Exploiting Graph Poisoning and Unlearning to Enhance Link Inference Attacks

Technical Appendix

Anonymous submission

1 Proof of Theorems

Proof of Theorem 1

For simplicity, we use K_{com} and K_{all} to denote $|\mathcal{N}_k^{G_\mathcal{A}}(u) \cap \mathcal{N}_k^{G_\mathcal{A}}(v)|$ and $|\mathcal{N}_k^{G_\mathcal{A}}(u) \cup \mathcal{N}_k^{G_\mathcal{A}}(v)|$, respectively, for any two nodes u, v. Consider E_P defined by Theorem 1. Following its definition, it must be true that

$$\mathsf{SS}(u, v, G_\mathcal{A} \cup E_P) = \frac{K_{com}}{K_{all} + |E_P|}.$$

Similarly, for any set of poisoning edges E'_P that are defined in Theorem 1, it must be true that

$$\mathsf{SS}(u, v, G_\mathcal{A} \cup E'_P) = \frac{K_{com} + n_1}{K_{all} + n_2},$$

where n_1 and n_2 are the number of new common neighbors of u and v and the number of neighbors of either u or v (but not both) added by insertion of E'_P, respectively. As Theorem 1 assumes $|E'_P| \leq |E_P|$, it must be true that

$$n_2 \leq |E'_P| \leq |E_P|.$$

Following the above definitions, the difference between the SS values by inserting E_P and E'_P is calculated as:

$$\begin{aligned} &\mathsf{SS}(u, v, G_\mathcal{A} \cup E_P) - \mathsf{SS}(u, v, G_\mathcal{A} \cup E'_P) \\ =&\frac{K_{com}}{K_{all} + |E_P|} - \frac{K_{com} + n_1}{K_{all} + n_2} \\ <&\frac{K_{com}}{K_{all} + |E_P|} - \frac{K_{com} + n_1}{K_{all} + |E_P|} \\ =&-\frac{n_1}{K_{all} + |E_P|} \leq 0 \end{aligned} \tag{1}$$

This completes the proof.

Proof of Theorem 2

Given a node pair (u, v), we consider their label distribution vectors:

$$\begin{aligned} \mathcal{L}_u^{G_\mathcal{A}} &= [p_1, p_2, \ldots, p_n], \\ \mathcal{L}_v^{G_\mathcal{A}} &= [q_1, q_2, \ldots, q_n], \end{aligned}$$

where p_i, q_i follow Equation (3) in the paper. The relationship between each p_i and q_i belongs to one of the two following cases:

$$|p_i - q_i| = \begin{cases} p_i - q_i, & p_i \geq q_i; \\ q_i - p_i & p_i < q_i. \end{cases} \tag{2}$$

Let N_u denote the size of u'k-hop neighborhood, and E_P be the poisoning edges defined in Theorem 2.After the injection of E_P, each element in $\mathcal{L}_u^{G_\mathcal{A} \cup E_P}$, denoted as p'_i, is updated as follows:

$$p'_i = \begin{cases} \frac{p_i N_u}{N_u + |E_P|} & y_i \neq y^*; \\ \frac{p_i N_u + K}{N_u + |E_P|} & y_i = y^*. \end{cases} \tag{3}$$

We have following six cases in terms of the relationship between p_i, p'_i, and q_i:

- **Case 1**: $y_i \neq y^*$, $p_i < q_i$ and $p'_i < q_i$
- **Case 2**: $y_i \neq y^*$, $p_i \geq q_i$ and $p'_i \leq q_i$
- **Case 3**: $y_i = y^*$, $p_i \geq q_i$ and $p'_i > q_i$
- **Case 4**: $y_i \neq y^*$, $p_i \geq q_i$ and $p'_i \geq q_i$
- **Case 5**: $y_i \neq y^*$, $p_i < q_i$ and $p'_i \geq q_i$
- **Case 6**: $y_i = y^*$, $p_i \geq q_i$ and $p'_i < q_i$

However, Case 5 should be eliminated because connecting with nodes associated with $y_i \neq y^*$ only decreases p_i (i.e., $p_i > p'_i$). Similarly, Case 6 should be eliminated as connecting with nodes associated with $y_i = y^*$ will increase p_i (i.e., $p_i < p'_i$).

By considering Cases 1 - 4, After edge insertion, the LS value is updated as:

$$\begin{aligned} \mathsf{LS}(u, v, G_\mathcal{A} \cup E_P) = 1 - &\frac{1}{2} \sum_{\substack{i=1 \\ (p'_i,\, q_i) \in \text{Case 1,2}}}^{n} (q_i - p'_i) - \\ &\frac{1}{2} \sum_{\substack{i=1 \\ (p'_i,\, q_i) \in \text{Case 3,4}}}^{n} (p'_i - q_i) \end{aligned} \tag{4}$$

Then for any set of edge E'_P that is defined in Theorem 2, Let $\mathcal{L}_u^{G_\mathcal{A} \cup E_P} = [..., p_i^{E_P}, ...]$ and $\mathcal{L}_u^{G_\mathcal{A} \cup E'_P} = [..., p_i^{E'_P}, ...]$ be the vectors of label distributions for u's neighborhood after injecting E_P and E'_P respectively. Note that $|E_P| \geq |E'_P|$, thus there might exist $i = 1$ that $p_i^{E_P} < q_i$ and $p_i^{E'_P} \geq q_i$. We name such case as **Case 7**. Then according to the first row in Eqn. (3), $\sum_{\text{Case 1,2}} p_i^* < \sum_{\text{Case 1,2}} p_i^{E'_P}$ when $y \neq y^*$. We have the following derivation:

$$
\begin{aligned}
&\mathsf{LS}_k(u,v,G_{\mathcal{A}}\cup E_P)-\mathsf{LS}(u,v,G_{\mathcal{A}}\cup E'_P)\\
=&\frac{1}{2}\sum_{\substack{i=1\\(p_i^{E_P},q_i)\in\text{Case 1,2}}}^{n}((p_i^{E_P}-q_i)-(p_i^{E'_P}-q_i))+\\
&\frac{1}{2}\sum_{\substack{i=1\\(p_i^{E_P},p_i^{E'_P},q_i)\in\text{Case 7}}}^{n}((p_i^{E_P}-q_i)-(q_i-p_i^{E'_P}))+\\
&\frac{1}{2}\sum_{\substack{i=1\\(p_i^{E'_P},q_i)\in\text{Case 3,4}}}^{n}((p_i^{E'_P}-q_i)-(p_i^{E_P}-q_i))\\
\le&\frac{1}{2}\sum_{\substack{i=1\\(p_i^{E_P},p_i^{E'_P},q_i)\in\text{Case 1,2,7}}}^{n}(p_i^{E_P}-p_i^{E'_P})+\\
&\frac{1}{2}\sum_{\substack{i=1\\(p_i^{E_P},p_i^{E'_P},q_i)\in\text{Case 3,4}}}^{n}(p_i^{E'_P}-p_i^{E_P})\qquad(5)\\
=&\frac{1}{2}(\sum_{\substack{i=1\\(p_i^{E_P},p_i^{E'_P},q_i)\in\text{Case 1,2,7}}}^{n}(p_i^{E_P}-p_i^{E'_P})+\\
&(1-\sum_{\substack{i=1\\(p_i^{E_P},p_i^{E'_P},q_i)\in\text{Case 1,2,7}}}^{n}p_i^{E'_P})-\\
&(1-\sum_{\substack{i=1\\(p_i^{E_P},p_i^{E'_P},q_i)\in\text{Case 1,2,7}}}^{n}(p_i^{E_P})))\\
=&\sum_{\substack{i=1\\(p_i^{E_P},p_i^{E'_P},q_i)\in\text{Case 1,2,7}}}^{n}(p_i^{E_P}-p_i^{E'_P})\le 0
\end{aligned}
$$

This finishes the proof.

Proof of Theorem 3

For simplicity, we use N_u and N_v to denote the size of u's neighborhood and v's neighborhood in $G_{\mathcal{A}}$, i.e., $N_u=|\mathcal{N}_k^{G_{\mathcal{A}}}(u)|, N_v=|\mathcal{N}_k^{G_{\mathcal{A}}}(v)|$. And we define $S_{E_P}=\sum_{(u,u^*)\in E_P} fs(u^*,v,G_{\mathcal{A}})$. Following the definition of V^* in Theorem 3, it must be true that

$$AFS(u,v,G_{\mathcal{A}})\ge\frac{S_{E_P}}{|E_P|}\qquad(6)$$

Next, we define $S_G=AFS(u,v,G_{\mathcal{A}})\times N_u$, and derive the following for any E'_P defined by Theorem 3:

$$
\begin{aligned}
&\frac{S_G+S_{E_P}}{(N_u+|E_P|)}-\frac{S_G+\frac{|E'_P|}{|E_P|}S_{E_P}}{(N_u+|E'_P|)}\\
=&\frac{S_G|E'_P|+S_{E_P}N_u-S_G|E_P|-\frac{|E'_P|N_u}{|E_P|}S_{E_P}}{(N_u+|E_P|)(N_u+|E'_P|)}\\
=&\frac{S_G(|E'_P|-|E_P|)+S_{E_P}N_u(1-\frac{|E'_P|}{|E_P|})}{(N_u+|E_P|)(N_u+|E'_P|)}\qquad(7)\\
=&\frac{(|E'_P|-|E_P|)(|E_P|S_G-S_{E_P}N_u)}{|E_P|(N_u+|E_P|)(N_u+|E'_P|)}\\
=&\frac{(|E'_P|-|E_P|)(AFS(u,v,G_{\mathcal{A}})-\frac{S_{E_P}}{|E_P|})}{N_u(N_u+|E_P|)(N_u+|E'_P|)}
\end{aligned}
$$

Since $|E'_P|\le|E_P|$ (assumption of Theorem 3) and $AFS(u,v,G_{\mathcal{A}})\ge\frac{S_{E_P}}{|E_P|}$) (Eqn. 6), we have

$$\frac{S_G+S_{E_P}}{(N_u+|E_P|)}\ge\frac{S_G+\frac{|E'_P|}{|E_P|}S_{E_P}}{(N_u+|E'_P|)}\qquad(8)$$

Next, we will prove $\mathsf{FS}(u,v,G_{\mathcal{A}}\cup E_P)\le\mathsf{FS}(u,v,G_{\mathcal{A}}\cup E'_P)$, for any E'_P defined by Theorem 3. Following the definition of FS, the difference between $\mathsf{FS}(u,v,G_{\mathcal{A}}\cup E_P)$ and $\mathsf{FS}(u,v,G_{\mathcal{A}}\cup E'_P)$ is:

$$
\begin{aligned}
&\mathsf{FS}(u,v,G_{\mathcal{A}}\cup E_P)-\mathsf{FS}(u,v,G_{\mathcal{A}}\cup E'_P)\\
=&\frac{S_GN_v+S_{E_P}N_v}{(N_u+|E_P|)N_v}-\frac{S_GN_v+S_{E'_P}N_v}{(N_u+|E'_P|)N_v}\\
=&\frac{S_G+S_{E_P}}{(N_u+|E_P|)}-\frac{S_G+S_{E'_P}}{(N_u+|E'_P|)}\\
\le&\frac{S_G+\frac{|E'_P|}{|E_P|}S_{E_P}}{(N_u+|E_P|)}-\frac{S_G+\frac{|E'_P|}{|E_P|}S_{E_P}}{(N_u+|E'_P|)}\qquad(9)\\
\le&\frac{S_G+\frac{|E'_P|}{|E_P|}S_{E_P}}{(N_u+|E'_P|)}-\frac{S_G+\frac{|E'_P|}{|E_P|}S_{E_P}}{(N_u+|E'_P|)}\\
=&\,0
\end{aligned}
$$

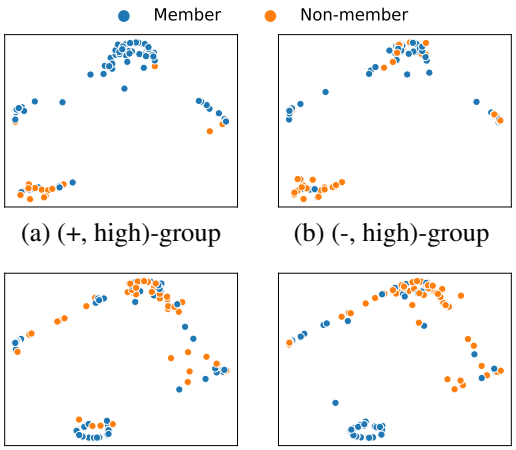
This finishes the proof.

2 Illustration of the Relationship between Change of Node Similarity and Membership Exposure

Figure 1 illustrates the relationship between Δ_{HS} and membership distinguishability. For presentation purposes, we randomly choose 150 member node pairs and 150 non-member node pairs from each subgroup. The features of these data samples are constructed from the similarity between the nodes' posterior probabilities output by both M_P and M_U; they are used by LMIA for membership inference (More details of LMIA features can be found in Section 5 of the main paper).

3 Illustration of the Attack

Figure 2 illustrates the three phases of our attack.



(a) (+, high)-group (b) (-, high)-group

(c) (+, low)-group (d) (-, low)-group

Figure 1: The impact of change in node similarity on privacy vulnerability to membership inference

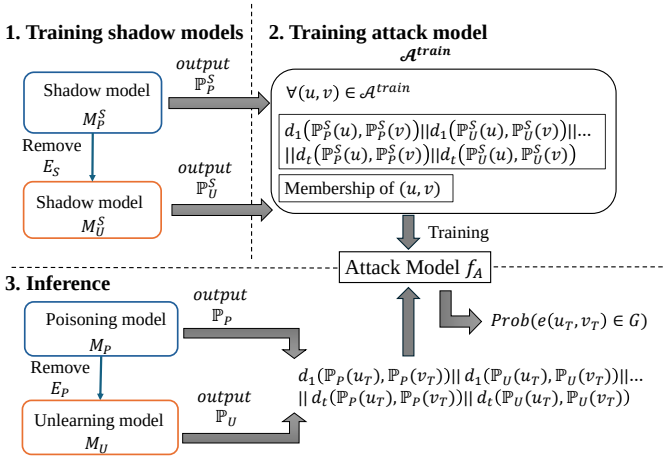


Figure 2: Illustration of the three phases of our attack

4 Pseudo Code of Our Algorithm

The pseudo code can be found in Algorithms 1 and 2. The main function of Algorithm 1 considers two alternative options to construct the poisoning edges: they are constructed as the edges connecting with either u_T or v_T that minimize SS, LS, and FS, respectively. The option that results in higher HS similarity is returned as the poisoning edges E_P.

Algorithm 1 calls the $ConstructEdges()$ function to construct the poisoning edges. As the poisoned edges affect the closer node pairs more than the faraway ones, $ConstructEdges()$ only constructs edges that are directly connected to the targets. It first identifies a set of nodes V_P whose connection to u_T can minimize $\mathsf{SS}(u, v, G_{\mathcal{A}})$. Next, it constructs a set of edges whose connection to u_T minimizes $\mathsf{LS}(u, v, G_{\mathcal{A}})$ (Lines 3 - 4). During this step, it prunes the nodes in V_P to ensure the nodes in the set are associated with the label of the largest disparate distribution score (Line 3). After the LS-driven search, it constructs a set of edges whose injection can minimize $\mathsf{FS}(u, v, G_{\mathcal{A}})$ (Lines 5-9) by further pruning the node set V_P and keeping only nodes whose features have the smallest feature score (Line 7). Finally, the edges between u_T and the picked nodes are constructed as the poisoned edges and returned by the algorithm.

Algorithm 1: Constructing poisoning edges

Input: Target (u_T, v_T), adversary graph $G_{\mathcal{A}}$, budget b
Output: A set of edges $E_P \subseteq G^{\mathcal{A}}$ for adversarial unlearning

1 $E_{u_T} \leftarrow \text{ConstructEdges}(u_T, v_T, G^{\mathcal{A}}, b)$
// Construct poisoning edges associated with u_T by Alg 2
2 $E_{v_T} \leftarrow \text{ConstructEdges}(v_T, u_T, G^{\mathcal{A}}, b)$
// Construct poisoning edges associated with v_T.
3 **if** $\mathsf{HS}(u_T, v_T, G^{\mathcal{A}} \cup E_{u_T}) < \mathsf{HS}(u_T, v_T, G^{\mathcal{A}} \cup E_{v_T})$ **then**
4 | **return** E_{u_T}
5 **else**
6 | **return** E_{v_T}
7 **end**

It is possible that $ConstructEdges()$ function cannot identify b edges (b: poisoning budget) that minimize SS, LS, and FS simultaneously. For this case, we identify the edges that minimize two of the three metrics (e.g., SS and LS), and return the one that returns the minimal value on the remaining metric (e.g., FS). However, in our experiments, we never met the case that fails to identify b poisoning edges that minimize SS, LS, and FS simultaneously.

5 Data Description and Link to Code & Datasets

Table 1 reports the statistical information of the three datasets used in the experiments.

Our code and datasets are available at: https://anonymous.4open.science/r/MEDUSA-4E51

6 Additional Experimental Results

Impact of number of attack targets on attack performance. To assess whether the proposed attack can be scaled to target a large number of nodes, we vary the number of attack targets as 0.1%, 0.5%, 1%, 5%, and 10% of the original graph. Given that adding poisoning edges for one target might affect other targets, we analyze the impact of varying numbers of attack targets in two scenarios: 1) the targets are dependent on each other, meaning there is at least one other target within the 2-hop neighborhood of each target; 2) the targets are independent, meaning each target is excluded from the 2-hop neighborhood of any other target.

Table 2 presents the attack performance of MEDUSA across different numbers of attack targets in both settings. Overall, MEDUSA demonstrates strong scalability in targeting multiple nodes simultaneously, maintaining an attack accuracy above 0.814 even when 10% of the edges are targeted. However, when the targets are interdependent, a slight

Algorithm 2: *ConstructEdges*()

Input: Target (u_T, v_T), adversary graph $G_{\mathcal{A}}$, poisoning budget b.
Output: A set of poisoning edges $E_P \notin G_{\mathcal{A}}$

Function `ConstructEdges` $(u_T, v_T, G_{\mathcal{A}}, b)$ **:**
  /* Decrease $\mathsf{SS}(u_T, v_T, G_{\mathcal{A}})$ */
  $V_P \leftarrow DN(u_T, v_T, G_{\mathcal{A}}) \cap \mathcal{N}_k^{G_{\mathcal{A}}}(u_T)$;
  /* $DN()$: Eqn. (11) in the paper */
  /* Decrease $\mathsf{LS}(u_T, v_T, G_{\mathcal{A}})$ */
  $y_i^* \leftarrow \underset{y_i, dis(u_T, v_T, y_i, G_{\mathcal{A}})>0}{\text{argmax}} dis(u_T, v_T, y_i, G_{\mathcal{A}})$ // $dis()$: Eqn. (13) in the paper.
  $V_P \leftarrow$ Nodes in V_P labeled with y_i^*;
  /* Alter $\mathsf{FS}(u_T, v_T, G_{\mathcal{A}})$ */
  $V_P^* \leftarrow \emptyset$;
  while $|V_P^*| < |V_P|$ **do**
    $u^* \leftarrow \underset{\substack{u' \in V_P \setminus V_P^*, \\ fs(u', v_T, G_{\mathcal{A}}) < AFS(u_T, v_T, G_{\mathcal{A}})}}{\text{argmin}} fs(u')$;
    /* $AFS()$: Eqn. (15)), $fs()$: Eqn. (16) in the paper */
    $V_P^* \leftarrow V_P^* \cup u^*$
  end
  $E_P \leftarrow \{e(u, u') | u' \in V_P^*\}$;
  return E_P;

Table 1: Description of datasets.

Dataset	# of nodes	# of edges	# of labels
Cora	2,708	5,278	7
Citeseer	3,327	4,452	6
Pubmed	19,717	44,325	3

Table 2: Attack accuracy (ACC) of MEDUSA when varying the number of attack targets ($|G_{\mathcal{A}}| = 15\%|G|$) Dependent targets: the targets are within 2-hop neighborhood of other targets. Independent targets: the targets are excluded in the 2-hop neighborhood of any other targets.

Dataset	Number of attack targets (% of $G_{\mathcal{A}}$)				
	0.1%	0.5%	1%	5%	10%
Citeseer	0.869	0.861	0.848	0.839	0.814
Cora	0.886	0.872	0.861	0.851	0.836
Pubmed	0.874	0.869	0.861	0.852	0.838

(a) Dependent targets

Dataset	Number of attack targets (% of $G_{\mathcal{A}}$)				
	0.1%	0.5%	1%	2%	3%
Citeseer	0.877	0.871	0.874	0.869	0.874
Cora	0.891	0.893	0.888	0.889	0.892
Pubmed	0.876	0.880	0.880	0.882	0.879

(b) Independent targets

decrease in attack accuracy is observed as the number of targets increases, ranging between 4% and 6%. This decline is likely due to the interaction between poisoning edges associated with different targets, particularly when these targets are closely located within the graph's neighborhood. Conversely, when the targets are independent of each other, the reduction in attack accuracy is minimal, with changes not exceeding 1% across all settings.