Secure Graph Convolutional Network on Vertically Split Data from Sparse Matrix Decomposition

Yu Zheng* [1] Qizhi Zhang* [2] Chenang Li [1] Lichun Li [2]
Kai Zhou[3] Shan Yin[2]

[1]University of California, Irvine [2]Ant Group [3]Hong Kong Polytechnic University

Background and Motivation

Table 1. MPC Frameworks for Secure Graph Learning. 'Inf' and 'Tra' denote inference and training. 'SS', 'HE', and 'OT' denote secret sharing, homomorphic encryption, and oblivious transfer, respectively.

Framework	Scenario	Inf.	Tra.	Crypto. Primitive	Security
OblivGNN (Sec' 24) [1]	MLaaS	✓	×	Function SS, SS	Semi-honest
CoGNN (CCS' 24) [2]	Horizontal	✓	✓	HE, OT, SS	Semi-honest
Grace	Vertical	✓	✓	SS	Semi-honest

Securely computing graph convolutional networks (GCNs) is critical for applying their analytical capabilities to privacy-sensitive data like social/credit networks. Multiplying a sparse yet large adjacency matrix of a graph in GCN—a core operation in training/inference—poses a performance bottleneck in secure GCNs. Consider a GCN with $|\mathcal{V}|$ **nodes** and $|\mathcal{E}|$ **edges**; it incurs a large $O(|\mathcal{V}|^2)$ communication overhead.

Challenges. i) Integrating existing cryptographic advances requires heavy communication or computational overhead. ii) Current MPC studies have primarily targeted horizontal partitioning, while MPC specialized for vertical partitioning remains lacking.

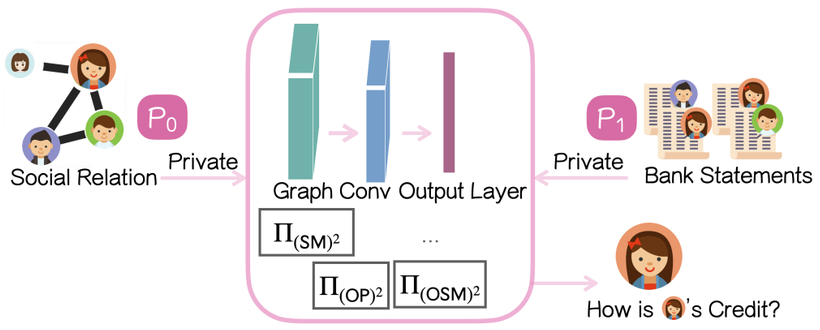


Figure 1. Ideal Functionality of Our Framework Grace

QUESTION: Can we propose a secure sparse matrix multiplication protocol, achieving *accurate*, ***efficient***, *and* ***secure GCN training*** *over* ***vertical*** *data* for the *first* time?

Our Contribution

Secure Sparse Matrix Multiplication ((SM)^2). Modeling bipartite graphs and leveraging the monotonicity of non-zero entry locations, we propose a co-design harmonizing secure multi-party computation (MPC) with matrix sparsity.

- **Sparsity Decomposition.** Our sparse matrix decomposition transforms an arbitrary sparse matrix into a product of structured matrices.
- **Specialized MPC Primitives.** Specialized MPC protocols for oblivious permutation and selection multiplication are then tailored, enabling our secure sparse matrix multiplication ((SM)^2) protocol, optimized for secure multiplication of these structured matrices.
- **(SM)^2 Protocol by Concatenation.** Together, we concatenate the MPC primitives in an ordered sequence; As a result, (SM)^2 protocol takes $O(|\mathcal{E}|)$ communication in constant rounds.

Private Graph Learning Framework (Grace). Supported by (SM)^2, we present Grace (Figure 1):

- building upon light 2PC secret sharing only;
- *communication-efficient* and *memory-friendly* with preserved accuracy.

Security Model

2PC Security. As in Figure 1, two parties $\mathcal{P}_0, \mathcal{P}_1$ aim to jointly train a $\mathsf{GCN}(\mathsf{A}, \mathsf{X}; \mathsf{W})$ without revealing their private inputs and intermediate computations.

- Graph owner $\mathcal{P}_0$: an adjacency matrix A corresponding to a private graph $\mathcal{G}$;
- Feature owner $\mathcal{P}_1$: private node features X.

The graph leakages are $|\mathcal{V}|, |\mathcal{E}|, |\mathcal{V}| + |\mathcal{E}|$, which are tolerable (and unavoidable) since the efficiency gain is correlated to $|\mathcal{E}|$. Pre-computations on correlated randomness are realized by a data-independent offline phase.

Theoretical Result

Theorem 1. (Sparse Matrix Decomposition). Let an $m \times n$ sparse matrix $\mathsf{A} \in \mathbb{M}_{m,n}(\mathcal{R})$ contain n_{row} non-zero rows, n_{col} non-zero columns, and t non-zero elements. Then, there exists a matrix decomposition $\mathsf{A} = \sigma_5 \delta_m^\top \Gamma_{\mathsf{out}} \sigma_4 \Sigma^\top \Lambda \sigma_3 \Sigma \sigma_2 \Gamma_{\mathsf{in}} \delta_n \sigma_1$, where $\sigma_5 \in \mathbb{S}_m$, $\sigma_4 \in \mathbb{S}_t, \sigma_3 \in \mathbb{S}_t, \sigma_2 \in \mathbb{S}_t, \sigma_1 \in \mathbb{S}_n$, and, 1) $\Sigma = (\Sigma[i,j])_{i,j=1}^{t}$ is the left-down triangle matrix such that $\Sigma[i,j] = 1$ if $i \geq j$ or 0 otherwise, 2) $\delta_k = (\delta_k[i,j])_{i,j=1}^{k}$ is the left-down triangle matrix such that $\delta_k[i,j] = 1$ for $i = j$ or -1 for $j = i-1$, or 0 otherwise, 3) $\Gamma_{\mathsf{in}} = (\Gamma_{\mathsf{in}}[i,j])_{i=1,j=1}^{t,n}$ is a matrix such that $\Gamma_{\mathsf{in}}[i,j] = 1$ for $1 \leq i = j \leq n_{\mathsf{col}}$ or 0 otherwise, 4) $\Gamma_{\mathsf{out}} = (\Gamma_{\mathsf{out}}[i,j])_{i=1,j=1}^{m,t}$ is a matrix such that $\Gamma_{\mathsf{out}}[i,j] = 1$ for $1 \leq i = j \leq n_{\mathsf{row}}$ or 0 otherwise.

Secure Sparse Matrix Multiplication

Our efficient MPC design avoids unnecessary secure computation over unrelated nodes by focusing on computing non-zero results while concealing the sparse topology.

1). Sparsity Decomposition.

- Decompose A of $\mathcal{G}$ into two bipartite graphs:
  - $\mathsf{A}_{\mathsf{out}}$: linking the out-degree nodes to edges;
  - A_{in}: linking edges to in-degree nodes.
- Represent the topology into a sequence of *linear transformations*. (Figure 2)

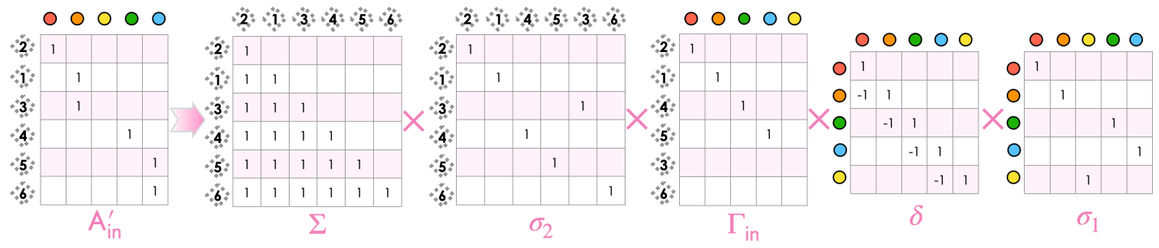
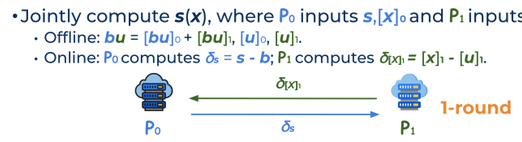


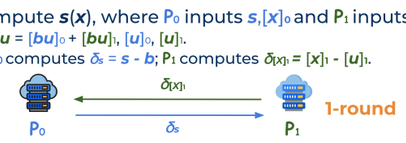
Figure 2. Re-decomposition of $\mathsf{A}'_{\mathsf{in}}$

2). Specialized MPC Primitives.

- *OP*: a private permutation σ over a private vector X to get $\langle \sigma \mathsf{X} \rangle$.
- *OSM*: a private bit s multiplying an arithmetic number x to get $\langle sx \rangle$.



(a) Oblivious Permutation (OP)



(b) Oblivious Selection-Multiplication (OSM)

3). (SM)^2 Protocol by Concatenation. (SM)^2 protocol (Figure 3) concatenates each sub-protocol one by one to get $\langle \mathsf{AX} \rangle$.

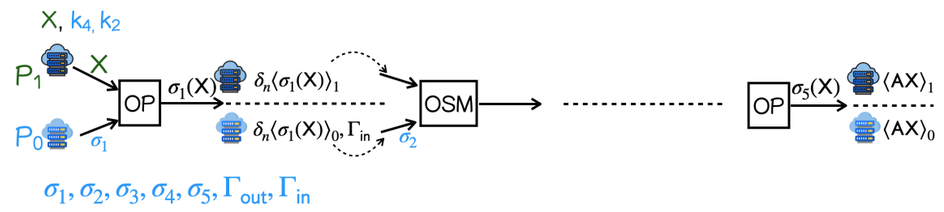


Figure 3. (SM)^2 Protocol Realization

Evaluation on Private GCN

- **Environment:** three Ubuntu servers with 16-core Intel(R) Xeon(R) Platinum 8163 2.50GHz CPUs of 62GB RAM and NVIDIA-T4 GPU of 16GB RAM;
- **Datasets:** Cora, Citeseer, and Pubmed.
- **Metrics:** communication, memory usage, accuracy, running time, and ablation study on (SM)^2.

1. Grace saves communication overhead by 62%-78% for training and 46%-81% for inference. (*cf.*, CoGNN [2], OblivGNN [1]).
2. Grace alleviates out-of-memory problems of using Beaver-triples for large datasets.
3. Grace achieves inference and training accuracy comparable to plaintext counterparts.
4. Grace is faster by 6%-45% in inference and 28%-95% in training across various networks and excels in narrow-bandwidth and low-latency ones.
5. (SM)^2 protocol shows a 10-42× speed-up for 5000×5000 matrices and saves 10%-21% memory for "small" datasets and up to 90%+ for larger ones.

Table 2. Communication Costs (GB/epoch) for Full-batch Training.

Framework	Phase	Dataset		
		Cora	Citeseer	Pubmed
CoGNN	Online	85.81	199.85	269.98
	Offline	1.18	2.96	3.27
CoGNN-Opt	Online	0.75	1.33	3.78
	Offline	0.07	0.07	0.55
Grace	Online	**0.23**	**0.51**	**0.84**
	Offline	0.08	0.14	0.33

Reference

[1] Zhibo Xu, Shangqi Lai, Xiaoning Liu, Alsharif Abuadbba, Xingliang Yuan, and Xun Yi. Oblivgnn: Oblivious inference on transductive and inductive graph neural network. In *USENIX Security*, 2024.

[2] Zhenhua Zou, Zhuotao Liu, Jinyong Shan, Qi Li, Ke Xu, and Mingwei Xu. CoGNN: Towards secure and efficient collaborative graph learning. In *CCS*, 2024.