GRAND : Graph Reconstruction from Potential Partial Adjacency and Neighborhood Data

Sofiane Azogagh
Université du Québec à Montréal
Montréal, Canada
azogagh.sofiane@courrier.uqam.ca

Zelma Aubin Birba
Université du Québec à Montréal
Montréal, Canada
birba.zelma_aubin@courrier.uqam.ca

Josée Desharnais
Université Laval
Québec, Canada
josee.desharnais@ift.ulaval.ca

Sébastien Gambs
Université du Québec à Montréal
Montréal, Canada
gambs.sebastien@uqam.ca

Marc-Olivier Killijian
Université du Québec à Montréal
Montréal, Canada
killijian.marc-olivier.2@uqam.ca

Nadia Tawbi
Université Laval
Québec, Canada
nadia.tawbi@ift.ulaval.ca

Abstract

Cryptographic approaches, such as secure multiparty computation, can be used to securely compute a function of a distributed graph without centralizing the data of each participant. However, the output of the protocol can leak sensitive information about the structure of the original graph. In particular, we propose an approach by which an adversary observing the result of a private protocol for the computation of the number of common neighbors between all pairs of vertices, can reconstruct the adjacency matrix of the graph. In fact, this can only be done up to co-squareness, a notion we introduce, as two different graphs can have the same matrix of common neighbors. To realize this, we consider two adversary models, one who observes the common neighbors matrix only and a more informed one that has partial knowledge of the original graph. Our results demonstrate that, from their common neighbors matrix, graphs can be reconstructed with high accuracy (up to co-squareness). The proposed reconstruction is also interesting in itself from the point of view of graph theory.

CCS Concepts

• **Security and privacy** → *Distributed systems security*; *Data anonymization and sanitization*; **Social network security and privacy**.

Keywords

Graph reconstruction, edge prediction, private computing, common neighbor matrix.

ACM Reference Format:
Sofiane Azogagh, Zelma Aubin Birba, Josée Desharnais, Sébastien Gambs, Marc-Olivier Killijian, and Nadia Tawbi. 2025. GRAND : Graph Reconstruction from Potential Partial Adjacency and Neighborhood Data. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3736988

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15572142.

1 Introduction

Graphs have emerged as the predominant format for representing relational data, as they naturally capture both the relationships and structures inherent in such datasets. Indeed, from social networks [24] to biological systems [18], the interconnection of entities can be easily visualized and understood through graphs. However, as data becomes increasingly decentralized, new challenges arise with respect to its analysis. For example, in a scenario in which the structure of a graph is distributed across multiple parties (*e.g.*, a social network distributed across multiple servers or a shared knowledge graph), the objective might be to study this structure without centralizing this data and without each party disclosing the private details of their segment.

For instance, in this setting, edge prediction and edge removal techniques can be used to improve the quality of local knowledge. More precisely, edge prediction aims to imply the existence of an edge between two vertices, with typical use cases being in recommendation systems or in social networks [9] but also in anomaly detection, influence analysis and community detection [5]. In contrast, edge removal has for objective to decide if an edge between two vertices should not exist. This can be used, for example, as a counter-measure to data poisoning, in which adversarial edges may have been introduced by an adversary to compromise graph integrity [25, 26, 28]. One of the classical techniques for predicting the existence/non-existence of an edge is based on the computation of the number of common neighbors and infers that two vertices sharing several, respectively few, neighbors should probably be linked, respectively unlinked. For instance, Crypto'Graph [1] enables the oblivious computation of the number of common neighbors between a pair of vertices from a graph distributed among two participants. From this number of common neighbors, each participant can decide whether or not any pair of vertices should share an edge in their own local graph, and thus perform local edge prediction or removal. In addition, a private protocol to compute the common neighbors matrix on a distributed graph could also be applied in the context of social media platforms trying to improve their recommendation algorithms by computing the number of common friends of their users across their joint networks, as explored previously in the literature [1, 6].

Summary of contributions. In this work, we address this issue by precisely characterizing what can be learned by an honest-but-curious participant from the output of the protocol as well as the knowledge of his own local graph. We also consider a less-informed adversary that only has access to the output of the computation of the number of common neighbors matrix, and tries to infer the structure of the graph. Our main contributions can be summarized as follows.

- We developed a novel graph reconstruction approach leveraging the common neighbors matrix. To achieve this, we designed topological attacks that infer information on the graph structure based on the number of common neighbors. The knowledge of the graph obtained from these topological attacks is then used to enhance the spectrum-based attack introduced in a seminal work by Erdős *et al.* [8].
- Additionally, we have introduced a new notion of equivalence between graphs based on their common neighbors matrices, as well as an appropriate metric to fairly compare the reconstructed graph and the original one.
- We demonstrate through experiments on real-world datasets that it is possible to perfectly reconstruct certain graphs even without a prior knowledge on their structure. We also show how a partial knowledge of such structure can be leveraged to achieve a more accurate reconstruction.

Outline. First, we formalize the problem addressed in Section 2 before reviewing the relevant literature in Section 3. Then, in Section 4, we introduce the theoretical background necessary to understand our attacks, such as the properties inferred from the common neighbors matrix that we rely on to reconstruct the graph. Afterwards, we describe the algorithms composing our graph reconstruction method named GRAND in Section 5. Then, we report on the experimental results in Section 6. before analyzing the performance of our reconstruction in the presence of a differential privacy-based defensive mechanism. Finally, we conclude the paper in Section 7 with a discussion on future work.

2 Problem definition

System overview. We consider an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with no self-loop. The neighborhood of a vertex $v \in \mathcal{V}$ on G is denoted as $\Gamma(v)$ and includes all vertices that share an edge with v on G. For any two vertices u and v in $\mathcal{V}$, the *common neighborhood* of $u, v \in V$ on G is the set $\Gamma(u) \cap \Gamma(v)$. In particular, we are interested in the *common neighbors matrix*, in which each entry (u, v) contains the number of common neighbors of u and v in the graph G. It is easy to see that this matrix is, in fact, the result of the matrix multiplication of the adjacency matrix of G by itself, which we denote by G^2. In addition, its diagonal is the degree sequence of G : $G^2(u, u) = |\Gamma(u)|$. Table 1 summarizes the notations that we will be using in the paper.

Adversary models. We consider two types of adversaries. The first one corresponds to an external attacker obtaining the result of a private computation of the common neighbors matrix on the graph G. This type of attacker does not have any prior knowledge about G and aims at reconstructing it solely based on G^2. This adversary model has already been explored in previous work [8]. We designate this type of attacker as *uninformed.*

The second adversary model is motivated by the more general setting in which the attacker has some prior knowledge of G (*e.g.*, the existence or non-existence of some of the edges of G). Such an adversary could be, for instance, an honest-but-curious participant who inputs their subgraph of the graph G to the private protocol to obtain G^2 as output, and tries to reconstruct G from the knowledge of their subgraph and G^2. We express the prior knowledge of the attacker as a set of edges that exist in the graph denoted $\mathcal{E}_1$, as well as a set of edges that do not exist called $\mathcal{E}_0$. The rest of the possible connections between vertices are considered unknown. Remark when $\mathcal{E}_1 = \mathcal{E}_0 = \emptyset$, the second adversary model corresponds to the first, hence leading to a more general problem. We call such an attacker an *informed* adversary.

Problem statement. From the above adversary models, we devise the following problem statement:

Problem 1 (Reconstructing G from $G^2, \mathcal{E}_0, \mathcal{E}_1$). *Let G be a graph, reconstruct G from the knowledge of its common neighbors matrix G^2 and the lists $\mathcal{E}_0$ and $\mathcal{E}_1$ of non-existing and existing edges in G.*

Table 1 gives a summary of the notations used in the paper.

$\mathcal{V}$	set of vertices (*i.e.,* vertices)
$\mathcal{E}$	set of edges (*i.e.,* links)
$\mathcal{E}_1$	set of existing edges known by adversary
$\mathcal{E}_0$	set of non-existing edges known by adversary
$G = (\mathcal{V}, \mathcal{E})$	graph of vertices $\mathcal{V}$ and edges $\mathcal{E}$
$\Gamma(x) \subseteq \mathcal{V}$	neighbors of x in G (*i.e.,* $\{v \in \mathcal{V} \mid (v, x) \in \mathcal{E}\}$)
$\Gamma^\star(x) \subseteq \mathcal{V}$	neighbors of x in $G^\star$
G^2	common neighbors matrix of G (*i.e.,* $G^2(u, v) = |\Gamma(u) \cap \Gamma(v)|$)
$\Gamma^2(x)$	neighbors of x in G^2, that is, vertices that have at least one common neighbor with u
$M|_E$	matrix M restricted to E
ρ	Proportion of existing and non-existing edges known by attacker

Table 1: Summary of notations.

Some related works presented in the next section have tackled the specific case of the reconstruction of G by an uninformed adversary. We suspect that this problem may be NP-hard, as it is really close to known NP-complete problems, such as the Intersection Pattern problem [3] and the Square Root Graph problem [15] (in the latter, the square root of a graph is different from the definition we state in this paper, as is discussed in the next section). However, we have not found a theoretical analysis of this exact problem, which is not standard in graph theory, as isomorphic graphs do not have the same square matrix in general (see the discussion in Section 4.2).

3 Related Work

The reconstruction of graphs has received much attention over the years, with the motivation coming from different use cases, such as recommendations in social networks [20, 24], uncovering hidden interactions between proteins or molecules [17] as well as the discovery of unknown relationships between criminal organizations [14]. While the objective remains to reconstruct the network as accurately as possible, the various approaches differ in their initial knowledge and the adopted approaches.

However, the problem of recovering G from G^2 as defined in Section 2 has received much less attention. More precisely, in the field of linear algebra, the problem of finding the square root of a matrix (the adjacency matrix of G is one square root of the matrix G^2) is a well-studied problem. However, the matrix studied does not necessarily correspond to a binary matrix (*i.e.*, the adjacency matrix of a graph). By construction, G^2 is a positive semi-definite matrix (psd), because there exists a matrix such that $BB^T = G^2$, this matrix being $B := G = B^T$. As G is symmetric, Theorem 7.26 in [11] states that there exists a unique symmetric positive semi-definite matrix serving as the square root of this G^2. However, this theorem does not solve our problem because G, the square root that we are looking for, may not be itself psd. Furthermore, the psd solution is not even a binary matrix in general.

In the field of graph theory, different definitions of G^2 co-exist, with many defining it as a binary matrix. For instance, in [16] the entries of the matrix represent the existence of a path of length at most 2 between the vertices, which is $G \cup G^2$ with 0-1 entries, while in [2] it represents the existence of a path of length exactly 2 between the vertices. As a result, the considered problem is quite different. In our setting, the value is the *number of paths of length exactly 2*, which means that G^2 may not be a binary matrix and G is not a subgraph of it.

To the best of our knowledge, the work closest to ours is [8], which proposes a method for the reconstruction of the adjacency matrix of a (bipartite) graph, from the knowledge of the common neighbors information between all pairs of vertices. Their approach, which we revisit and improve as part of our contribution, relies on the spectral domain of the adjacency matrix. More precisely, it exploits the properties of the singular value decomposition of the common neighbors matrix to iteratively reconstruct an approximation of the desired graph. However, relying on the graph's spectral domain makes them susceptible to reconstruct a co-spectral graph [21], which is a graph exhibiting an identical spectrum but non-isomorphic by definition. The reconstructed graph may not even have the same common neighbors matrix, as discussed later.

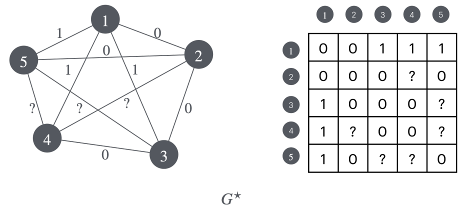
The problem of reconstructing G from its sequence of degrees (the diagonal of G^2) has also received a lot of attention [4, 12] but the proposed solutions to solve this problem potentially require exploring an exponential number of solutions [12]. Note that this problem is more general than the one we tackle, since we have more information. Also, in some settings we do not study, the diagonal might not be available to the attacker.

This paper goes beyond existing work by considering a more powerful adversary than previously studied in works such as [4, 8]. We propose a solution that achieves the best known reconstruction performance of an undirected graph solely based on its common neighbors matrix. In addition to that setting, we study the adversarial model in which, besides the common neighbors matrix, the adversary possesses some partial knowledge of the original graph. Our experiments demonstrate that this additional knowledge grants the adversary greater capacity to reconstruct the graph.

4 Theoretical building blocks

Recall that our objective is, given some partial knowledge of the adjacency on G, as well as G^2 the common neighbors matrix, to reconstruct G. Our attack reconstructs G through different steps. Figure 1 depicts our representation of the reconstructed graph $G^\star$ during the attacks. It can be viewed as a complete graph in which the edges are labeled respectively as 0, 1 or ? if in G, they exist, do not exist or we do not know if this is the case.

During the reconstruction, the cells of $G^\star$ are updated to reflect the information learned. Hence, cells that have value ?s are turned into $1s$ or $0s$ depending on the inference performed in $G^\star$ when we learn the existence or absence of an edge in G. The final reconstructed graph obtained at the end of the attacks, which we denote $\hat{G}$, contains only 0s and 1s.



	1	2	3	4	5
1	0	0	1	1	1
2	0	0	0	?	0
3	1	0	0	0	?
4	1	?	0	0	?
5	1	0	?	?	0

Figure 1: Example of a graph $G^\star$ and its adjacency matrix.

4.1 Spectrality vs topology

Consider the adjacency matrix of the graph G, also denoted as G. It can be expressed as:

$$G = U \Lambda U^T \tag{1}$$

in which U is an orthogonal matrix and Λ is a diagonal matrix containing the eigenvalues of G. This decomposition shows that there are two equivalent representations of the graph information: the left-hand side of Equation (1), which we call the *topology domain*, and the right-hand side, which we call the *spectral domain.* As demonstrated by this equation, these two domains encode the same information in different forms. As mentioned in [22], similar to how signals can be analyzed in both time and frequency domains via Fourier transforms, graphs can be studied in both topological and spectral domains, with each domain better suited for analyzing different properties of the graph.

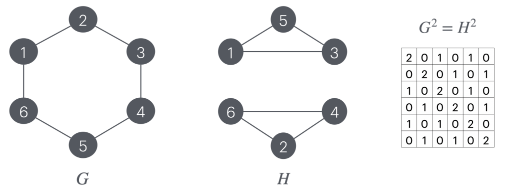
The topology domain is the most intuitive representation of a graph, in which vertices and their connections can be directly visualized and understood by humans. In this domain, it is easy to reason about structural properties like paths, cycles, and connectivity patterns between vertices. Thus, this representation naturally lends itself to analyzing local properties and deriving insights about the graph's structure through direct observation.

In contrast, the spectral domain studies graphs through the eigenvalues and eigenvectors of their adjacency matrices. While less intuitive to human understanding, this domain has given rise to spectral graph theory - a rich field that connects graph properties to linear algebra. The interpretation of eigenvalues and eigenvectors is not immediately obvious, but their study has revealed deep connections to many graph properties like connectivity, clustering and symmetry.

4.2 Non-unicity of the solution

Two non-isomorphic graphs can share the same common neighbors matrix, as do those illustrated in Figure 2, together with their common neighbors matrix. We call this new notion of equivalence between graphs *co-square equivalence.* Let us make precise that co-square graphs must have exactly the same common neighbors matrices *without permuting any lines or columns.* This is uncommon in graph theory but it is necessary in our practical setting, since the vertices are labeled and distinctive. One implication of this is that co-squareness is incomparable with the notion of isomorphism of graphs. Indeed, there are isomorphic graphs whose matrices of common neighbors are only equal after a symmetric permutation of lines and columns (*e.g.*, a well-chosen permutation of vertices of a graph with vertices of different degrees). Conversely, there are non-isomorphic graphs with the same matrix of common neighbors (such as those of Figure 2). Even without the interdiction of permuting lines and columns, to the best of our knowledge, co-square graphs do not belong to any known graph theory category.

Co-squareness is also incomparable with co-spectrality. The graphs in Figure 2 are not co-spectral as they have different eigenvalues, which are $\{-2,-1,-1,1,1,2\}$ for G and $\{-1,-1,-1,-1,2,2\}$ for H. Conversely, there exist non-isomorphic graphs sharing the same eigenvalues [21, 23]. These graphs may not even have the same sequences of degrees, which results in them not being co-square. More details about co-spectrality will be given in Section 4.3.2.



2	0	1	0	1	0
0	2	0	1	0	1
1	0	2	0	1	0
0	1	0	2	0	1
1	0	1	0	2	0
0	1	0	1	0	2

Figure 2: Two co-square (and non co-spectral) graphs

In summary, we are in the presence of three notions of equivalences between graphs. Isomorphism and co-spectrality are incomparable by definition, with co-squareness being also incomparable to both notions. As non-unicity can lure us to construct a co-square graph, prior knowledge of the target graph's structure might help to distinguish which is the right one. For instance, the *a priori* information that $(1,3)$ does not exist in G removes the possibility that H would be the target graph. For instance, in our work, such information is exploited by the informed adversary.

4.3 Inferring properties of G

Hereafter, we leverage the common neighbors information G^2, as well as the partial knowledge $\mathcal{E}_0, \mathcal{E}_1$ to deduce properties about G. Recall that the partial reconstruction of G is denoted by $G^\star$ while the number of common neighbors between vertices $u, v \in \mathcal{V}$ is given by $G^2(u,v)$ and the degree of vertex u is $G^2(u,u)$.

4.3.1 *Topological properties.*

Proposition 1 (Rows of G^2 and degrees of neighbors). *Let $u \in \mathcal{V}$ be a vertex on G. Then, the sum of the degrees of its neighbors is equal to the sum of the u^{th} row in G^2:*

$$\forall u \in \mathcal{V}, \sum_{v\in\mathcal{V}} G^2(u,v) = \sum_{w\in\Gamma(u)} |\Gamma(w)|. \tag{2}$$

Remark that this is not a complete characterization of a common neighbors matrix, as there are matrices satisfying this property that do not correspond to any graph. However, we can exploit Proposition 1 to infer unknown existing and non-existing edges in G.

Proposition 2 (Completeness of a neighborhood in $G^\star$). *Let $u \in \mathcal{V}$ be a vertex of G. If u has the same degree on G and $G^\star$, then the neighborhood of u on $G^\star$ is complete :*

$$\forall u \in \mathcal{V}, |\Gamma^\star(u)| = |\Gamma(u)| \implies \forall v \in \mathcal{V}\backslash\Gamma^\star(u), (u,v) \notin \mathcal{E}. \tag{3}$$

Similarly, if two vertices u and v have the same number of common neighbors on G and $G^\star$, then the common neighborhood of u and v is complete:

$$\begin{aligned}\forall u,v \in \mathcal{V}, |\Gamma^\star(u) \cap \Gamma^\star(v)| = |\Gamma(u) \cap \Gamma(v)| \implies \\ [\forall w \in \Gamma^\star(u)\backslash\Gamma^\star(v), (w,v) \notin \mathcal{E}].\end{aligned} \tag{4}$$

Proposition 3 (Completing a neighborhood in $G^\star$). *Let $u \in \mathcal{V}$ be a vertex on G. If the following conditions are met :*

1. *u is missing k neighbors (*i.e.*, $|\Gamma(u)| - |\Gamma^\star(u)| = k$);*
2. *there are exactly k vertices $v_1, \ldots, v_k$ such that we do not know if (u, v_i) exists;*

then all edges (u, v_i) must exist in G.

Similarly, for common neighborhoods, if:

1. *u and v are missing k vertices in their common neighborhood (*i.e.*, $|\Gamma(u) \cap \Gamma(v)| - |\Gamma^\star(u) \cap \Gamma^\star(v)| = k$);*
2. *there are exactly k vertices $w_1, \ldots, w_k$ such that we do not know if (u, w_i) exists;*

then all edges $(u, w_i), (v, w_i)$ must exist.

Proposition 3 can be seen as the complementary of Proposition 2, in the sense that the latter identifies some 0s in $G^\star$ based on the number of missing neighbors (or common neighbors), while the former uses the same information to identify some 1s.

Proposition 4 (Triangles). *Let u, v be two vertices on $\mathcal{V}$, if the following conditions are satisfied:*

1. *u and v are connected in $G^\star$ (*i.e.*, $(u,v) \in \mathcal{E}_1^\star$);*
2. *u and v have at least one common neighbor (*i.e.*, $G^2(u,v) > 0$);*

then u, v are in $k = G^2(u,v)$ triangles. Their common neighbors are then vertices $w_1, \ldots, w_k \in \mathcal{V}$ such that

$$\forall i \in \{1,\ldots,k\}, G^2(u, w_i) > 0 \text{ and } G^2(v, w_i) > 0.$$

Proposition 5 (Bi-cliques[1]). *Let $U = \{u_1, \ldots, u_p\}$ and $V = \{v_1, \ldots, v_q\}$ be two sets of vertices in $\mathcal{V}$. If the following conditions are satisfied:*

1. *the neighborhood of u_1 is complete : $\Gamma(u_1) = \Gamma^\star(u_1) = \{v_1, \ldots, v_q\}$;*

[1]A bi-clique, or complete bipartite graph, is a bipartite graph in which each vertex is connected to all the vertices in the other partition.

(2) *each vertex u_i, $i \in \{2, \ldots, p\}$ shares exactly q common neighbors with u_1;*

then $U \cup V$ forms a bi-clique, in which U and V are the associated partitions.

Proposition 6 (Bipartition or disconnectedness of G). *G^2 is disconnected if, and only if, G is either disconnected or bipartite.*

An example of such a situation is given in Figure 2, in which both a bipartite graph G and a disconnected graph H have the same common (disconnected) neighbors matrix. Thus, given a disconnected matrix G^2 (which can be determined in linear time), one cannot decide with certainty if the studied graph is disconnected or if it is bipartite. There are examples in which only a bipartite graph exists, so one cannot focus on each of the components to find a co-square graph (which is of course the best one can do if no prior knowledge is given).

4.3.2 *Spectral properties.* G being a real symmetric matrix, its eigenvalue decomposition can be written as :

$$G = U \Sigma U^T.$$

As U is orthogonal, we have $U^T = U^{-1}$, which translates to

$$G^2 = (U \Sigma U^T)^2 = U \Sigma^2 U^T = U \Lambda U^T.$$

Consequently, the problem of reconstructing a matrix G from the spectrum of its square G^2 can be solved by finding the right sign for each eigenvalue. Indeed, for each eigenvalue λ_i of G^2, we need to choose between $+\sqrt{\lambda_i}$ and $-\sqrt{\lambda_i}$ as the corresponding eigenvalue σ_i of G, leading to an exponential number of possibilities to explore. This sign assignment problem is likely what could make the reconstruction problem NP-hard, since there are 2^n possible sign combinations for the eigenvalues. Erdős and co-authors provided in [8] a heuristic method exploiting this insight. More precisely, their Greedy algorithm chooses the right eigenvalue based on the following results from Eckart and Young theorem [7] :

Theorem 4.1 (Eckart and Young). *The best rank-k approximation of G in the Frobenius norm $\|\cdot\|_F$ is given by*

$$\sum_{i=1}^{k} u_i \sigma_i u_i^T$$

in which σ_i is the i-th largest eigenvalue of G and u_i is the corresponding eigenvector.

This powerful theorem allows to reconstruct G iteratively by choosing the sign of $\sqrt{\lambda_i}$ that minimizes the Frobenius distance at each step for $i = 1, \ldots, n$. Additionally, the early iterations are the most important ones, as the largest eigenvalues (which contain the most information) are the first ones. Therefore, any prior knowledge of the graph should be used iteratively to "guide" the reconstruction in the right direction, and the more knowledge the adversary has, the better he can potentially guide the reconstruction.

5 GRAND

Hereafter, we propose several attacks for the reconstruction of G given its common neighbors matrix G^2. We also describe how this reconstruction can be enhanced from the knowledge of the existing and non-existing edges $\mathcal{E}_1, \mathcal{E}_0$. More precisely, we first describe deterministic attacks that can infer *exactly* the existence or non-existence of edges in G based on topological properties. Then, to further improve the reconstruction, we then present heuristic attacks for the reconstruction of G based on the spectrum of G^2 while leveraging the partial reconstruction obtained after the topological attacks. The algorithms for each of the procedures described below are provided in detail in Appendix B.

5.1 Topological attacks

We now present how to leverage the propositions described previously to reconstruct G. These inferences recover G in an exact manner with respect to G^2, which means that they guarantee that the common neighbors in the reconstructed graph match G^2. Note that some of them use prior knowledge $(\mathcal{E}_0, \mathcal{E}_1)$. We start by incorporating the prior knowledge (if any) into the reconstructed graph $G^\star$ as follows:

- All existing edges in the prior knowledge get a label of 1 in $G^\star$, which means that $\forall (u, v) \in \mathcal{E}_1, G^\star(u, v) \longleftarrow 1$.
- Similarly, non-existing edges in the prior knowledge get assigned 0 in $G^\star$, which means that $\forall (u, v) \in \mathcal{E}_0, G^\star(u, v) \longleftarrow 0$.
- G does not contain self-loops, so $\forall u \in V, G^\star(u, u) \longleftarrow 0$.
- All the other possible edges get a label of ?.

DegreeCombinationAttack. Proposition 1 enables the following inference : given a vertex u, if there is only one candidate set of vertices in $\mathcal{V}$, $S = \{w_1, \ldots, w_{|\Gamma(u)|}\}$ such that $\sum_{v \in \mathcal{V}} G^2(u, v) = \sum_{w_i \in S} G^2(w_i, w_i)$, then $(u, w_i) \in G$. One can use this observation to identify potential vertices that belong to the neighborhood of a certain vertex. Similarly, this proposition can also be used to infer non-existing edges in G: given two vertices u, v, if the degree of a vertex v exceeds the sum of the u^{th} row in G^2 (that is, without the degree of u), then $(u, v) \notin \mathcal{E}$. Figure 3 gives an illustration of this attack by inferring the presence of edges in G.

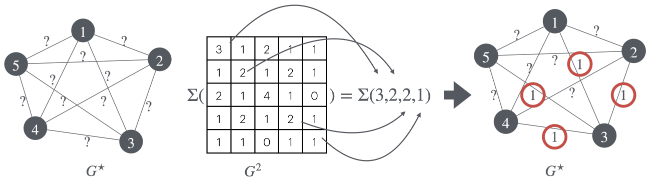
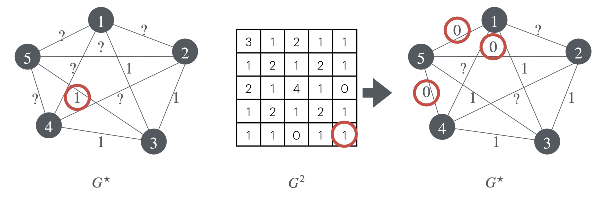


Figure 3: DegreeCombinationAttack. The sum of the row of vertex 3 in G^2 is equal to the sum of the degrees of 1, 2, 4 and 5. Thus, the edges (1, 3), (2, 3), (4, 3), (5, 3) do exist in G.

For this attack to work, one needs to have the degree of each vertex in G. When G^2 is fully available to the adversary, this is given by the diagonal of G^2. Moreover, this attack does not assume that the attacker has prior knowledge of G, allowing them to infer information about G even when $\mathcal{E}_0 = \mathcal{E}_1 = \emptyset$ (*i.e.* an uninformed adversary).

. DegreeMatchingAttack. From Proposition 2, we can derive a procedure to identify the vertices whose neighborhoods have been completely recovered. This attack establishes the non-existence of edges in G and puts 0's in $G^\star$ by comparing the degrees of vertices on $G^\star$ and G. An example of such inference is depicted in Figure 4.



3	1	2	1	1
1	2	1	2	1
2	1	4	1	0
1	2	1	2	1
1	1	0	1	1

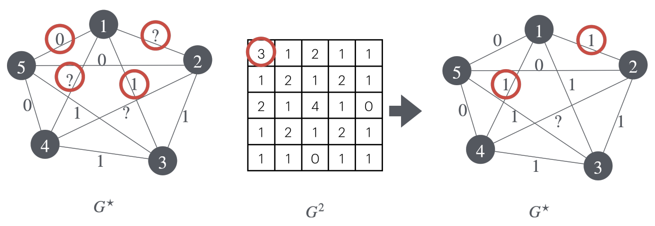
Figure 4: DegreeMatchingAttack. The degree of vertex 5 is equal to the number of its currently known neighbors in $G^\star$. Thus all unknown edges in the neighborhood of 5 cannot exist.

NeighborMatchingAttack. The previous attack can also be applied to elements other than the diagonal of G^2, namely $G^2(u,v), u \neq v$. The intuition behind this reconstruction is the same: if the number of common neighbors between two vertices u, v is the same on $G^\star$ and G, it must be that there is no common neighbor on G other than the ones known in $G^\star$. This allows to insert 0s in $G^\star$ by looking at the known neighbors.

To realize this, consider for instance the special case of zero values in G^2. Such a situation reflects the absence of common neighbors between the corresponding vertices: $\Gamma(u) \cap \Gamma(v) = \emptyset$. Since the edges of $G^\star$ are included in G, we also have $\Gamma^\star(u) \cap \Gamma^\star(v) = \emptyset$. More precisely :

$$G^2(u,v) = 0 \implies [\forall w \in \Gamma^\star(u), (w,v) \notin \mathcal{E}].$$

DegreeCompletionAttack. If u is missing k edges in $G^\star$ to reach a certain degree given by $G^2(u,u)$ and its neighborhood also has k unknown edges, then all the unknown edges have to exist. This attack is depicted in Figure 5.



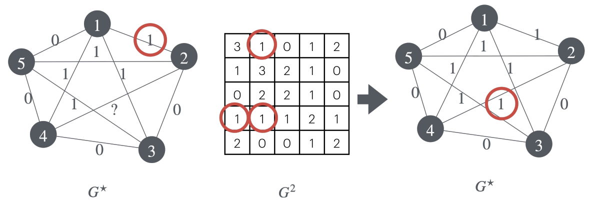
3	1	2	1	1
1	2	1	2	1
2	1	4	1	0
1	2	1	2	1
1	1	0	1	1

Figure 5: DegreeCompletionAttack. Vertex 1 has one known neighbor which is 3, and two unknown edges. Since its degree is 3, all its unknown edges exist in G.

NeighborCompletionAttack. The DegreeCompletionAttack can be adapted to the other information in G^2. If u, v have m common neighbors on $G^\star$ and $m+k$ on G, and u has exactly k unknown edges in its neighborhood, then all k potential neighbors of u must actually be common neighbors of u and v in G.

TriangleAttack. . Proposition 4 allows the identification of triangles from the known edges in G as well as G^2. In Figure 6, triangles can be identified by looking for vertices that share common neighbors with both vertices of an existing edge (u, v).

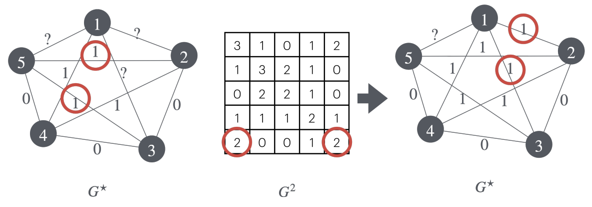
BicliqueAttack. When the neighborhood of a certain vertex u is completely known, one can infer additional information about



3	1	0	1	2
1	3	2	1	0
0	2	2	1	0
1	1	1	2	1
2	0	0	1	2

Figure 6: TriangleAttack. Edge (1, 2) is known and vertices 1 and 2 have one common neighbor. Then, there exists one triangle containing 1 and 2. Notably, 4 is the only vertex that has common neighbors with both 1 and 2. Therefore 4 is the vertex forming a triangle with 1 and 2.

connections between other vertices and u. For instance, if all the $G^2(u,u)$ neighbors of u are known, and v has $G^2(u,u)$ common neighbors with u, then all the neighbors of u are also neighbors of v. Inversely, if the number of missing neighbors for v is equal to the number of missing common neighbors between v and u, then all the missing neighbors of v are among the neighbors of u. Therefore, v does not have another edge with any of the other vertices in the graph. Figure 7 provides an illustration of this inference.



3	1	0	1	2
1	3	2	1	0
0	2	2	1	0
1	1	1	2	1
2	0	0	1	2

Figure 7: BicliqueAttack. The neighbors of vertex 5 (which are 2 and 3) are all known because its degree on $G^\star$ is equal to $G^2(5,5)$. Therefore, vertex 1, which has two common neighbors with 5, is also connected to 2 and 3.

Combining the attacks. Our topological attacks come in two flavors. Some of them, like the DegreeMatchingAttack and NeighborMatchingAttack identify non-existing edges in the graph, changing ?s into 0s in $G^\star$. Complementarily, attacks such as the DegreeCompletionAttack and NeighborCompletionAttack identify existing edges, changing ?s into 1s states. Since the changes made by one type of attack can influence the output of the other, the process needs to be repeated until none of the procedures update $G^\star$.

5.2 Spectral attack

Hereafter, we introduce algorithms for the reconstruction of G based on its spectrum. Our solution revisits the greedy sign assignment algorithm of [8], and improves it by taking into account an additional constraint related to the partial knowledge of the graph G. The main idea here is to iteratively find the best sign assignment for each eigenvalue, starting from the first. Let us first denote by $\mathcal{E}_0^\star$ the 0s in $G^\star$ so far, $\mathcal{E}_1^\star$ the 1s, and $\mathcal{E}^\star = \mathcal{E}_1^\star \cup \mathcal{E}_0^\star$. For each eigenvalue, the best sign is the one which satisfies the following two constraints :

- *Closeness to a binary matrix :* Similarly to [8], we impose that the reconstructed matrix be a binary one.
- *Closeness to the known adjacency information:* To account for the already known information on the graph, we impose that our reconstruction be close to it. Namely, for a given potential edge (u, v), our reconstructed adjacency matrix should contain a value of 1 (respectively 0) at index (u, v) if $(u, v) \in \mathcal{E}_1^\star$ (respectively $\mathcal{E}_0^\star$).

We formalize the above constraints as a minimization problem, in which for the *i-th* singular value, we choose the sign $s \in \{+, -\}$ that minimizes the joint distance with our objectives :

$$min_{s\in\{+,-\}} \alpha \left\|M^s - BM^s\right\|_F + \beta \left\|M^s|_{\mathcal{E}^\star} - G^\star|_{\mathcal{E}^\star}\right\|_F,$$

in which M^s denotes the reconstructed matrix computed with singular value $s\sqrt{\Lambda(i,i)}$. BM^s is the binary version of the reconstructed matrix M^s. Precisely, for each row i and column j of M^s,

$$BM^s(i,j) = \begin{cases} 1 & \text{if } M^s(i,j) > t, \\ 0 & \text{otherwise,} \end{cases}$$

in which t is a chosen binarization threshold. The α term for the reconstruction of a binary matrix, and the β term allows the reconstruction of a matrix close to the known information in $G^\star$. Thus, α and β are weighting factors that enables to set a trade-off between the two constraints. This is especially suitable for the case when we already know a significant part of the matrix, as we can boost the second term to benefit from that partial knowledge. $M^s|_{\mathcal{E}^\star}$ denotes M^s restricted to the indices (rows and columns) contained in $\mathcal{E}^\star$.

5.3 Targeted error forgetting

Often, the spectral attack will reconstruct G with errors. While correcting errors in the reconstruction is not straightforward, we can leverage our knowledge of G^2 to detect some of them. Indeed, consider two vertices u, v that do not have the right number of common neighbors in the reconstruction. Then, it must be the case that there is an error either in the neighborhood of u, in the neighborhood of v or both. Therefore, in such cases we can rollback the updates of the spectral attack, and put ?s in the neighborhood of u and v. This heuristic cannot delete all the errors in the reconstruction (since u and v could have the right number of common neighbors, but the wrong common neighbors), but we experimentally observed that it allows to correct most of them, at the expense of some errors.

5.4 Co-square graph instantiation

Because of the possible existence of a co-square graph, some entries in G^2 can be achieved with different attributions of edges in $G^\star$. An example of such a scenario, extracted from Netscience, is depicted in Figure 8–the number labels are the ones of Netscience.

For each separate component, the position of dotted edges with respect to the blue vertices can be in three possible ways to yield the values in G^2: as they are pictured, or vertically or in diagonal. For instance for the left example, the edges can be replaced by edges $(53, 226)$ and $(54, 227)$ or by $(53, 227)$ and $(54, 226)$. The first graph of 5 vertices (hence detached from Netscience) and its two co-square graphs (which we just described) are the smallest co-square graphs. In addition, they are isomorphic, in contrast to the first

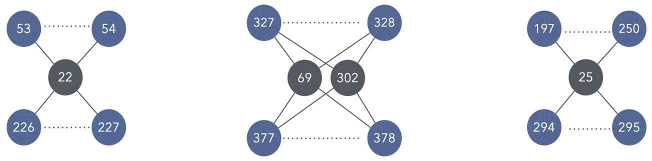


Figure 8: Co-square sub-graphs in Netscience. Plain lines are reconstructed edges while dotted lines are missing edges in the reconstruction. For each component, adding the two missing edges between any combination of two of the four blue vertices yields a graph with equal matrix of common neighbors.

co-square graphs presented in Section 4.2, which are the smallest non-isomorphic co-square graphs.

In presence of co-square graphs or subgraphs, prior knowledge of these components of G is essential for reconstruction. In our method, when no prior knowledge is given, or when it does not contain any information about the co-square-inducing vertices, we instantiate one co-square of the target graph after identifying the co-square-inducing vertices based on G^2.

5.5 Pipeline

Our attack pipeline, depicted in Figure 9, unfolds as follows:

(1) We initialize $G^\star$ with the known information $\mathcal{E}_0, \mathcal{E}_1$. When $\mathcal{E}_0 = \mathcal{E}_1 = \emptyset$, the adjacency matrix of this graph contains ?s everywhere, except on the main diagonal in which there are 0s due to the fact that there is no loop in the graph;
(2) We run the topological attacks until no new information is discovered on G.
(3) We then pass the partial information gathered so far $G^\star$ to the spectral attack, which optimizes the reconstruction to be close to the known edges and non-edges learnt.
(4) Since the spectral method might introduce errors, we perform a targeted error forgetting that removes the adjacency information of vertices in $G^\star$ based on $G^{\star 2}$ and G^2.
(5) The error forgetting introduces ?s in $G^\star$. To recover new values, we run another round of topological attacks.
(6) To identify edges that were not recovered so far because of co-squareness, we perform co-square subgraph instantiation.

6 Experiments

In the following, we present the performance of our algorithms evaluated on various real-world datasets presented in Table 2. More precisely, we compare our reconstruction performance with the one reached by [8], while taking into account the prior knowledge $\mathcal{E}_0, \mathcal{E}_1$. We simulate this prior knowledge by a uniform sampling of the matrix G, with a proportion of ρ. In other words, the attacker is allowed to know a proportion ρ of 1s (edges) and 0s (non-edges) from G (*i.e.*, informed adversary).

We evaluate the reconstruction performance of both adversary models detailed in Section 2 with an amount of prior knowledge on G between 0 (*i.e.*, uninformed adversary) and 1 (*i.e.*, fully informed adversary). The following two metrics are used to quantify the reconstruction performance:

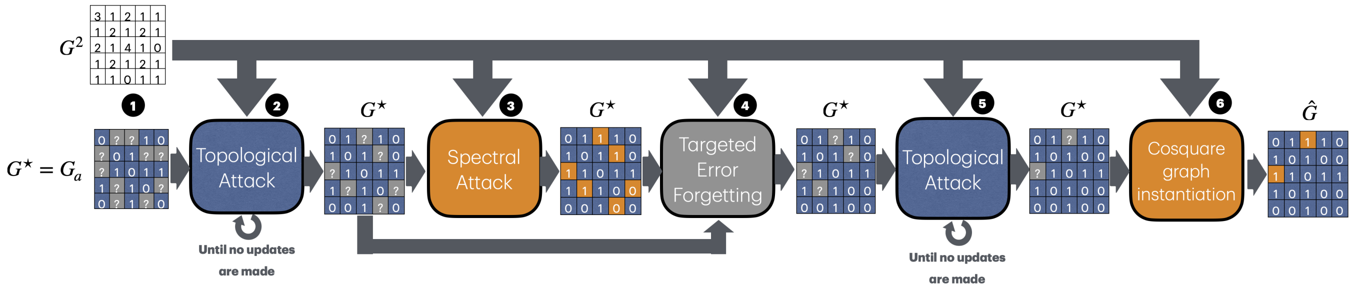


Figure 9: GRAND

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Density	Domain
Netscience [19]	379	914	0.0127	Co-authorship
Bio-diseasome [19]	516	1188	0.0089	Human diseases
Polblogs [13]	1490	16715	0.0151	Political blogs
Cora [13]	2485	5069	0.0016	Citations

Table 2: Datasets characteristics.

- *Relative Absolute Error (RAE).* The RAE measures the amount of wrong predictions (positives and negatives) in proportion of the number of edges in G (also used in [8]).

$$RAE = \frac{\|G - \hat{G}\|_F}{\|G\|_F}.$$

- We introduce the *Common Neighbors Error*, as a measure of error with regard to the common neighbors matrix. This measure serves as an evaluation metric that takes into account the co-square equivalence explained in Section 4.2. To a real attacker, it can also serve as an indicator of how close they are to the real graph without knowing it beforehand.

$$CNE = \frac{\|G^2 - \hat{G}^2\|_F}{\|G^2\|_F}.$$

The parameters of the spectral attack are chosen as follows :

- $\beta = \frac{2\cdot|\mathcal{E}^\star|}{|\mathcal{V}|^2}$. Since $\mathcal{E}^\star$ contains the indices of the known information in the reconstructed matrix, this choice of β ensures that the more we know about G, the more we use that information in our reconstruction. When G is completely known, $\beta = 1$.
- $\alpha = 1 - \beta$, to complement for the unknown information.
- $t = 0.5$. This choice acts as a middle ground for binarizing values between 0 and 1. In [8], Erdős and co-authors also reported this value as the best one for their approach.

The reported values are averaged over 10 experiments.

Reconstruction by an uninformed adversary. In this scenario, the attacker gets no prior knowledge, which means that all the edges in $G^\star$ are labeled as ?s.

Reconstruction by an informed adversary. For this scenario, the common neighbors matrix G^2 is provided to our attack pipeline, as well as $G^\star$, initialized as described previously. However, again we want to clarify that the approach of [8] does not take into account the prior knowledge of the graph, and there is no straightforward way to include it either. To have a fair comparison with that approach, we consider in this scenario that when using the approach of [8], the attacker reconstructs a graph $\hat{G}$ that they overwrite using prior knowledge. Namely, edges in $\mathcal{E}_0$ are removed from $\hat{G}$ and edges in $\mathcal{E}_1$ are added to $\hat{G}$. All other possible edges stay as they were in $\hat{G}$. We call this modified approach of [8] the *informed Erdös approach.*

From the perspective of G. Figure 10 presents the RAE of both approaches with respect to the amount of prior knowledge. The shade for GRAND denotes the maximum and minimum RAE depending on the instantiated co-square subgraph. Both approaches benefit from additional prior knowledge, and GRAND consistently reaches a lower RAE on all datasets, for all values of ρ.

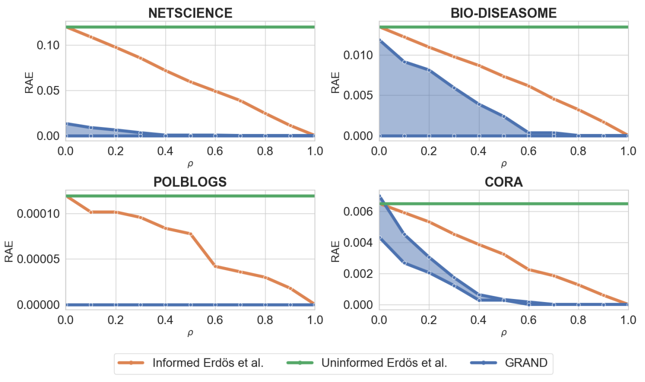


Figure 10: Relative Absolute Error (RAE) with respect to the amount of prior knowledge ρ

From the perspective of G^2. The results in terms of CNE, presented in Figure 11, complement previous ones by alleviating the consideration for co-square graphs. Without any prior knowledge, GRAND reconstructs perfect co-square graphs of the target graph for Polblogs, bringing the CNE to 0. Similar results have also been observed on Netscience and Bio-diseasome. This shows that GRAND actually reaches the theoretical maximum reconstruction performance, since without prior knowledge the attacker cannot guess which co-square graph corresponds to the target graph.

Complexity. Most of our deterministic attacks iterate on all the possible edges (in the worst case), making their complexity $O(|\mathcal{V}|^2)$.

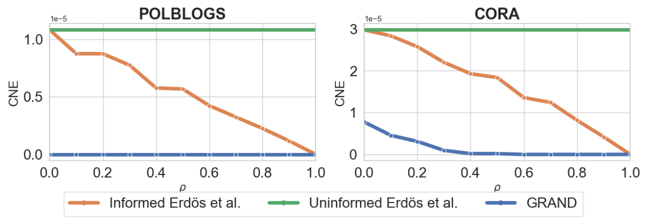


Figure 11: Common Neighbors Error with respect to the amount of prior knowledge ρ

The only exception to that is DegreeCombinationAttack, which might result in exponential complexity. In practice, this issue is dealt with by performing that attack only for vertices with degree in $\{1, 2\}$. However, the complexity of the complete pipeline is dominated by the spectral attack, which, similar to [8] is in $O(|\mathcal{V}|^3)$ because of the singular value decomposition. Regarding the SVD, its truncated version can be used to reduce the computational complexity to $O(k|\mathcal{V}|^2)$, with k being the rank of the decomposition, which establishes a tradeoff between efficiency and accuracy.

In our experiments, the complete pipeline takes 18s for Polblogs and 5min for Cora which are the most time-consuming, and takes a few seconds for Netscience and Bio-diseasome. These timings were obtained on a Macbook Pro 2020 with 16GB memory and an M1 chip. Additional experimental results can be found in Appendix A.

GRAND in the presence of defenses. We also study the reconstruction capacity of our method in the presence of existing defensive mechanisms. In this area, differential privacy-based methods [10, 27] constitute the state-of-the-art.

We choose to study our reconstruction capacity for a graph protected using the PrivGraph [27] method. PrivGraph uses noise addition to privately determine graph properties such as inter-community and intra community edge weights before reconstructing a similar graph to the original based on the (private) learned properties, with the aim of achieving edge-privacy.

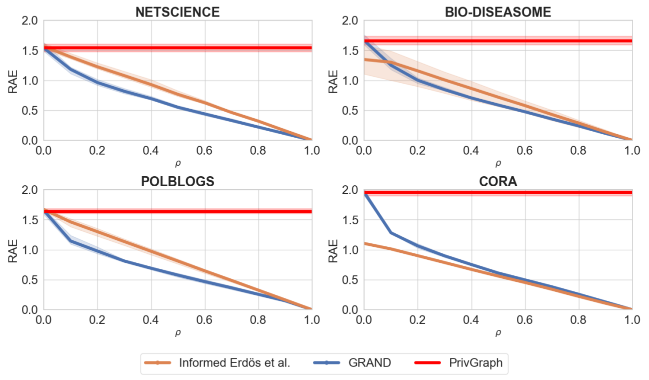


Figure 12: Attacks on a graph protected by PrivGraph. The blue and orange lines denote the RAE after reconstruction. The red line is the RAE measure between G' and G.

This experiment was designed for the following scenario: a party produces an edge-private graph G' from their initial graph G, then outputs the common neighbors matrix G'^2 of the edge-private graph. The adversary then tries to reconstruct the initial graph from the common neighbors matrix of the edge-private one. A before, we consider that the attacker can have some prior knowledge on the initial graph. We vary the privacy parameter ϵ of PrivGraph between 0.5 and 3.5, and present average values of the RAE for each dataset in Figure 12. Again, the partial knowledge of the graph allows GRAND to obtain a better reconstruction than [8].

7 Conclusion

In this paper, we introduced GRAND, a novel approach for reconstructing graphs from their neighborhood data given by the common neighbors matrix. This can have significant implications in real-world scenarios, such as revealing hidden connections in social networks or communication systems. More precisely, the topological angle of attacks allows to infer information about the existence or non-existence of edges in the target graph based on the number of common neighbors of these vertices and on their degrees. In particular, we have identified properties that can be leveraged to infer this information about the graph's structure. The spectral angle uses eigendecomposition by incorporating information from the topological attack, iteratively reconstructing the graph through its eigenvalues and minimizing the distance to the partially recovered graph. We have shown, through multiple experiments on various datasets, that our approach is, for most of the cases, able to reconstruct graphs up to co-squareness, even for an uninformed adversary.

We have also defined co-squareness, a new notion of equivalence between graphs that is specific to the content of their common neighbor matrices, a property not being shared by most isomorphic graphs. Moreover, we have introduced a novel reconstruction metric, which focuses on the values of the square matrix, since in the absence of prior knowledge, graphs that are co-square to the original one are as good as one can hope.

Given the vast diversity of graphs, it is very challenging to generalize any attack to all types of graphs. Therefore, one of the future research avenues will be to extend the attack to directed graphs, bipartite graphs and other types of graphs. Another direction will be to study the NP-hardness of the problem by reducing it to a known NP-hard problem.

Acknowledgments

This work is supported by the DEEL Project CRDPJ 537462-18 funded by the National Science and Engineering Research Council of Canada (NSERC) and the Consortium for Research and Innovation in Aerospace in Québec (CRIAQ), together with its industrial partners Thales Canada inc, Bell Textron Canada Limited, CAE inc and Bombardier inc[2]. Sébastien Gambs is also supported by the Canada Research Chair program and a Discovery Grant from NSERC while Josée Desharnais, Marc-Olivier Killijian and Nadia Tawbi are supported by Discovery Grants from NSERC.

[2]https://deel.quebec

References

[1] Sofiane Azogagh, Zelma Aubin Birba, Sébastien Gambs, and Marc-Olivier Killijian. 2024. Crypto'Graph: Leveraging Privacy-Preserving Distributed Link Prediction for Robust Graph Learning.

[2] Yandong Bai, Pedro P Cortés, Reza Naserasr, and Daniel A Quiroz. 2024. Characterizing and recognizing exact-distance squares of graphs. *Discrete Mathematics* 347, 8 (2024), 113493.

[3] V. Chvétal. 1980. Recognizing Intersection Patterns. In *Combinatorics 79 Part I*, M. Deza and I.G. Rosenberg (Eds.). Annals of Discrete Mathematics, Vol. 8. Elsevier, 249–251. doi:10.1016/S0167-5060(08)70883-5

[4] Brian Cloteaux. 2016. Fast Sequential Creation of Random Realizations of Degree Sequences. *Internet Mathematics* 12, 3 (2016), 205–219.

[5] Nur Nasuha Daud, Siti Hafizah Ab Hamid, Muntadher Saadoon, Firdaus Sahran, and Nor Badrul Anuar. 2020. Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications* 166 (2020), 102716.

[6] Didem Demirag, Mina Namazi, Erman Ayday, and Jeremy Clark. 2023. Privacy-Preserving Link Prediction. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, Joaquin Garcia-Alfaro, Guillermo Navarro-Arribas, and Nicola Dragoni (Eds.). Springer International Publishing, Cham, 35–50.

[7] Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* 1, 3 (1936), 211–218.

[8] Dóra Erdös, Rainer Gemulla, and Evimaria Terzi. 2012. Reconstructing Graphs from Neighborhood Data. In *2012 IEEE 12th International Conference on Data Mining*. 231–240. doi:10.1109/ICDM.2012.154

[9] Mohammad Al Hasan and Mohammed J Zaki. 2011. A survey of link prediction in social networks. *Social network data analytics* (2011), 243–275.

[10] Yizhang He, Kai Wang, Wenjie Zhang, Xuemin Lin, and Ying Zhang. 2024. Common Neighborhood Estimation over Bipartite Graphs under Local Differential Privacy. *Proc. ACM Manag. Data* 2, 6, Article 228 (Dec. 2024), 26 pages. doi:10.1145/3698803

[11] Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

[12] Zoltán Király. 2012. Recognizing graphic degree sequences and generating all realizations. *Eötvös Loránd University, Tech. Rep. Egres TR-2011-11* (2012).

[13] Yaxin Li, Wei Jin, Han Xu, and Jiliang Tang. 2020. Deeprobust: A pytorch library for adversarial attacks and defenses. *arXiv preprint arXiv:2005.06149* (2020).

[14] Duncan McAndrew. 2021. The structural analysis of criminal networks. *The Social Psychology of Crime* (2021).

[15] Rajeev Motwani and Madhu Sudan. 1994. Computing roots of graphs is hard. *Discrete Applied Mathematics* 54, 1 (1994), 81–88.

[16] Rajeev Motwani and Madhu Sudan. 1994. Computing roots of graphs is hard. *Discrete Applied Mathematics* 54, 1 (1994), 81–88. doi:10.1016/0166-218X(94)00023-9

[17] Giulia Muzio, Leslie O'Bray, and Karsten Borgwardt. 2021. Biological network analysis with deep learning. *Briefings in bioinformatics* 22, 2 (2021), 1515–1530.

[18] Georgios A Pavlopoulos, Maria Secrier, Charalampos N Moschopoulos, Theodoros G Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G Bagos. 2011. Using graph theory to analyze biological networks. *BioData mining* 4 (2011), 1–27.

[19] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. https://networkrepository.com

[20] Nitai B. Silva, Ing-Ren Tsang, George D. C. Cavalcanti, and Ing-Jyh Tsang. 2010. A graph-based friend recommendation system using Genetic Algorithm. In *IEEE Congress on Evolutionary Computation*. 1–7.

[21] Edwin R. van Dam and Willem H. Haemers. 2003. Which graphs are determined by their spectrum? *Linear Algebra Appl.* 373 (2003), 241–272. doi:10.1016/S0024-3795(03)00483-X Combinatorial Matrix Theory Conference (Pohang, 2002).

[22] Piet Van Mieghem. 2023. *Graph spectra for complex networks*. Cambridge university press.

[23] Piet Van Mieghem and Ivan Jokić. 2024. Co-eigenvector graphs. *Linear Algebra Appl.* 689 (2024), 34–59. doi:10.1016/j.laa.2024.02.008

[24] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P.N. Puttaswamy, and Ben Y. Zhao. 2009. User Interactions in Social Networks and Their Implications. In *Proceedings of the 4th ACM European Conference on Computer Systems* (Nuremberg, Germany) *(EuroSys '09)*. Association for Computing Machinery, New York, NY, USA, 205–218. doi:10.1145/1519065.1519089

[25] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. 2019. Adversarial Examples for Graph Data: Deep Insights into Attack and Defense. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4816–4823. doi:10.24963/ijcai.2019/669

[26] Xiaojun Xu, Hanzhang Wang, Alok Lal, Carl A. Gunter, and Bo Li. 2023. EDoG: Adversarial Edge Detection For Graph Neural Networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. 291–305. doi:10.1109/SaTML54575.2023.00027

[27] Quan Yuan, Zhikun Zhang, Linkang Du, Min Chen, Peng Cheng, and Mingyang Sun. 2023. PrivGraph: differentially private graph data publication by exploiting community information. In *Proceedings of the 32nd USENIX Conference on Security Symposium* (Anaheim, CA, USA) *(SEC '23)*. USENIX Association, USA, Article 182, 18 pages.

[28] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial Attacks on Neural Networks for Graph Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 2847–2856. doi:10.1145/3219819.3220078

A Experimental results

A.1 Errors made by an uninformed attacker

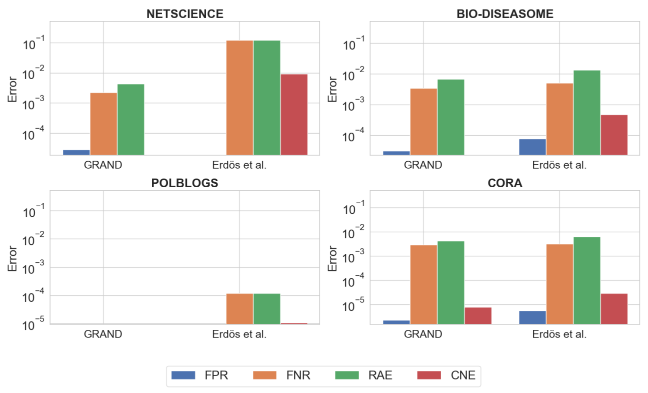


Figure 13: False Positives Rate, False Negatives Rate, Relative Absolute Error and Common Neighbors Error with ρ=0.

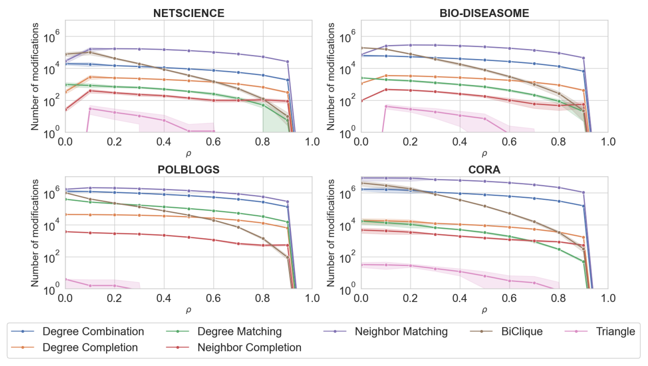


Figure 14: Number of modifications made by the topological attacks.

In figure 13, in addition to the RAE and CNE, we present the False Positives Rate (FPR) and the False Negatives Rate (FNR), where positives are edges and negatives are non edges.

A.2 Comparative importance of topological attacks

Figure 14 showcases the fact that since adjacency matrices are very sparse, attacks that only find 1s (e.g. TriangleAttack, NeighborCompletionAttack) tend to make less modifications.

Algorithm 3 Identifies absent edges in G based on the degrees of vertices in G and $G^\star$

```
Require: Partial reconstruction $G^\star$, common neighbors matrix $G^2$
function DegreeMatchingAttack($G^\star, G^2$)
    for $u \in \mathcal{V}$ do
        if $G^2(u,u) = |\Gamma^\star(u)|$ then
                                ▷ The neighborhood of $u$ is complete
            for $v \in \mathcal{V} \backslash \Gamma^\star(u)$ do
                $G^\star(u,v) \leftarrow 0$            ▷ Update $G^\star$
            end for
        end if
    end for
    return $G^\star$
end function
```

Algorithm 4 Identifies triangles in G.

```
Require: Reconstructed graph $G^\star$, common neighbors matrix $G^2$
function TriangleAttack($G^\star, G^2$)
    $edges \leftarrow \{(u,v), G^\star(u,v) = 1\}$        ▷ Extract edges
    for all $(u,v) \in edges$ do
        $S_u = \{w : G^2(u,w) > 0$ and $u \neq w\}$
        $S_v = \{w : G^2(v,w) > 0$ and $v \neq w\}$
              ▷ Vertices that share common neighbors with $u$ and $v$
        $S = S_u \cap S_v$
        if $|S| = G^2(u,v)$ then
            for all $w \in S$ do
                $G^\star(u,w) \leftarrow 1$
                $G^\star(v,w) \leftarrow 1$
            end for
        end if
    end for
    return $G^\star$
end function
```

Algorithm 5 Identifies present edges by comparing the common neighbors of pairs of vertices on G and $G^\star$.

```
Require: Reconstructed graph $G^\star$, common neighbors matrix $G^2$
function NeighborCompletionAttack($G^\star, G^2$)
    for $(u,v) \in \mathcal{V}^2$ do
        $U \leftarrow \{w \in \mathcal{V}, G^\star(u,w) =?\}$
        $V \leftarrow \{v \in \mathcal{V}, G^\star(v,w) =?\}$
        if $G^2(u,v) = |\Gamma^\star(u)| + |U|$ then
            for $w \in U$ do
                $G^\star(u,w) \leftarrow 1$
                $G^\star(v,w) \leftarrow 1$
            end for
        end if
        if $G^2(u,v) = |\Gamma^\star(v)| + |V|$ then
            for $w \in V$ do
                $G^\star(u,w) \leftarrow 1$
                $G^\star(v,w) \leftarrow 1$
            end for
        end if
    end for
    return $G^\star$
end function
```

Algorithm 6 Identifies bi-cliques and their complements G.

```
Require: Reconstructed graph $G^\star$, common neighbors matrix $G^2$
function BicliqueAttack($G^\star, G^2$ )
    $U \leftarrow \{u, G^2(u,u) = |\Gamma^\star(u)|\}$ ▷ Nodes with satisfied degree
    for all $u \in U$ do
        for all $v \in \mathcal{V}$ do
            if $G^2(u,v) = G^2(u,u)$ then          ▷ Bi-clique found
                for all $w \in \Gamma^\star(u)$ do
                    $G^\star(v,w) \leftarrow 1$
                end for
            else
                $m_G = G^2(v,v) - |\Gamma^\star(v)|$
                $m_{G^2} = G^2(u,v) - |\Gamma^\star(u) \cap \Gamma^*(v)|$
                if $m_G = m_{G^2}$ then
                    for all $w \in \mathcal{V} \backslash \Gamma^\star(u)$ do
                        $G^\star(v,w) \leftarrow 0$
                    end for
                end if
            end if
        end for
    end for
    return $G^\star$
end function
```

Algorithm 7 ToplogicalAttack

```
Require: Lists of edges and non-edges $\mathcal{E}_1, \mathcal{E}_0$, common neighbors
    matrix $G^2$
function TopologicalAttack($\mathcal{E}_1, \mathcal{E}_0, G^2$)
    $G^\star \leftarrow$ Init($\mathcal{E}_1, \mathcal{E}_0$)          ▷ Copy 1s and 0s into $G^\star$
    while $G^\star$ has been updated do
        $G^\star \leftarrow$ DegreeCombinationAttack($G^\star, G^2$)
        $G^\star \leftarrow$ DegreeMatchingAttack($G^\star, G^2$)
        $G^\star \leftarrow$ NeighborMatchingAttack($G^\star, G^2$)
        $G^\star \leftarrow$ DegreeCompletionAttack($G^\star, G^2$)
        $G^\star \leftarrow$ NeighborCompletionAttack($G^\star, G^2$)
        $G^\star \leftarrow$ TriangleAttack($G^\star, G^2$)
        $G^\star \leftarrow$ BicliqueAttack($G^\star, G^2$)
    end while
    return $G^\star$
end function
```

Algorithm 8 Identifies existing edges by comparing the neighborhoods of vertices on G and $G^\star$

```
Require: Reconstructed graph $G^\star$, common neighbors matrix $G^2$
function DegreeCompletionAttack($G^\star, G^2$)
    for $u \in \mathcal{V}$ do
        $V \leftarrow \{v \in \mathcal{V}, G^\star(u,v) =?\}$
        if $G^2(u,u) = |\Gamma^\star(u)| + |V|$ then
            for $v \in V$ do
                $G^\star(u,v) \leftarrow 1$                ▷ Update $G^\star$
            end for
        end if
    end for
    return $G^\star$
end function
```

Algorithm 9 SpectralAttack

```
Require: Partial reconstructed graph $G^\star$, SVD of $G^2$ ($U, \Lambda, V$ such that $G^2 = U\Lambda V$), $\alpha, \beta$ weighting factors.
1: function SpectralAttack($G^\star, U, \Lambda, V, \alpha, \beta$)
2:     $M_0 \leftarrow \mathbf{0}^{n\times n}$
3:     $\mathcal{E}^\star \leftarrow \{(i,j) \mid G^\star(i,j) \in \{0,1\}\}$    ▷ Extract edges and non-edges from $G^\star$
4:     for $i = 1$ to $|\mathcal{V}|$ do
5:         $M_i^+ \leftarrow M_{i-1} + U(:,i) \cdot \sqrt{\Lambda(i,i)} \cdot V(:,i)^T$
6:         $M_i^- \leftarrow M_{i-1} - U(:,i) \cdot \sqrt{\Lambda(i,i)} \cdot V(:,i)^T$
7:         $d^+ \leftarrow \alpha \left\|M_i^+ - BM_i^+\right\|_F + \beta \left\|M_i^+|_{\mathcal{E}^\star} - G^\star|_{\mathcal{E}^\star}\right\|_F$
8:         $d^- \leftarrow \alpha \left\|M_i^- - BM_i^-\right\|_F + \beta \left\|M_i^-|_{\mathcal{E}^\star} - G^\star|_{\mathcal{E}^\star}\right\|_F$
9:         if $d^+ < d^-$ then
10:            $M_i \leftarrow M_i^+$
11:        else
12:            $M_i \leftarrow M_i^-$
13:        end if
14:    end for
15:    return $BM$
16: end function
```

Algorithm 10 TargetedErrorForgetting

```
Require: Graph reconstructed by spectral attack $G_s^\star$, Graph reconstructed by topological attacks $G_t^\star$, common neighbors matrix $G^2$
1: function TargetedErrorForgetting($G_s^\star, G_t^\star, G^2$)
           ▷ Common neighbors matrix on the reconstructed graph
2:     $G_s^{\star 2} \leftarrow G_s^\star \cdot G_s^\star$
3:     for $(u,v) \in \mathcal{V}^2$ do
                                        ▷ Incorrect common neighbors
4:         if $G^2(u,v) \neq G_s^{\star 2}(u,v)$ then
5:             for $w \in \mathcal{V}$ do
6:                 $G_s^\star(u,w) \leftarrow G_t^\star(u,w)$
7:                 $G_s^\star(v,w) \leftarrow G_t^\star(v,w)$
8:             end for
9:         end if
10:    end for
11:    return $G_s^\star$
12: end function
```

B Algorithms

Algorithm 1 Determining existence of edges of G from the degrees of vertices.

```
Require: Partial reconstruction $G^\star$, common neighbors matrix $G^2$
1: function DegreeCombinationAttack($G^\star, G^2$)
2:     for $u \in \mathcal{V}$ do
3:         $s \leftarrow \Sigma_{v\in\mathcal{V}} G^2(u,v)$          ▷ Sum of u-th row
4:         $d \leftarrow G^2(u,u)$                          ▷ Degree of vertex u
5:         $S = \{\}$                                       ▷ Candidate set
6:         $candidate = \{v \mid G^2(v,v) < s - d\}$
7:         $non\_candidate = \mathcal{V} \setminus candidate$
8:         for all $C \subseteq candidate$ of size $d$ do        ▷ Search set
9:             if $S = \{\}$ then
10:                if $\Sigma_{w\in C} G^2(w,w) = s$ then
11:                    $S \leftarrow C$                      ▷ Candidate set found
12:                end if
13:            else                                          ▷ Multiple sets found
14:                $S \leftarrow \{\}$
15:                break
16:            end if
17:        end for
18:        for all $v \in S$ do                              ▷ Update $G^\star$
19:            $G^\star(u,v) \leftarrow 1$
20:        end for
21:        for all $v \in non\_candidate$ do
22:            $G^\star(u,v) \leftarrow 0$
23:        end for
24:    end for
25:    return $G^\star$
26: end function
```

Algorithm 2 Identifies absent edges in $G^\star$ based on the numbers of common neighbors in G.

```
Require: Reconstructed graph $G^\star$, common neighbors matrix $G^2$
1: function NeighborMatchingAttack($G^\star, G^2$)
2:     for $(u,v) \in \mathcal{V}^2$ do
3:         if $G^2(u,v) = |\Gamma^\star(u) \cap \Gamma^\star(v)|$ then
4:             ▷ The common neighborhood of $u$ and $v$ is complete
5:             for $w \in \Gamma^\star(u) \setminus (\Gamma^\star(u) \cap \Gamma^\star(v))$ do
6:                 $G^*(w,v) \leftarrow 0$                   ▷ Update $G^\star$
7:             end for
8:             for $w \in \Gamma^\star(v) \setminus (\Gamma^\star(u) \cap \Gamma^\star(v))$ do
9:                 $G^\star(w,u) \leftarrow 0$               ▷ Update $G^\star$
10:            end for
11:        end if
12:    end for
13:    return $G^\star$
14: end function
```