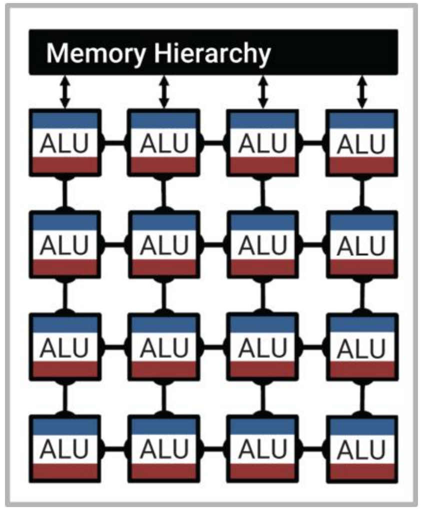
LISA: Graph Neural Network based Portable Mapping on Spatial Accelerators

Zhaoying Li, Dan Wu, Dhananjaya Wijerathne, Tulika Mitra

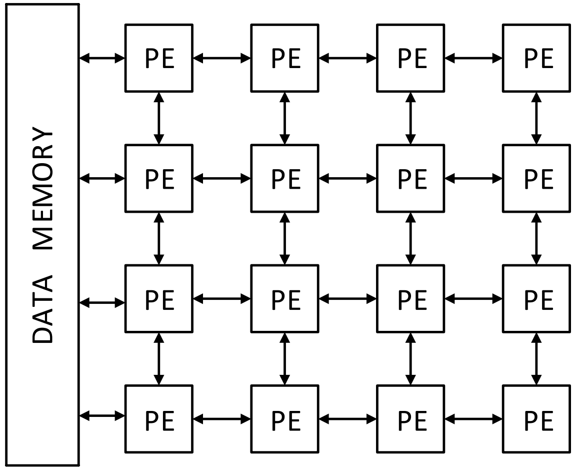
HPCA 2022

NUS
National University of Singapore

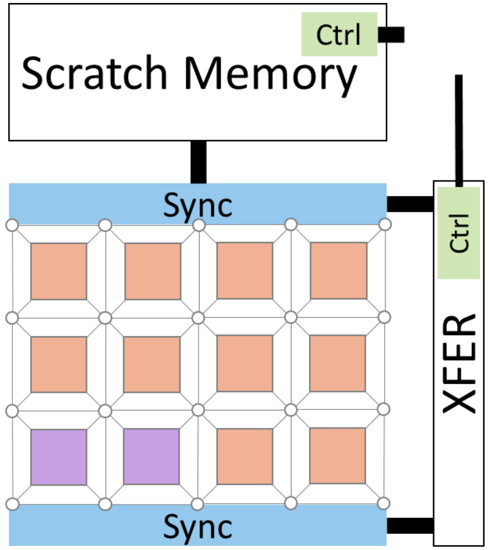
The Rapid Increase of Spatial Accelerator



Deep Learning Accelerators[1]



Coarse Grained Reconfigurable Architecture (CGRA) for compute-intensive kernels



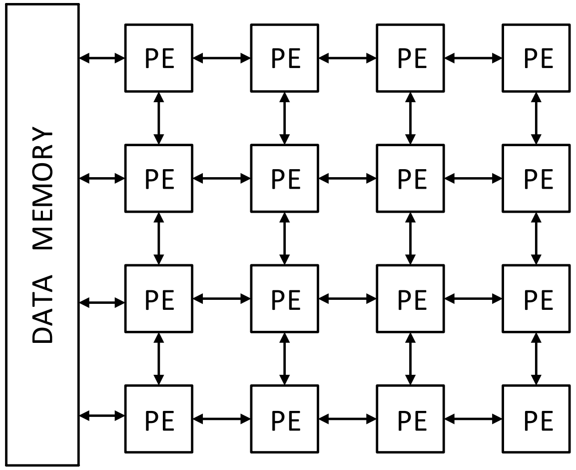
Matrix Multiplication Accelerators[2]

Common characteristics: Programmable Processing Elements (PE), Network on-chip, and Scratch Memory.

High performance, low power consumption, for various kernels

1. https://www.synopsys.com/designware-ip/technical-bulletin/building-efficient-deep-learning-dwtb_q318.html
2. Weng, Jian, et al. "A hybrid systolic-dataflow architecture for inductive matrix algorithms." 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2020.

CGRAs

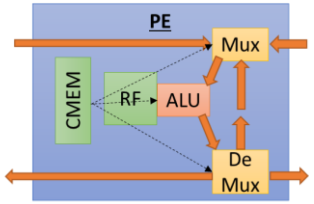


4x4 CGRA

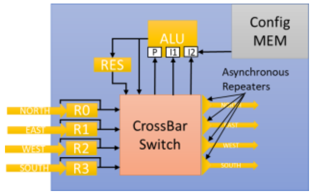
Different Size

2x2, 4x4, 8x8, 16x16 ...

Different On-chip Network



N2N
Communicate with neighbors



HyCUBE[1]
Single cycle multiple hops

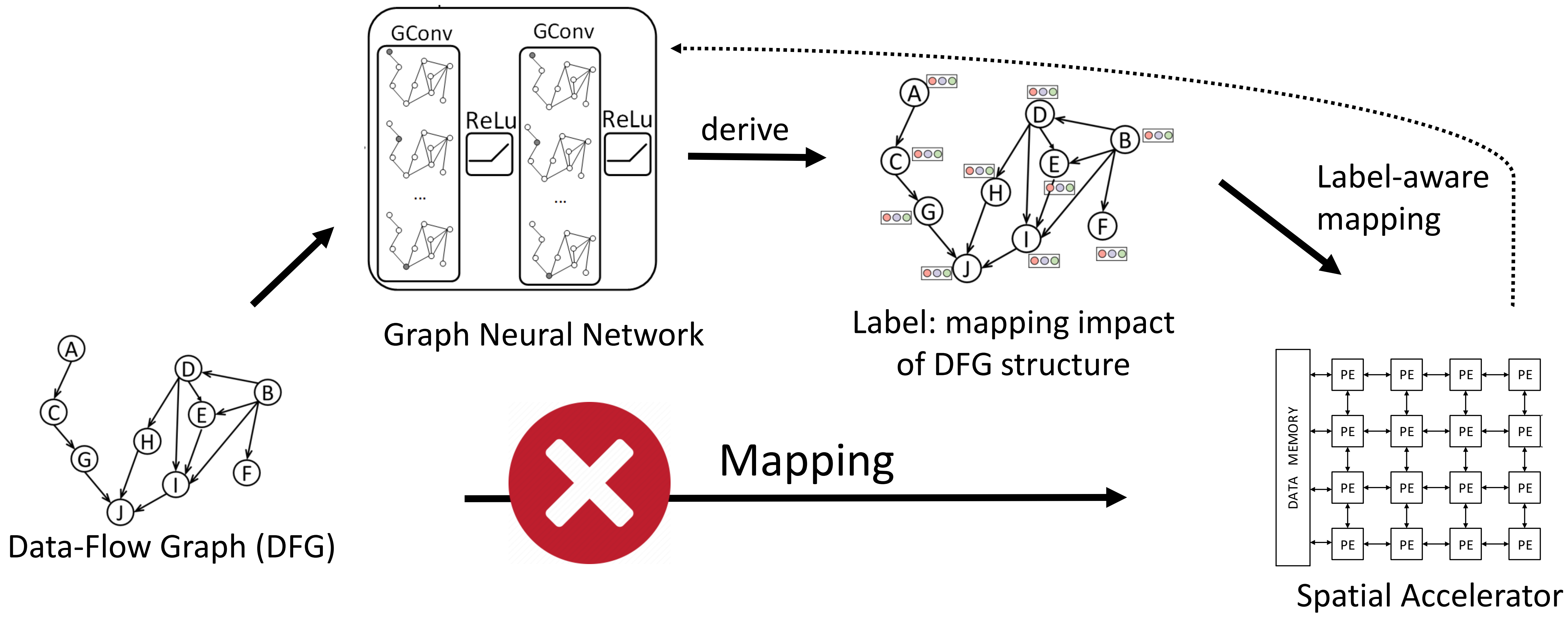
Different Functionality

Homogeneous vs Heterogeneous

1. Karunaratne, Manupa, et al. "HyCUBE: A CGRA with reconfigurable single-cycle multi-hop interconnect." Proceedings of the 54th Annual Design Automation Conference 2017.
2. Weng, Jian, et al. "A hybrid systolic-dataflow architecture for inductive matrix algorithms." 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2020

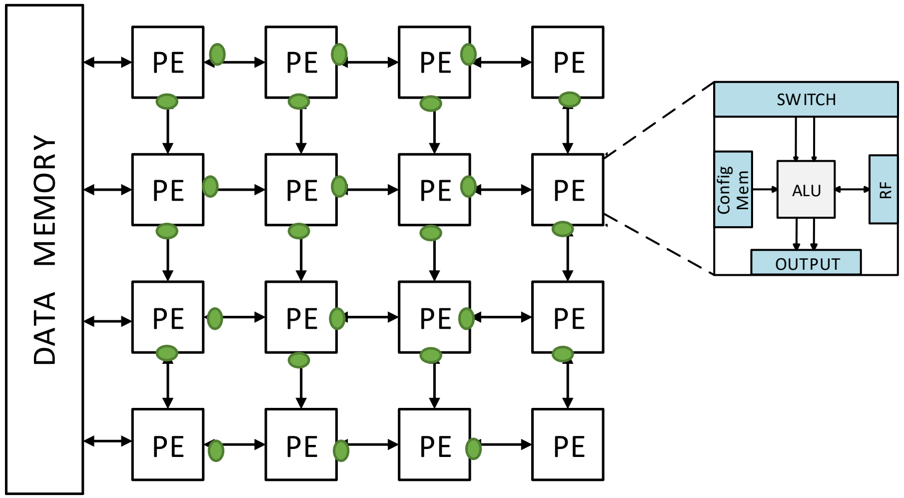
LISA Overview

Automatically tune the parameter for different accelerators 😇



Need to handcraft the mapper for each accelerator 🙁

Background



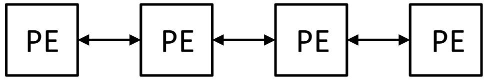
Coarse Grained Reconfigurable Architecture (CGRA)

- The PE executes the **operation** (add, load, store, etc.)
- The on-chip network sends **data** among PEs
- The network and PE can be **reconfigured every cycle**

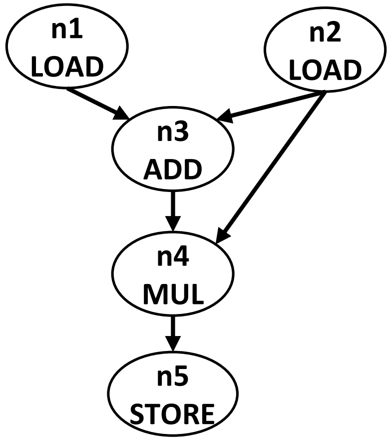
Background

```
for ( int i = 0 ; i <= N ; i++ ) {
      a [ i ] = b [ i ] + c [ i ] ;
      d [ i ] = a [ i ] * c [ i ] ;
}
```

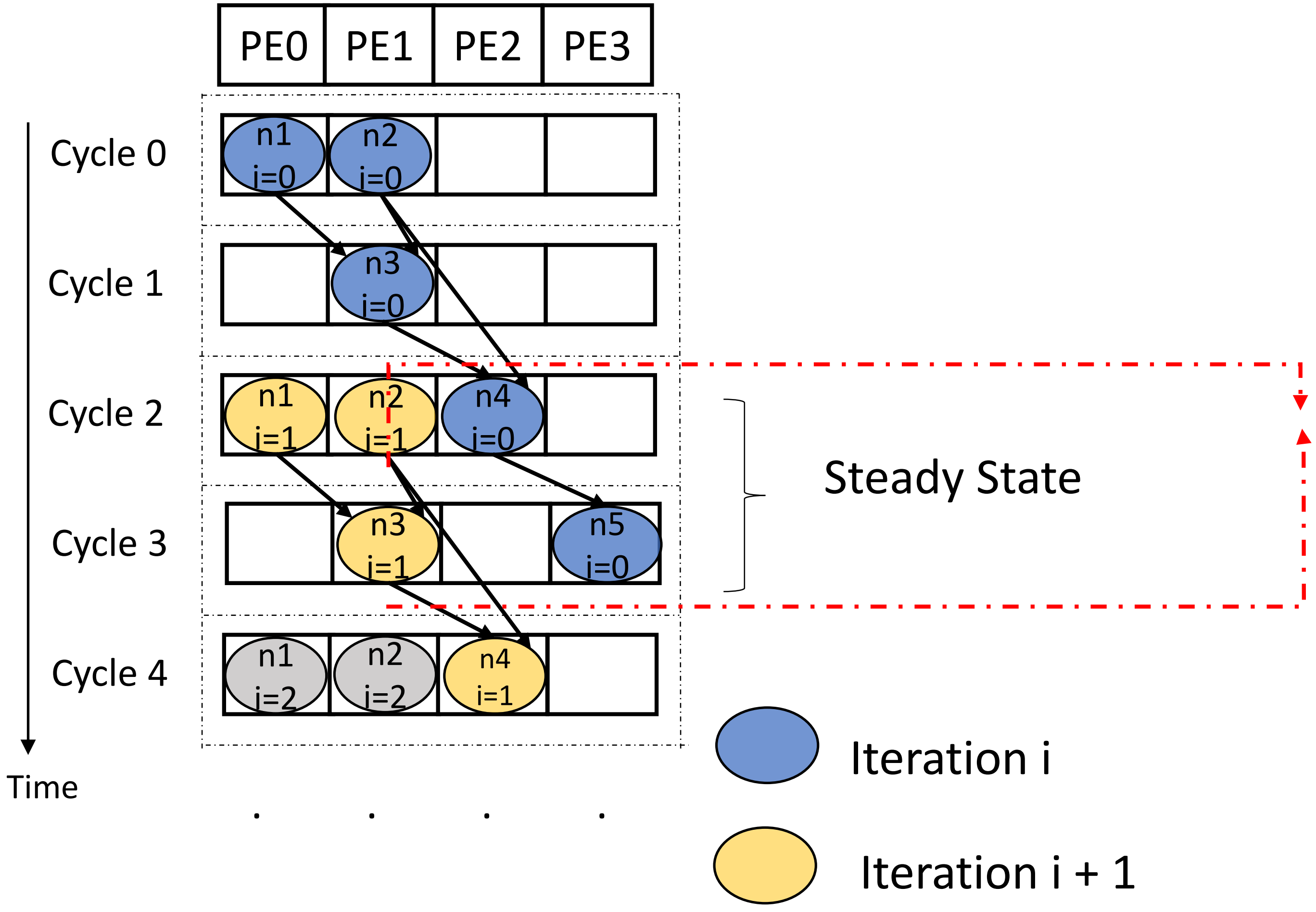
Kernel Code



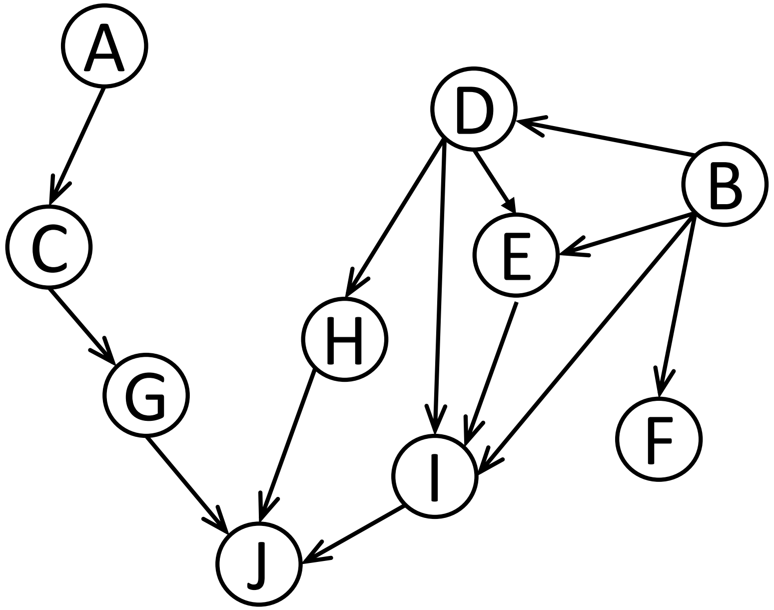
CGRA



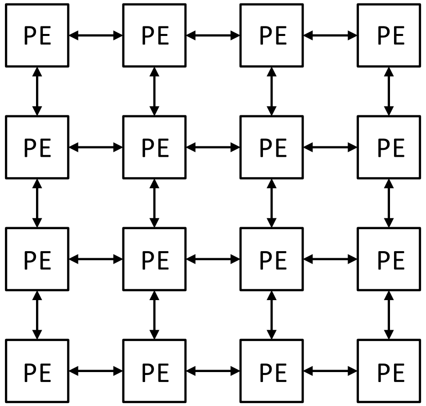
Data Flow Graph



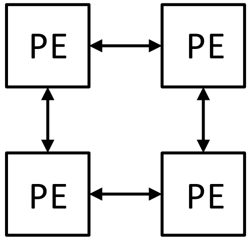
Accelerator Characteristic affect on mapping



DFG



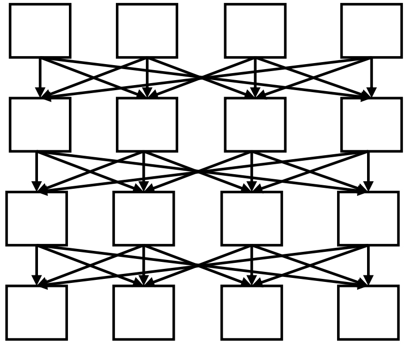
4x4 CGRA



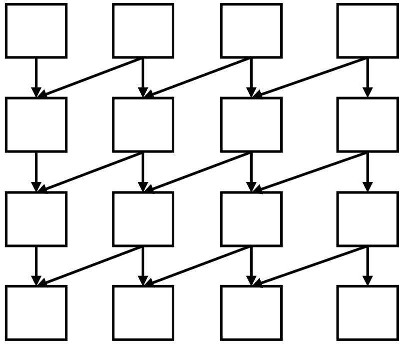
2x2 CGRA

- Different CGRA size -> different hardware resource
- To map DFG on 4x4 CGRA, we need to place nodes spatially to utilize hardware resource.
- To map DFG on 2x2 CGRA, we need to ``stretch`` the DFG along the time dimension.

Accelerator Characteristic affect on mapping



CGRA with normal routing resource

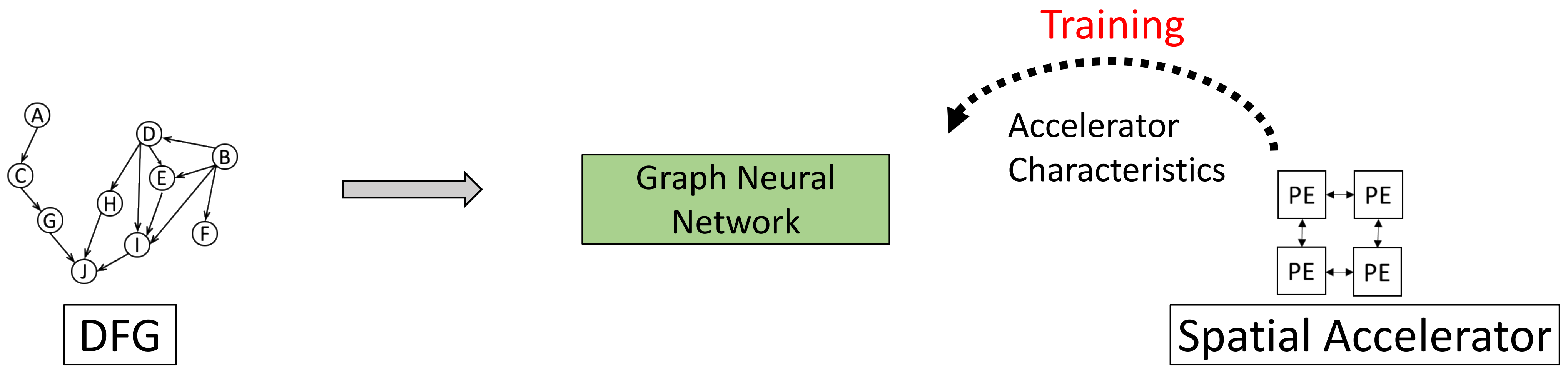


CGRA with less routing resource

- Less routing resources make it hard to route complex data dependency.
- The mapper needs to ``be aware of `` the insufficient routing resource and solve it.

The spatial accelerator is so diverse, and we need a smart mapper!

GNN (Graph Neural Network)



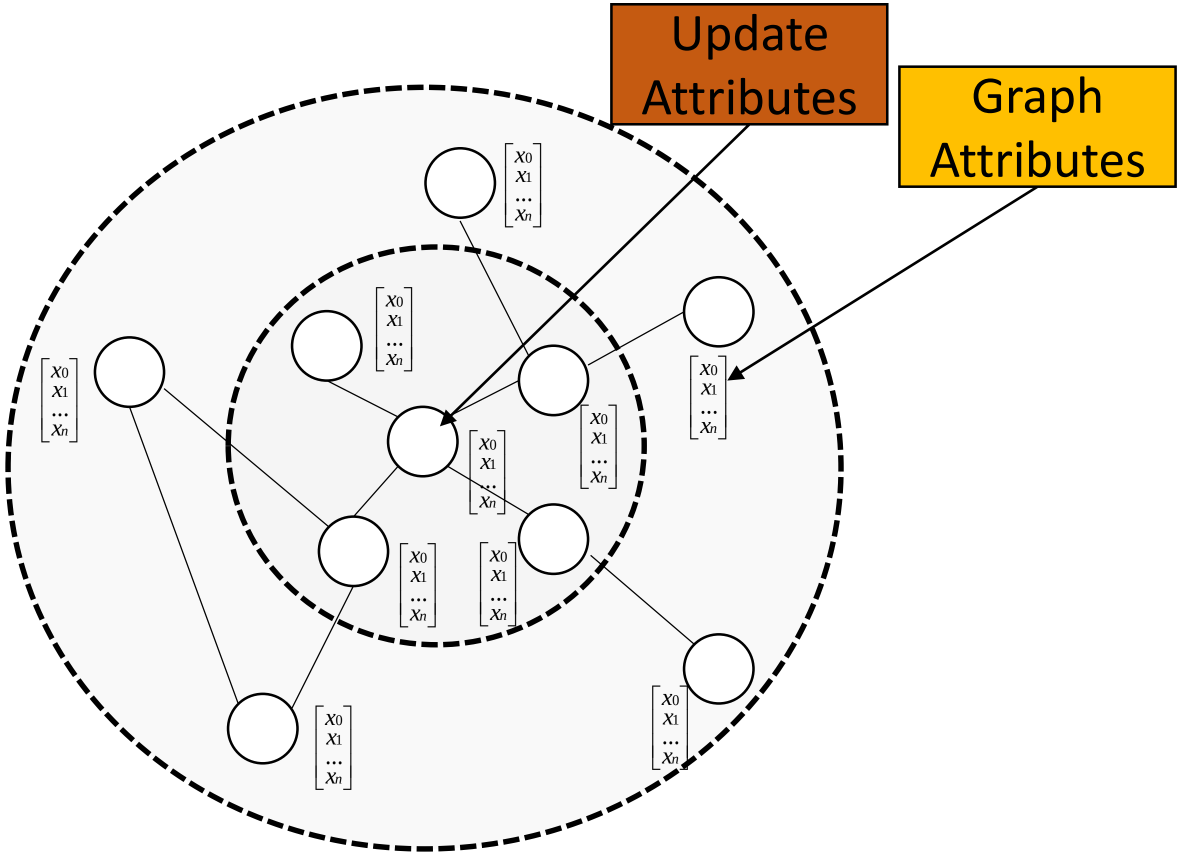
Why we use GNN

- Flexible methods to traverse DFG and collect information
- Re-train the GNN model for different accelerators to automatically tune the parameters.

GNN (Graph Neural Network)

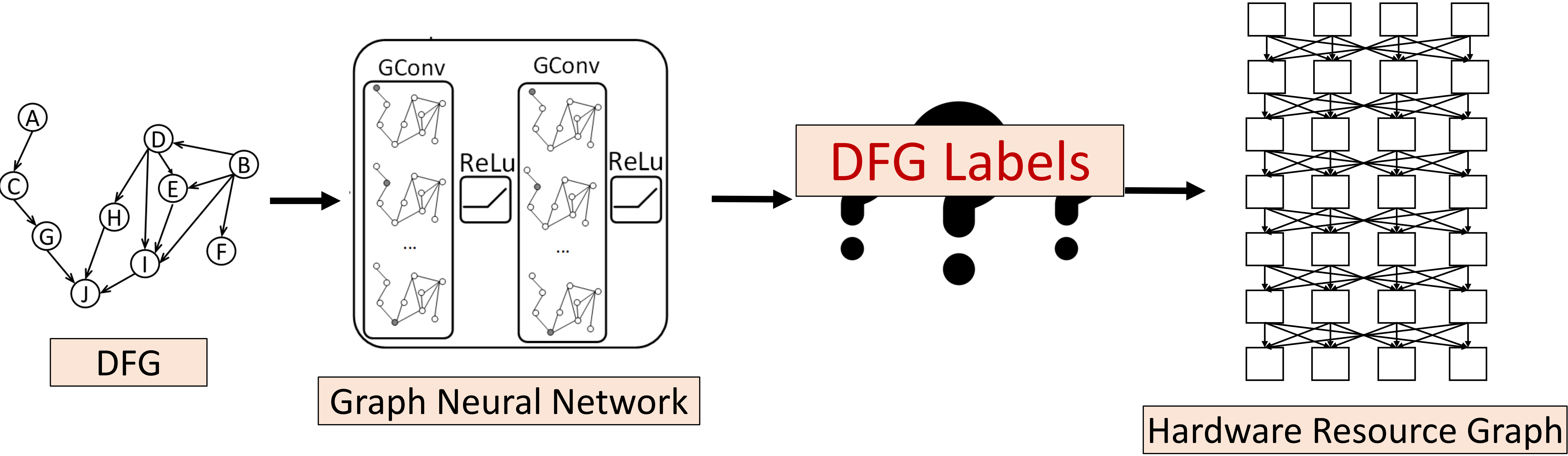
How GNN works

- Collect nodes and edges information
- Use aggregator to process information
- Update itself information and then propagate it



Graph Neural Network (GNN) Example

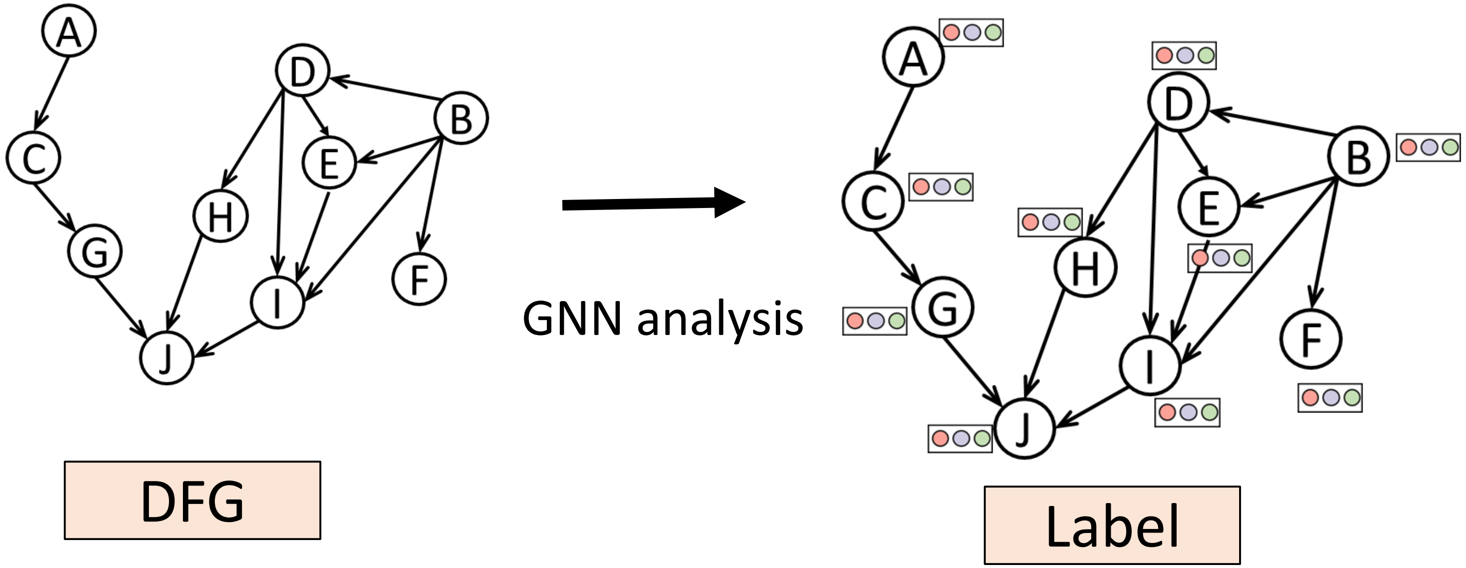
The Gap between Mapping and GNN



Label

- Labels are the estimated mapping information of DFG nodes and edges.

--> Describe how DFG should be mapped on Spatial Accelerators.

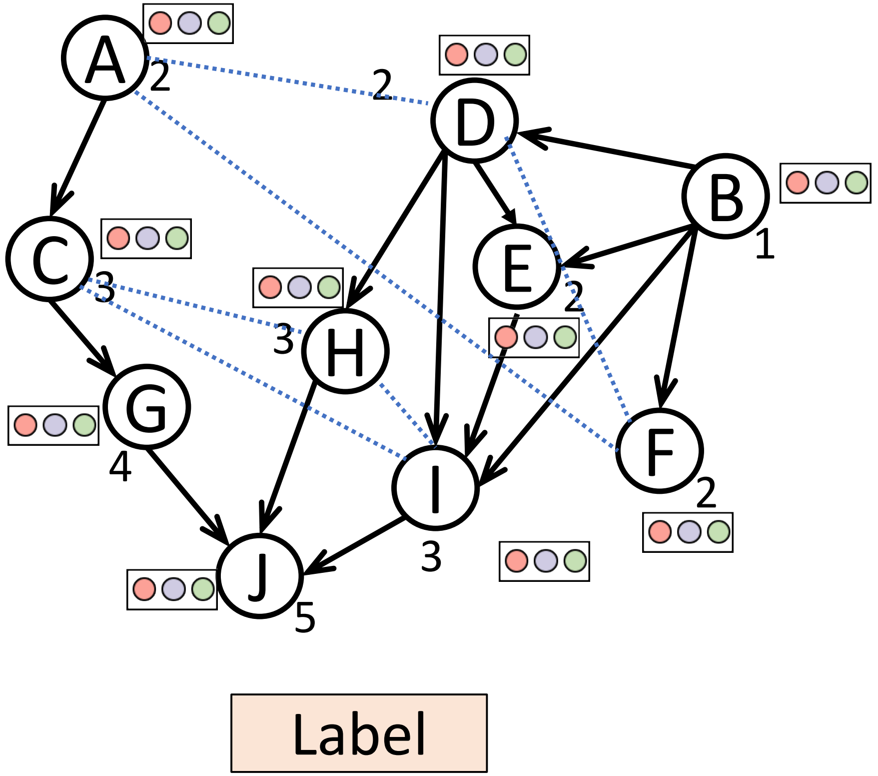


Label

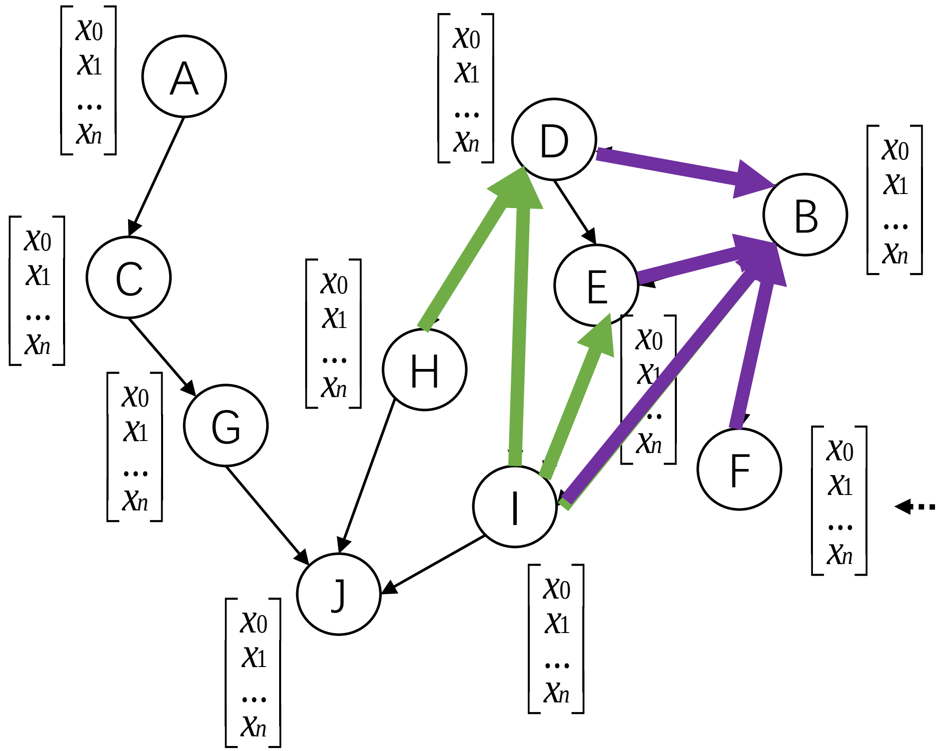
- Labels describes mapping information of DFG nodes and edges.

--> how DFG should be mapped on Spatial Accelerators.

Label	Type
Schedule Order	Node
Spatial Mapping Distance	Edge
Temporal Mapping Distance	Edge
Same-level Node Association	Edge



GNN – DFG analysis example



- LISA designs a network for each label
- Example: use GNN to derive schedule order label
- Each DFG node has attributes:
  - # of child nodes
  - # of parent nodes
  - ASAP value
  - # of nodes in two hops
  - …

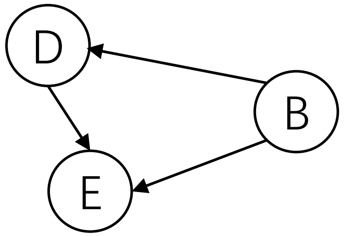
DFG structure information

Aggregator：

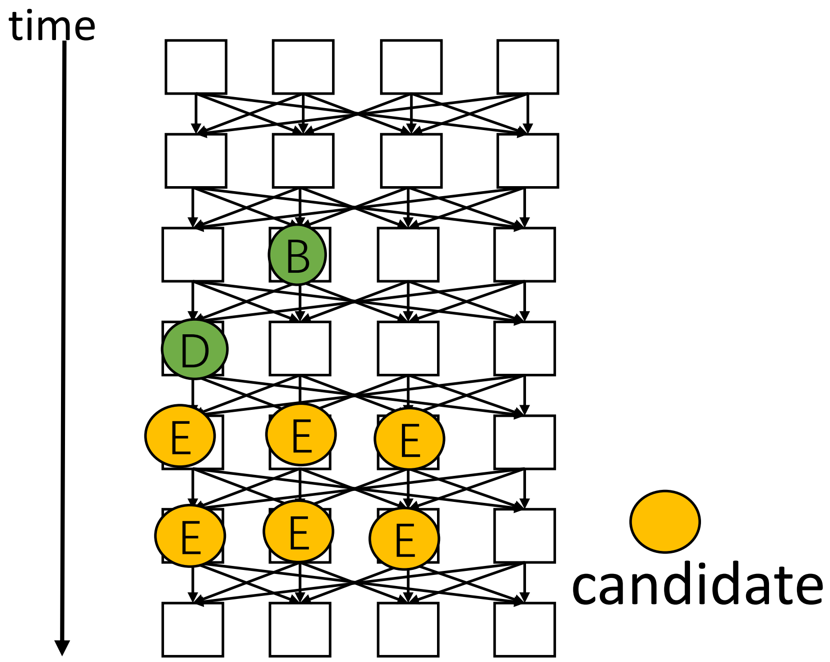
Collect neighbor information $\mathbf{m}_v^{(t+1)} = \mathbf{W_1}\left[\mathbf{mean}(\mathbf{m}_u^{(t)}), \mathbf{max}(\mathbf{m}_u^{(t)}), \mathbf{min}(\mathbf{m}_u^{(t)})\right]$

Update itself schedule order $\mathbf{h}_v^{(t+1)} = \mathbf{W_2}\left(\mathbf{W_3}\mathbf{h}_v^{(t)} + \mathbf{m}_v^{(t+1)}\right)$

Label-aware Simulated Annealing



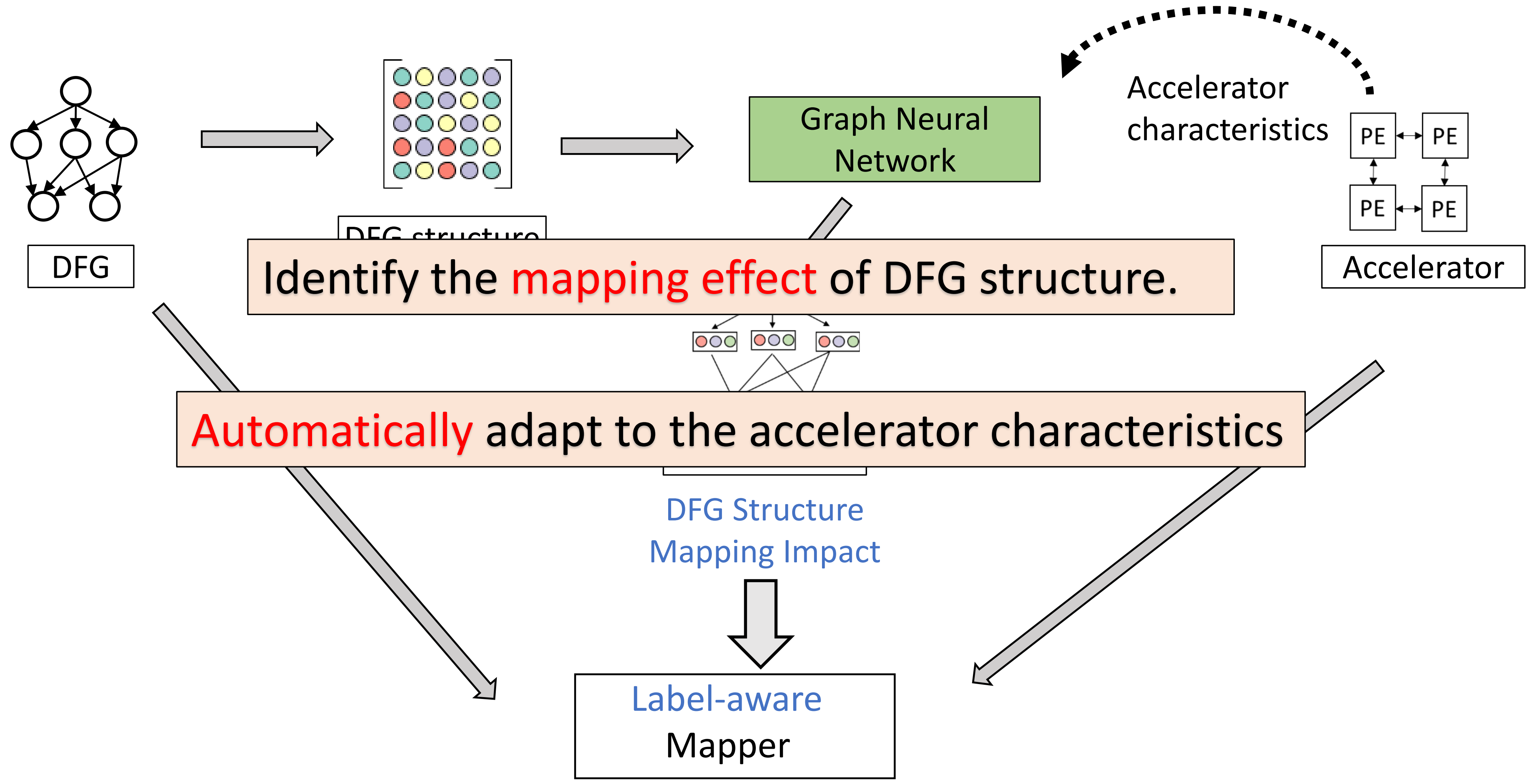
Part of DFG



Mapping Example

- Use simulated annealing to construct mapping
- Use schedule order to place nodes one by one.
- Select PE candidate according to mapping distance label cost (with normal distribution on selection)
- We can customize combinations of labels to guide mapping decisions.

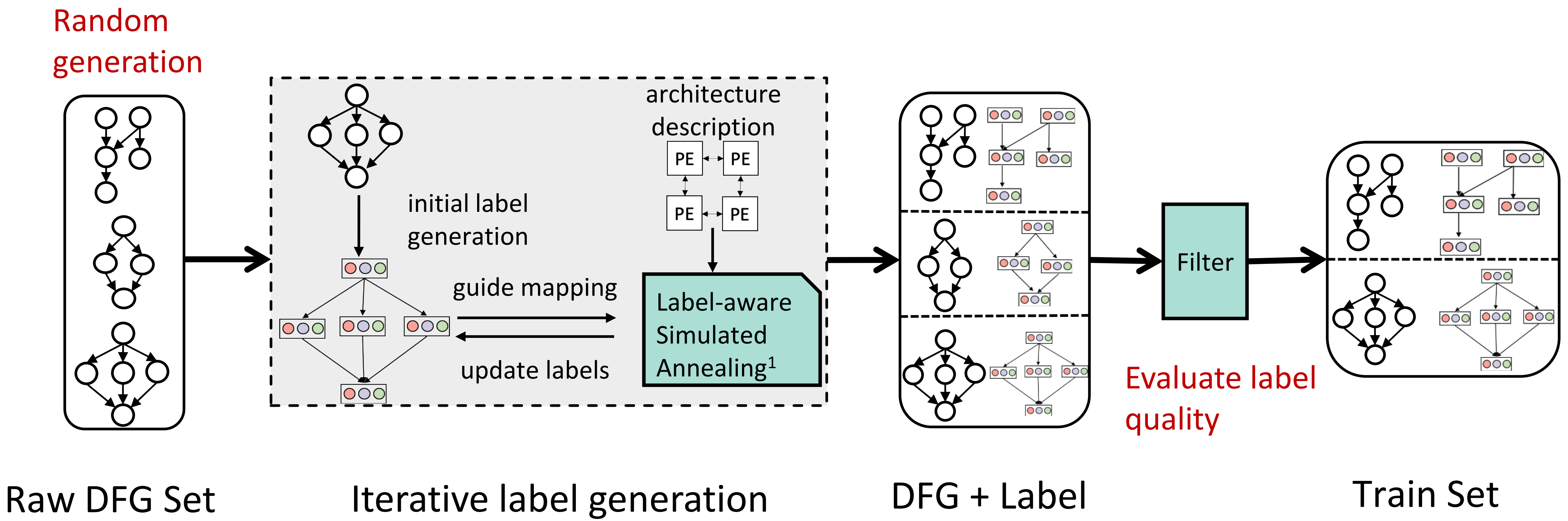
LISA: Learning Induced Mapping for Spatial Accelerators



Identify the mapping effect of DFG structure.

Automatically adapt to the accelerator characteristics

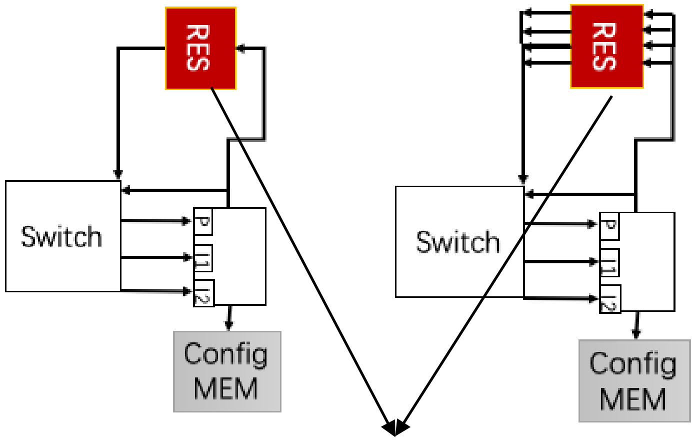
Train Data Set



The train data generation ``profiles`` the accelerator characteristics.

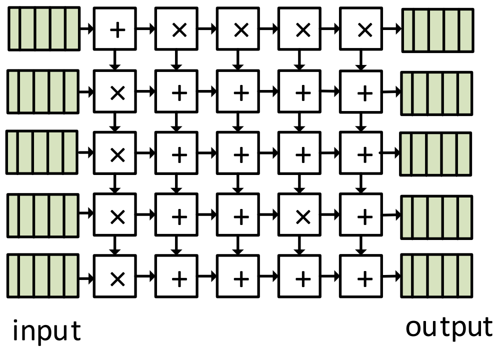
Experiments

- Various Architectures
  - 4x4 CGRA
  - 3x3 CGRA
  - 4x4 CGRA with less routing resource
  - 4x4 CGRA with more memory connectivity
  - 8x8 CGRA
  - Systolic Array



PE with Different Routing Resource

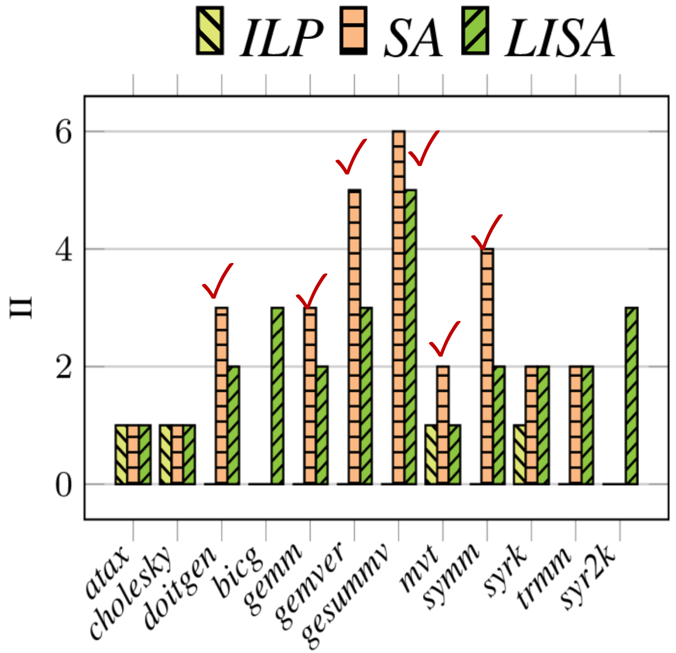
- Benchmarks: Polybench and unrolled version
- Platform: CGRA-ME
- Baseline:
  - ILP (integer linear programming)
  - Simulated Annealing



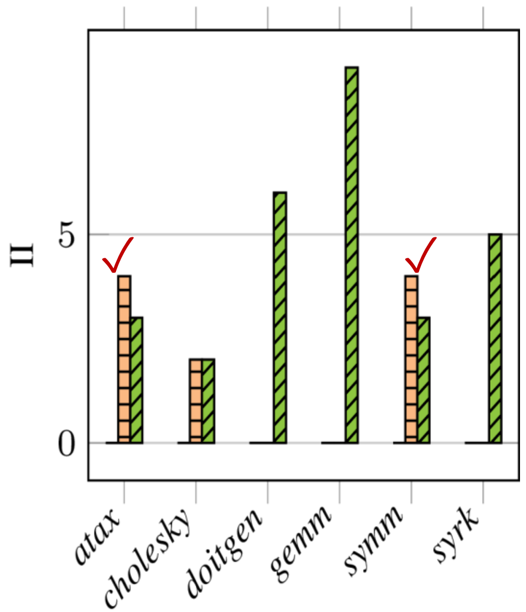
Systolic Array Unit[1]

1. Weng, Jian, et al. "A hybrid systolic-dataflow architecture for inductive matrix algorithms." 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2020.

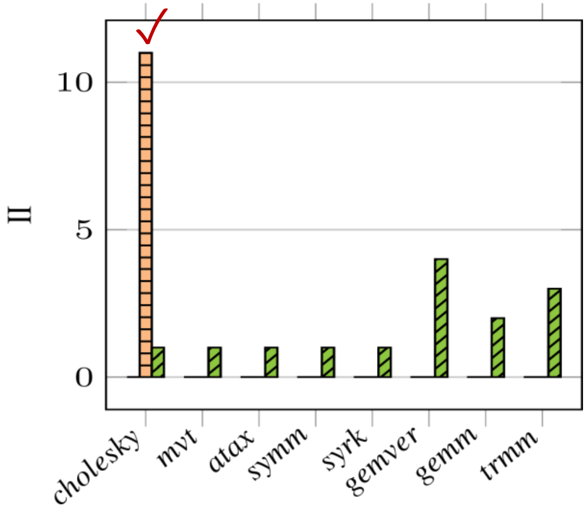
Performance



4×4 baseline CGRA



4×4 CGRA for unrolled DFG



8×8 CGRA for unrolled DFG

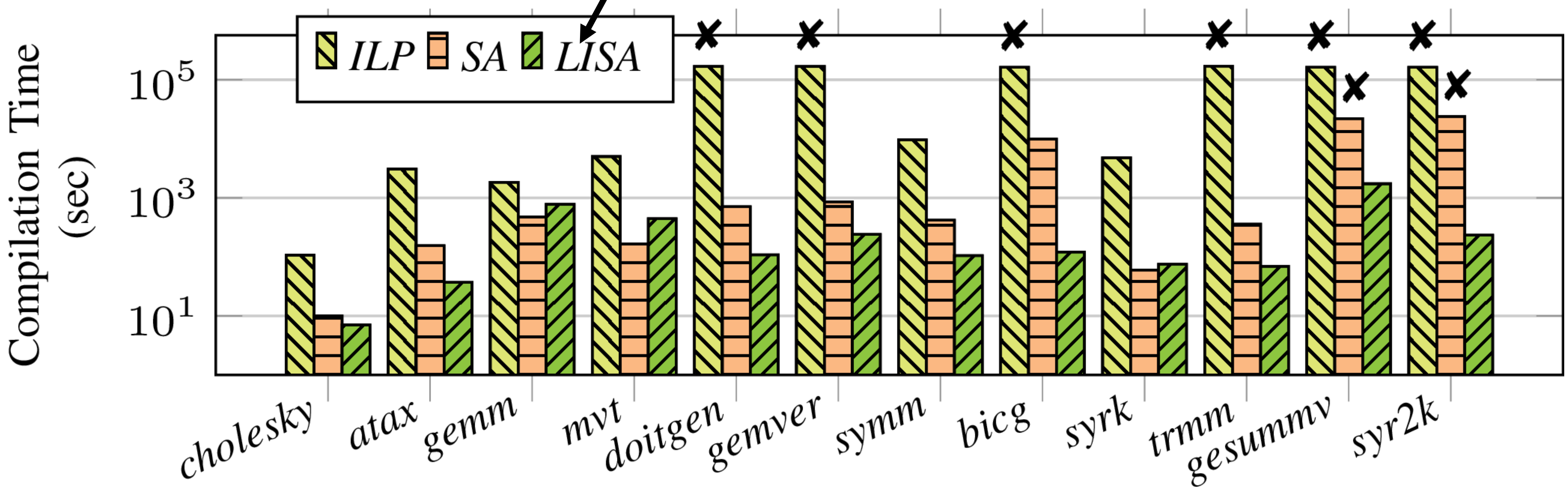
- In total, 71 combinations of benchmarks and accelerators. (The figure does not show all the combinations)
- ILP maps 23 combinations
- SA maps 49 combinations
- LISA maps 70 combinations
- LISA also achieves significant improvement on mapped DFGs.

Reasons:

- ❖ LISA maps the DFG with an all-encompassing global view
- ❖ LISA is better aware of accelerator characteristics and resources

Compilation Time Comparison

Include GNN inference time and mapping time

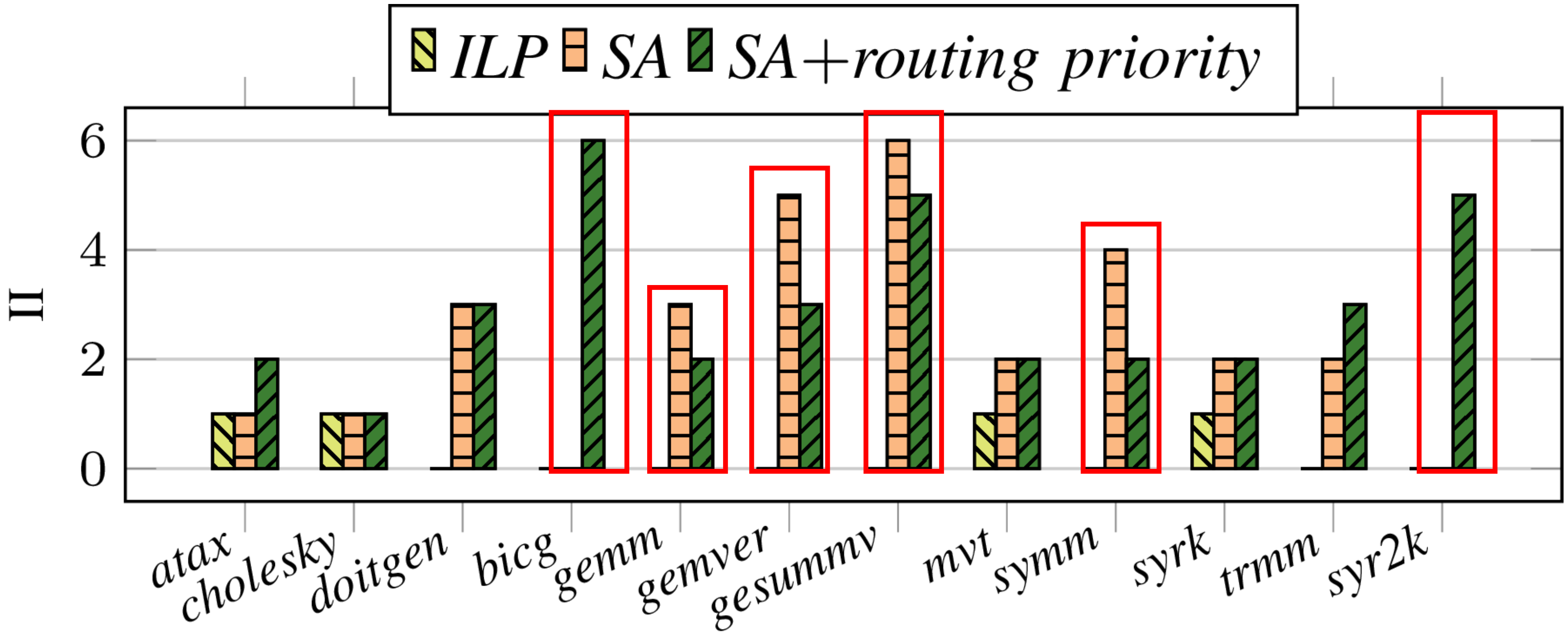


Compilation time comparison on 3x3 CGRA

On 3×3 CGRA, LISA achieves 594x and 17x compilation time reduction compared to ILP and SA

LISA uses labels to guide mapping and greatly reduces the search space

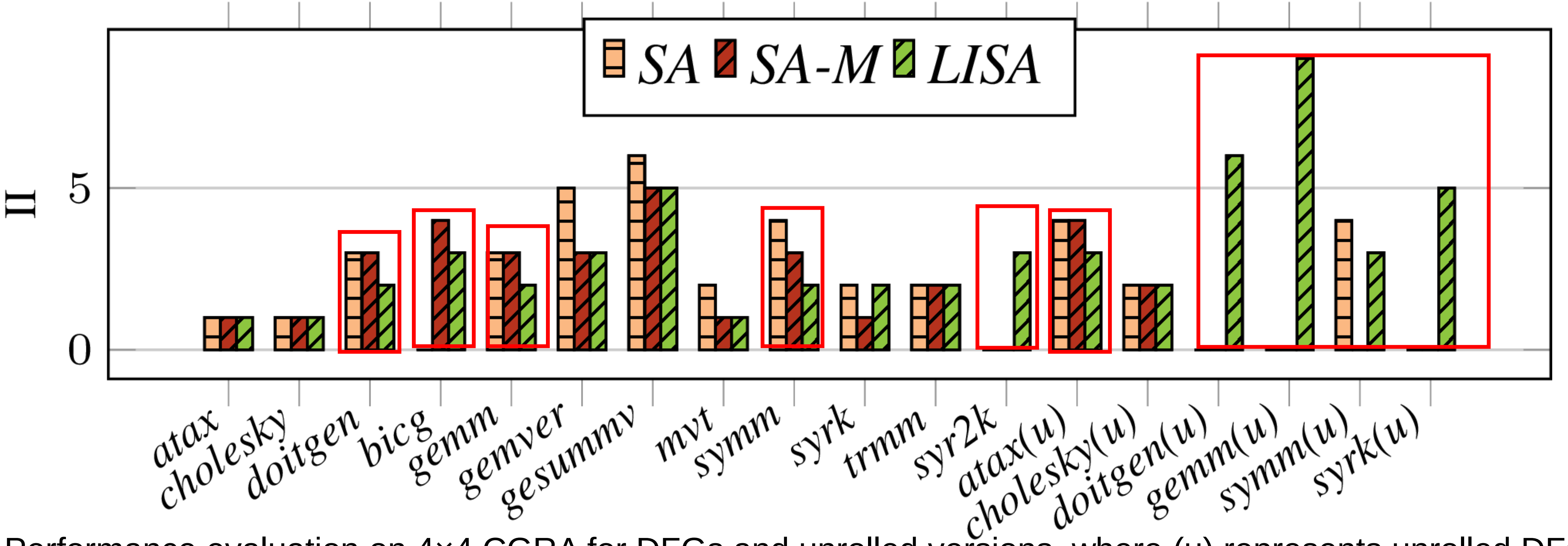
Effectiveness of LISA label



Effectiveness evaluation for temporal mapping distance (label 4) on 4×4 CGRA

- LISA uses multiple labels for mapping and temporal mapping distance (label 4) to decide routing priority
- We add routing priority to the original SA
- SA with routing priority can map two more benchmarks.

Effectiveness of Label-aware Mapping



Performance evaluation on 4×4 CGRA for DFGs and unrolled versions, where (u) represents unrolled DFGs.

- We create a new version of SA with 10× movements, called SA-M
- LISA still achieves lower II for four original DFGs and maps one more DFG (syr2k) compared to SA-M
- SA-M cannot map any unrolled DFGs.

GNN model accuracy

Spatial accelerator architecture	Prediction accuracy			
	label1	label2	label3	label4
4×4 baseline	0.788	0.856	0.932	0.992
3×3 baseline	0.648	0.939	0.992	0.938
4×4 with less routing resource	0.758	0.885	0.951	0.977
4×4 with less memory connectivity	0.738	0.852	0.941	0.988
8×8 baseline	0.685	0.716	0.914	0.990
systolic accelerator	0.759	0.768	0.907	1.000

Achieve high accuracy for most labels on different accelerators

Summary

- This is the first work to employ GNN to map DFG.
- We propose a portable mapping solution for spatial accelerators
- LISA can achieve high-quality mapping for different accelerators.

SCAN ME

Thank You!

LISA ToolChain: **https://github.com/ecolab-nus/lisa**

This research is partially supported by the National Research Foundation, Singapore under its Competitive Research Programme Award NRF-CRP23-2019-0003