Appendix of
Deep Efficient Private Neighbor Generation for Subgraph Federated Learning

Anonymous

I. Proof for Embedding-fused Graph Convolution

A. Proof for Statement 1

Statement 1 (Correctness of embedding-fused graph convolution)**.** *For a node v, at each layer of embedding-fused graph convolution, it aggregates nodes on the impaired ego-graph with the corresponding mended deep neighbor embeddings with separate learnable weights.*

Proof. At k-th layer of embedding-fused graph convolution, for every node u in v's one-hop ego-graph $G^1(v)$, we denote its mean averaged node representations as $\bar{x}_u^k \in \mathbb{R}^{1\times d_h}$ and embeddings as $\bar{z}_u \in \mathbb{R}^{1\times d_z}$.

According to our description in Section IV, we have

$$x_u^k = \sigma(W^{(k)} \times [\bar{x}_u^{k-1}||\bar{z}_u]^\top)^\top,$$

where $W^{(k)} \in \mathbb{R}^{d_h\times(d_h+d_z)}$ is the learnable matrix in the convolution.

As x_u^k can also be regarded as

$$\sigma(\begin{bmatrix} W_{1,1}^{x(k)} & ... & W_{1,d_h}^{x(k)} & W_{1,1}^{z(k)} & ... & W_{1,d_z}^{z(k)} \\ \vdots & ... & \vdots & \vdots & ... & \vdots \\ W_{d_h,1}^{x(k)} & ... & W_{d_h,d_h}^{x(k)} & W_{d_h,1}^{z(k)} & ... & W_{d_h,d_z}^{z(k)} \end{bmatrix} \times \begin{bmatrix} \bar{x}_{u,1}^{k-1} \\ \vdots \\ \bar{x}_{u,d_x}^{k-1} \\ \bar{z}_{u,1} \\ \vdots \\ \bar{z}_{u,d_z} \end{bmatrix})^\top,$$

which equals to $\sigma(W^{x(k)} \times \bar{x}_u^{k-1\top} + W^{z(k)} \times \bar{z}_u^\top)^\top$, where $W^{x(k)} \in \mathbb{R}^{d_h\times d_h}$ and $W^{z(k)} \in \mathbb{R}^{d_h\times d_z}$ are learnable weights in the convolution.

Therefore, we justify the correctness of embedding-fused graph convolution where the mended deep neighbors and the representations/features contribute to the convolution with respective learnable parameters, and conclude the proof. □

B. Proof for Statement 2

Lemma 1. *For a node v, we denote the prediction, computed by one layer of embedding-fused graph convolution on its 1-hop impaired ego-graph, where every node is mended with deep neighbors computed on the respective L-hop missing context, as $\tilde{y}'_v$, and the prediction, computed by (L+1) layers of graph convolution on its (L+1)-hop ego-graph, as $\tilde{y}_v$, where $L \in \mathbb{N}^*$. $\tilde{y}'_v$ and $\tilde{y}_v$ are the compound vectors for the same local context of v.*

Proof. For node v, we compute its prediction $\tilde{y}'_v$ as

$$\tilde{y}'_v = x_v^1 = \sigma(W^{(1)} \times [mean(\{x_u^0|u \in G^1(v)\})||\bar{z}_v]^\top,$$

where for every $u \in G^1(v)$,

$$x_u^0 = \sigma(W^{(0)} \times [x_u||\bar{z}_u])^\top = \sigma(W^{(0)} \times [x_u||mean(z_u)])^\top$$

Since x_u^0 contains $\{x_u, z_u\}$, and $\tilde{y}'_v$ is then computed based on $\{x_u, z_u|u \in G^1(v)\} \cup \{\bar{z}_v\}$. We only need to verify $\{x_u, z_u|u \in G^1(v)\} \cup \{\bar{z}_v\}$ containing the same information as the $\{x_u|u \in G^{L+1}(v)\}$.

First we have $\{\bar{z}_v\}$ computed from the L-hop neighbors of v, *i.e.*, $\{x_u|u \in G^L(v)\}$. Then we only need to consider whether the content of $\{x_u, z_u|u \in G^1(v)\}$ covers the $\{x_u|u \in G^{L+1}(v) \setminus G^L(v)\}$. Since every $z_u^p \in z_u$ is computed on the L-hop ego-graph of node u with original graph convolution mechanism, z_u^p contains the information of $\{x_p|p \in G^L(u)\}$. Thus, the union of z_u for $u \in G^1(v)$ covers $\{x_p|p \in G^L(u), u \in G^1(v)\} = \{x_p|p \in G^{L+1}(v)\}$, which includes $\{x_u|u \in G^{L+1}(v) \setminus G^L(v)\}$.

Obviously, $\{x_u, z_u|u \in G^1(v)\} \cup \{\bar{z}_v\}$ contains the same $L+1$ ego-graph content as $\{x_u|u \in G^{L+1}(v)\}$ does, we have Lemma 1 proved. □

Statement 2 (Comparison between embedding-fused graph convolution and original graph convolution)**.** *For a node v, we denote the prediction, computed by K layers of embedding-fused graph convolution on its K-hop impaired ego-graph mended with deep neighbors of L-hop local contexts, as $\tilde{y}'_v$, and the prediction, computed by (K+L) layers of graph convolution on its (K+L)-hop ego-graph, as $\tilde{y}_v$, where $K, L \in \mathbb{N}^*$. $\tilde{y}'_v$ and $\tilde{y}_v$ are the compound vectors for the same local context of v.*

Proof. To prove Statement 2, we extend Lemma 1 from 1-hop impaired ego-graph to the K-hop impaired ego-graph mended with L-hop local missing context embeddings.

By iterativly applying Lemma 1 K-L times, we have node v's prediction $\tilde{y}'_v$ computed on $\{x_u, z_u|u \in G^K(v)\}$ with z_u containing the information of $\{x_p|p \in G^L(u)\}$. The entire content is the same as where $\tilde{y}_v$ is retrieved with original graph convolution, *i.e.*, $\{x_p|p \in G^{K+L}(u)\}$. In this way, we have Statement 2 proved.

□

II. Proof for Theorem 1

Lemma 2 (Noise-Free Edge-LDP of mini-batch GCN after one epoch on 1-hop ego-graphs)**.** *Given a graph, with its nodes' degrees by at least D, and a GCN model for embedding computation, after one epoch of mini-batch training on 1-hop ego-graphs drawn from the graph with sampling size as d, the GCN achieves at most $(\ln \frac{D+1}{D+1-d}, \frac{d}{D})$-edge-LDP when $d < D$, and at least $(d \ln \frac{D+1}{D}, 1-(\frac{D-1}{D})^d)$-edge-LDP otherwise.*

Proof. To prove Lemma 2, we first revisit the NFDP mechanisms [1] on (ε, δ)-differential privacy of different sampling policies.

Theorem B.1 (NFDP mechanism [1]: (ε, δ)-differential privacy of sampling without replacement)**.** *Given a training dataset of size D, sampling without replacement achieves $(\ln \frac{D+1}{D+1-d}, \frac{d}{D})$- differential privacy, where d is the subsample size.*

Theorem B.2 (NFDP mechanism [1]: (ε, δ)-differential privacy of sampling with replacement)**.** *Given a training dataset of size D, sampling with replacement achieves $(d \ln \frac{D+1}{D}, 1-(\frac{D-1}{D})^d)$- differential privacy, where d is the subsample size.*

To apply Theorem B.1 and Theorem B.2 in Lemma 2, we can regard the 1-hop neighbor list of the target node v, *i.e.*, the neighbors on the 1-hop ego-graph of v, as the entire dataset with size D, and the mini-batch sampling node size is the subsampling size d.

In this way, one epoch of training the GCN model with the mini-batch sampling has two cases. One case is when $d < D$, while the other is $d \geq D$. For the neighbor sampling method, we follow the implementation of FederatedScope [2], where the former case uses the sampling without replacement, and the latter case uses the sampling with replacement. Therefore, when $d < D$, the sampling can achieve $(\ln \frac{D+1}{D+1-d}, \frac{d}{D})$-differential privacy for the neighbor list, and $(d \ln \frac{D+1}{D}, 1-(\frac{D-1}{D})^d)$-differential privacy otherwise.

To transfer the general DP to the edge-LDP, we need to analyze it according to the definition of edge-LDP and differential privacy. We revisit the definition of general DP as follows.

Definition B.1 ((ε, δ)-differential privacy)**.** *A randomized mechanism $\mathcal{M} : \mathcal{A} \to B$ with domain $\mathcal{A}$ and range B satisfies (ε, δ)-differential privacy if for all two neighboring inputs $U, U' \in \mathcal{A}$ that differ by one record, and any measurable subset of outputs $S \subseteq B$ it holds that*

$$Pr[\mathcal{M}(U) \in S] \leq e^{\varepsilon} Pr[\mathcal{M}(U') \in S] + \delta \tag{1}$$

Then we revisit the definition of edge-LDP as below.

Definition 1. *For a graph with n nodes, denote its node v's neighbor list as $(b_1, \ldots, b_n)$. For $u \in [n]$, if v is linked with v, b_u is 1. Otherwise, b_u is 1. Let $\varepsilon, \delta \in \mathbb{R}_{\geq 0}$, and $R : \mathcal{G} \to \mathbb{R}$ is a randomized algorithm. R provides (ε, δ)-edge-LDP if for any two local neighbor lists γ, γ' that differ in one bit and any $S \subseteq R$,*

$$Pr[R(\gamma) \in S] \leq e^{\varepsilon} Pr[R(\gamma') \in S] + \delta. \tag{2}$$

By regarding the input dataset U, U' in Eq. (1) as two neighbor lists γ, γ' in Eq. (2), we have general differential privacy transferred to edge-LDP. As the mini-batch sampling GCN can achieve γ, γ' in Eq. (2) through whether sampling a neighbor node in the ego-graph, we transfer the sampling in NFDP of (ε, δ)-differential privacy to the equal effect of the mini-batching sampling in noise-free (ε, δ)-edge-LDP.

Since nodes on a graph can have different degrees, and the lower bound of the protection implies the privacy of this mechanism, we choose the max values of (ε, δ) by calculating them using the minimum degree among all nodes. In this way, Lemma 2 is proved. □

Lemma 3 (Noise-Free Edge-LDP after N epochs of L-hop mini-batch embedding computation)**.** *For a subgraph, given every node's L-hop ego-graph with its every L-1 hop nodes of degrees by at least D, and a GCN model for embedding computation, after N epochs of mini-batch training with each hop of sampling size as d, the GCN achieves $(\tilde{\varepsilon}, \tilde{\delta})$-edge-LDP, where*

$$\tilde{\varepsilon} = \min\{LN\varepsilon, LN\varepsilon\frac{(e^{\varepsilon}-1)}{e^{\varepsilon}+1} + \varepsilon U\sqrt{2LN}\},$$
$$\tilde{\delta} = (1-\delta)^{LN}(1-\delta'),$$

and $U = \min\{\sqrt{\ln(e + \frac{\varepsilon\sqrt{LN}}{\delta'})}, \sqrt{\ln(\frac{1}{\delta'})}\}$, for $\delta' \in [0, 1]$, and (ε, δ) are $(\ln \frac{D+1}{D+1-d}, \frac{d}{D})$ and $(d \ln \frac{D+1}{D}, 1-(\frac{D-1}{D})^d)$ in Lemma 2 for respective cases.

Proof. To prove Lemma 3, we need to adaptively apply Lemma 2 by N epochs on the L times of graph convolution, *i.e.*, total LN times. Thus, we revisit the Composition of Differentially Private Mechanisms [3] as follows.

Theorem B.3 (Composition of Differentially Private [3])**.** *For any $\varepsilon > 0$, $\delta, \delta' \in [0, 1] > 0$, the class of (ε, δ)-differential private mechanisms satisfies $(\tilde{\varepsilon}, 1-(1-\delta)^k(1-\delta'))$-differential private under k-fold adaptive composition, for*

$$\tilde{\varepsilon} = \min\{k\varepsilon, k\varepsilon\frac{(e^{\varepsilon}-1)}{e^{\varepsilon}+1} + \varepsilon\sqrt{2k}\min\{\sqrt{\ln(e + \frac{\varepsilon\sqrt{k}}{\delta'})}, \sqrt{\ln(\frac{1}{\delta'})}\}\}$$

By firstly aligning general differential privacy to edge-LDP as we described in the proof of Lemma 2, obviously, we have the same conclusion of the composition rule for edge-LDP as Theorem B.3. Then we substitute the k in the composition rule to LN, and specifying the (ϵ, δ) as the pairs in Lemma 2. Thus, Lemma 3 is proved.

□

Theorem 1 (Noise-free edge-LDP of FedDEP)**.** *For a distributed subgraph system, on each subgraph, given every node's L-hop ego-graph with its every L-1 hop neighbors of degrees by at least D, FedDEP unifies all subgraphs in the system to federally train a joint model of a classifier and a cross-subgraph deep neighbor generator, as specified in Section IV. By learning from deep neighbor embeddings that are obtained from locally trained GNNs in N epochs of* *mini-batch training with a sampling size for each hop as* d*, FedDEP achieves* $(\log(1+r(e^{\tilde{\varepsilon}}\text{-}1), r\tilde{\delta})$*-edge-LDP, where*

$$\tilde{\varepsilon} = \min\{LN\varepsilon, LN\varepsilon\frac{(e^{\varepsilon}-1)}{e^{\varepsilon}+1} + \varepsilon U\sqrt{2LN}\},$$
$$\tilde{\delta} = (1-\delta)^{LN}(1-\delta'), \quad \delta' \in [0,1],$$

and $U = \min\{\sqrt{\ln(e+\frac{\varepsilon\sqrt{LN}}{\delta'})}, \sqrt{\ln(\frac{1}{\delta'})}\}$*.* r *is the expected value of the Bernoulli sampler in DGen. When* $d < D$*,* (ε, δ) *are tighter than* $(\ln\frac{D+1}{D+1-d}, \frac{d}{D})$*; when* $d \geq D$*,* (ε, δ) *are tighter than* $(d\ln\frac{D+1}{D}, 1-(\frac{D-1}{D})^{d})$*. Both pairs of* (ε, δ) *serve as the lower bounds of the edge-LDP protection under the corresponding cases.*

Proof. FedDEP framework first pre-calculates the embeddings from a mini-batch trained GCN to retrieve prototype sets, then it leverages the deep neighbor generator that employs a Bernoulli sampler R with expected value r to jointly train a classifier on subgraphs mended with generated deep neighbor prototypes.

To prove Theorem 1, we revisit the privacy amplification by subsampling in the general DP [4].

Theorem B.4 (privacy amplification by subsampling [4])**.** *Given a dataset* U *with* n *data records, subsampling mechanism* $\mathcal{S}$ *subsamples a subset of data* $\{d_i|\sigma_i = 1, i \in [n]\}$ *by sampling* $\sigma_i \sim Ber(p)$ *independently for* $i \in [n]$*. If mechanism* $\mathcal{M}$ *satisfied* (ε, δ)*-differential privacy, mechanism* $\mathcal{M} \circ \mathcal{S}$ *is* $(\log(1+p(e^{\varepsilon-1}), p\delta)$*-differential private.*

We prove Theorem 1 by applying Theorem B.4 and Lemma 3 in four steps.

We first transfer the conclusion of Theorem B.4 into edge-LDP by following the proof of Lemma 2. Then we specify the (ε, δ)-differential privacy mechanism $\mathcal{M}$ in Theorem B.4 as the edge-LDP embedding computation GCN model in Lemma 3 with respective privacy-related parameters. Next, we specify the subsampling mechanism $\mathcal{S}$ in Theorem B.4 as the Bernoulli sampler in FedDEP with DGen on prototypes. By substituting the p in Theorem B.4 to r, we have Theorem 1 proved.

□

References

[1] L. Sun and L. Lyu, “Federated model distillation with noise-free differential privacy,” in *IJCAI*, 2021.

[2] Z. Wang, W. Kuang, Y. Xie, L. Yao, Y. Li, B. Ding, and J. Zhou, “Federatedscope-gnn: Towards a unified, comprehensive and efficient package for federated graph learning,” in *KDD*, 2022.

[3] P. Kairouz, S. Oh, and P. Viswanath, “The composition theorem for differential privacy,” in *ICML*, 2015.

[4] B. Balle, G. Barthe, and M. Gaboardi, “Privacy profiles and amplification by subsampling,” *Journal of Privacy and Confidentiality*, vol. 10, no. 1, 2020.