Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Computer Engineering Project

---oOo---

Acceleration of Graph Attention Network on FPGA

Council: CE - CC05
Supervisor: Assoc. Prof. Pham Quoc Cuong
Member Secretary: Mr. Nguyen Thanh Loc

Authors:

- Hoang Tien Duc - 2152520
- Dang Hoang Gia - 2153312
- Nguyen Duc Bao Huy - 2152089

Outline

01 Introduction

Graph Neural Network

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

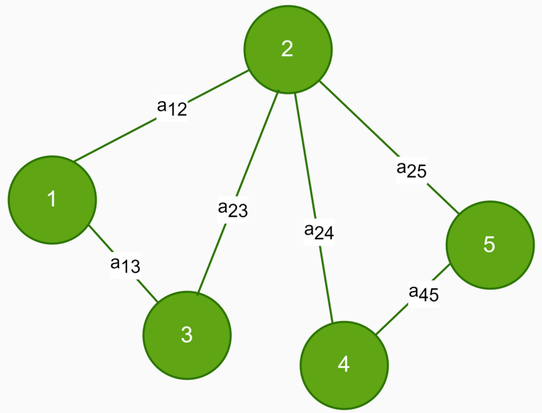
GCN:

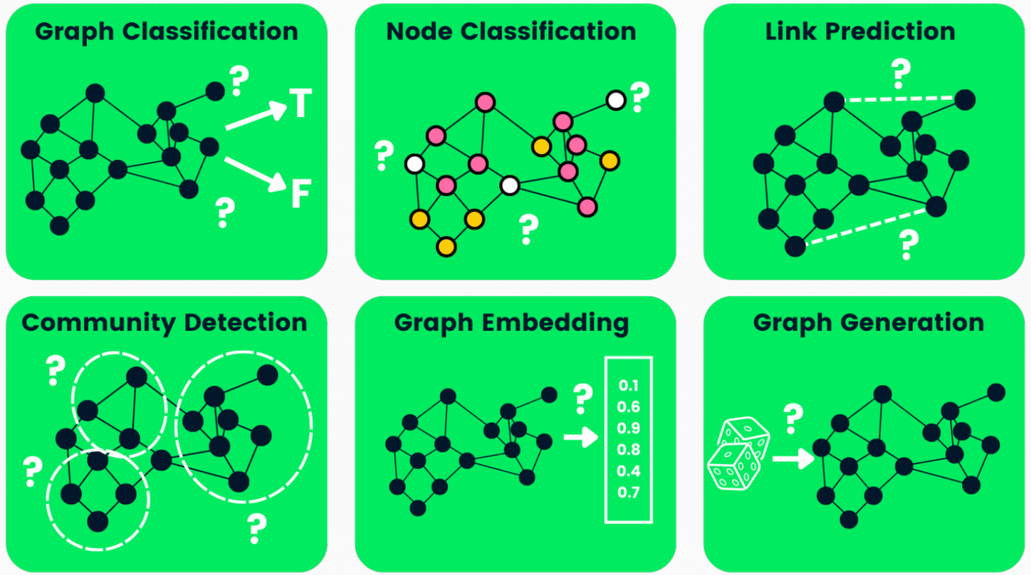
- Used in tasks like **node classification** and **link prediction**
- But **poor performance** on dynamic and inductive tasks.

GAT:

- Improves flexibility with an **attention mechanism**, focusing on **important node relationships**,
- Make it better for complex, dynamic, and real-time tasks.

==> **FPGA-based accelerators for GAT are needed to accelerate the computation of GAT operations.**

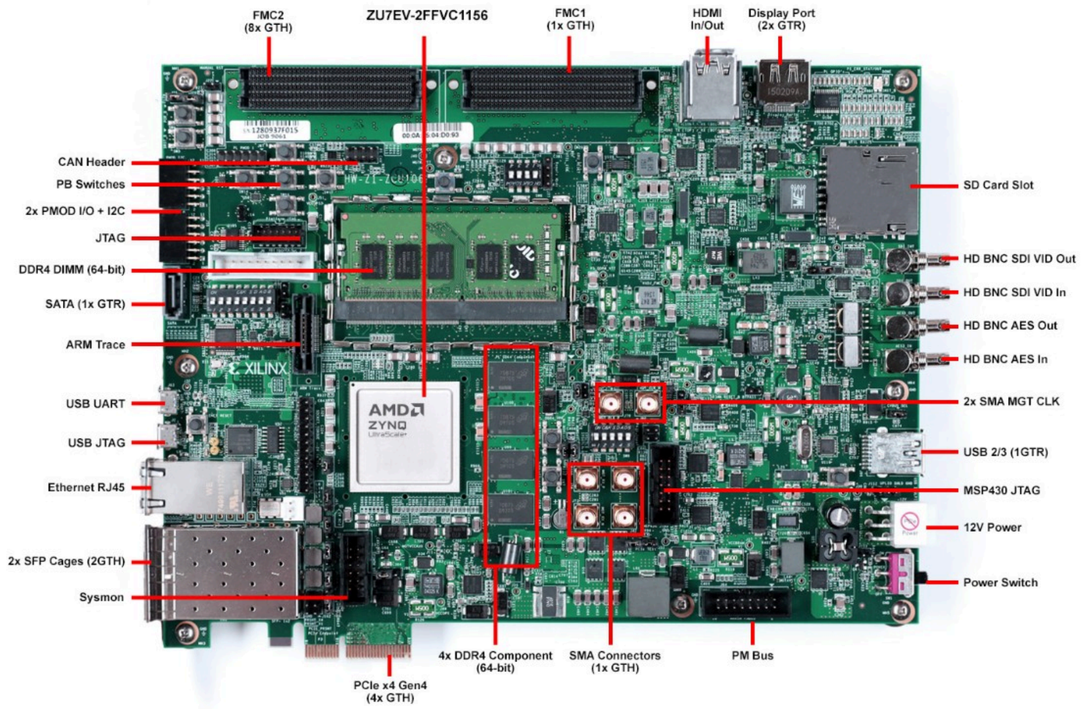




02 Background

ZCU106 Zynq SoC Ultrascale+

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024



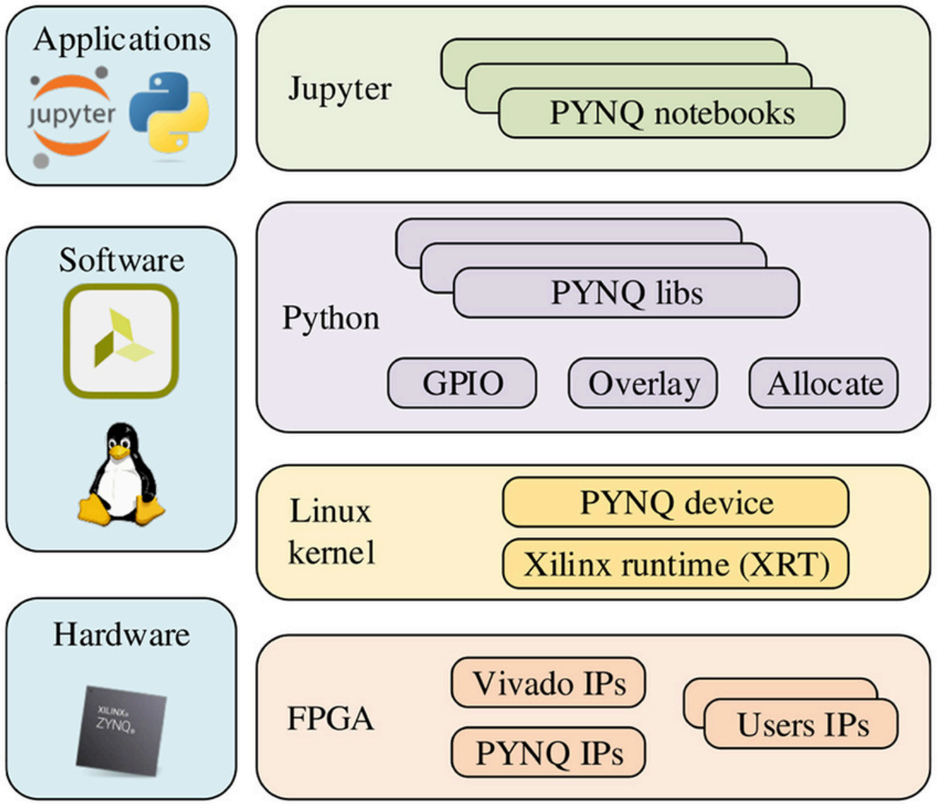
Zynq UltraScale+ MPSoC ZCU106

Resource	Information
Core Device	XCZU7EV
Available IOBs	260
LUT Elements	230,400
Flip-Flops	460,800
Block RAMs	312
Ultra RAMs	96
DSPs	1728

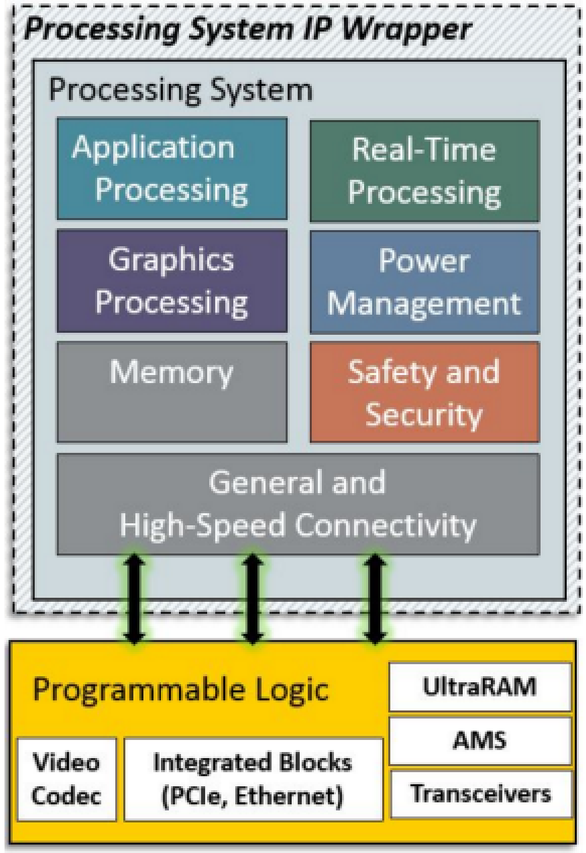
Resource on XCZU7EV

PYNQ Processing System

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024



Pynq Framework



Xilinx PS-PL Connectivity

Graph Attention Network

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

GAT Algorithms:

Linear transformation: $$\vec{z_{ij}} = W\vec{h}_{ij}$$

Attention coefficient: $$e_{ij} = a(W\vec{h_i}, W\vec{h_j})$$

Softmax: $$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k\in\mathcal{N}_i}\exp(\text{LeakyReLU}(e_{ik}))}$$

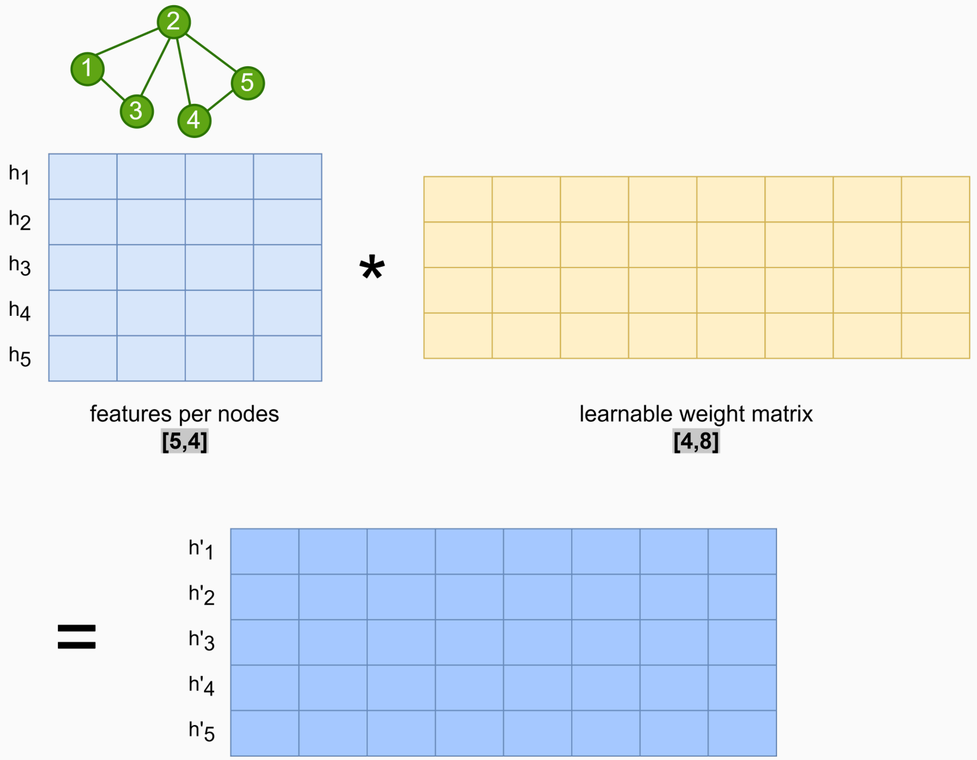
Aggregation: $$\vec{\mathbf{h}'_i} = \sigma\left(\sum_{j\in\mathcal{N}(i)}\alpha_{ij}W\vec{\mathbf{h}_j}\right)$$

Graph Attention Network

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Weight and feature linear transformation

$$\vec{z_{ij}} = W\vec{h}_{ij}$$

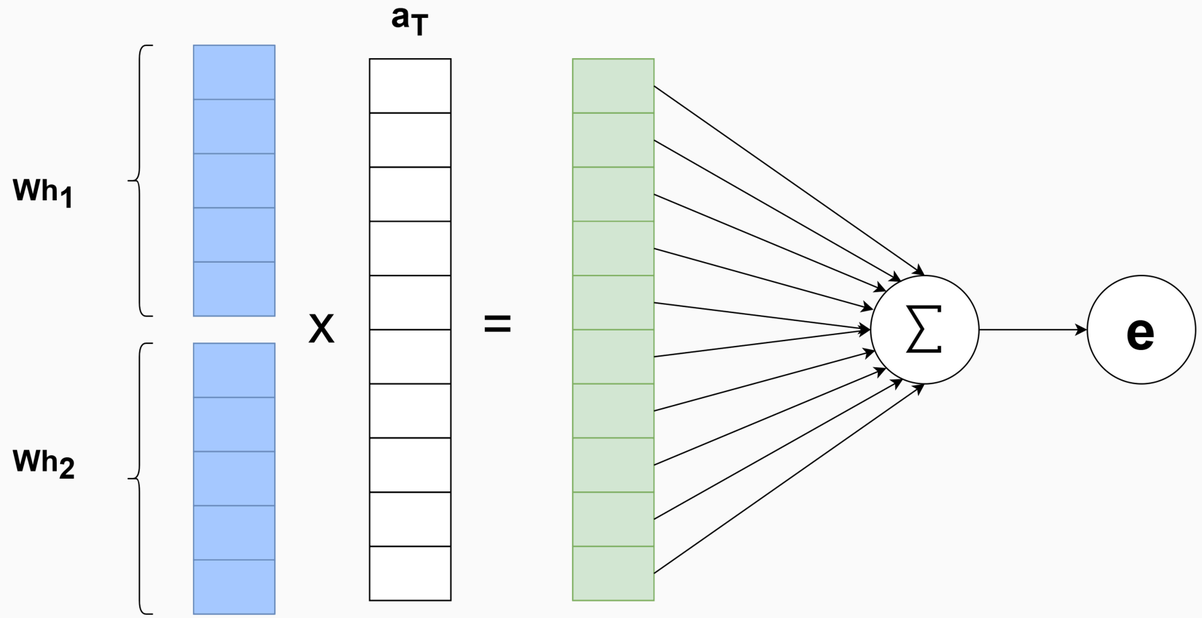


Graph Attention Network

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Attention coefficient

$$e_{ij} = a(W\vec{h_i}, W\vec{h_j})$$

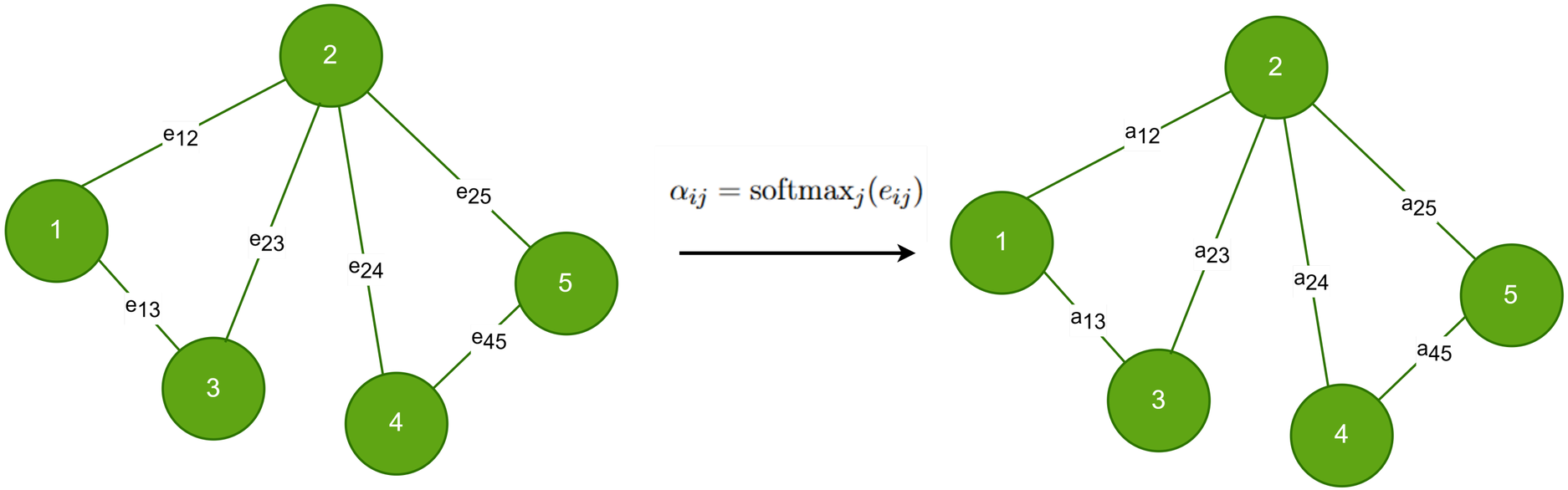


Graph Attention Network

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Softmax

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(e_{ik}))}$$

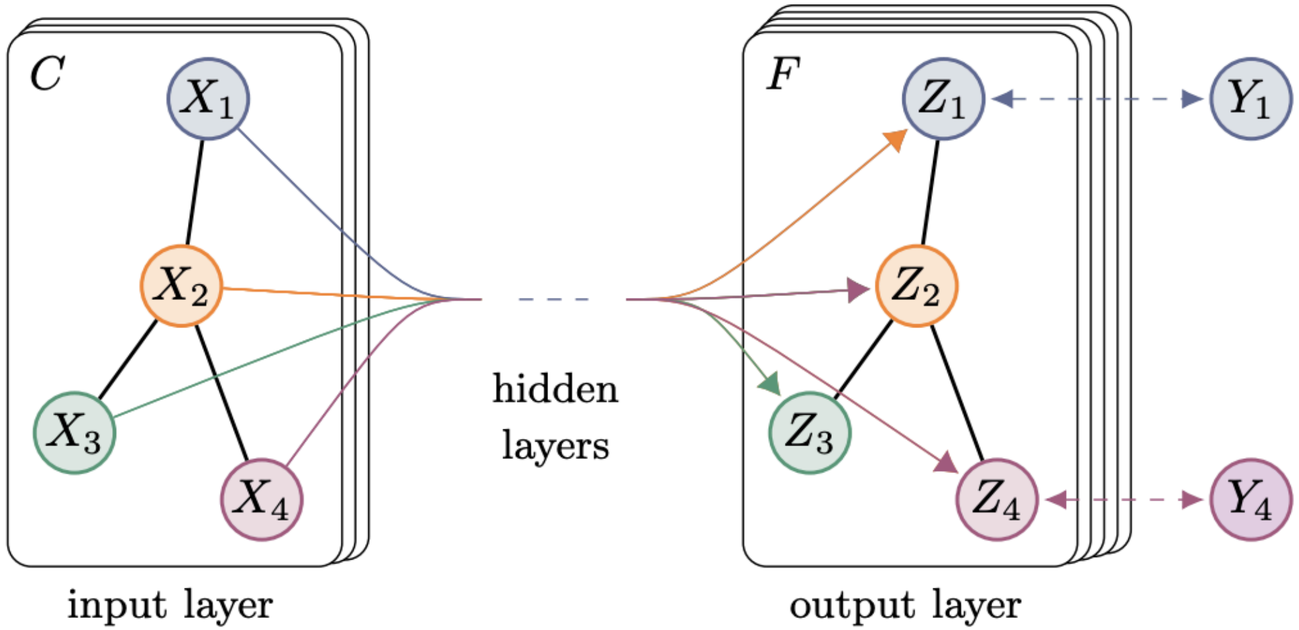


Graph Attention Network

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Aggregation

$$\vec{\mathbf{h}}_i' = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W \vec{\mathbf{h}}_j\right)$$

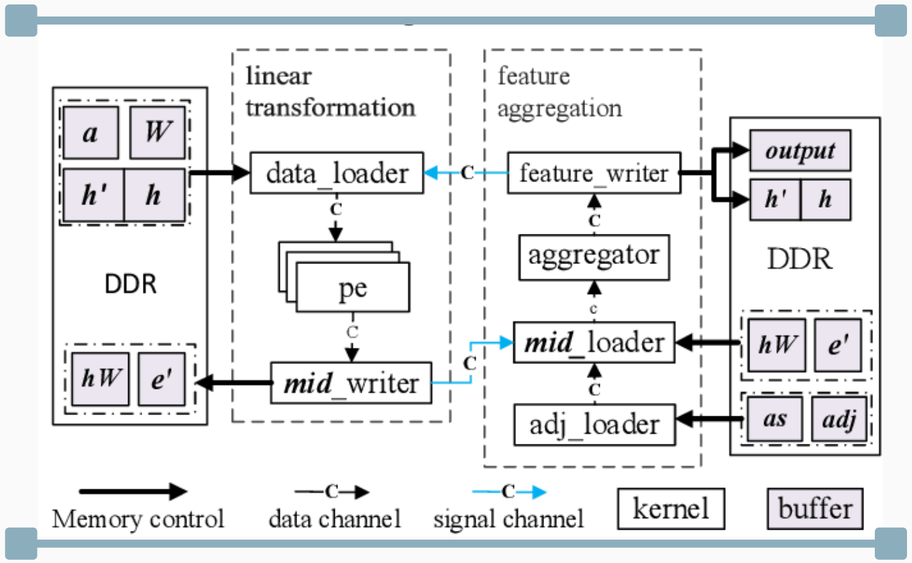


Related Work

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

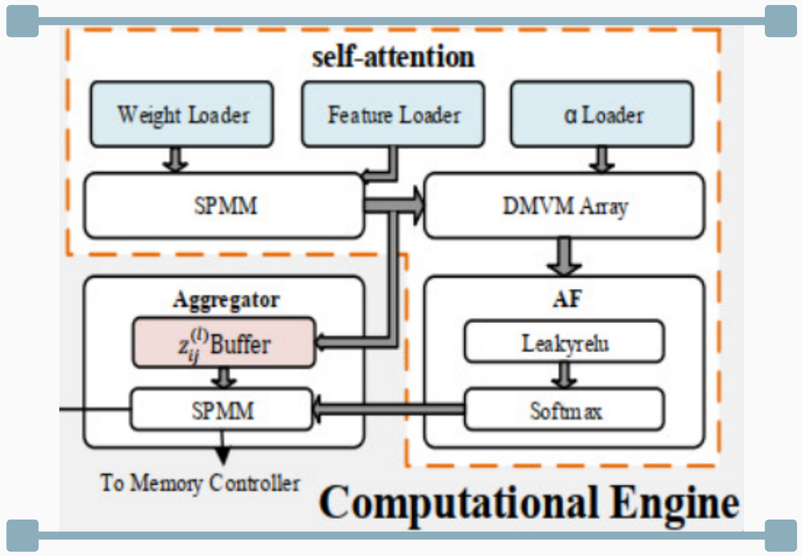
S-GAT: (W. Yan, W. Tong and X. Zhi)

- Model compression, feature quantization, shift-based multiplication.
- Performance: 593x CPU, 7.34x Nvdia Tesla V100



SH-GAT: (Wang. et.al)

- Split weights, softmax approximation, load-balanced SPMM, pre-fetching to address memory irregularities.
- Performance: 3283x CPU. 13x GPU



03 Methodological Approach

Algorithm Optimization

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Original GAT Calculation

$$\vec{z_{ij}} = W\vec{h}_{ij}$$

$$e_{ij} = \text{LeakyReLU}\left(\vec{a}^{\top}[\vec{z}_i \| \vec{z}_j]\right)$$

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(e_{ik}\right)}$$

$$\vec{h}_i = \sigma\left(\sum_{j\in\mathcal{N}(i)}\alpha_{ij}\vec{z}_j\right)$$

Proposed GAT Calculation

$$\vec{z_{ij}} = W\vec{h}_{ij}$$

$$e_{ij} = \text{ReLU}\left(\vec{a}_1^{\top}\vec{z}_i + \vec{a}_2^{\top}\vec{z}_j\right)$$

$$\alpha_{ij} = \frac{2^{e_{ij}}}{\sum_{k\in\mathcal{N}(i)}2^{e_{ik}}}$$

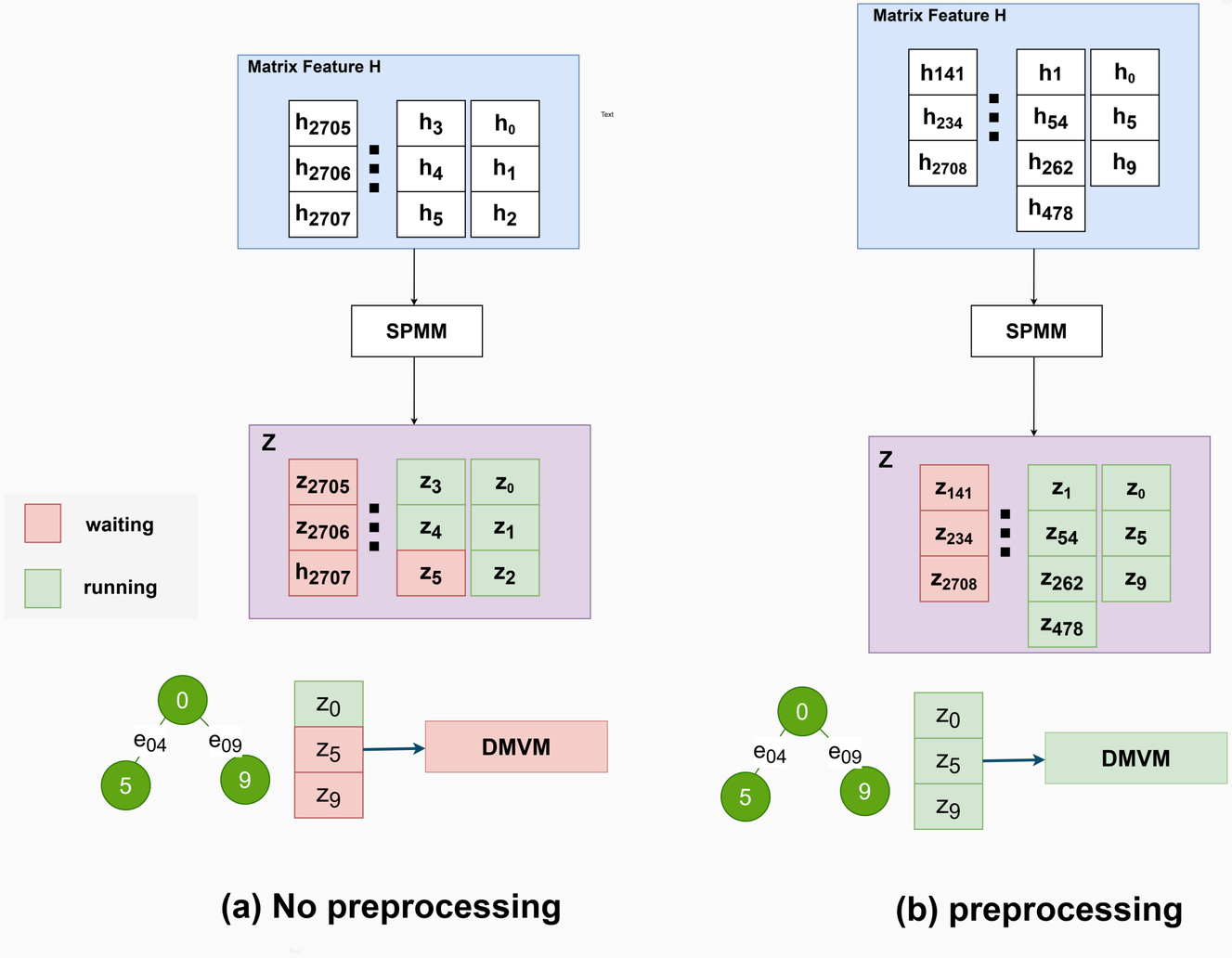
$$\vec{h}_i = \text{ReLU}\left(\sum_{j\in\mathcal{N}(i)}\alpha_{ij}\vec{z}_j\right)$$

Data pre-processing

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Pipeline stalls occur due to **unorganized data** flow through computational modules.

==> Group **source node** and **neighboring nodes** into a **subgraph** to streamline data flow and prevent stalls.

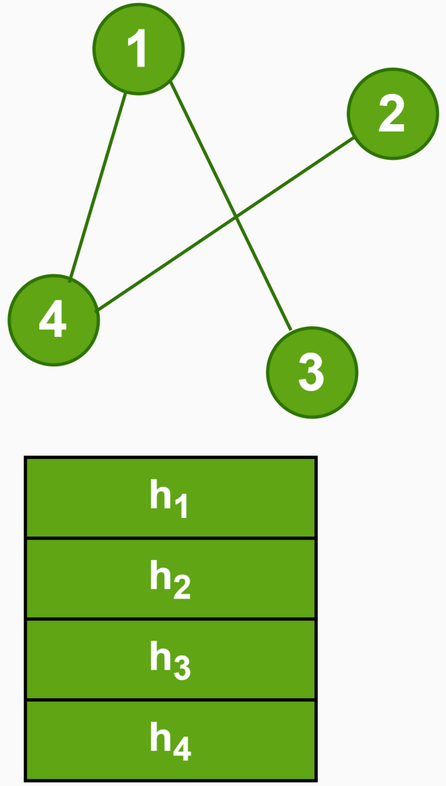


Data pre-processing

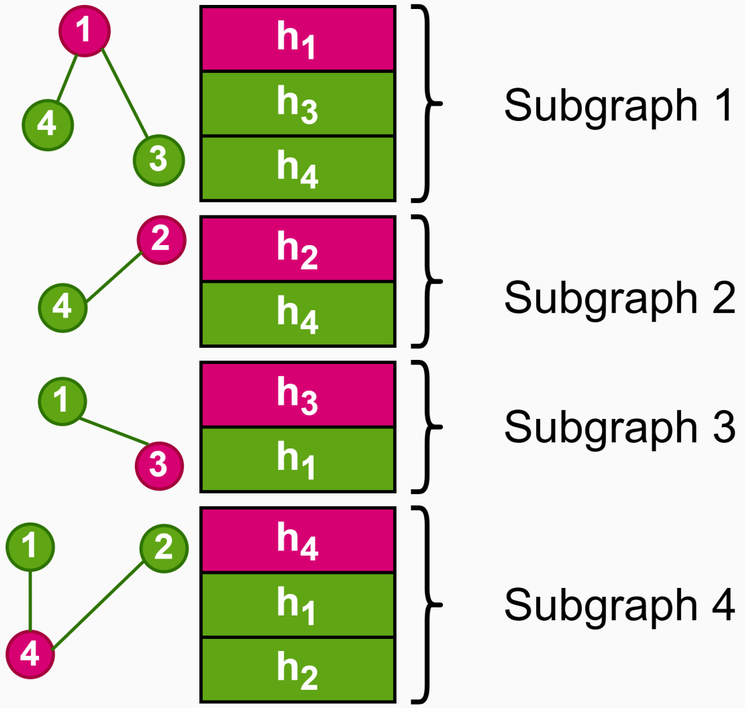
Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

A 4-node graph can represent as 4 subgraphs.

Source node: header of each subgraph.
Neighbor node: node adjacent to source node.



Original Graph and H matrix



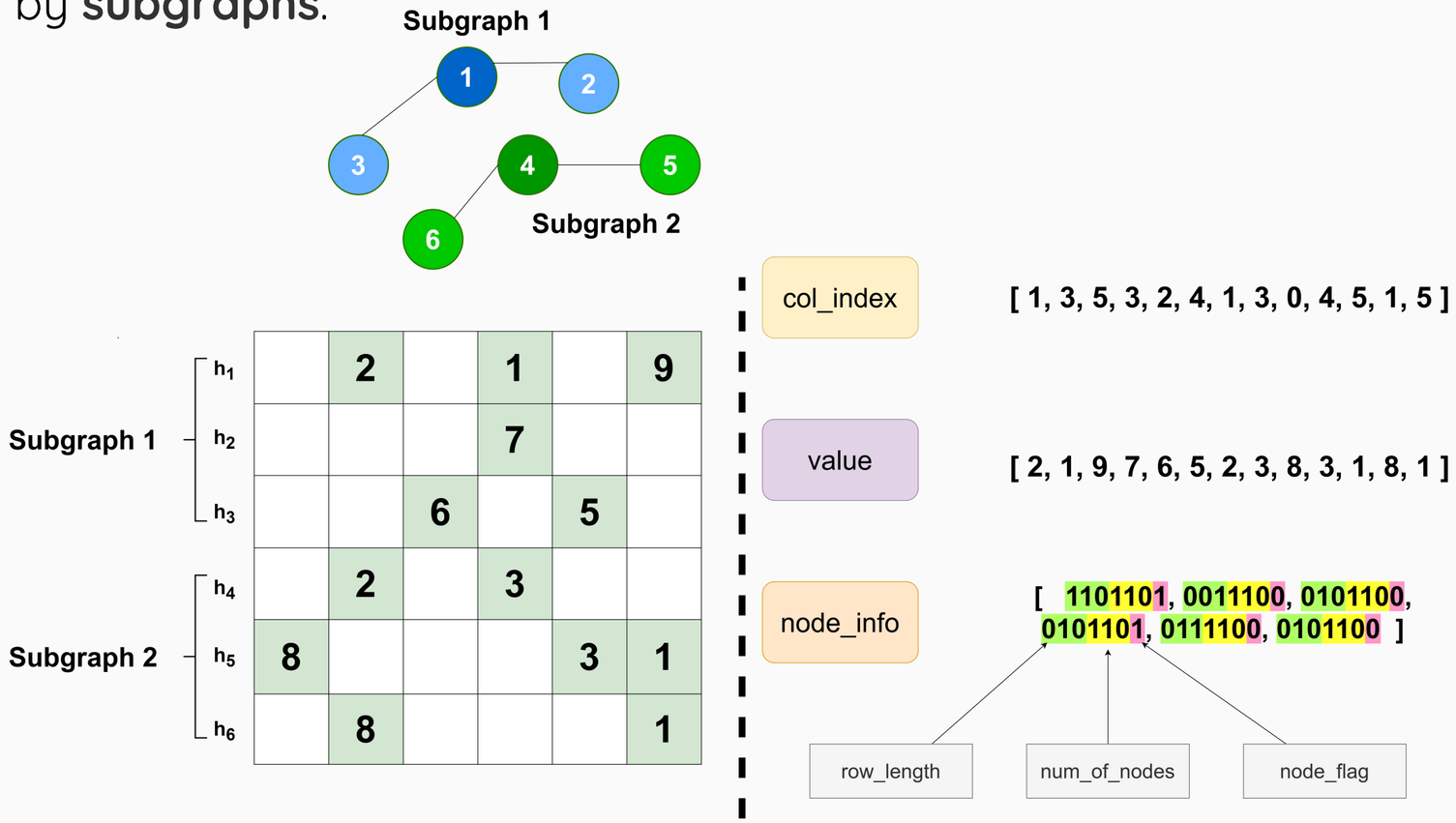
Proposed subgraphs

GCSR Format

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

GCSR stands for **Graph Compressed Sparse Row.**

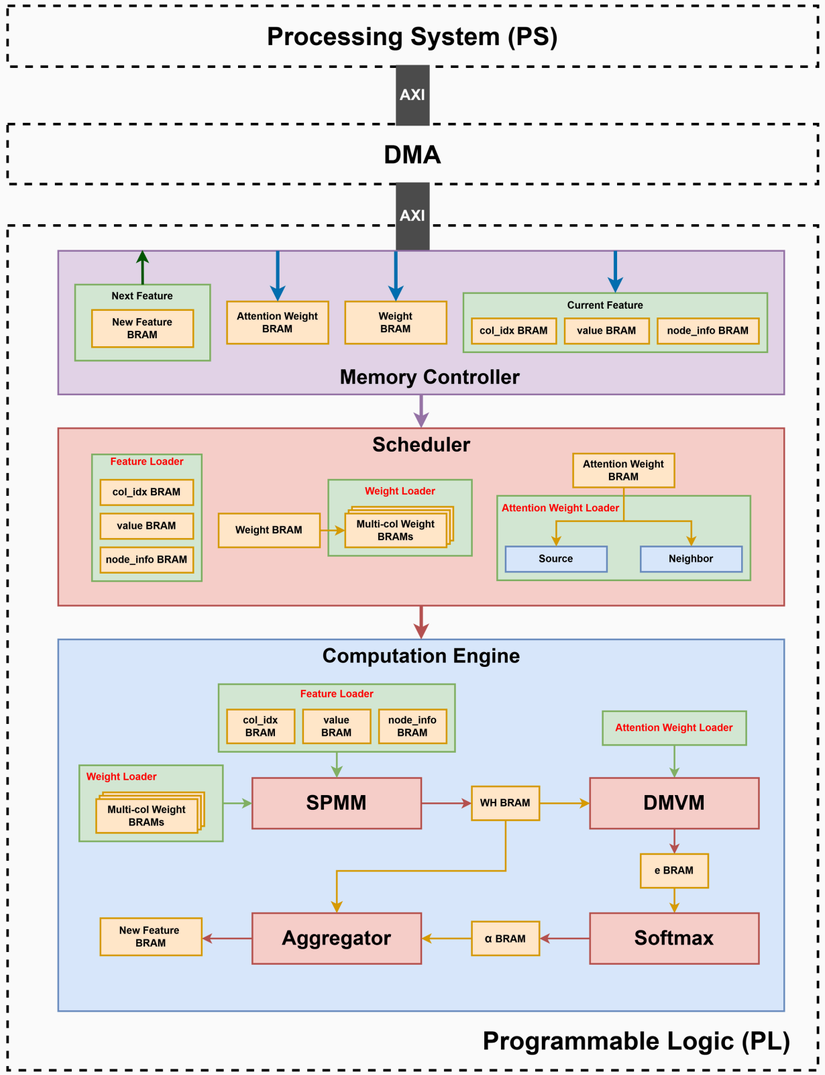
- Collect information of **index and value,** compressed it into 3 components.
- Organized by **subgraphs**.



04 Overall Architecture

General Architecture

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024



PS is responsible for **running software application** (training GAT model and transferring input data).

PL is responsible for **operating GAT algorithm.**

- Memory Controller (**6** BRAMs).
- Scheduler (**3** Loaders).
- SPMM (Sparse Matrix Multiplication).
- DMVM (Dense Matrix Vector Multiplication).
- Softmax.
- Aggregator.

SPMM/SP-PE

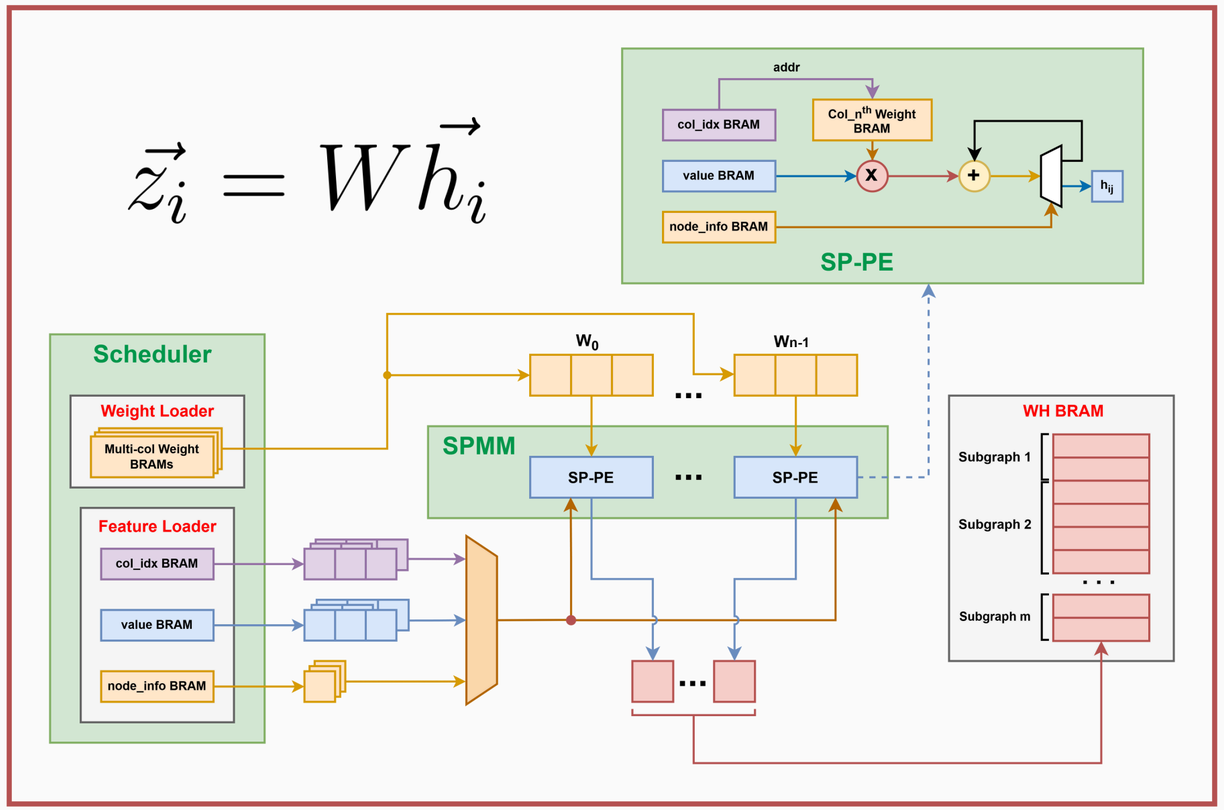
(Sparse Matrix Vector Multiplication/Sparse Processing Element)

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Each **SP-PE** contains a **column-specific Weight BRAM**.

col_idx is used as **read address** to retrieve the corresponding **weight value** from BRAM.

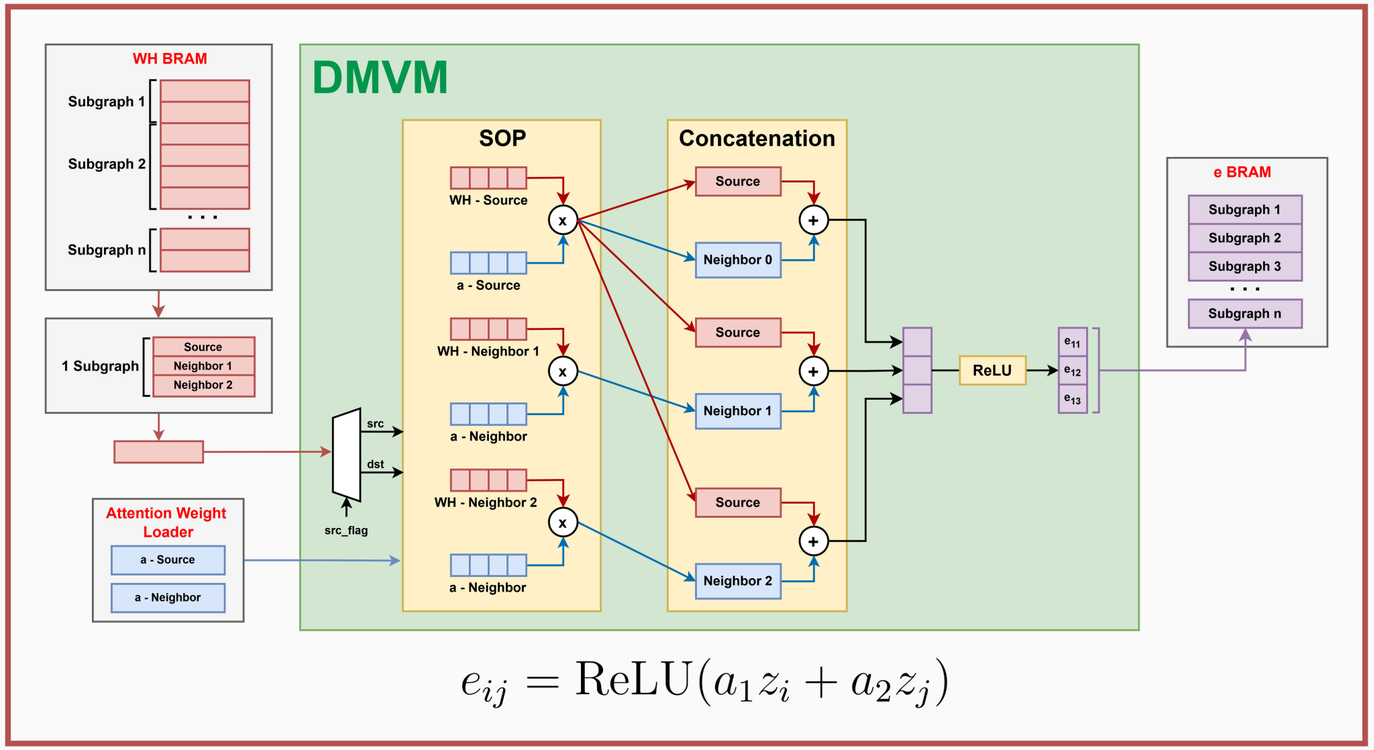
When the number of non-zero feature values reaches **row_length**, **pe_ready_o** will be asserted.



DMVM

(Dense Matrix Vector Multiplication)

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024



$$e_{ij} = \mathrm{ReLU}(a_1 z_i + a_2 z_j)$$

Each **WH entry** is multiplied with either **a_source** or **a_neighbor** based on **src_flag**.

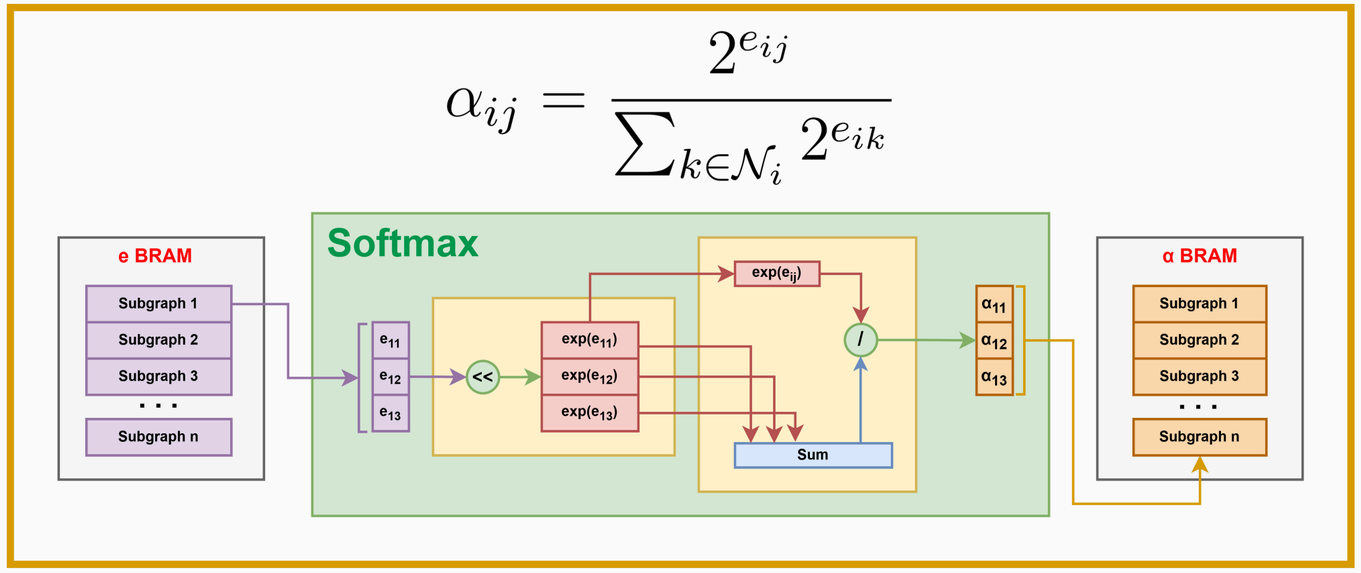
The result is then concatenated with the **source node's product**.

Finally, it passes through the **ReLU** function to retrieve **Attention Cofficient e**.

Softmax

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

$$\alpha_{ij} = \frac{2^{e_{ij}}}{\sum_{k \in \mathcal{N}_i} 2^{e_{ik}}}$$



Each entry from **e BRAM** contains **attention coefficients** of a **subgraph**.

After calculating the **base-2 exponentials** for each coefficient, their **sum** is computed sequentially.

Finally, each base-2 exponential coefficient is **divided** by the sum to obtain the **normalized coefficient**.

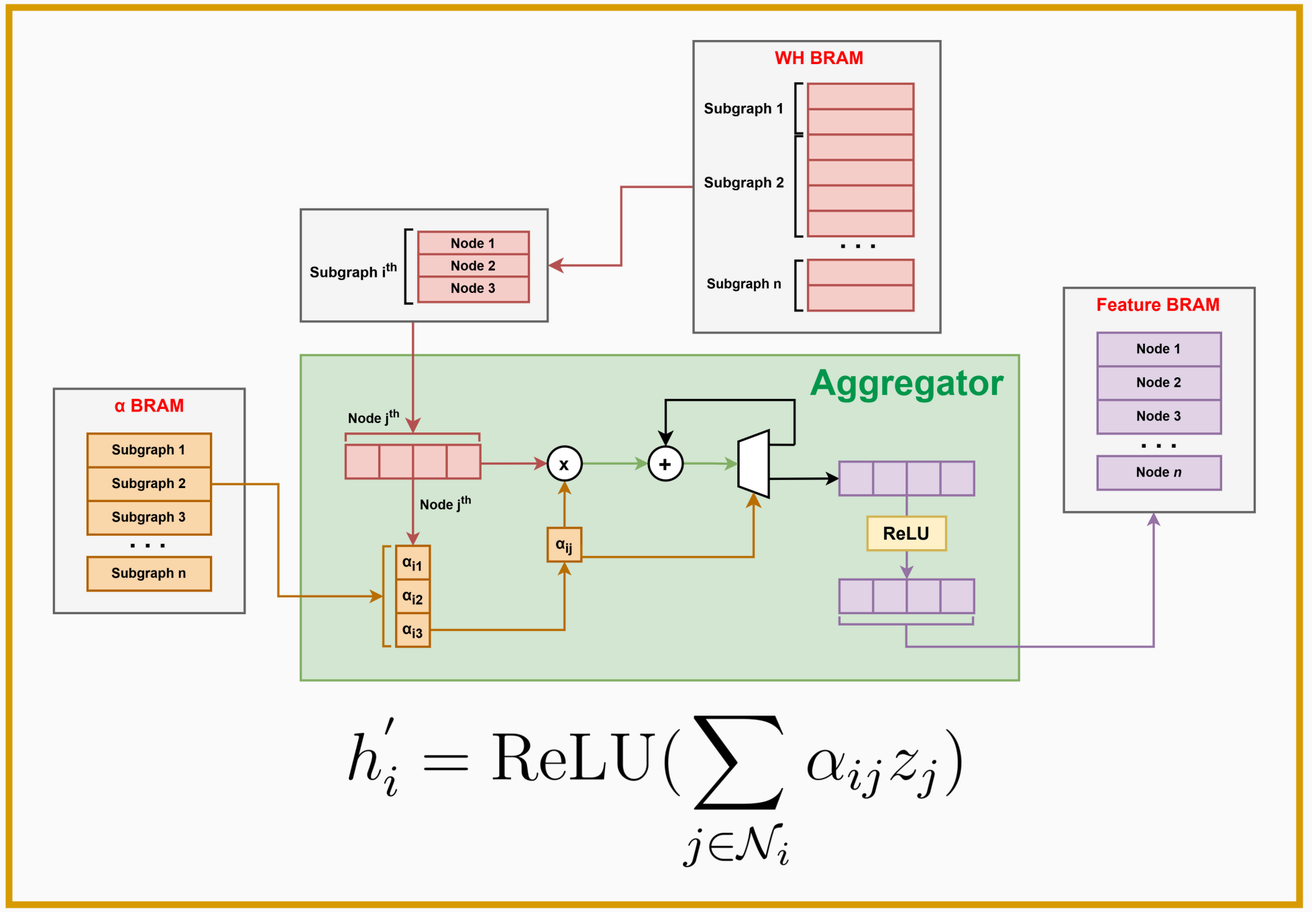
Aggregator

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Each vector from **WH BRAM** is **multiplied** with corresponding normalized coefficient from **α BRAM**.

The results are **accumulated** until all nodes in the subgraph are processed.

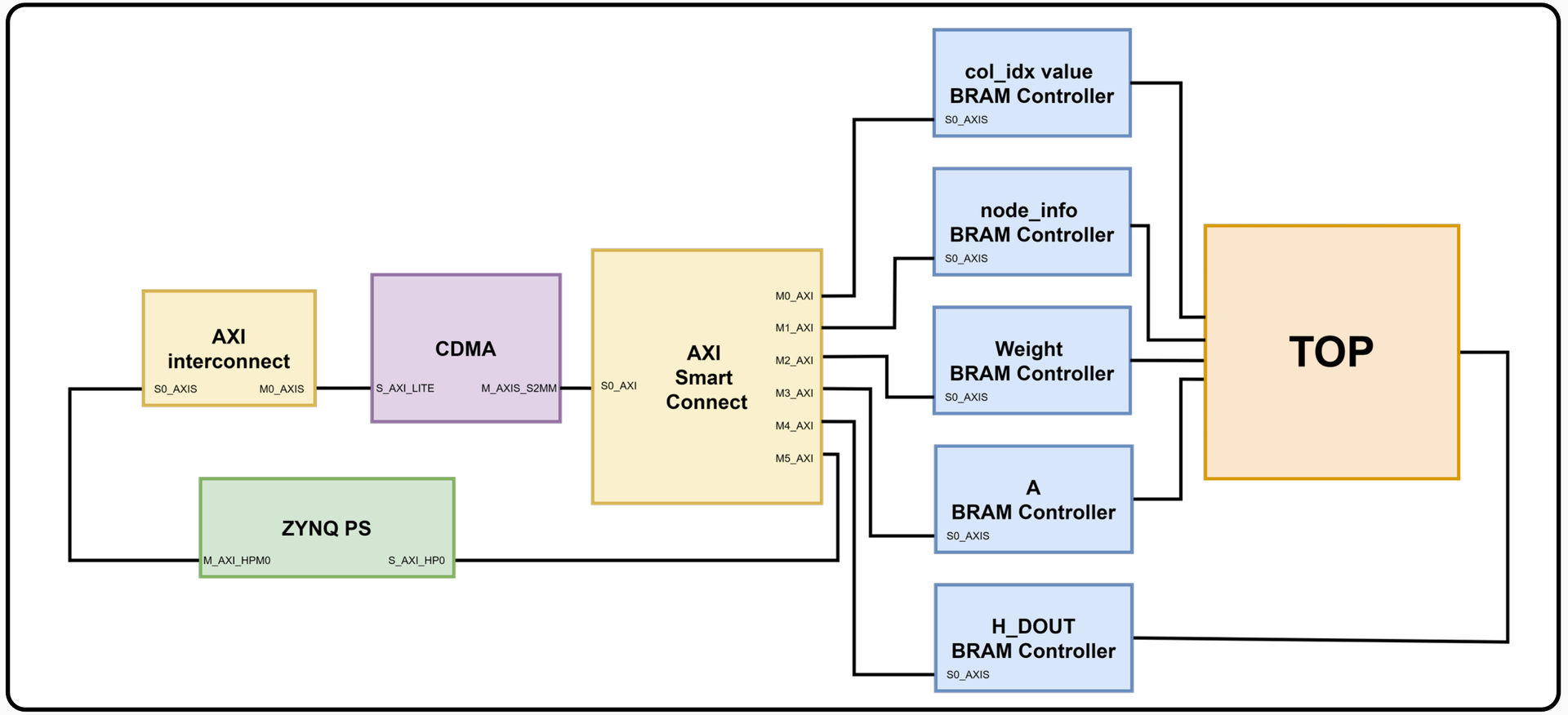
Finally, the accumulated values pass through the **ReLU** function to produce the **feature vector** for the **next layer**.



05 Implementation

System Implementation

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024



System Block Design (Vivado)

06 Results

SPMM Simulation

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

11	0	0	0	3
0	3	0	0	1
2	9	4	0	0
0	13	8	0	5

X

24	14	7	3	28	-28
14	21	16	2	2	-29
20	9	22	-9	-22	19
16	13	18	13	2	-3
-26	20	31	11	1	2

=

186	214	170	66	311	-302
16	83	79	17	7	-85
254	253	246	-12	-14	-241
212	445	539	9	-145	-215

Feature (4x5) Weight (5x6) WH (4x6)

col_idx	0	4	1	4	0	1	2	1	2	4
value	11	3	3	1	2	9	4	13	8	5

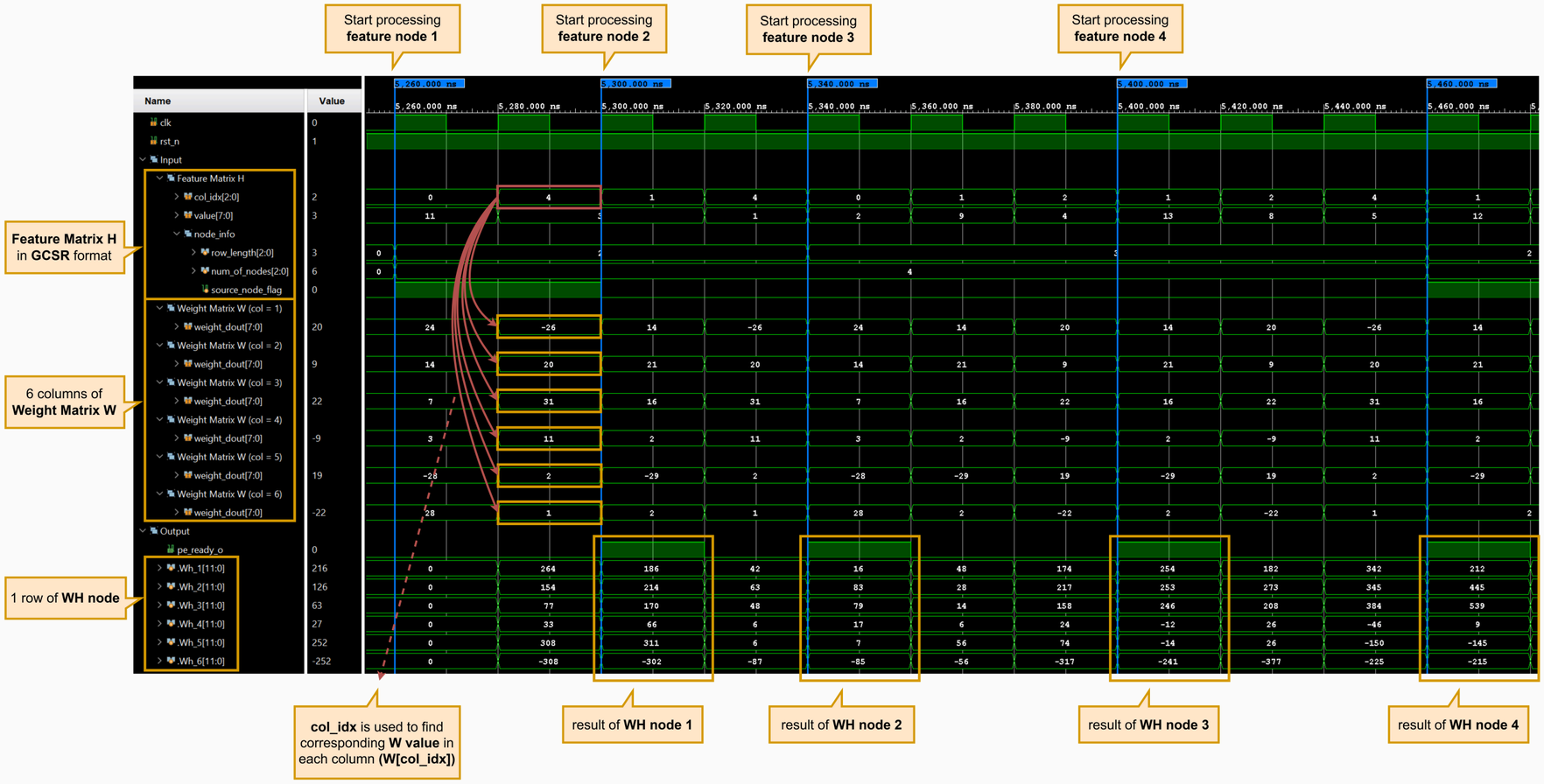
node_info	2 || 4 || 1	2 || 4 || 0	3 || 4 || 0	3 || 4 || 0

SPMM Simulation

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

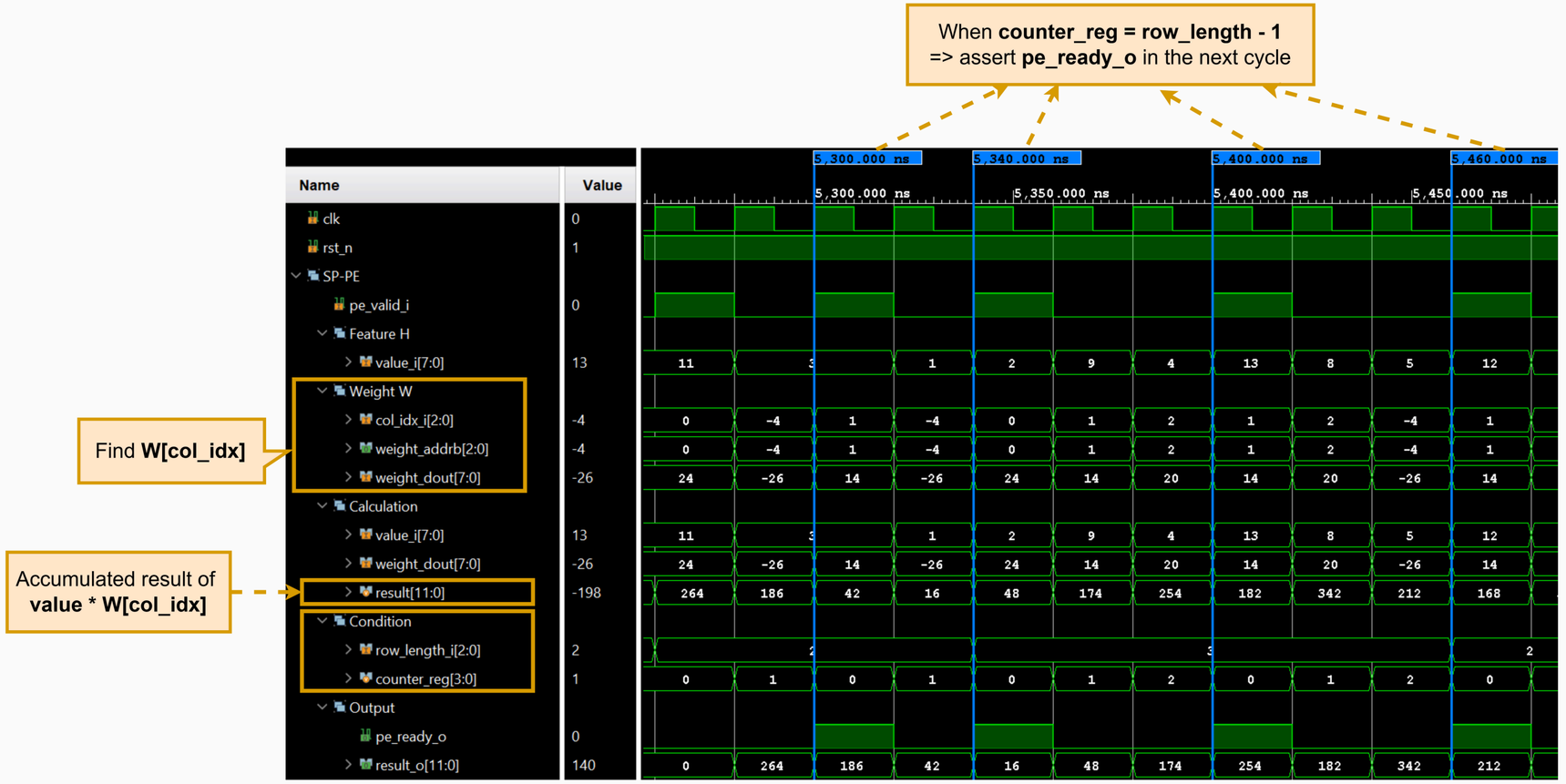
186	214	170	66	311	-302
16	83	79	17	7	-85
254	253	246	-12	-14	-241
212	445	539	9	-145	-215

WH (4x6)



SPMM Simulation

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024



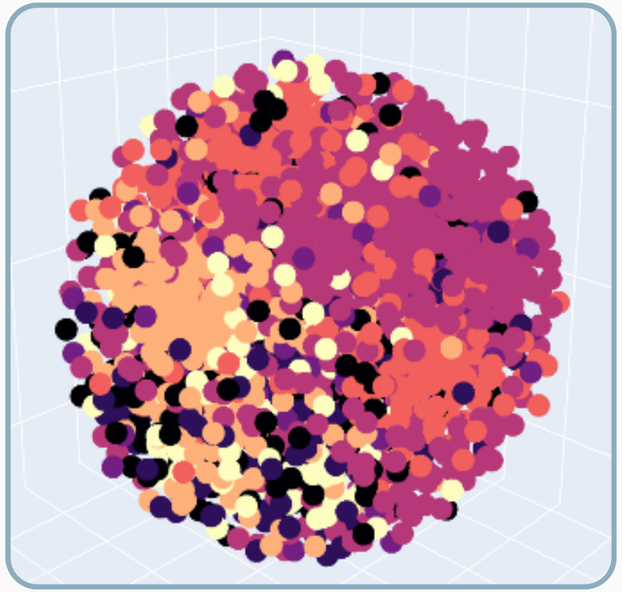
Calculation operation in one SP-PE module

Model Training

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Visualize Cora Dataset (Without Training)

2-D Visualization

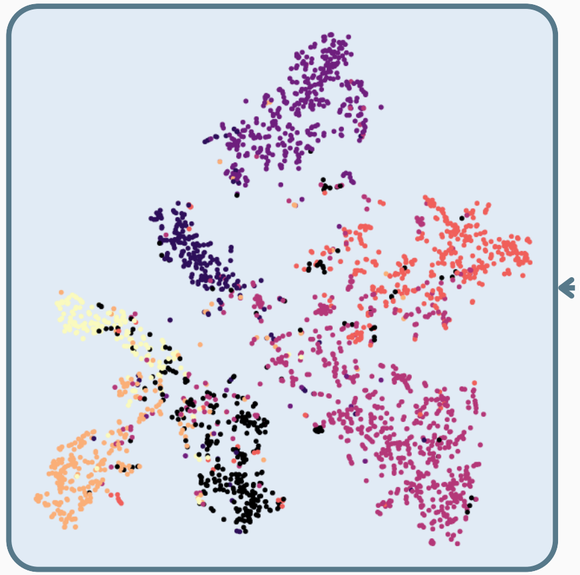


3-D Visualization

Model Training

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Visualization (Training GAT Model + Without Quantization)



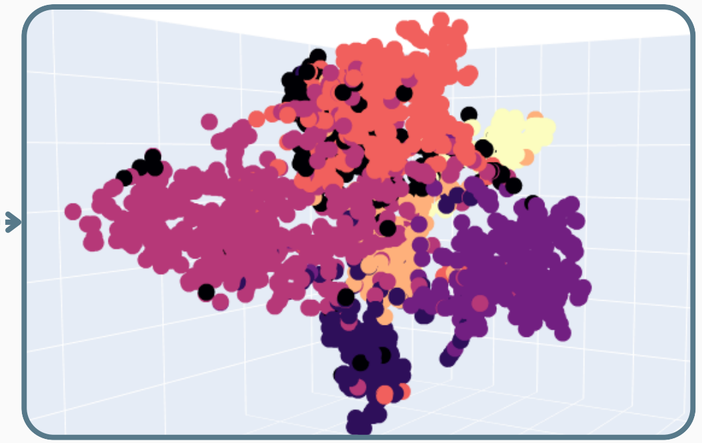
2-D Visualization

82.5%

```
test_accuracy = buildGATModel.test()
print(f'Test Accuracy without Quantization:

Test Accuracy without Quantization: 0.825
```

Accuracy



3-D Visualization

Model Training

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Visualization (Training GAT Model + Quantization)

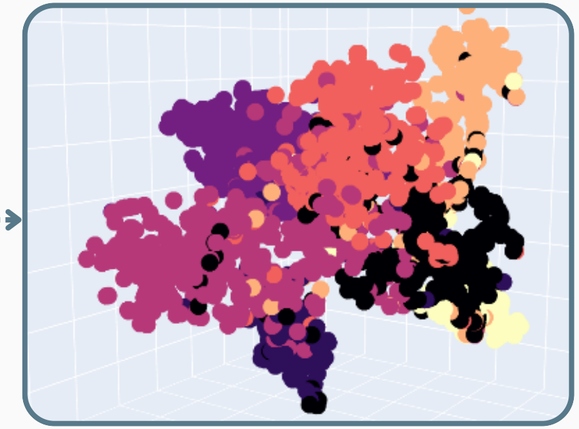
2-D Visualization

82.2%

```
test_accuracy = buildGATModel.test()
print(f'Test Accuracy with Quantization Aware Training (QAT):
```

```
Test Accuracy with Quantization Aware Training (QAT): 0.822
```

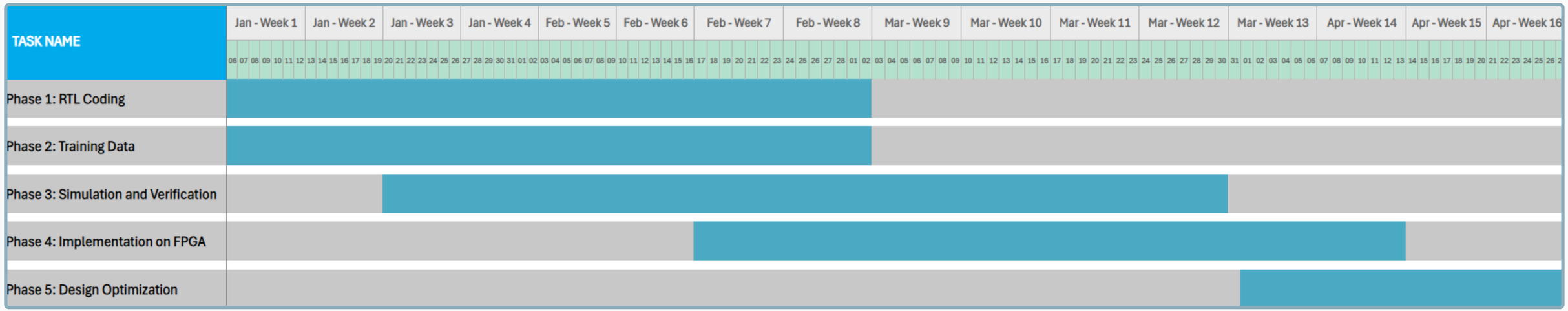
Accuracy



3-D Visualization

Future Plan

Future Plan



Proposed Timeline And Goals

Thank you for your Attention

System Accuracy

Ho Chi Minh University of Technology
Faculty of Computer Science and Engineering
Computer Engineering Project | 2024

Visualization (Full Flow Simulation On Software)

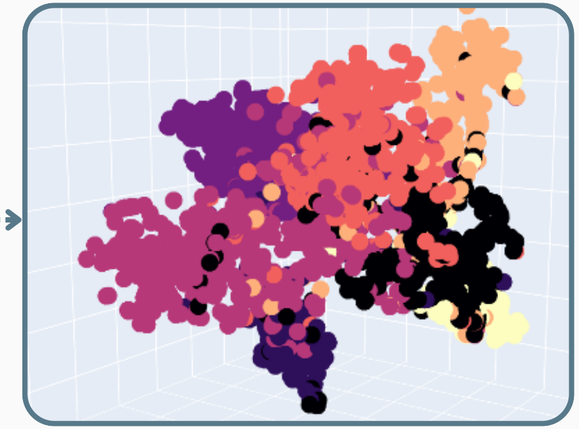
2-D Visualization

82.2%

```
test_accuracy = buildGATModel.test()
print(f'Test Accuracy with Quantization Aware Training (QAT):
```

```
Test Accuracy with Quantization Aware Training (QAT): 0.822
```

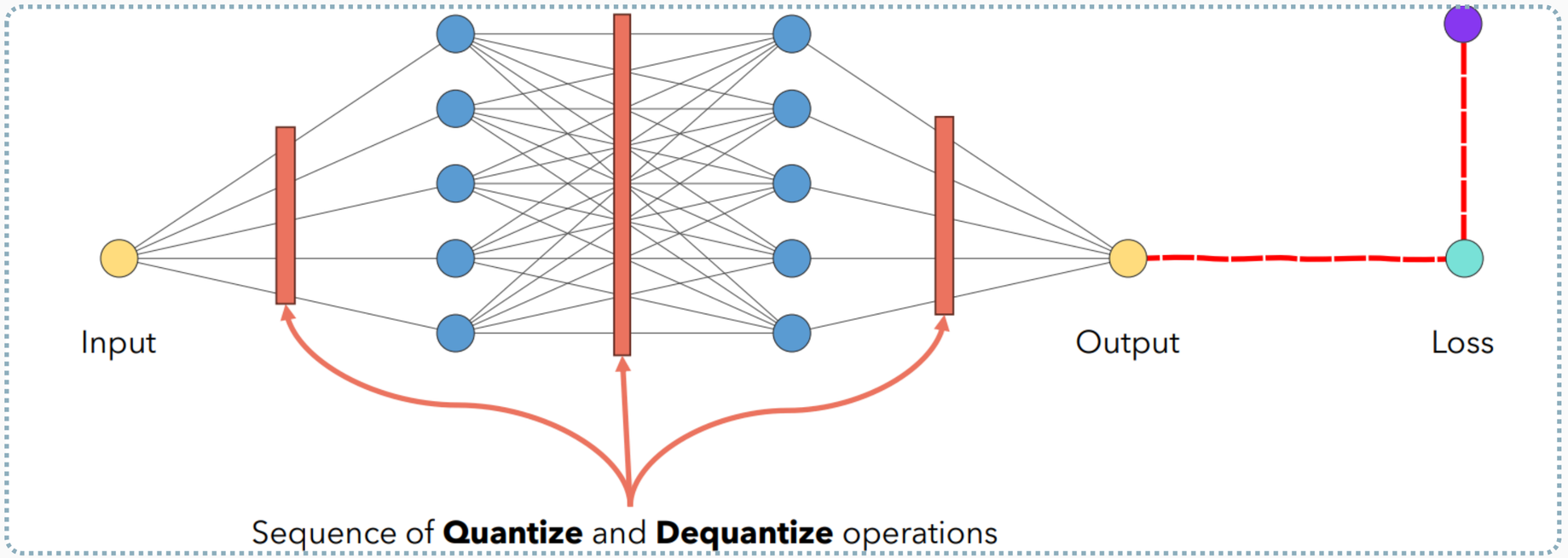
Accuracy



3-D Visualization

Quantization Process

- Convert parameters from Float32 to Int8
- Apply Quantization Aware Training (QAT) technique
- Detailed Flow



Softmax Accuracy Loss

Dataset	Cora	Citeseer	Pubmed
Accuracy Loss	0.15	0.006	0.012

Accuracy loss across datasets

Latency Comparison

Datasets	Latency (us)				
	SH-GAT (Ours)	FTW-GAT [10]	FP-GNN [9]	H-GAT [8]	S-GAT [7]
Cora	19.4	44.9	46.3	600	6400
CiteSeer	22.1	50.8	71.4	800	N/A
PubMed	150.2	339	616	5700	N/A

Comparison with FPGA-based accelerators on latency

GAT With Dataset

- Cora (Previous Slide)
- CiteSeer
- PubMed

Dataset	Nodes	Edges	Input Feature	Classes	Feature Density	Edge Density	Weight Density
Cora	2708	10,556	1433	7	1.3%	0.14%	100%
CiteSeer	3327	9104	3703	6	0.8%	0.08%	100%
PubMed	19,717	88,648	500	3	10.4%	0.02%	100%

Dataset information with Nodes, Edges and Features