Penguin: Parallel-Packed Homomorphic Encryption for Fast Graph Convolutional Network Inference

Ran Ran[1], Nuo Xu[2], Tao Liu[3], Wei Wang[4],
Gang Quan[5], Wujie Wen[1]
North Carolina State University[1], Lehigh University[2],
Lawrence Technological University[3], Anonym, Inc[4], Florida International University[5],

Background

1. Homomorphic Encryption (CKKS)
2. Graph Convolution Neural Network
3. Threat Model setting

CKKS

Encoding Process

- Message vector
- Notated as: $(Z_0,\ldots,Z_{N/2-1})$

→

- Find $m(x) \in R[X]/(X^N+1)$
- Such that $m(\zeta_j)=Z_j$

→

- Scale by Δ and round
- $m'(x)= round(\Delta * m(x))$

→

- Packed in polynomial
- Notated as: $p(x)$

Encryption/Decryption Process

- Generate public key p_k and secret key $s_k=(1,s)$
- $p_k=(b,a)$
- s.t. $b+a*s = e$ for small e

→

- Encrypt to Ciphertext $c(x)$
- $c(x)=(c_0(x),c_1(x)) = v*(b,a) +(m+e_0, e_1)$

→

- Decrypt to plaintext $m(x)$
- $m(x)=c_0(x)+c_1(x)*s$

Where v, e_0, e_1 are random polynomials

Orders of Magnitude Data Size Increase after Encryption

[1] original data in 8bits, slot number is 2^15, $p(x)$ in 8 * 2^15 = 32 KB

Encryption adding one more noise polynomial, size 2X

Parameter Q=1920bits and encrypt p(x) to c(x), memory overhead increase from 32KB to **30MB (~1000X)**

Background

1. Homomorphic Encryption (CKKS)
2. Graph Convolution Neural Network
3. Threat Model setting

GCN layer

$$X^{(l+1)} = \sigma\left(AX^{(l)}W^{(l)}\right)$$

GCN layer

$$X^{(l+1)} = \sigma(AX^{(l)}W^{(l)})$$

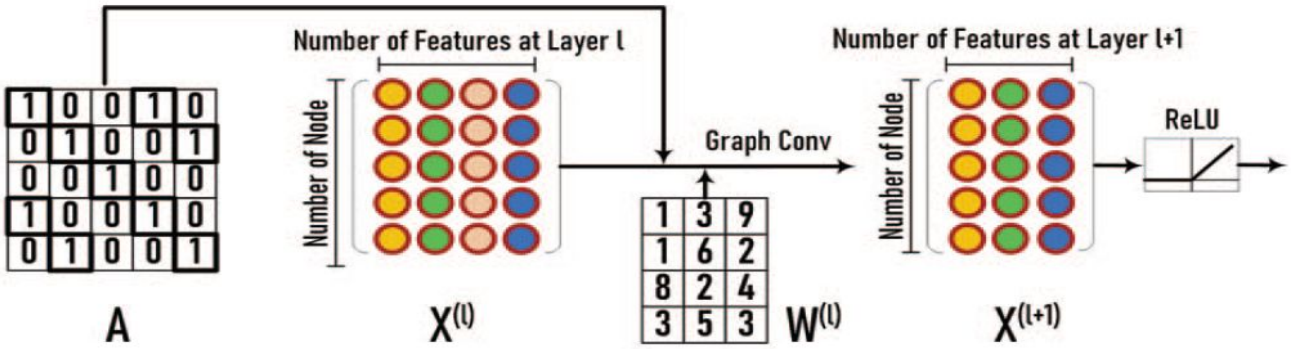
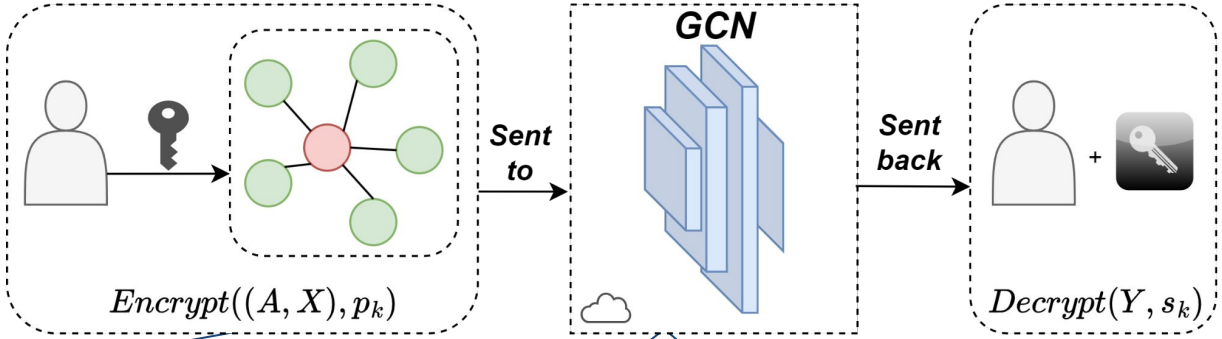


Figure source: Geng, Tong, et al. "AWB-GCN: A graph convolutional network accelerator with runtime workload rebalancing." *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2020.

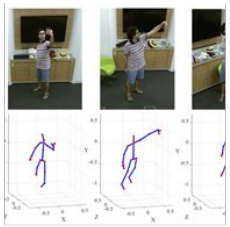
Background

1. Homomorphic Encryption (CKKS)
2. Graph Convolution Neural Network
3. Threat Model setting

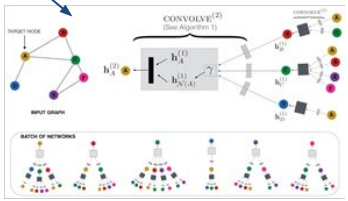
Threat Model



A - Adjacency matrix
X - Features



Human action recognition [1]



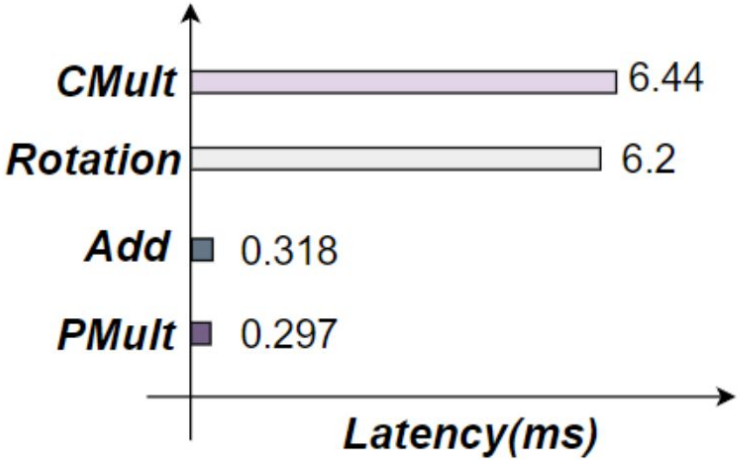
Recommendation System [2]

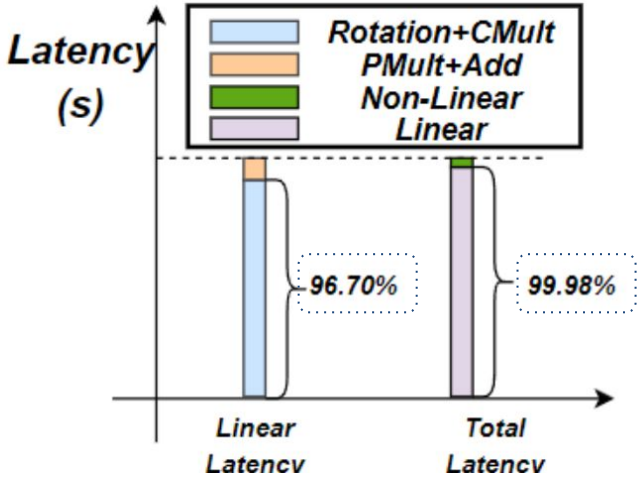
[1] Zhang, Pengfei, et al. "View adaptive neural networks for high performance skeleton-based human action recognition." *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2019): 1963-1978.
[2] Ying, Rex, et al. "Graph convolutional neural networks for web-scale recommender systems." *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018.

Motivation and Challenge

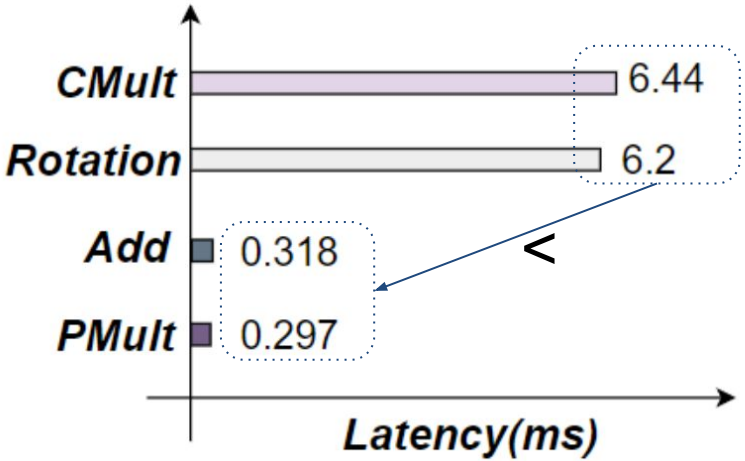
1. High-latency HE operations
2. Large memory consumption
3. Execution Order

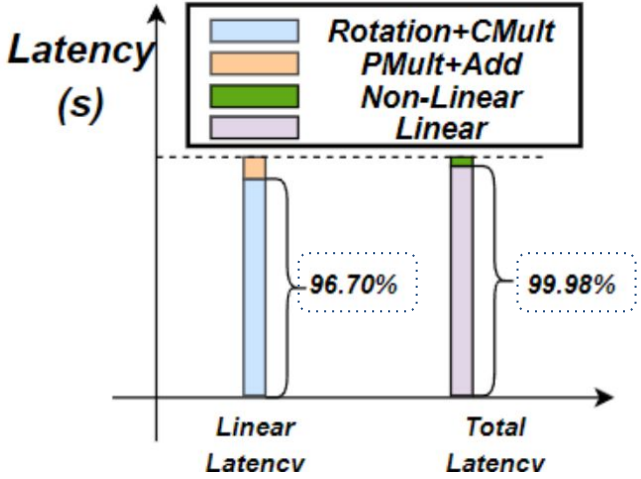
Two Operations Dominate



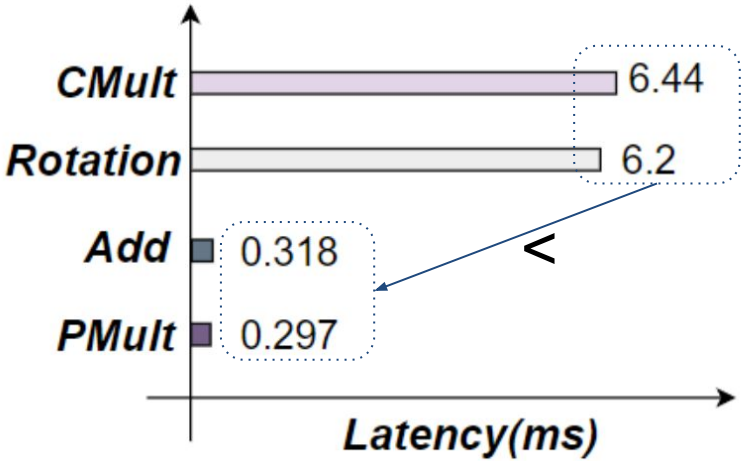


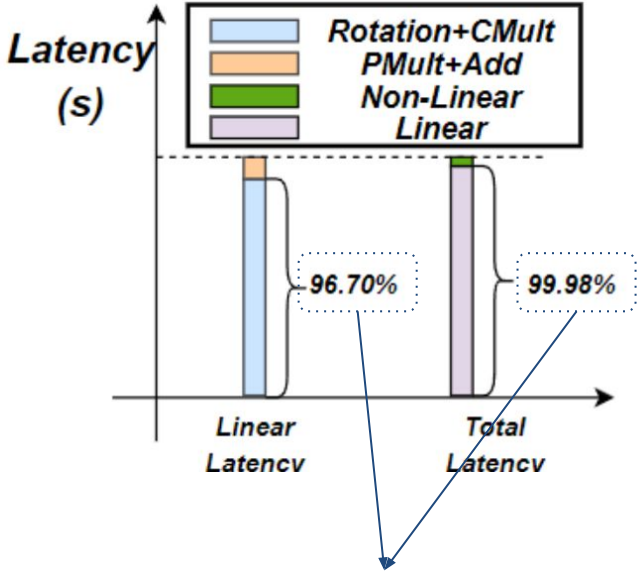
Two Operations Dominate





Two Operations Dominate





Rotation+CMult
dominate the latency

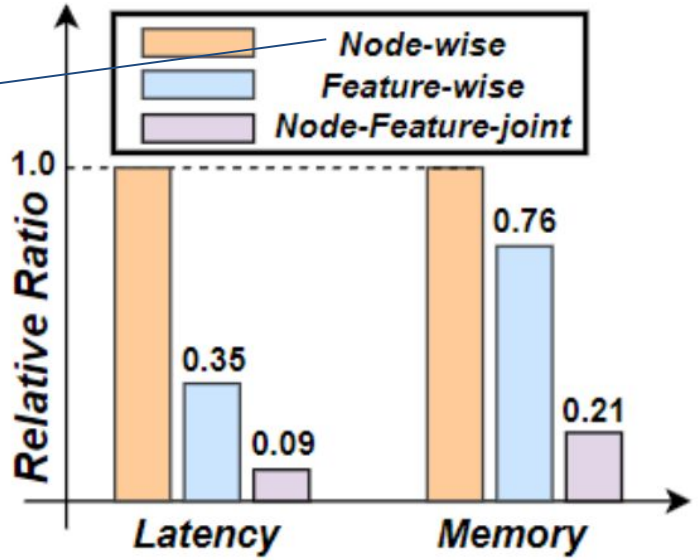
Motivation and Challenge

1. High-latency HE operations
2. Large memory consumption
3. Execution Order

Execution Order

(A · X) optimized

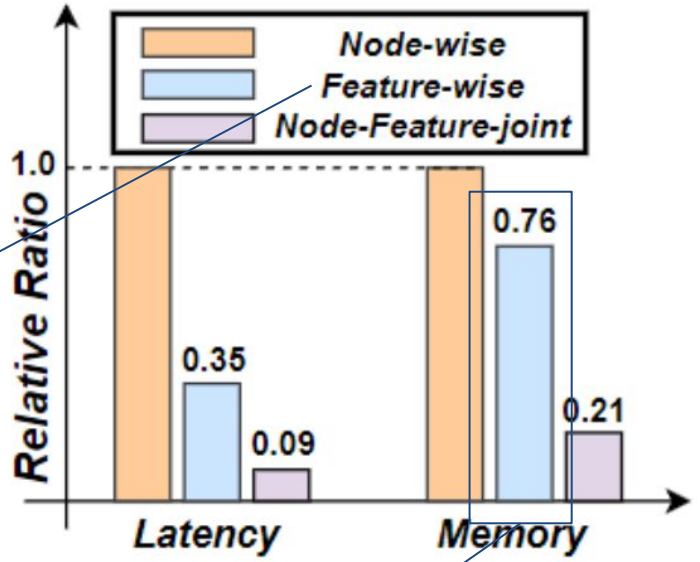
(X· W) optimized



Execution Order

(A · X) optimized

(X· W) optimized

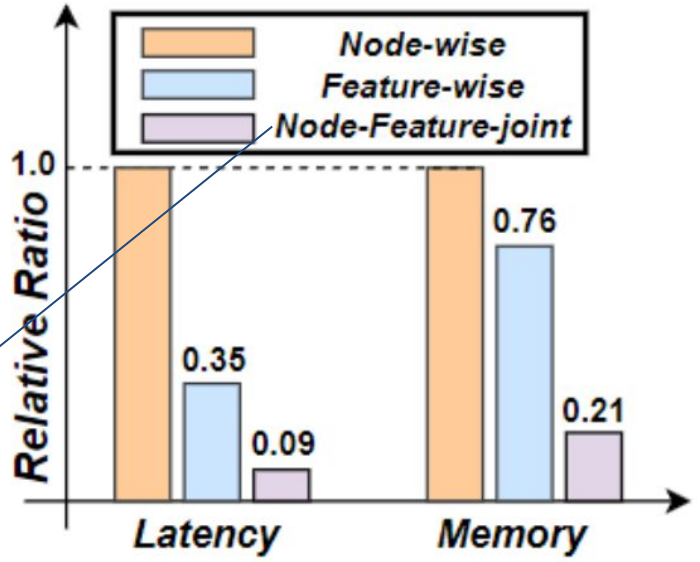


Less memory as feature reduction

Execution Order

$(A \cdot X)$ optimized

$(X \cdot W)$ optimized

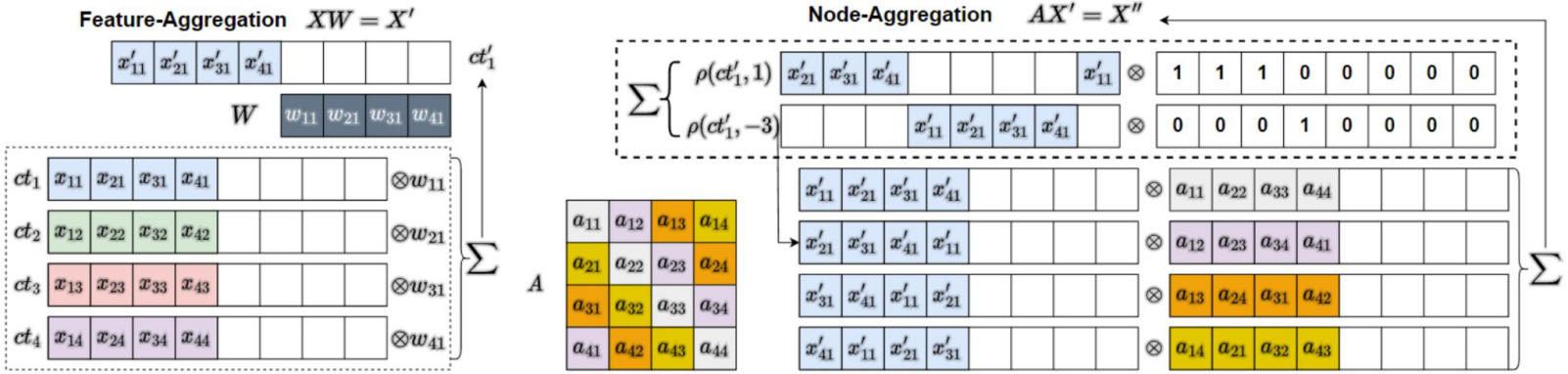


Need a global optimized solution

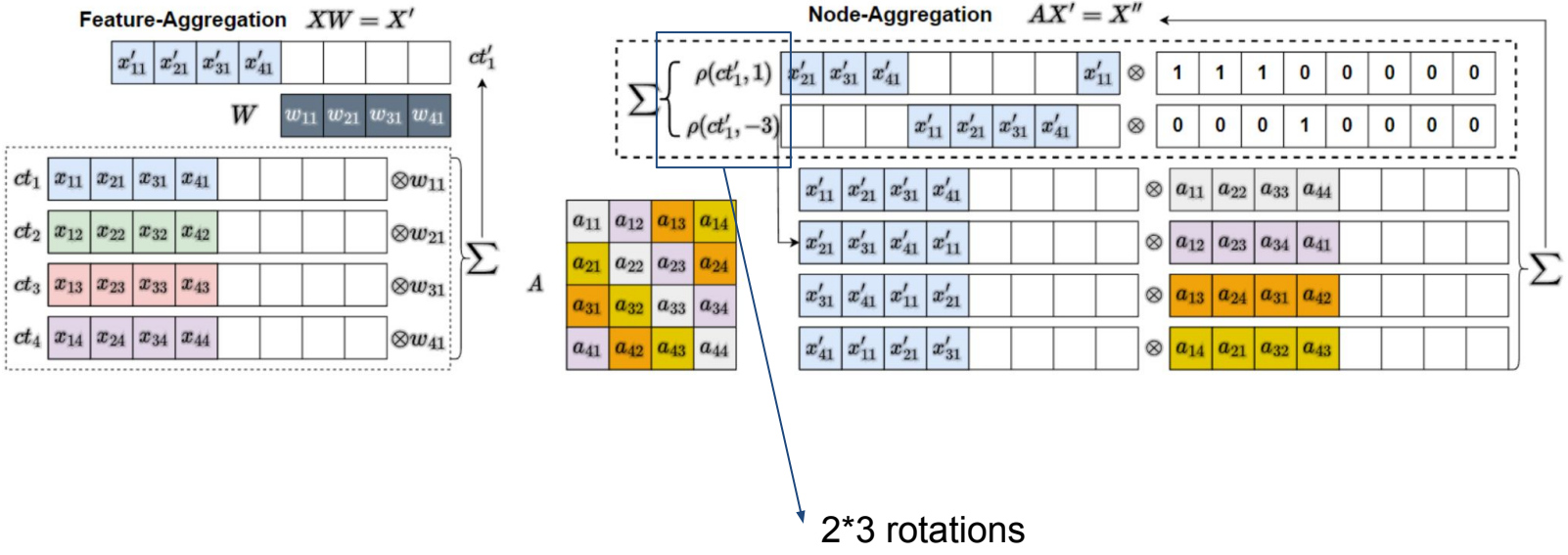
Proposed Techniques

1. Parallel-Packing
2. Interleaved Assembling

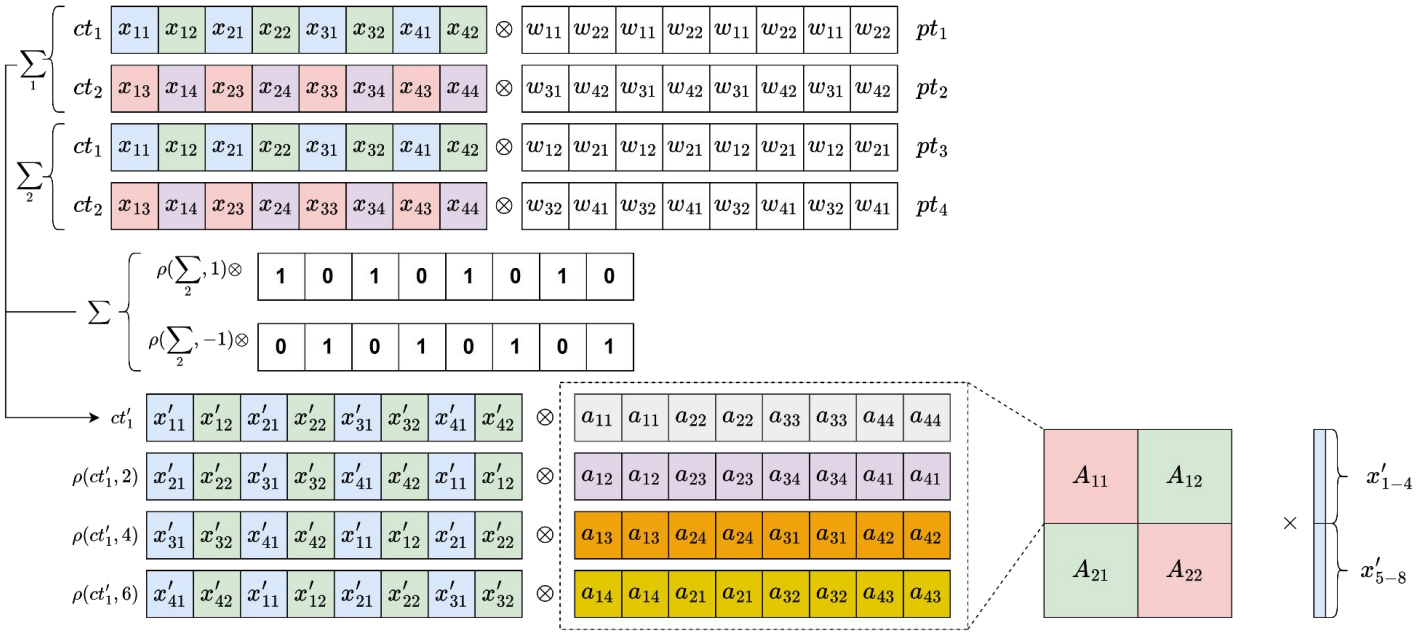
Feature-optimized packing



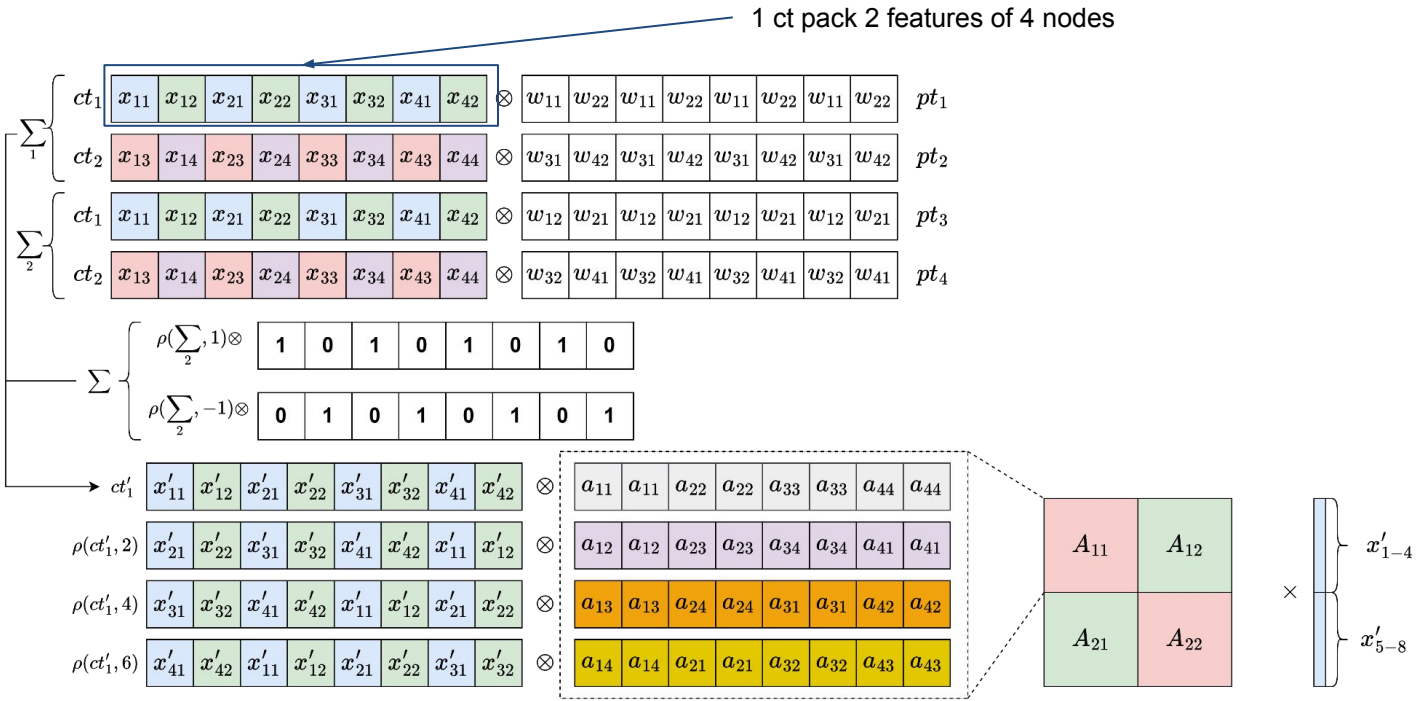
Feature-optimized packing



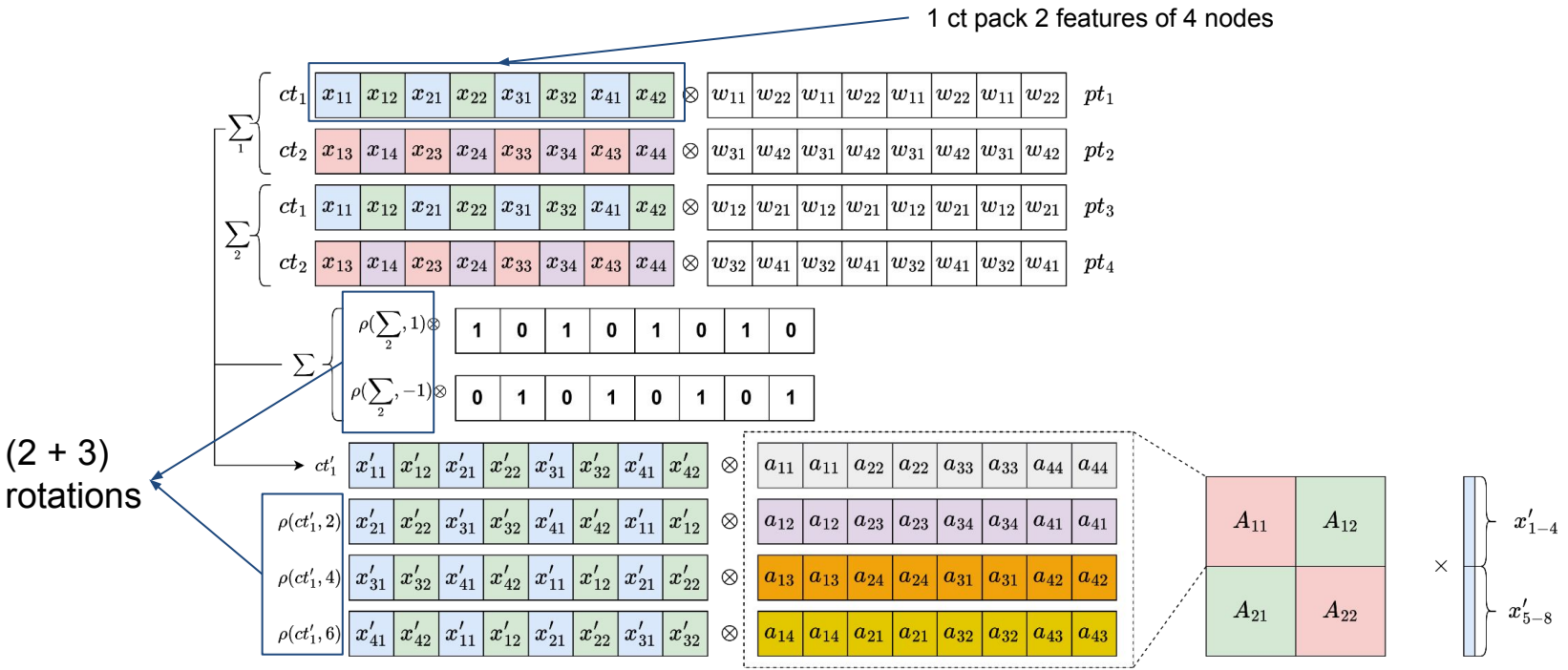
Parallel-Packing



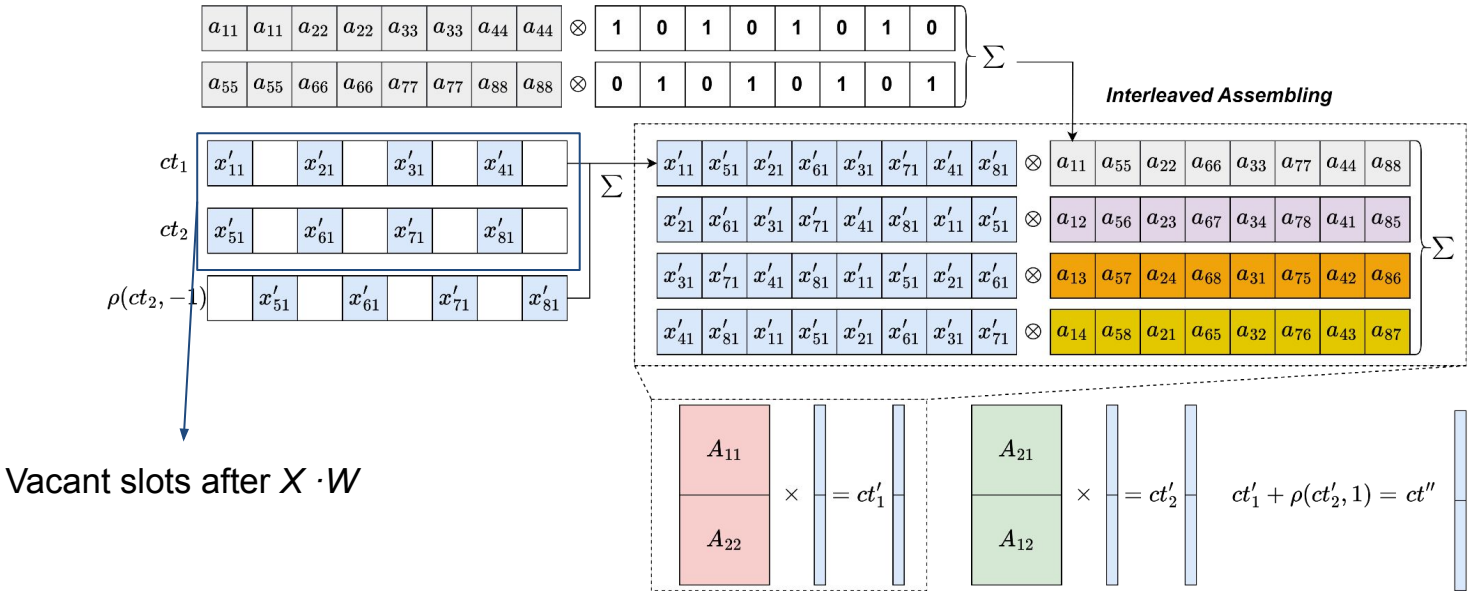
Parallel-Packing



Parallel-Packing



Interleaved Assembling



Experiment

Dataset	Method	Security Level	Latency (s)	Amortized Latency	Speedup (×)
Cora	Gazelle [17]	128-bit	3832.36	1.42	-
	E2DM(64) [15]	98-bit	3150.74	1.16	1.22
	HElayers [1]	128-bit	2102.47	0.78	1.82
	uSCORE(32,128) [13]	98-bit	1727.12	0.64	2.22
	Penguin(32,128)+IA	128-bit	**660.57**	**0.24**	**5.92**
Citeseer	Gazelle [17]	128-bit	4727.94	1.42	-
	E2DM(64) [15]	98-bit	4561.15	1.37	1.04
	HElayers [1]	128-bit	3044.58	0.92	1.54
	uSCORE(32,128) [13]	98-bit	2377.50	0.72	1.97
	Penguin(32,128)+IA	128-bit	**928.10**	**0.28**	**5.07**
Pubmed	Gazelle [17]	128-bit	158655.54	8.05	-
	E2DM(64) [15]	98-bit	154530.49	7.84	1.03
	HElayers [1]	128-bit	103283.56	5.24	1.54
	uSCORE(32,128) [13]	98-bit	78843.49	4.00	2.01
	Penguin(32,128)+IA	128-bit	**30522.43**	**1.55**	**5.19**

Conclusion

1. In this work, we propose a two-dimension parallel packing technique with a interleaved assembling technique to speed up the HE-GCN inference.

2. These techniques can better save ciphertext memory and effectively reduce the number of homomorphic operations required.

3. Experimental results based on the GAEs for link prediction task have shown roughly 5x speedup to previous SOTAs.

Thanks!. Q & A