LinGCN: Structural Linearized Graph Convolutional Network for Homomorphically Encrypted Inference

Hongwu Peng[1] · Ran Ran[2] · Yukui Luo[3] · Jiahui Zhao[1] · Shaoyi Huang[1] · Kiran Thorat[1] · Tong Geng[4] · Chenghong Wang[5] · Xiaolin Xu[6] · Wujie Wen[2] · Caiwen Ding[1]

[1]University of Connecticut, [2]North Carolina State University,

[3]University of Massachusetts Dartmouth, [4]University of Rochester,

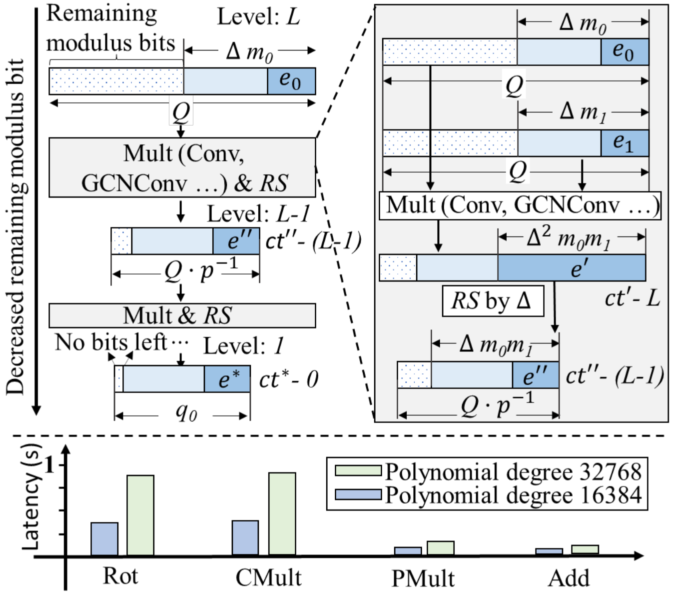
[5]Indiana University Bloomington, [6]Northeastern University

Background

- Machine-Learning-As-A-Service (MLaaS) faces security challenges.
- Secure private inference (PI): multi-party computation (MPC) and homomorphic encryption (HE).
- HE requires much less communication cost compared to MPC, but still faces computational overhead.

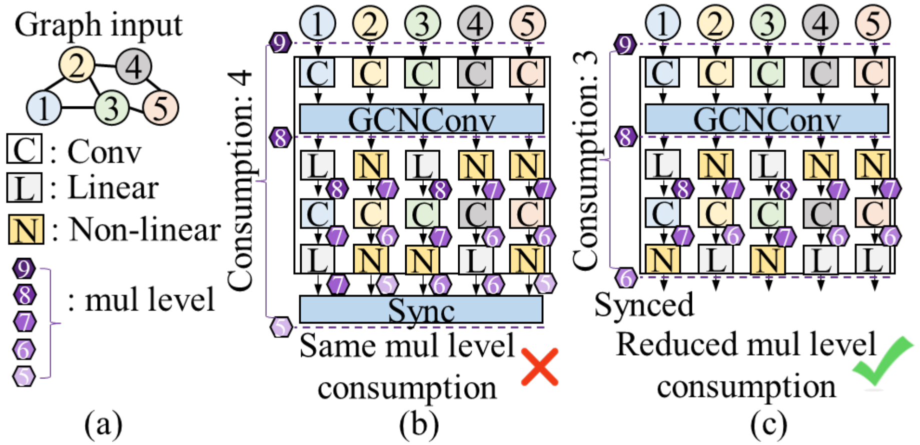
Motivation

Observation 1: Conserving Levels in CKKS



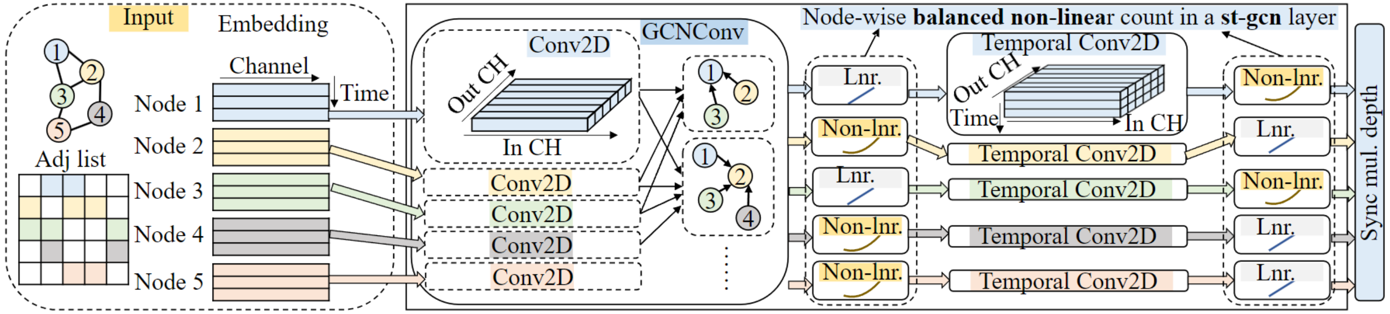
Top: Rescale decreases the ciphertext level. Bottom: Higher polynomial degree leads to longer HE operator's latency.

Observation 2: Structural/synchronized linearization matters!

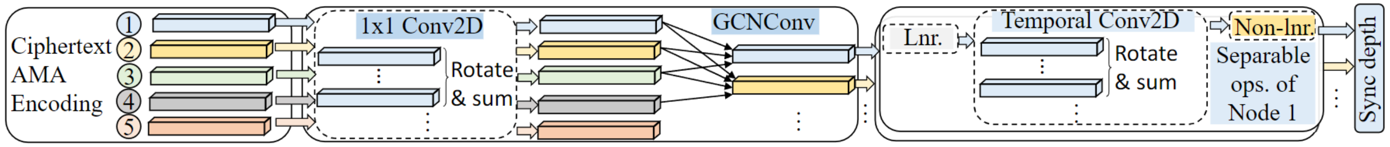


Unstructured vs. structural linearization. Unstructured one doesn't lead to effective level reduction.

Structural Linearized GCN



(a) Single st-gcn layer example with structure pruned non-linear operator



(b) Single st-gcn layer computation for homomorphically encrypted inference

Structural linearization:

$$\underset{W,h}{\operatorname{argmin}}\mathcal{L} = \underset{W,h}{\operatorname{argmin}} \;\; \mathcal{L}_{acc}(f_W(X_0), Y) + \mu \cdot \sum_{i=1}^{2L} ||h_i||_0 \quad (2)$$

subject to $\forall j,k \in [1,V],\ (h_{2i,j} + h_{2i+1,j}) = (h_{2i,k} + h_{2i+1,k})$

$$\frac{\partial \mathcal{L}}{\partial h_{w(i,k)}} = \frac{\partial \mathcal{L}_{acc}}{\partial X_{i,k}}(\sigma_n(Z_{i-1}) - Z_{i-1})\frac{\partial h_{i,k}}{\partial h_{w(i,k)}} + \mu \frac{\partial h_{i,k}}{\partial h_{w(i,k)}},$$

$$\frac{\partial h_{i,k}}{\partial h_{w(i,k)}} = Softplus(h_{w(i,k)}) \quad (3)$$

Algorithm 1 Structural Polarization.

Input: h_w: auxiliary parameter
Output: h: final indicator
for $i = 0$ to L **do**
  $s_h, s_l = 0$ and $ind_h, ind_l \leftarrow \emptyset$
  for $j = 1$ to V **do**
    if $h_{w(2i,j)} > h_{w(2i+1,j)}$ **then**
      $s_h \mathrel{+}= h_{w(2i,j)}$, $s_l \mathrel{+}= h_{w(2i+1,j)}$
      $ind_h \leftarrow (2i, j)$, $ind_l \leftarrow (2i+1, j)$
    else
      $s_h \mathrel{+}= h_{w(2i+1,j)}$, $s_l \mathrel{+}= h_{w(2i,j)}$
      $ind_h \leftarrow (2i+1, j)$, $ind_l \leftarrow (2i, j)$
    end if
  end for
  $h_{ind_h} = s_h > 0$ and $h_{ind_l} = s_l > 0$
end for

Polynomial replacement & overall workflow

$$\sigma_n(x) = c \cdot w_2 x^2 + w_1 x + b \quad (4)$$

$$\mathcal{L}_p = (1-\eta)\mathcal{L}_{CE}(f_{W,s}(X_0), Y) + \eta \mathcal{L}_{KL}(f_{W,s}(X_0), f_{W,t}(X_0)) + \frac{\varphi}{2}\sum_{i=1}^{L} MSE\left(\frac{X_{i,s}}{||X_{i,s}||_2}, \frac{X_{i,t}}{||X_{i,t}||_2}\right) \quad (5)$$

Algorithm 2 LinGCN Framework Workflow.

Input: Pretrain ReLU-based model M_T, linearization penalty μ and optim. OP_L, polynomial replacement optim. OP_P
Output: Level-reduced polynomial model
Copy M_S from M_T
Initialize h_w for M_S
for Structural linearization iterations **do**
  Calculate $\mathcal{L}$ via Eq. 2 and Algorithm 1
  Update W and h_w through back propagation (Eq. 3) by minimizing $\mathcal{L}$ using OP_L
end for
Freeze h_w and h
Replace ReLU in M_s with polynomial
Initialize w_{poly}
for Polynomial replacement iterations **do**
  Calculate $\mathcal{L}_p$ via Eq. 5
  Update W through back propagation by minimizing $\mathcal{L}_p$ using OP_P
end for

Experiment Result

LinGCN is evaluated on the NTU-RGB+D dataset, and excels in the following aspects:

- Reduced multiplication depth: lower encryption level, lower latency
- Minimal accuracy loss
- 10% accuracy improvement over CryptoGCN.

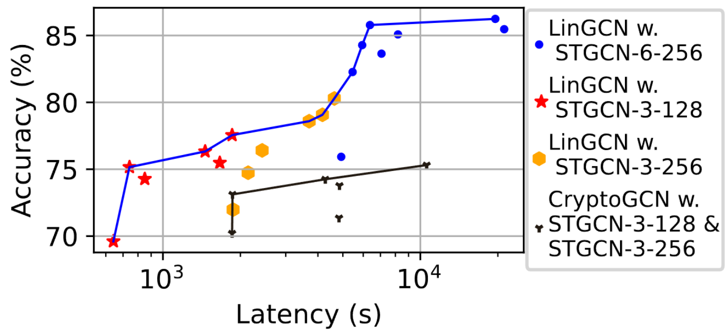


Figure 1: Frontier of LinGCN vs. CryptoGCN [12]

Table 2: STGCN-3-128 comparison

Model	STGCN-3-128		
Methods	Non-linear layers	Test acc (%)	Latency (s)
LinGCN	6	77.55	1856.95
LinGCN	5	75.48	1663.13
LinGCN	4	76.33	1458.95
LinGCN	3	74.27	850.22
LinGCN	**2**	**75.16**	**741.55**
LinGCN	1	69.61	642.06
CryptoGCN	6	74.25	4273.89
CryptoGCN	5	73.12	1863.95
CryptoGCN	4	70.21	1856.36

Table 3: STGCN-3-256 comparison

Model	STGCN-3-256		
Methods	Non-linear layers	Test acc (%)	Latency (s)
LinGCN	6	80.29	4632.05
LinGCN	5	79.07	4166.12
LinGCN	**4**	**78.59**	**3699.49**
LinGCN	3	76.41	2428.88
LinGCN	2	74.74	2143.46
LinGCN	1	71.98	1873.40
CryptoGCN	6	75.31	10580.41
CryptoGCN	5	73.78	4850.93
CryptoGCN	4	71.36	4831.93

Table 4: LinGCN for STGCN-6-256 model.

Model	STGCN-6-256		
Methods	Non-linear layers	Test acc (%)	Latency (s)
LinGCN	12	85.47	21171.80
LinGCN	11	86.24	19553.96
LinGCN	7	85.08	8186.35
LinGCN	5	83.64	7063.51
LinGCN	**4**	**85.78**	**6371.39**
LinGCN	3	84.28	5944.81
LinGCN	2	82.27	5456.12
LinGCN	1	75.93	4927.26