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

Yu Zheng, *University of California, Irvine*
Qizhi Zhang, *ByteDane*
Lichun Li, *Ant Group*
Kai Zhou, *Hong Kong Polytechnic University*
Shan Yin, *Ant Group*

***Abstract*—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.**

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. Our sparse matrix decomposition transforms an arbitrary sparse matrix into a product of structured matrices. Specialized MPC protocols for oblivious permutation and selection multiplication are then tailored, enabling our secure sparse matrix multiplication ($(\text{SM})^2$) protocol, optimized for secure multiplication of these structured matrices. Together, these techniques take $O(|\mathcal{E}|)$ communication in constant rounds. Supported by $(\text{SM})^2$, we present GRACE, a secure 2-party framework that is communication-efficient and memory-friendly on standard vertically-partitioned graph datasets. Performance of GRACE has been empirically validated across diverse network conditions.

1. Introduction

Graphs, representing structural data and topology, are widely used across various domains, such as social networks and merchandising transactions. Graph convolutional networks (GCN) [1] have significantly enhanced model training on these interconnected nodes. However, these graphs often contain sensitive information that should not be leaked to untrusted parties. For example, companies may analyze sensitive demographic and behavioral data about users for applications ranging from targeted advertising to personalized medicine. Given the data-centric nature and analytical power of GCN training, addressing these privacy concerns is imperative.

Secure multi-party computation (MPC) [2], [3], [4] is a critical tool for privacy-preserving machine learning, enabling mutually distrustful parties to collaboratively train models with privacy protection over inputs and (intermediate) computations. While research advances (*e.g.*, [5], [6], [7], [8], [9], [10], [11]) support secure training on convolutional neural networks (CNNs) efficiently, private GCN training with MPC over graphs remains challenging.

TABLE 1: MPC Frameworks for Secure Graph Learning

Framework	Scenario	Inference	Training	Security
OblivGNN [15]	MLaaS	✓	×	Semi-honest
LinGCN [16]	MLaaS	✓	×	Semi-honest
Penguin [13]	MLaaS	✓	×	Semi-honest
CoGNN [14]	Horizontal	✓	✓	Semi-honest
GRACE	Vertical	✓	✓	Semi-honest

Graph convolutional layers in GCNs involve multiplications with a (normalized) adjacency matrix containing $|\mathcal{E}|$ non-zero values in a $|\mathcal{V}| \times |\mathcal{V}|$ matrix for a graph with $|\mathcal{V}|$ nodes and $|\mathcal{E}|$ edges. The graphs are typically sparse but large. One could use the standard Beaver-triple-based protocol to securely perform these sparse matrix multiplications by treating graph convolution as ordinary dense matrix multiplication. However, this approach incurs $O(|\mathcal{V}|^2)$ communication and memory costs due to computations on irrelevant nodes. Integrating existing cryptographic advances, the initial effort of SecGNN [12], [13] requires heavy communication or computational overhead. Recently, CoGNN [14] optimizes the overhead in terms of horizontal data partitioning, proposing a semi-honest secure framework. As in Table 1, research for secure GCN over vertical data remains nascent.

Current MPC studies, for GCN or not, have primarily targeted settings where participants own different data samples, *i.e.*, horizontally partitioned data [14]. MPC specialized for scenarios where parties hold different types of features [17], [18], [19] is rare. This paper studies 2-party secure GCN training for these vertical partition cases, where one party holds private graph topology (*e.g.*, edges) while the other owns private node features. For instance, LinkedIn holds private social relationships between users, while banks own users' private bank statements. Such real-world graph structures underpin the relevance of our focus. To our knowledge, no prior work tackles secure GCN training in this context, which is crucial for cross-silo collaboration.

To realize secure GCN over vertically split data, we tailor MPC protocols for sparse graph convolution, which fundamentally involves sparse (adjacency) matrix multiplication. Recent studies have begun exploring MPC protocols for sparse matrix multiplication (SMM). ROOM [20], a

seminal work on SMM, requires foreknowledge of sparsity types: whether the input matrices are row-sparse or column-sparse. Unfortunately, GCN typically trains on graphs with arbitrary sparsity, where nodes have varying degrees and no specific sparsity constraints. Moreover, the adjacency matrix in GCN often contains a self-loop operation represented by adding the identity matrix, which is neither row- nor column-sparse. Araki *et al.* [21] avoid this limitation in their scalable, secure graph analysis work, yet it does not cover vertical partition.

To bridge this gap, we propose a secure sparse matrix multiplication protocol, $(\mathrm{SM})^2$, achieving *accurate, efficient, and secure GCN training over vertical data* for the first time.

2. Methods

2.1. New Techniques for Sparse Matrices

The cost of evaluating a GCN layer is dominated by SMM in the form of AX, where A is a sparse adjacency matrix of a (directed) graph $\mathcal{G}$ and X is a dense matrix of node features. For unrelated nodes, which often constitute a substantial portion, the element-wise products $0 \cdot x$ are always zero. Our efficient MPC design avoids unnecessary secure computation over unrelated nodes by focusing on computing non-zero results while concealing the sparse topology. We achieve this by: 1) decomposing the sparse matrix A into a product of matrices, including permutation and binary diagonal matrices, that can *faithfully* represent the original graph topology; 2) devising specialized protocols for efficiently multiplying the structured matrices while hiding sparsity topology.

2.1.1. Sparse Matrix Decomposition. We decompose adjacency matrix A of $\mathcal{G}$ into two bipartite graphs: one represented by sparse matrix $\mathsf{A}_{\mathsf{out}}$, linking the out-degree nodes to edges, the other by sparse matrix A_{in}, linking edges to in-degree nodes.

We then permute the columns of $\mathsf{A}_{\mathsf{out}}$ and the rows of A_{in} so that the permuted matrices $\mathsf{A}'_{\mathsf{out}}$ and $\mathsf{A}'_{\mathsf{in}}$ have non-zero positions with *monotonically non-decreasing* row and column indices. A permutation σ is used to preserve the edge topology, leading to an initial decomposition of $\mathsf{A} = \mathsf{A}'_{\mathsf{out}}\sigma\mathsf{A}'_{\mathsf{in}}$. This is further refined into a sequence of *linear transformations*, which can be efficiently computed by our MPC protocols for *oblivious permutation* and *oblivious selection-multiplication*. Our decomposition approach is not limited to GCNs but also general sparse matrices.

2.1.2. New Protocols for Linear Transformations. *Oblivious permutation* (OP) is a two-party protocol taking a private permutation σ and a private vector X from the two parties, respectively, and generating a secret share $\langle\sigma\mathsf{X}\rangle$ between them. Our OP protocol employs correlated randomnesses generated in an input-independent offline phase to mask σ and X for secure computations on intermediate results, requiring only 1 round in the online phase (*cf.*, ≥ 2 in previous works [21], [22]).

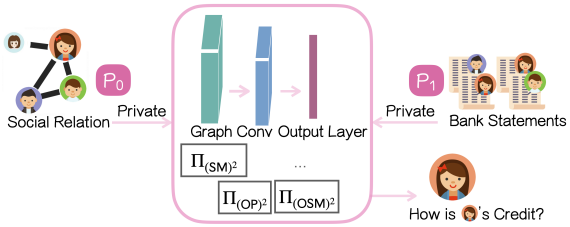

Figure 1: Ideal Functionality of GRACE

Another crucial two-party protocol in our work is *oblivious selection-multiplication* (OSM). It takes a private bit s from a party and secret share $\langle x\rangle$ of an arithmetic number x owned by the two parties as input and generates secret share $\langle sx\rangle$. Our 1-round OSM protocol also uses precomputed randomnesses to mask s and x. Compared to the Beaver-triple-based [23] and oblivious-transfer (OT)-based approaches [24], our protocol saves $\sim$50% of online communication while having the same offline communication and round complexities.

By decomposing the sparse matrix into linear transformations and applying our specialized protocols, our $(\mathrm{SM})^2$ protocol reduces the complexity of evaluating $|\mathcal{V}| \times |\mathcal{V}|$ sparse matrices with $|\mathcal{E}|$ non-zero values from $O(|\mathcal{V}|^2)$ to $O(|\mathcal{E}|)$.

2.2. GRACE: Secure GCN made Efficient

Supported by our new sparsity techniques, we build GRACE, a two-party computation (2PC) framework for GCN inference and training over vertical data.

2.2.1. Workflow of GRACE. Figure 1 outlines GRACE's function. A graph owner $\mathcal{P}_0$, with an adjacency matrix A corresponding to a private graph $\mathcal{G}$, and a feature owner $\mathcal{P}_1$ with private node features X, aim to jointly train a GCN without revealing their private inputs. This involves computing a parameterized function $\mathsf{GCN}(\mathsf{A}, \mathsf{X}; \mathsf{W})$, where the weights W are secret-shared over the two parties.

The GRACE framework includes a sparse matrix decomposition method and secure 2PC protocols for permutation (Π_{OP}), selection-multiplication (Π_{OSM}), and SMM ($\Pi_{(\mathsf{SM})^2}$). The sparse matrix decomposition is performed solely by the graph owner, while all 2PC protocols are executed by both parties without disclosing any intermediate computations.

In practical cross-institution collaboration, graph owners can be social networking platforms (*e.g.*, Facebook) holding social relationships as a graph, and feature owners can be banks holding users' bank statements as node features. As a motivating example, they may want to build a credit-investigation model for predicting the credit of a loaner for future repayment while keeping their data confidential. Our setting can be extended to multi-party, where different types of node features are learned from different parties (*e.g.*, bank statements from banks and transactions from online-shopping companies). Usually, the graph structure is fixed to represent a specific relationship, such as a social circle, in

real-world scenarios. Thus, we focus on single-party graph ownership without limiting feature ownerships.

2.2.2. Security Model. GRACE can be instantiated with any type of security models offered by the corresponding MPC protocols. Following advances [7], [8], [14], [25], [26], [27], GRACE focuses on 2PC security against the static semi-honest probabilistic polynomial time (PPT) adversary $\mathcal{A}$ regarding the real/ideal-world simulation paradigm [28]. Specifically, two parties, $\mathcal{P}_0$ and $\mathcal{P}_1$, with inputs $\langle x\rangle_0$ and $\langle x\rangle_1$, want to compute a function $y = f(\langle x\rangle_0, \langle x\rangle_1)$ without revealing anything other than y. $\mathcal{A}$ corrupts either $\mathcal{P}_0$ or $\mathcal{P}_1$ at the start, following the protocol, but tries to learn the other's private inputs. $\mathcal{A}$ can only learn data from the corrupted party but nothing from honest ones.

Many protocols utilize pre-computations for improving efficiency, *e.g.*, Beaver triples [23] for multiplication. They can be realized by a data-independent offline phase run by a semi-honest dealer $\mathcal{T}$ or 2PC protocols from homomorphic encryption [29] or oblivious transfer [24], [30] or oblivious shuffle [31], [32]. We adopt the first common approach (also called client-aided setting [25]) for simplicity. The $\mathcal{T}$ does not interact with any party (particularly, receives nothing) online. It only generates pseudo-randomnesses in an input-independent offline phase by counter-indexed computations of pseudorandom function (PRF), where $\mathcal{T}$ and $\mathcal{P}_i$ share a PRF key (denoted by key_i) for $i \in \{0, 1\}$ and a counter ctr are synchronized among all parties.

2.2.3. Scope of Graph Protection. Like existing MPC works, GRACE protects the entry values stored in the graph and (intermediate) computations. For metadata, most secure matrix multiplication protocols (without sparse structure) reveal input dimensionality (*e.g.*, $|\mathcal{V}|$ in GCN) that is typically considered public knowledge. When sparsity is explored, it is normal to leak reasonable knowledge, such as $|\mathcal{V}| + |\mathcal{E}|$ in GraphSC [33]. In GRACE, the only additional metadata revealed is $|\mathcal{E}|$. This leakage is tolerable (and unavoidable) since the efficiency gain is correlated to $|\mathcal{E}|$. Corresponding to GRACE's GCN training, the *dimension* of adjacency matrix A (*i.e.*, equal to $|\mathcal{V}|$) and the *dimension* of feature matrix X are assumed to be public.

Privacy leakages from training/inference results, *e.g.*, embedding inversion and sensitive attribute inference, also appear in plaintext computations and are beyond our scope. These can be protected via orthogonal techniques like (local) differential privacy and robustness training, which are compatible with our work. In the semi-honest settings, the attacker can only view the well-formed secret shares and not actively perform the malicious attacks like model inversion.

3. Results & Evaluation

We are the first to explore sparsity over vertically split, secret-shared data in MPC, enabling decompositions of sparse matrices with arbitrary sparsity and isolating computations that can be performed in plaintext without sacrificing privacy.

We propose two efficient 2PC primitives for OP and OSM, both optimally single-round. Combined with our sparse matrix decomposition approach, our $(\mathrm{SM})^2$ protocol ($\Pi_{(\mathsf{SM})^2}$) achieves constant-round communication costs of $O(|\mathcal{E}|)$, reducing memory requirements and avoiding out-of-memory errors for large matrices. In practice, it saves $99\%+$ communication and reduces $\sim 72\%$ memory usage over large (5000×5000) matrices compared with using Beaver triples.

We build an end-to-end secure GCN framework for inference and training over vertically split data, maintaining accuracy on par with plaintext computations. We evaluate the performance of our $(\mathrm{SM})^2$ protocol and GRACE's private GCN inference/training on 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. To evaluate GRACE, we conducted extensive experiments over three standard graph datasets (Cora [34], Citeseer [35], and Pubmed [36]), reporting communication, memory usage, accuracy, and running time under varying network conditions, along with an ablation study with or without $(\mathrm{SM})^2$. Below, we highlight our key achievements.

Communication. GRACE saves communication overhead by 62%-78% for training and 46%-81% for inference. (*cf.*, CoGNN [27], OblivGNN [15]).

Memory usage. GRACE alleviates out-of-memory problems of using Beaver-triples [23] for large datasets.

Accuracy. GRACE achieves inference and training accuracy comparable to plaintext counterparts.

Computational efficiency. 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.

Impact of $(\mathrm{SM})^2$. Our $(\mathrm{SM})^2$ protocol shows a 10-$42\times$ speed-up for 5000×5000 matrices and saves 10-21% memory for "small" datasets and up to 90%+ for larger ones.

4. Conclusion

We propose GRACE, a secure 2PC framework for GCN inference and training over vertically partitioned data, a neglected MPC scenario motivated by cross-institutional business collaboration. It is supported by our $(\mathrm{SM})^2$ protocol using a sparse matrix decomposition method for converting an arbitrary-sparse matrix into a sequence of linear transformations and employing 1-round MPC protocols of oblivious permutation and selection-multiplication for efficient secure evaluation of these linear transformations.

Our work provides an open-source baseline and extensive benchmarks for practical usage. Theoretical and empirical analysis demonstrate GRACE's superior communication and memory efficiency in private GCN computations. For further research with various partitioning, researchers can streamline the hybrid MPC protocols by integrating plaintext handcrafts, leveraging secure computation as a pragmatic alternative for cross-organizational collaboration. Hopefully, our insight could motivate further research on private graph learning.

References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[2] D. Chaum, I. Damgard, and J. van de Graaf, "Multiparty computations ensuring privacy of each party's input and correctness of the result," in *CRYPTO*, 1987, pp. 87–119.

[3] H. Chen and R. Cramer, "Algebraic geometric secret sharing schemes and secure multi-party computations over small fields," in *CRYPTO*, 2006, pp. 521–536.

[4] M. Ciampi, D. Ravi, L. Siniscalchi, and H. Waldner, "Round-optimal multi-party computation with identifiable abort," in *EUROCRYPT*, 2022, pp. 335–364.

[5] D. Rathee, M. Rathee, N. Kumar, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma, "CrypTFlow2: Practical 2-party secure inference," in *CCS*, 2020, pp. 325–342.

[6] L. K. L. Ng and S. S. M. Chow, "GForce: GPU-friendly oblivious and rapid neural network inference," in *Usenix Security*, 2021, pp. 2147–2164.

[7] S. Tan, B. Knott, Y. Tian, and D. J. Wu, "CRYPTGPU: Fast privacy-preserving machine learning on the GPU," in *S&P*, 2021, pp. 1021–1038.

[8] J. Watson, S. Wagh, and R. A. Popa, "Piranha: A GPU platform for secure computation," in *Usenix Security*, 2022, pp. 827–844.

[9] M. Keller and K. Sun, "Secure quantized training for deep learning," in *ICML*, 2022, pp. 10 912–10 938.

[10] P. Mohassel and P. Rindal, "ABY3: A mixed protocol framework for machine learning," in *CCS*, 2018, pp. 35–52.

[11] L. Folkerts, C. Gouert, and N. G. Tsoutsos, "Redsec: Running encrypted discretized neural networks in seconds," in *NDSS*, 2023.

[12] S. Wang, Y. Zheng, and X. Jia, "SecGNN: Privacy-preserving graph neural network training and inference as a cloud service," *IEEE Trans. Serv. Comput.*, vol. 16, no. 4, pp. 2923–2938, 2023.

[13] R. Ran, N. Xu, T. Liu, W. Wang, G. Quan, and W. Wen, "Penguin: Parallel-packed homomorphic encryption for fast graph convolutional network inference," in *NeurIPS*, 2023.

[14] Z. Zou, Z. Liu, J. Shan, Q. Li, K. Xu, and M. Xu, "CoGNN: Towards secure and efficient collaborative graph learning," in *CCS*, 2024.

[15] Z. Xu, S. Lai, X. Liu, A. Abuadbba, X. Yuan, and X. Yi, "Oblivgnn: Oblivious inference on transductive and inductive graph neural network," in *33rd USENIX Security Symposium, USENIX Security*. USENIX Association, 2024.

[16] H. Peng, R. Ran, Y. Luo, J. Zhao, S. Huang, K. Thorat, T. Geng, C. Wang, X. Xu, W. Wen, and C. Ding, "LinGCN: Structural linearized graph convolutional network for homomorphically encrypted inference," in *NeurIPS*, 2023.

[17] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y. Zhang, and Q. Yang, "Vertical federated learning: Concepts, advances, and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3615–3634, 2024.

[18] T. Castiglia, Y. Zhou, S. Wang, S. Kadhe, N. Baracaldo, and S. Patterson, "LESS-VFL: Communication-efficient feature selection for vertical federated learning," in *ICML*, 2023, pp. 3757–3781.

[19] G. Wang, B. Gu, Q. Zhang, X. Li, B. Wang, and C. X. Ling, "A unified solution for privacy and communication efficiency in vertical federated learning," in *NeurIPS*, 2023.

[20] P. Schoppmann, A. Gascón, M. Raykova, and B. Pinkas, "Make some ROOM for the zeros: Data sparsity in secure distributed machine learning," in *CCS*, 2019, pp. 1335–1350.

[21] T. Araki, J. Furukawa, K. Ohara, B. Pinkas, H. Rosemarin, and H. Tsuchida, "Secure graph analysis at scale," in *CCS*, 2021, pp. 610–629.

[22] G. Asharov, K. Hamada, D. Ikarashi, R. Kikuchi, A. Nof, B. Pinkas, K. Takahashi, and J. Tomida, "Efficient secure three-party sorting with applications to data analysis and heavy hitters," in *CCS*, 2022, pp. 125–138.

[23] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *CRYPTO*, 1991, pp. 420–432.

[24] W. Tzeng, "Efficient 1-out-n oblivious transfer schemes," in *PKC*, 2002, pp. 159–171.

[25] N. Attrapadung, G. Hanaoka, T. Matsuda, H. Morita, K. Ohara, J. C. N. Schuldt, T. Teruya, and K. Tozawa, "Oblivious linear group actions and applications," in *CCS*, 2021, pp. 630–650.

[26] B. Knott, S. Venkataraman, A. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten, "CrypTen: Secure multi-party computation meets machine learning," in *NeurIPS*, 2021, pp. 4961–4973.

[27] N. Koti, V. B. Kukkala, A. Patra, and B. R. Gopal, "Graphiti: Secure graph computation made more scalable," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS*. ACM, 2024, pp. 4017–4031.

[28] Y. Lindell, "How to simulate it - A tutorial on the simulation proof technique," in *Tutorials on the Foundations of Cryptography*, 2017, pp. 277–346.

[29] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT*, vol. 6110, 2010, pp. 1–23.

[30] M. Keller, E. Orsini, and P. Scholl, "MASCOT: Faster malicious arithmetic secure computation with oblivious transfer," in *CCS*, 2016, pp. 830–842.

[31] M. Chase, E. Ghosh, and O. Poburinnaya, "Secret-shared shuffle," in *ASIACRYPT*, 2020, pp. 342–372.

[32] X. Song, D. Yin, J. Bai, C. Dong, and E. Chang, "Secret-shared shuffle with malicious security," in *NDSS*, 2024.

[33] K. Nayak, X. S. Wang, S. Ioannidis, U. Weinsberg, N. Taft, and E. Shi, "GraphSC: Parallel secure computation made easy," in *S&P*, 2015, pp. 377–394.

[34] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, pp. 93–106, 2008.

[35] C. L. Giles, K. D. Bollacker, and S. Lawrence, "CiteSeer: An automatic citation indexing system," in *ACM Dig. Lib.*, 1998, pp. 89–98.

[36] F. Dernoncourt and J. Y. Lee, "Pubmed 200k RCT: a dataset for sequential sentence classification in medical abstracts," in *IJCNLP*, 2017, pp. 308–313.