{shadow} is unavailable to the adversary, and **Attack 2**, when G_{shadow} is accessible. We will now delve into the specifics of these two attacks.

4.1 Attack 1: without Shadow Graph

When G_{shadow} is unavailable, the adversary only has access to the embeddings Z_{target} of a set of nodes in G_{target}. To predict the membership of the target link, the adversary can exploit the node embedding similarity from Z_{target}. Following this idea, we propose an unsupervised attack that employs the k-means clustering algorithm to infer whether the target node pair (v_i, v_j) has a link in G_{target} based on Z_{target}. The attack consists of two phases:

Clustering phase. The adversary measures the embedding similarity for each node pair in Z_{target}, where the similarity metrics include Dot product similarity, Cosine similarity, and Euclidean distance. Then for each node pair, the attacker generates its *similarity feature* by concatenating these similarity metrics. Next, the adversary applies k-means clustering with $k = 2$ to partition all node pairs into two classes: members and non-members. Since member links exhibit higher similarity in their embeddings compared to non-member links, the cluster with a higher average pairwise embedding similarity is labeled as the member cluster, while the other cluster is labeled as the non-member cluster.

Inference phase. For the target node pair (v_i, v_j) and their embeddings $\vec{z_i}$ and $\vec{z_j}$, the adversary constructs its similarity feature by concatenating the three similarity values (Dot product similarity, Cosine similarity, and Euclidean distance). Then, the attacker assigns the pair to the cluster whose average pairwise embedding similarity, measured on the similarity feature, is closer to that of the target node pair (v_i, v_j).

4.2 Attack 2: with Shadow Graph

When G_{shadow} is available, we adapt the shadow model approach [47] to our setting, enabling us to train a set of shadow models from G_{shadow} that can replicate the functionality of the target UGRL model. By utilizing the shadow models, we design Attack 2 as a binary supervised classifier f trained on the node embeddings generated by the shadow models and the membership information of links in G_{shadow}. Then given the target node pair (v_i, v_j) and their embeddings $\vec{z_i}$ and $\vec{z_j}$, the classifier f predicts whether the edge $e(v_i, v_j)$ belongs to G_{target}. Following this idea, Attack 2 comprises three phases: *shadow model training*, *attack model training*, and

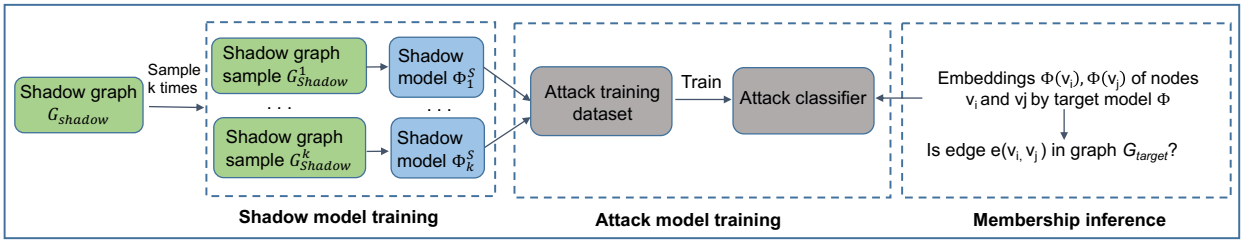


Figure 1: Overview of Attack 2.

membership inference (Figure 1). Next, we explain the details of these phases.

Shadow model training phase. To collect training data for the classifier f, the adversary randomly samples k subgraphs from the shadow graph G_{shadow}. Subsequently, k shadow models $\{\Phi^S_1, \ldots, \Phi^S_k\}$ are trained using the corresponding subgraphs.

Attack model training phase. For each shadow sample graph G^i_{Shadow} and its corresponding trained shadow model Φ^S_i, the adversary selects a set of node pairs $\{(v_j, v_k)\}$ from G^i_{Shadow}, and obtains the embeddings $\Phi^S_i(v_j)$ and $\Phi^S_i(v_k)$ of v_j and v_k for each node pair. Next, the adversary computes the similarity $sim(\Phi^S_i(v_j), \Phi^S_i(v_k))$ using different similarity metrics, including Dot product similarity, Cosine similarity, and Euclidean distance. The adversary concatenates all measured similarity values to create the LMIA feature of the node pair (v_j, v_k), and labels $e(v_j, v_k)$ as *member* ("1") if it is an edge in G^i_{Shadow}, and *non-member* ("0") otherwise. Consequently, the adversary constructs an *attack training dataset* consisting of the derived similarity features and their member/non-member labels. Finally, the adversary trains the binary membership attack classifier f on the attack training dataset.

Membership inference phase. During inference, the adversary computes the three similarity values (Dot product similarity, Cosine similarity, and Euclidean distance) between the embedding $\vec{z_i}$ and $\vec{z_j}$ of the target node pair. The adversary then feeds the concatenation of these three similarity values into the trained attack classifier f and obtains probabilities for each class. By selecting a probability threshold of 0.5, the target node pair is predicted with the label of the class whose probability exceeds 0.5.

5 EVALUATION OF ATTACK PERFORMANCE

We conduct a comprehensive set of empirical studies with the objective of addressing two key research questions:

- **RQ1** - How effective is LMIA against the representative UGRL models?
- **RQ2** - How do the different amounts of preserved structural information in the embedding affect LMIA performance?

5.1 Experimental Setup

All the experiments are performed on a machine with 2 × Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz, 12 cores, 24 processors, and 128 GB memory. All the algorithms are implemented in Python along with PyTorch.

Datasets. We use five real-world graph datasets (DBLP, LastFm, Cora, Citeseer, and Pubmed datasets) that are popularly used in the literature for graph learning. More details of these datasets can be found in Appendix B.

UGRL algorithms. We employ four state-of-the-art UGRL algorithms, namely DeepWalk [41], node2vec [15], LINE [51], and GAE [27]. The encoder of the GAE model is set as a 2-layer neural network with 64 neurons in each layer. Both DeepWalk and node2vec models preserve the 1st- to 5th-order proximity in the embeddings, while LINE preserves the 1st-order proximity, and GAE preserves the 1st- to 2nd-order proximity (with a 2-layer neural network).

Shadow graph. Two settings are considered in the experiments: (1) *Non-transfer setting*: where both shadow graph and target graph are sampled from the same dataset, and (2) *Transfer setting*: where the shadow graph and the target graph are sampled from two different datasets. The performance of LMIA is evaluated under both settings.

Metrics. For assessing the attack effectiveness of LMIA, we employ three metrics: (1) *Attack accuracy*: The ratio of correctly predicted member/non-member edges by LMIA over the total number of node pairs in the testing data; (2) *Area Under the Curve (AUC)*: Measured over the true positive rate (TPR) and false positive rate (FPR) at various settings of thresholds of the attack classifier; and (3) *True-Positive Rate at False-Positive Rates (TPR@FPR)* [6]: Measures TPR at various FPR values.

Regarding the performance of UGRL models, we measure the performance of downstream tasks on node embeddings, specifically focusing on node classification. We train an MLP classifier on the node embeddings and measure the AUC as the embedding quality.

Setup of attack classifier. We use three types of attack classifiers, namely Multi-layer Perceptron (MLP), Random Forest (RF), and Support Vector Machine (SVM). The MLP consists of three hidden layers with 64, 32, and 16 neurons, respectively. The activation functions used are Relu for the hidden layers and Sigmoid for the output layer. The classifier is trained for 1,000 epochs with a learning rate of 0.001 using cross-entropy loss and the Adam optimizer. For RF, the maximum depth is set as 150, and for SVM, the radial basis measurement (RBF) kernel is used with a regularization parameter of 1 and a degree of the polynomial kernel measurement set to 3. The kernel coefficient γ is set to 1. The implementation of these classifiers is obtained from the sklearn package [3].

Training and testing data of LMIA. In terms of the training and testing data for LMIA, we first generate a data pool that consists of 70% of the edges (E_{mem}) randomly sampled from the original graph and the same number of disconnected node pairs (E_{non}) that

[3] https://scikit-learn.org/

are randomly sampled from the original graph. Next, we generate the LMIA training dataset which consists of 30% edges randomly sampled from E_{mem} as members and an equal number of edges randomly chosen from E_{non} as non-members. Similarly, we generate the LMIA testing dataset that consists of an equal number of member and non-member edges. The size ratio between the LMIA training and testing data is 7:3.

Baselines. The paper considers four baselines for comparison with the LMIA attack:

- **Baseline-1.** This baseline involves an *ensemble* of three sub-attacks, each using a different similarity metric (Dot product similarity, Cosine similarity, and Euclidean distance). For each sub-attack, we use the threshold-based MIA model [11] to mark the target link as a non-member/member by comparing its node similarity with a given threshold. The best threshold value that delivers the optimal attack performance is chosen empirically. Finally, links in the testing data with node embedding similarity higher than the threshold are marked as members, and the rest as non-members.
- **Baseline-2.** This baseline uses the link inference attack proposed in [11][4]. It involves reconstructing the adjacency matrix of the target graph using an encoder-decoder network and adversary knowledge of the auxiliary graph. The decoder setup is the same as in [11], and the encoder architecture is the same as the target model. An auxiliary graph containing 70% of the edges from the target graph is used.
- **Baseline-3.** In this baseline, we modify the input feature of Attack 1 by employing the concatenation of embeddings (i.e., placing the embeddings side-by-side) instead of the concatenation of embedding similarities.
- **Baseline-4.** This baseline utilizes the concatenation of node embeddings as the input feature for the Attack 2 classifier.

5.2 Effectiveness of LMIA (RQ1)

Before proceeding with the attacks on the UGRL algorithms, we first evaluate the quality of the embeddings generated by these algorithms to justify why these algorithms are worthy to be attacked. The evaluation results can be found in Appendix C. We observe that, in general, the AUC values for all four UGRL algorithms are substantially higher than what would be achieved by random guessing for classification. This indicates that the embeddings produced by these algorithms exhibit commendable performance. Given the satisfactory performance of the UGRL models, we can proceed with launching the LMIA on these models.

Next, we assess the effectiveness of both Attacks 1 and Attacks 2 by evaluating their performance. Figure 2 illustrates the accuracy of both attacks across the five datasets. A cursory look reveals that the attack accuracy consistently exceeds 0.5, which is the accuracy obtained by random guessing. This substantiates the effectiveness of LMIA in targeting network embeddings.

Moving forward, we delve into a detailed discussion on the performance of Attacks 1 and 2.

[4]We use the implementation provided by the authors of [11], which is available at https://github.com/vasishtduddu/GraphLeaks.

5.2.1 Performance of Attack 1. Figure 2 presents the results of Attack 1. Remarkably, as an unsupervised attack model, Attack 1 demonstrates outstanding performance against node2vec, DeepWalk, LINE, and GAE across most datasets, yielding attack accuracy ranging from 0.62 to 0.93. Notably, in certain settings, such as when the LINE model is trained on the Cora dataset (Figure 2 (c)), the attack accuracy can reach as high as 0.93. Moreover, our attack outperforms all four baseline methods in terms of attack accuracy in the majority of settings. Notably, its performance surpasses that of Baseline-3 across all settings significantly. Besides the overall accuracy of the attack, we measure the attack accuracy of the clustering phase of Attack 1 and include the results in Appendix D.

5.2.2 Performance of Attack 2. Moving on to the performance of Attack 2, we consider two distinct settings for the shadow graph:

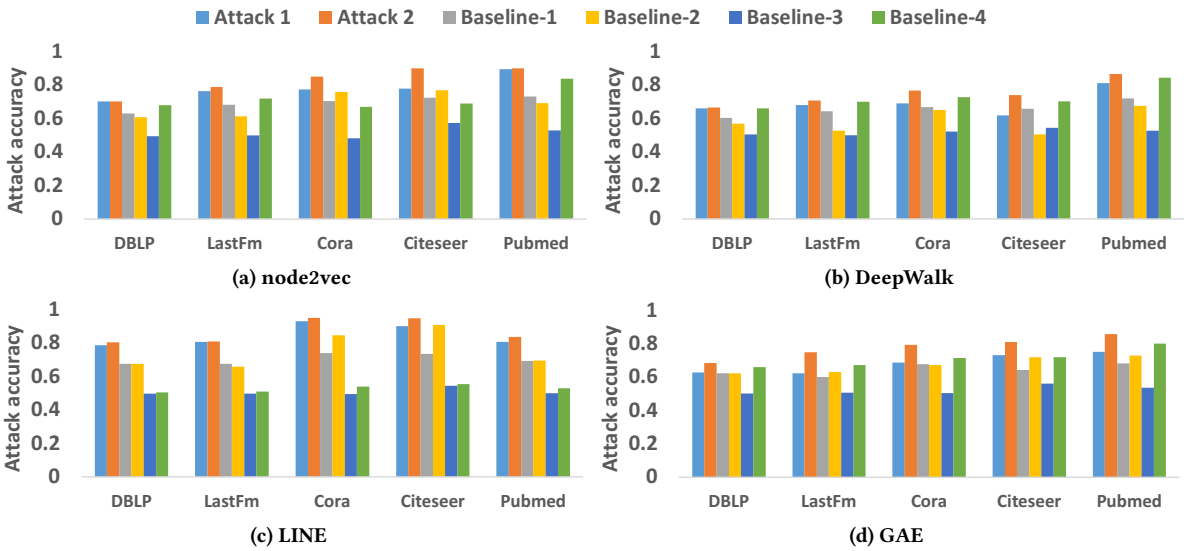
- **Setting 1 (Non-transfer setting)**: In this setting, both the shadow and target graphs are sampled from the same dataset.
- **Setting 2 (Transfer setting)**: This setting involves sampling the shadow and target graphs from different datasets.

Among the three types of attack classifiers (MLP, SVM, RF), we observe that the MLP attack consistently achieves the highest attack accuracy in the majority of settings. Hence, we solely present the attack accuracy of MLP in the subsequent discussions.

Setting 1 (Non-transfer setting). Figure 2 illustrates the results of Attack 2 under Setting 1 for the four UGRL models. First, similar to Attack 1, Attack 2 proves to be effective against node2vec, DeepWalk, LINE, and GAE, achieving attack accuracy within the range of $[0.67, 0.95]$. Notably, the attack accuracy can reach as high as 0.95, as observed in the case of the LINE model on the Cora dataset (Figure 2 (c)). Second, our attack outperforms the baselines in the majority of settings. Particularly, our attack's accuracy can be 0.13 higher than that of Baseline-2 [11]. Moreover, we observe that Baseline-4 has comparable performance as ours in some settings. This is attributed to the utilization of all embeddings by Baseline-4. However, as its feature dimension is much larger than ours, Baseline-4 suffers from overfitting in some settings and consequently has worse performance than ours in these settings. Additionally, we observe that LINE is more vulnerable to our attack compared to node2vec, DeepWalk, and GAE. We will delve into the reasons behind LINE's high vulnerability in Section 5.3.

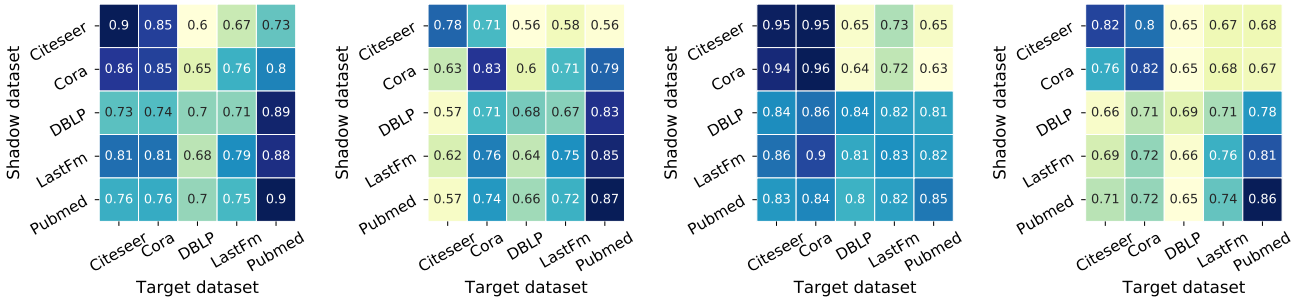
Apart from attack accuracy, we also measure TPR@FPR with FPR=1% of our Attack 2, Baseline-2, and Baseline-4. We do not consider Attack 1, Baseline-1, and Baseline-3 as they are not supervised MIA attacks and thus cannot be evaluated with TPR and FPR. The results are presented in Table 3. Overall, our Attack 2 outperforms both Baseline-2 and Baseline-4 methods in all the settings in terms of TPR@1% FPR. Furthermore, LINE consistently exhibits higher vulnerability to LMIA than the other three UGRL models, which aligns with the attack accuracy results (Figure 2). We will elaborate on LINE's vulnerability to LMIA in Section 5.3.

Beyond attack accuracy and TPR@FPR, we measured the attack AUC of our attacks and compare it with the baselines. The results can be found in Appendix E. Our main observation is that the attack AUC of our attacks (ranging from 0.71 to 0.99) surpass the performance of both baseline methods across all settings.



(a) node2vec (b) DeepWalk (c) LINE (d) GAE

Figure 2: Attack accuracy of Attack 1 & Attack 2 under Setting 1 (non-transfer setting).



(a) node2vec (b) DeepWalk (c) LINE (d) GAE

Figure 3: Accuracy of Attack 2 under Setting 2 (transfer setting).

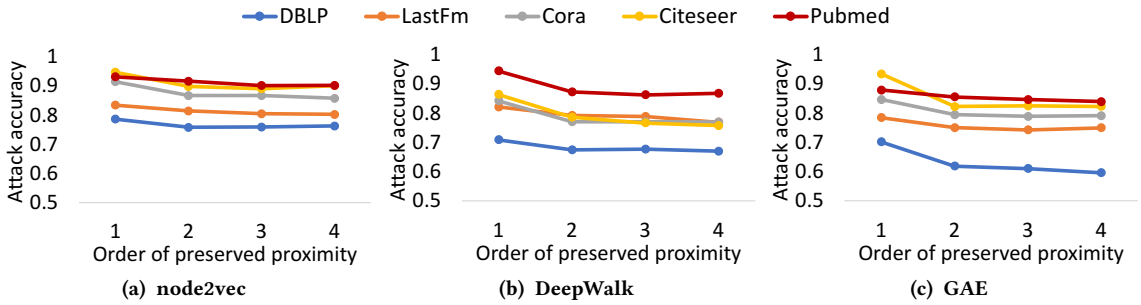
Settings	node2vec			DeepWalk			LINE			GAE		
	Attack 2	B2	B4	Attack 2	B2	B4	Attack 2	B2	B4	Attack 2	B2	B4
DBLP	0.18	0.05	0.11	0.14	0.03	0.1	0.46	0.23	0.05	0.08	0.04	0.06
LastFm	0.24	0.12	0.19	0.18	0.05	0.17	0.43	0.23	0.05	0.12	0.06	0.12
Cora	0.24	0.21	0.13	0.16	0.04	0.09	0.99	0.95	0.05	0.16	0.06	0.04
Citeseer	0.35	0.27	0.05	0.12	0.19	0.06	0.99	0.96	0.03	0.17	0.07	0.03
Pubmed	0.32	0.31	0.28	0.27	0.21	0.24	0.42	0.25	0.05	0.37	0.05	0.13

Table 3: TPR@1% FPR performance of our attack under Setting 1. We do not consider Attack 1, Baseline-1, and Baseline-3 as they are not supervised MIA attacks and thus cannot be evaluated with TPR and FPR. The best TPR@1% FPR performance for each UGRL model and each dataset is highlighted with olive color.

Setting 2 (Transfer setting). Figure 3 showcases the results of MLP Attack 2 under Setting 2. The following observations can be made. First, for each target dataset, LMIA achieves the highest attack accuracy when the shadow and target graphs are sampled from the same dataset (i.e., within the same column). Second, Even under the transfer setting (i.e., when the shadow and target graphs are sampled from different datasets), the attack remains successful, with attack accuracy ranging from 0.56 to 0.95. Notably, the accuracy can reach as high as 0.954 when the target dataset is sampled from the Cora dataset and the shadow dataset is sampled from the Citeseer dataset (Figure 3 (c)). This demonstrates LMIA's ability to learn knowledge from the shadow data, where node embeddings of member links exhibit higher similarity than non-member links, and successfully transfer such knowledge to the target graph.

5.3 Impact of Preserved Structural Information in Embeddings on LMIA Performance (RQ2)

It is evident from our observations that different UGRL methods exhibit varying levels of vulnerability to LMIA. In particular, LINE consistently displays higher vulnerability compared to the other



(a) node2vec (b) DeepWalk (c) GAE

Figure 4: The impact of the order of the proximity preserved in embeddings on attack accuracy.

three UGRL methods (Figure 2). This suggests that the vulnerability of these UGRL models is influenced by the amount of preserved structural information in their embeddings. Hence, in this section, we investigate how the preservation of structural information in embeddings affects LMIA performance.

Considering the stochastic nature of UGRL models, the output of their embeddings depends on several parameters, such as the neighborhood size, dimension of the target vector space, and length of random walks (for random-walk based models). These parameters determine the amount of preserved structural information in the embeddings. For instance, the neighborhood size determines the order of proximity preserved in the embedding. Consequently, we focus on two types of UGRL model parameters: (1) the order of proximity preserved in the embeddings, and (2) the number of dimensions of the embedding vector space.

Embeddings that preserve a lower order of proximity are more vulnerable. To examine the impact of preserved proximity order in embeddings on attack accuracy, we vary the neighborhood size parameter for node2vec and DeepWalk, as well as the number of layers for GAE, to control the order of proximity preserved in their embeddings. We exclude LINE as it only preserves 1st-order proximity.

Figure 4 presents the attack accuracy results of node2vec, DeepWalk, and GAE embeddings that preserve proximity from 1st to 4th order. A consistent observation across the three models is that embeddings preserving lower orders of proximity are more vulnerable than those preserving higher orders. Notably, the largest vulnerability disparity occurs when the preserved proximity changes from 1st- to 2nd-order. After reaching 2nd-order proximity, the vulnerability of embeddings becomes stable irrespective of the increase in proximity order for most cases. Consequently, we expect LMIA to be more successful against UGRL models that solely preserve 1st-order proximity compared to those preserving higher-order proximity. This finding also explains why LINE consistently exhibits the highest vulnerability among the four studied models (Figure 2).

Embeddings of higher dimensions are more vulnerable. To examine the impact of embedding dimensions on attack performance, we vary the embedding size (e.g., 32, 64, 128, and 256) for the four UGRL models and measure the attack accuracy. The results, presented in Figure 5, indicate that embeddings with higher dimensions are more vulnerable to LMIA. This can be attributed to the fact that embeddings with more dimensions encode additional structural information that LMIA can leverage during the attack.

6 DEFENSE MECHANISMS

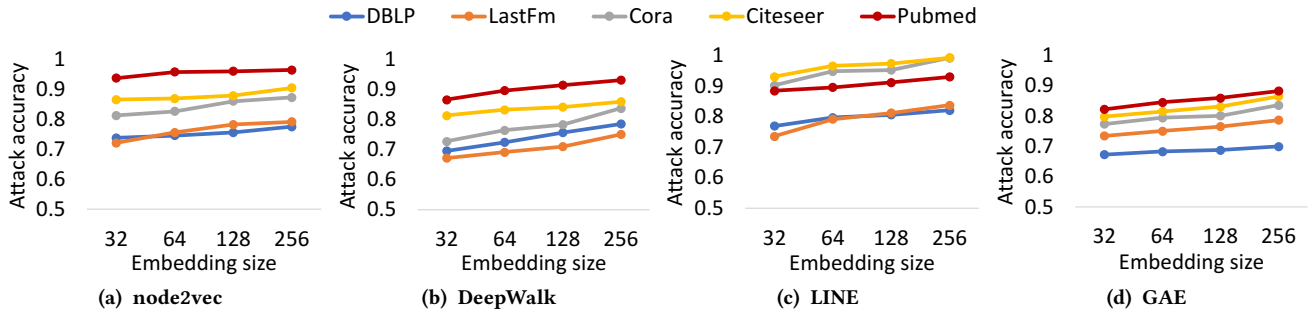
In order to defend against LMIA, one intuitive approach is to add noise to the UGRL models, which can confuse the adversary and provide privacy protection. However, our empirical evaluation revealed that existing defense mechanisms, such as differential privacy, which add noise to gradients or parameters during model training, or adding noise directly to posteriors/embeddings, can result in significant accuracy loss on node embeddings. To address this issue, we propose a defense mechanism that selectively adds noise only to the dimensions of embeddings that are deemed least important. This aims to minimize the quality loss on embeddings while providing sufficient privacy protection against LMIA. Next, we first present the details of estimating the importance of embedding dimensions (Sec. 6.1). Then we describe our defense mechanism (Sec. 6.2) and the empirical results of the performance of the defense mechanism (Sec. 6.3).

6.1 Estimating Importance of Embedding Dimensions

To estimate the importance of embedding dimensions, we consider three widely used methods: the *permutation-based* method, the *SHapley Additive exPlanations (SHAP)* value-based method, and the *mean decrease in impurity (MDI)* based method. We adapt these methods to our setting to find the importance of embedding dimensions by treating node embeddings as features for the node classifier in the downstream task.

Permutation-based importance (**PERM**). At the high level, the permutation-based methods [2] assign a score to each feature. A higher score indicates that perturbing that feature would result in a greater accuracy loss for the prediction. In other words, the model heavily relies on the information provided by that feature to make accurate predictions.

SHAP value based importance. SHAP is a technique used to explain the prediction of a data sample by computing the contribution of each feature to the prediction. It is based on *Shapley values*, which employ game theory to assign credit to features or feature values [32]. Higher Shapley values indicate greater importance of the corresponding features to the model's output.



(a) node2vec (b) DeepWalk (c) LINE (d) GAE

Figure 5: The impact of the number of embedding dimensions on attack accuracy.

MDI-based importance. The MDI method [5] is commonly used for decision tree models. It measures the reduction in impurity (e.g., entropy or Gini index) when a feature is used to split a branch in the decision tree. To assess the importance of embedding dimensions, we construct a node classification classifier in the form of a decision tree, with the embedding as the input.

6.2 Perturbation-based Defense

To defend against LMIA, we introduce a perturbation-based defense by adding noise to the node embeddings. To mitigate the embedding quality loss of the UGRL models, we selectively add Laplace noise to the least important dimensions of the embeddings. Here's an overview of our defense mechanism: First, we measure the importance of each embedding dimension by using one of the three estimation methods described in Section 6.1. Second, we rank the embedding dimensions based on their importance. Third, we add noise to a subset of the embedding dimensions with the lowest importance. Specifically, we perturb $[d \times r]$ dimensions of the lowest importance, where d is the total number of dimensions in the embedding, and $r \in (0, 1]$ is the *perturbation ratio*. A higher perturbation ratio leads to perturbing more dimensions, resulting in a higher embedding quality loss for the UGRL models. The added noise Δ follows the Laplace distribution with a density function defined as below.

$$\Delta = \frac{1}{2b} e^{-\frac{x-\mu}{b}} \tag{2}$$

where b is the noise scale, and μ is the location parameter of the Laplace distribution A higher b indicates a stronger noise scale, leading to a stronger defense against LMIA.

6.3 Performance of Defense

In this section, we present the performance evaluation of our defense mechanisms. Specifically, we focus on the defense performance of Attack 2, as it exhibits superior attack performance compared to Attack 1.

Baselines. To provide a comparative analysis, we consider the following two baselines alongside our LMIA defense mechanisms.

- **Baseline-1 (DP).** Differential privacy (DP) [13] has demonstrated effectiveness against inference attacks on ML models [23, 47]. Hence, we adopt a differentially private deep learning method [1] as our first baseline. This method incorporates Laplace noise into the gradients during model training, where the Laplace noise follows the same distribution as our methods (Eqn. 2). The ϵ-differential privacy can be achieved by using the noise scale $b = 1/\epsilon$. At each iteration of the model training, the process involves computing the gradients for a batch of examples, clipping each gradient based on the L2 norm, adding Laplace noise to the gradients with a privacy budget, and subsequently updating the UGRL model parameters using stochastic gradient descent.
- **Baseline-2 (AdvR).** We employ an adversarial regularizer, which utilizes a discriminator to predict the link membership from the node embeddings. This approach has been used as an attack defense mechanism in prior literature [37]. The objective function is fomulated as a min-max problem:

$$\min_{\theta_E} \max_{\theta_D} L_{emb} - \lambda L_{dis}, \tag{3}$$

Here, L_{emb} represents the loss of a UGRL algorithm, and L_{dis} represents the loss of the discriminator used to infer link membership. The parameters θ_E and θ_D correspond to the UGRL model and the discriminator, respectively, while λ controls the strength of regularization. During the training of θ_E, we first fix θ_E and train θ_D several times until L_{dis} no longer decreases over the last 50 epochs. Then, with θ_D fixed, we calculate the total loss of θ_E as $L_{emb} - \lambda L_{dis}$ and update θ_E using stochastic gradient descent computed from the total loss. The training process terminates when the total loss of θ_E no longer decreases over the last 50 epochs. The loss function of the discriminator is Binary Cross Entropy.

Setup. We evaluate the performance of our defense mechanisms using different perturbation ratios, $r = \{0.2, 0.4, 0.6, 0.8, 1\}$ (where $r = 1$ implies noise added to all embedding dimensions without considering their importance), and noise scale $b = \{0.1, 0.5, 1, 5, 10\}$. Higher values of r and b indicate stronger privacy protection. For Baseline-1, we use the same noise scale b values as ours, while for Baseline-2, we employ regularization parameter values $\lambda = \{0.5, 1, 5, 10, 20\}$. Lower values of ϵ and higher values of λ correspond to stronger privacy protection.

Metrics. To evaluate the performance of our defense mechanisms, we employ the *attack accuracy* measurement (Section 5.1) to assess the attack performance after defense. Additionally, we measure the *embedding quality* as the AUC of node classification(Section 5.1). Higher AUC values indicate better embedding quality.

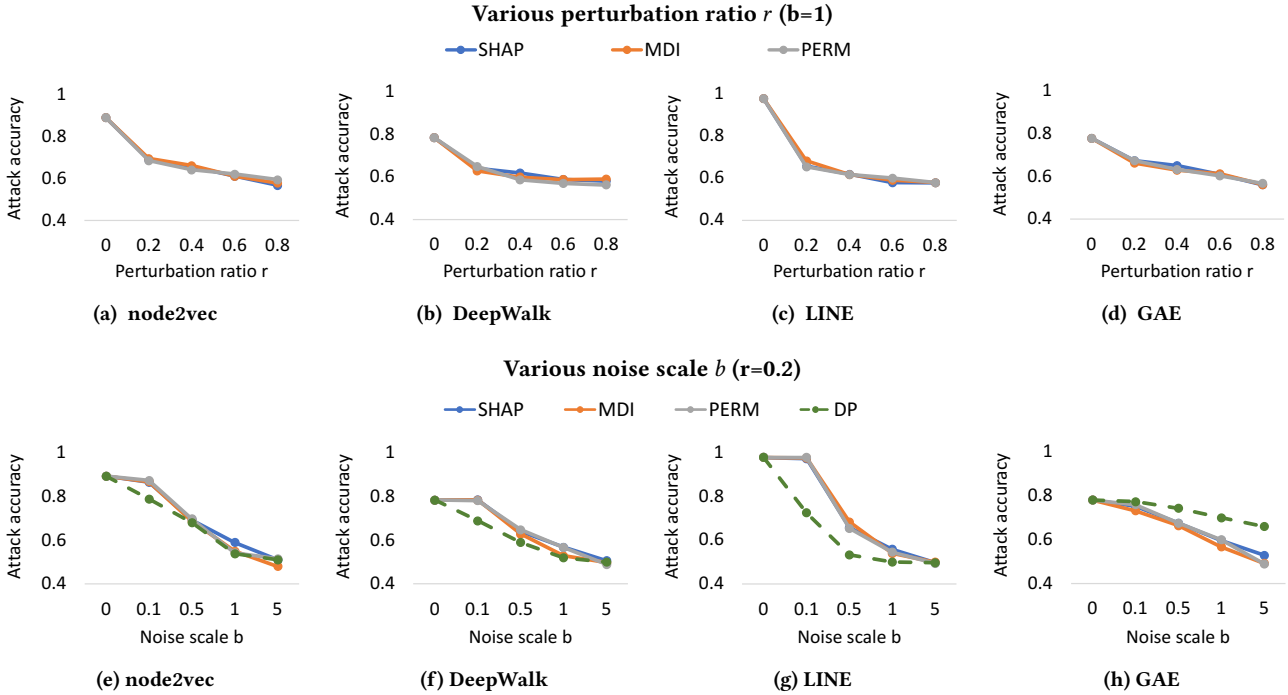


Figure 6: Impact of the perturbation ratio r and noise scale b on defense performance (Cora dataset).

Defense effectiveness. We present the defense effectiveness of our approach in Figure 6. Firstly, we explore the impact of varying the perturbation ratio r and present the corresponding attack accuracy results in Figure 6 (a) to (d). Our observations are as follows. First, our perturbation methods exhibit significant effectiveness against LMIA, notably reducing the attack accuracy. The defense strength increases with higher perturbation ratios (r). Remarkably, even with a modest perturbation ratio of 0.2 (perturbing only 20% of embedding dimensions), the attack accuracy can be reduced to 0.62. The three importance-based methods demonstrate similar defense performance.

Next, we assess the effectiveness of our defense methods by varying the noise scale b and comparing it with the baseline methods. Since our methods employ a different privacy parameter than Baseline-2 (AdvR), we focus our comparison on Baseline-1 (DP). The defense performance of AdvR can be found in Appendix F. Figure 6 (e) to (h) reveal the following insights. First, our perturbation methods remain effective against LMIA as the noise scale varies, with their defense power increasing at higher noise scales. For noise scale values equal to or greater than 1, the attack accuracy can be reduced to below 0.6. By perturbing only 20% of dimensions ($r = 0.2$), our methods provide defense capabilities comparable to those of Baseline-1 in most settings, even though Baseline-1 can offer a formal guarantee of differential privacy. Our methods exhibit stronger defense capabilities than Baseline-1 for the GAE model. This is attributed to the fact that, despite the presence of noise, GAE embeddings preserve certain fundamental structural information since the noise is injected into the gradients directly, leaving the adjacency matrix unaltered during computation. Consequently, GAE embeddings remain vulnerable to attacks.

Trade-off between defense effectiveness and embedding quality. While perturbation is effective in protecting against LMIAs, it carries the potential risk of degrading the quality of embeddings generated by UGRL models. To assess the delicate balance between defense effectiveness and embedding quality, we have gathered a set of results concerning defense effectiveness (quantified as 1−Attack Accuracy, where AC represents the attack accuracy) and embedding quality for the defense methods across various privacy parameters. These results are employed to construct the defense-quality ROC curve, where each point corresponds to a collected pair of defense effectiveness and embedding quality values. Ultimately, we measure the trade-off between defense effectiveness and embedding quality as the Area Under the Curve (AUC) of the defense-quality ROC curve.

Table 4 showcases the trade-off between attack accuracy and embedding quality for our three defense mechanisms (PERM, MDI, and SHAP-based perturbations) as well as the two baselines (DP and AdvR). Our observations indicate that our defense mechanisms consistently outperform both baselines concerning the trade-off between attack performance and embedding quality. Notably, our MDI-based perturbation methods exhibit the most favorable trade-off in most scenarios, underscoring the advantage of selectively introducing noise into the dimensions considered least important for defense purposes.

7 RELATED WORK

Membership inference attacks and defense. Shokri et al. [47] initialized the research on membership inference attacks (MIAs) against ML models. Yeom et al. explored the relationship between overfitting and MIAs, while Salem et al. [45] relaxed the initial

Method	Cora				Citeseer				LastFm			
	node2vec	DeepWalk	LINE	GAE	node2vec	DeepWalk	LINE	GAE	node2vec	DeepWalk	LINE	GAE
PERM	0.253	0.326	0.365	0.247	0.303	0.229	0.274	0.243	0.153	0.207	0.204	0.162
MDI	0.258	0.341	0.366	0.252	0.401	0.258	0.282	0.258	0.171	0.207	0.205	0.182
SHAP	0.236	0.318	0.366	0.25	0.327	0.192	0.283	0.249	0.147	0.201	0.199	0.173
DP	0.211	0.271	0.361	0.042	0.257	0.155	0.265	0.04	0.129	0.186	0.177	0.008
AdvR	0.177	0.271	0.312	0.159	0.259	0.157	0.269	0.18	0.146	0.184	0.184	0.162

Table 4: Trade-off between defense effectiveness and embedding quality of our defense and baselines (DP and AdvR). The best trade-off for each UGRL model and each dataset is marked with olive color.

assumptions of MIA. MIA has been applied to various domains, including federated learning [38], generative models [17], language models [48, 49], augmentation based DNN models [26], contrastive models [30], recommender system [61], and graph neural networks [19, 40]. Recent variants of MIAs, such as label-based MIAs [8, 29], have also been developed and evaluated.

Several defense mechanisms have been proposed to defend against MIAs, including the application of differential privacy [12] to ML models [7, 8, 17, 47], Confidence Masking [37], regularization [37], and techniques such as dropout, model stacking, and noise injection [24, 45]. For a comprehensive survey on MIAs and defenses, we refer readers to [22].

Privacy attacks against GNNs. Several types of privacy attacks have been proposed to infer sensitive information about the training graph of GNNs. These attacks can be categorized into three subtypes: *Membership Inference Attacks* (MIA), *Node Attribute Inference Attacks* (AIA), and *Property Inference Attacks* (PIA). MIA attacks have been addressed by various researchers [11, 19, 20, 40, 56] and can be divided into node-level and link-level attacks. For node-level MIA, both [20, 40] propose a black-box attack model that trains a Multi-Layer Perceptron (MLP) classifier using the posteriors obtained from shadow graphs. In the case of link-level MIA, He et al. [19] propose several black-box attack models with different levels of adversary knowledge. Wu et al. [56] develop a black-box attack against GNNs by quantifying the influence of one node on another node's posterior. Duddu et al. [11] propose an encoder-decoder-based attack model that reconstructs the graph structure using node embeddings, enabling link-level MIA based on the reconstructed graph. All these works utilize the fact that the nodes on the member edges have more similar posterior outputs than those on the non-member edges. Inspired by these works, our attacks also employ the node similarity to distinguish member and non-member edges. However, our attacks are fundamentally different from these works in the way that our attack features are derived from the similarity in node embeddings instead from posterior probabilities.

Regarding AIA, Duddu et al. [11] train an MLP on the node embeddings, assuming that the attacker possesses knowledge about a small fraction of the sensitive attribute of the target graph. For PIA, Zhang et al. [63] train a multi-task classifier based on the posteriors to infer graph properties (graph density, size, etc.). Suri et al. [50] propose a generic definition for PIA, aiming to distinguish between two possible training distributions. Zhang et al. [62] investigate the leakage of properties of node group distribution. Wang et al. [54] propose a set of black-box and white-box attacks to infer both node-level and link-level group properties of the training graph.

Privacy attacks on embeddings. Recent research has investigated the privacy vulnerabilities of various types of embeddings. Most of these studies have focused on privacy leakage in text data [33, 48] and images [30]. There is limited work addressing the privacy risks of graph embeddings [11, 63]. In particular, Zhang et al. [63] explore the privacy risks of graph embeddings, which represent the entire graph as a vector generated by GNNs. Their work aims to infer the links present in the original graph using graph embeddings. In contrast, our approach focuses on node embeddings for privacy inference. Of the few existing works on privacy risks in node embeddings, [11] is particularly relevant to our research. This work performs link inference through a graph reconstruction attack which employs an encoder-decoder model to reconstruct the adjacency matrix of the original graph. To perform this attack, they assume the availability of an auxiliary subgraph sampled from the same distribution as the target graph. However, in practice, such an auxiliary subgraph may not be readily available, and our attack does not rely on its presence.

8 CONCLUSION

In this paper, we conduct the first comprehensive study on the privacy leakage of UGRL models concerning link-level membership inference attacks (LMIAs). We specifically focus on designing LMIAs for two different settings and evaluating their effectiveness against four state-of-the-art UGRL algorithms. We investigate how the amount of preserved structural information in UGRL embeddings affects the accuracy of LMIA attacks. Furthermore, we propose simple yet effective defense mechanisms that introduce perturbation to the least important dimensions of embeddings, aiming to mitigate the privacy leakage caused by LMIAs. Our experimental results demonstrate the efficacy of our defense mechanism.

Future work in this area includes exploring the vulnerability of UGRL models to other types of attacks, such as attribute inference attacks [11, 48] and model inversion attacks [14, 55]. Investigating the potential of designing new attacks that leverage knowledge transfer techniques for UGRL models is another interesting direction. This involves utilizing knowledge acquired from a teacher model to attack a student model [46].

9 ACKNOWLEDGEMENT

We thank the anonymous reviewers for their feedback. This project was supported by the National Science Foundation (#CNS-2029038; #CNS-2135988). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

[3] Michael Backes, Mathias Humbert, Jun Pang, and Yang Zhang. walk2friends: Inferring social links from mobility profiles. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 1943–1957, 2017.

[4] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, pages 1–13, 2018.

[5] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[6] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *International Conference on Symposium on Security and Privacy*, pages 1897–1914, 2022.

[7] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.

[8] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974, 2021.

[9] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE transactions on knowledge and data engineering*, 31(5):833–852, 2018.

[10] Yushun Dong, Song Wang, Yu Wang, Tyler Derr, and Jundong Li. On structural explanation of bias in graph neural networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 316–326, 2022.

[11] Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. Quantifying privacy leakage in graph embedding. In *MobiQuitous EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 76–85, 2020.

[12] Cynthia Dwork. Differential privacy. In *the International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–12. Springer, 2006.

[13] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4), 2014.

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

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

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[17] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *In Proceedings of the Privacy Enhancing Technologies (PoPETs)*, 2017.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing links from graph neural networks. In *Proceedings of the USENIX Security Symposium*, pages 2669–2686, 2021.

[20] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. Node-level membership inference attacks against graph neural networks. *arXiv preprint arXiv:2102.05429*, 2021.

[21] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019.

[22] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022.

[23] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *Proceedings of the USENIX Security Symposium*, pages 1895–1912, 2019.

[24] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the ACM SIGSAC conference on computer and communications security*, pages 259–274, 2019.

[25] Jinyuan Jia, Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. Attriinfer: Inferring user attributes in online social networks using markov random fields. In *Proceedings of the International Conference on World Wide Web*, pages 1561–1569, 2017.

[26] Yigitcan Kaya and Tudor Dumitras. When does data augmentation help with membership inference attacks? In *Proceedings of the International conference on machine learning*, pages 5345–5355, 2021.

[27] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, pages 1–3, 2016.

[28] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, pages 1561–1569, 2017.

[29] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 880–895, 2021.

[30] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. Encodermi: Membership inference against pre-trained encoders in contrastive learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 2081–2095, 2021.

[31] Zhiyuan Liu, Yankai Lin, and Maosong Sun. *Representation learning for natural language processing*. Springer Nature, 2023.

[32] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[33] Saeed Mahloujifar, Huseyin A Inan, Melissa Chase, Esha Ghosh, and Marcello Hasegawa. Membership inference on word embedding and beyond. *arXiv preprint arXiv:2106.11384*, 2021.

[34] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE symposium on security and privacy (SP)*, pages 691–706, 2019.

[35] Xupeng Miao, Wentao Zhang, Yuezihan Jiang, Fangcheng Fu, Yingxia Shao, Lei Chen, Yangyu Tao, Gang Cao, and Bin Cui. P2cg: a privacy preserving collaborative graph neural network training framework. *The VLDB Journal*, pages 1–20, 2022.

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, pages 1–12, 2013.

[37] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the ACM SIGSAC conference on computer and communications security*, pages 634–646, 2018.

[38] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE symposium on security and privacy (SP)*, pages 739–753, 2019.

[39] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[40] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. In *International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 11–20, 2021.

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

[42] Tahleen A Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards fair graph embedding. In *International Joint Conferences on Artificial Intelligence*, IJCAI, pages 3289–3295, 2019.

[43] Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 2020.

[44] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567, 2019.

[45] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *Network and Distributed Systems Security Symposium*, 9(5):24–27, 2018.

[46] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9549–9557, 2021.

[47] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE symposium on security and privacy (SP)*, pages 3–18, 2017.

[48] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 377–390, 2020.

[49] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

[50] Anshuman Suri and David Evans. Formalizing and estimating distribution inference risks. *Privacy Enhancing Technologies Symposium (PETS)*, (4):528–551, 2022.

[51] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the international*

conference on world wide web, pages 1067–1077, 2015.
[52] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.
[53] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2508–2515, 2018.
[54] Xiuling Wang and Wendy Hui Wang. Group property inference attacks against graph neural networks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 2871–2884, 2022.
[55] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. Model extraction attacks on graph neural networks: Taxonomy and realisation. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security*, pages 337–350, 2022.
[56] Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. Linkteller: Recovering private edges from graph neural networks via influence analysis. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pages 2005–2024, 2022.
[57] Tong Wu, Yunlong Wang, Yue Wang, Emily Zhao, and Yilian Yuan. Leveraging graph-based hierarchical medical entity embedding for healthcare applications. *Scientific reports*, 11(1):5858, 2021.
[58] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022.
[59] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of the IEEE conference on computer security foundations symposium (CSF)*, pages 268–282, 2018.
[60] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 6(1):3–28, 2018.
[61] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhunmin Chen, Pengfei Hu, and Yang Zhang. Membership inference attacks against recommender systems. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 864–879, 2021.
[62] Wanrong Zhang, Shruti Tople, and Olga Ohrimenko. Leakage of dataset properties in multi-party machine learning. In *USENIX Security Symposium*, pages 2687–2704, 2021.
[63] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference attacks against graph neural networks. In *USENIX Security Symposium*, pages 4543–4560, 2022.
[64] Xiaoli Zhao, Minping Jia, and Zheng Liu. Semisupervised graph convolution deep belief network for fault diagnosis of electormechanical system with limited labeled data. *IEEE Transactions on Industrial Informatics*, 17(8):5450–5460, 2020.
[65] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

APPENDIX

A PRE-ATTACK ANALYSIS: PROPERTIES OF NODE EMBEDDINGS

Before delving into the details of the attacks, we first investigate the properties of node embeddings that can be exploited by LMIA. Specifically, we execute four UGRL algorithms (node2vec [15], DeepWalk [41], LINE [51], and GAE [27] on five popular network datasets: DBLP, LastFm, Cora, Citeseer, and Pubmed.

Since most UGRL algorithms aim to preserve the structural similarity of nodes in the network, we measure the similarity of embeddings for all pairs of nodes, both in member and non-member links. We consider three widely-used similarity measurements: *dot product similarity*, *cosine similarity*, and *Euclidean distance*. The dot product and cosine similarity measure the similarity between two vectors, while the Euclidean distance measures the distance between two vectors. Larger dot product and cosine similarity values indicate higher similarity, while smaller Euclidean distance values indicate closer proximity.

Figure 7 depicts the distribution of the embedding similarity feature for both member and non-member links in the Cora and LastFm datasets. The similarity feature of each link is represented as the concatenation of dot product similarity, cosine similarity, and Euclidean distance between the embeddings of the two nodes in the link. In the plot, we can observe that for all four UGRL models, the member and non-member links are distinguishable based on the similarity feature. This indicates that there are discernible differences in the similarity measurements of embeddings between nodes that are connected and nodes that are not connected in the original graph, across all the evaluated models.

Dataset	# of nodes	# of edges	# of classes
DBLP	10,000	37,430	2
LastFm	7,624	27,806	18
Cora	2,708	5,429	6
Citeseer	3,312	4,732	7
Pubmed	19,717	44,338	3

Table 5: Description of datasets

B DETAILS OF DATASETS

We conduct our experiments on five datasets, each with its own characteristics. The datasets used in our experiments are as follows:

- **DBLP co-authorship** dataset[5]: The DBLP co-authorship graph is constructed from the DBLP bibliography database. Each node in the graph represents a unique author and is associated with a gender feature. An edge exists between two author nodes if they have collaborated on a publication together. We randomly sampled a subgraph with 10,000 nodes and 45,646 edges from the dataset. The subgraph consists of a mix of male and female authors.
- **LastFm** dataset[6]: The LastFm dataset represents a social network of users. It contains 7,624 nodes representing users, and the edges indicate mutual follower relationships between users. This dataset provides insights into user interactions in the LastFm music platform.
- **Cora** dataset: The Cora dataset is a citation network where the nodes represent scientific publications, and the edges represent citation links between publications. It consists of 2,708 scientific publications classified into seven classes, along with 5,429 citation links.
- **Citeseer** dataset: The Citeseer dataset is another citation network where the nodes represent scientific documents, and the edges represent citation links between documents. It consists of 3,312 nodes classified into six classes, with 4,732 citation links.
- **Pubmed** dataset: The Pubmed Diabetes dataset is a citation network constructed from scientific publications in the field of diabetes. The nodes represent documents, and the edges represent citation links. The dataset includes 19,717 scientific publications from the PubMed database, classified into three classes. It contains a total of 44,338 citation links. Each publication in the dataset is described by a TF/IDF weighted word vector.

Table 5 provides statistical information about these datasets, including the number of nodes, number of edges, and number of classes.

[5]https://data.mendeley.com/datasets/3p9w84t5mr/1
[6]http://snap.stanford.edu/data/feather-lastfm-social.html

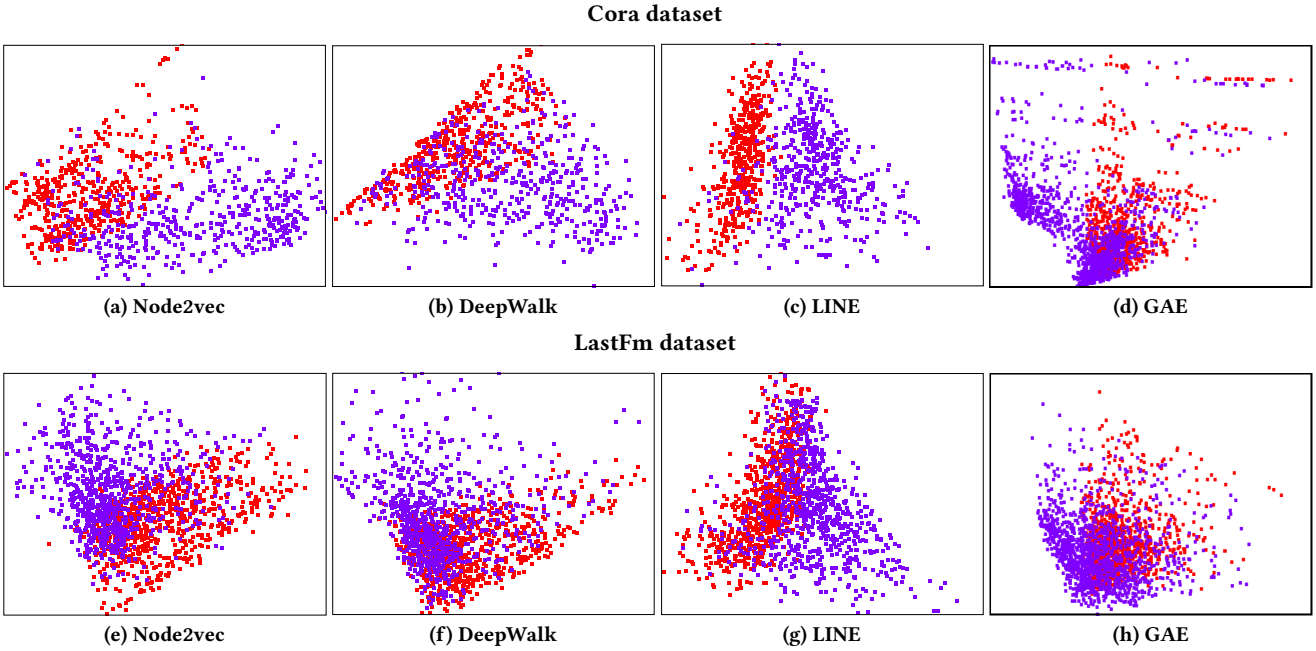


Figure 7: Distribution of LMIA input features for node pairs in both member and non-member links on both Cora and LastFm datasets. The blue dots are member node pairs, red dots are non-member node pairs.

Dataset	node2vec	DeepWalk	LINE	GAE
DBLP	0.65	0.66	0.70	0.71
LastFM	0.70	0.69	0.63	0.68
Cora	0.87	0.86	0.79	0.9
Citeseer	0.84	0.85	0.73	0.85
Pubmed	0.83	0.81	0.83	0.84

Table 6: Quality of the embedding by four UGRL models. Embedding quality is measured as AUC of node classification.

Dataset	node2vec	DeepWalk	LINE	GAE
DBLP	0.72	0.7	0.82	0.65
LastFm	0.78	0.71	0.82	0.64
Cora	0.8	0.72	0.94	0.71
Citeseer	0.81	0.67	0.91	0.74
Pubmed	0.91	0.83	0.82	0.77

Table 7: Attack accuracy of the clustering phase in Attack 1.

These datasets serve as representative examples for evaluating the performance and privacy implications of our proposed methods.

C ACCURACY OF UGRL MODELS

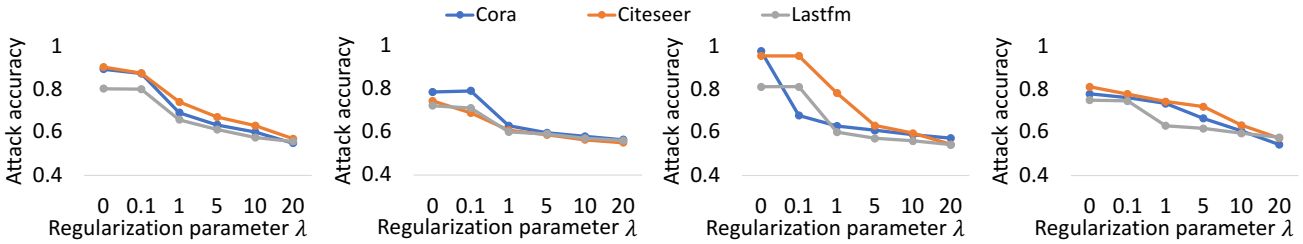
Before conducting attacks on the UGRL algorithms, we first evaluate the quality of the embeddings generated by these algorithms. This evaluation helps to justify why these algorithms are worth targeting. Table 6 presents the results of the embedding quality evaluation. Since the classification tasks for the LastFm, Cora, Citeseer, and Pubmed datasets are multi-label classifications, we measure the macro average AUC as an indicator of embedding quality. We observe that the AUC values for all four UGRL algorithms are significantly higher than random guesses. Note that the macro average AUC of random guesses for multi-label classification is 0.5.

D ATTACK ACCURACY OF CLUSTERING PHASE IN ATTACK 1

Table 7 presents the attack accuracy of the clustering phase of Attack 1 (Section 4.1). The accuracy is assessed on the attack training dataset. We observe that the clustering accuracy ranges from 0.64 to 0.94. There is a strong correlation between the accuracy of the clustering phase and the accuracy of the attack. In particular, we observe higher (lower, resp.) attack accuracy during the inference phase when the clustering accuracy is higher (lower, resp.). For instance, when GAE model is used as the target model and DBLP dataset as the target graph, the clustering accuracy and the attack accuracy are 0.65 and 0.63 respectively. Similarly, with LINE as the target model and Cora dataset as the target graph, the clustering accuracy and the attack accuracy are 0.94 and 0.93 respectively. Furthermore, we consistently observe that the attack accuracy remains lower than that of the clustering accuracy.

E ATTACK AUC OF ATTACK 2 UNDER SETTING 1

Table 8 displays the AUC values of Attack 2, Baseline-2 and Baseline-4 methods under Setting 1. We do not consider Attack 1, Baseline-1, and Baseline-3 as they are not supervised MIA attacks and thus cannot be evaluated with AUC. We have the following observations. First, our attack is highly effective as the attack AUC values are within the range of [0.71, 0.99], which is significantly higher than



(a) node2vec **(b) DeepWalk** **(c) LINE** **(d) GAE**

Figure 8: Defense performance of Baseline-2 with different regularization parameter λ.

Settings	node2vec			DeepWalk			LINE			GAE		
	Attack 2	B2	B4	Attack 2	B2	B4	Attack 2	B2	B4	Attack 2	B2	B4
DBLP	0.74	0.70	0.71	0.71	0.68	0.67	0.98	0.64	0.61	0.73	0.65	0.65
LastFm	0.81	0.75	0.78	0.81	0.69	0.68	0.93	0.62	0.60	0.81	0.69	0.68
Cora	0.93	0.85	0.77	0.87	0.72	0.75	0.99	0.6	0.62	0.94	0.73	0.67
Citeseer	0.95	0.87	0.76	0.89	0.77	0.76	0.99	0.58	0.58	0.85	0.73	0.71
Pubmed	0.93	0.88	0.89	0.91	0.82	0.86	0.93	0.61	0.61	0.89	0.73	0.74

Table 8: Attack AUC of Attack 2, Baseline-2, and Baseline-4 under Setting 1 (non-transfer setting). We do not consider Attack 1, Baseline-1, and Baseline-3 as they are not supervised MIA attacks and thus cannot be evaluated with AUC. The best AUC performance for each UGRL model and each dataset is highlighted with olive color.

the random guess values (0.5). Second, similar to the results shown in Figure 2, LMIA demonstrates significantly higher AUC against LINE compared to node2vec, DeepWalk, and GAE. This is consistent with our observations of other accuracy metrics (attack accuracy and TPR@FPR) in Section 5. Third, our attack outperforms the two baselines in all the settings. The difference in AUC values between our method and the baselines can be as high as 0.41 (e.g., Attack 2 vs. Baseline-2 with the Citeseer dataset in Table 8.).

F DEFENSE PERFORMANCE OF BASELINE-2

We present the defense performance of Baseline-2 (AdvR) in Figure 8. We observe that, while the defensive effectiveness strengthens with increasing values of λ, AdvR can downgrade the attack performance to around 0.5 (random guess) when the regularization parameter λ grows to no less than 10.