A Generic FPGA Accelerator Framework for Ultra-Fast GNN Inference

Rishov Sarkar, Cong Hao
rishov.sarkar@gatech.edu, callie.hao@ece.gatech.edu

Rishov Sarkar
Ph.D. Student

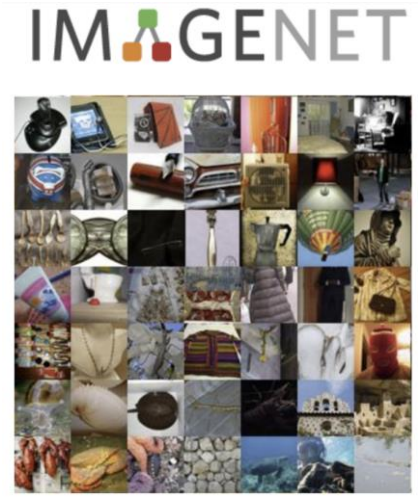
Sharc Lab @ Georgia Tech https://sharclab.ece.gatech.edu/

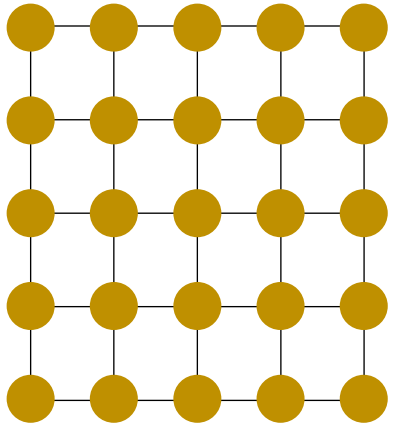
Outline

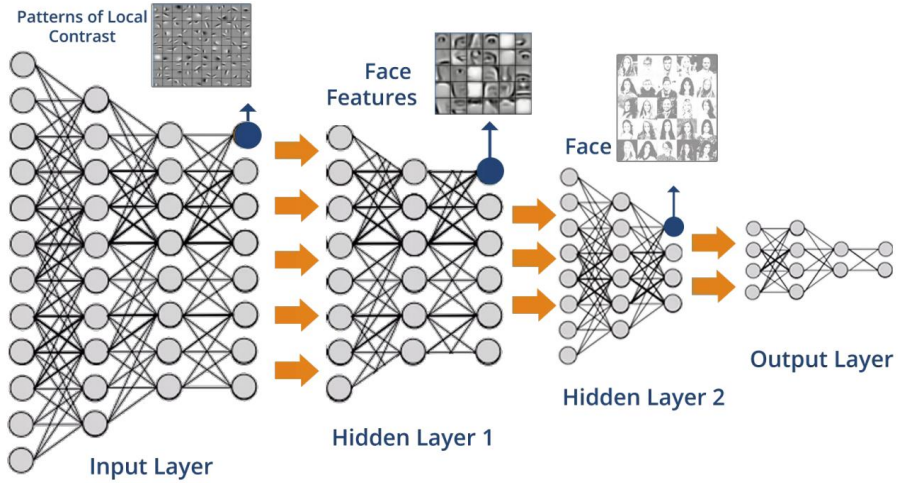
- **Background and Motivation**
  - What are Graph Neural Networks (GNNs)?
  - Why accelerate GNNs?
- **Existing Accelerator Limitations**
  - Not generic, not real-time
- **Generic and Parallelized GNN Accelerator**
  - Generic message passing framework — GenGNN
  - Node-level and edge-level parallelism
- **Evaluation: CPU/GPU**

What are Graph Neural Networks (GNNs)?

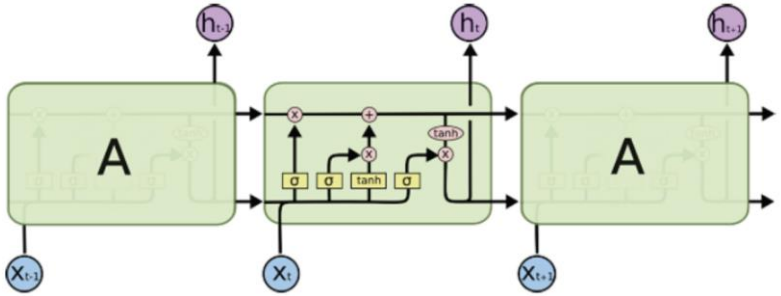
- **Traditional neural networks are designed for simple sequences & grids**







Speech/Text



[Slide credit: http://web.stanford.edu/class/cs224w]

What are Graph Neural Networks (GNNs)?

- **Reality: A lot of real-world data does not "live" on grids**
  - Arbitrary size and complex topological structure
  - No fixed node ordering or reference point

Image credit: Medium

Social Networks

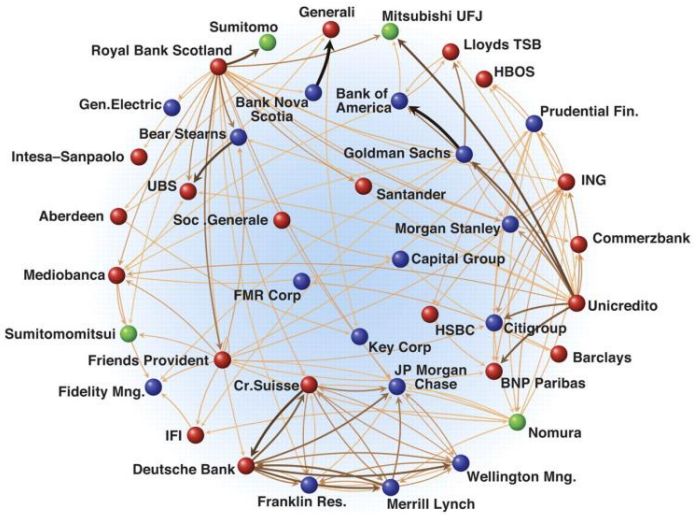


Image credit: Science

Economic Networks

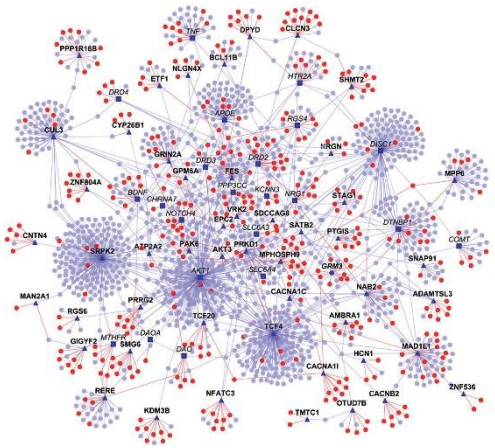


Image credit: Madhavicmu / Wikimedia Commons/CC-BY-SA-4.0

Protein Interaction Networks

What are Graph Neural Networks (GNNs)?

- **Mainstream:** Pass **messages** between pairs of nodes, **aggregate**, and **transform**

Image credit: Medium

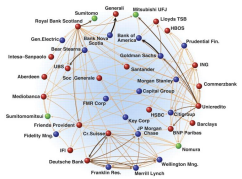
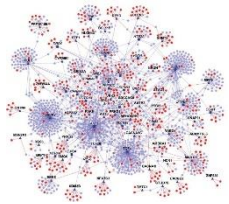
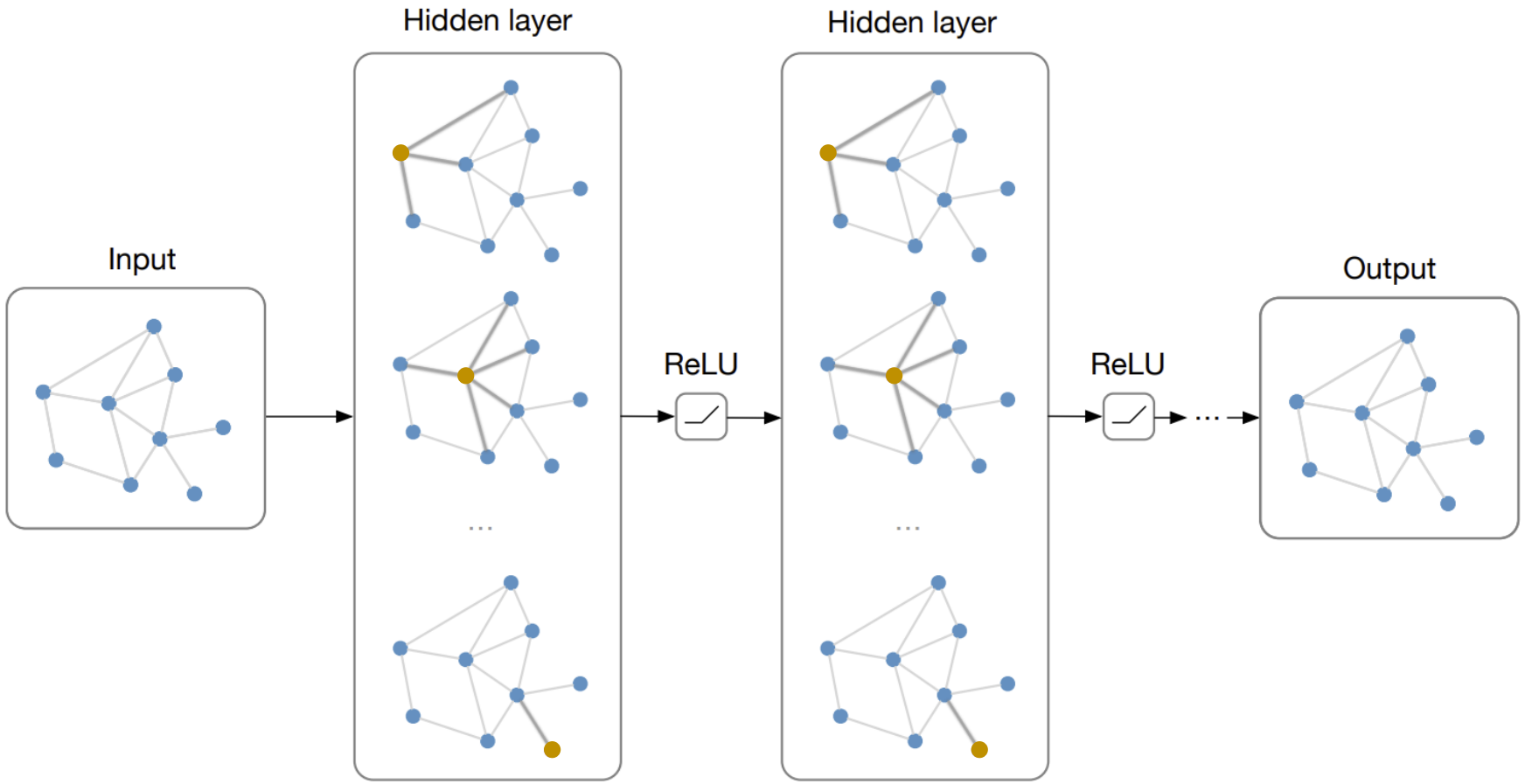


Image credit: Science

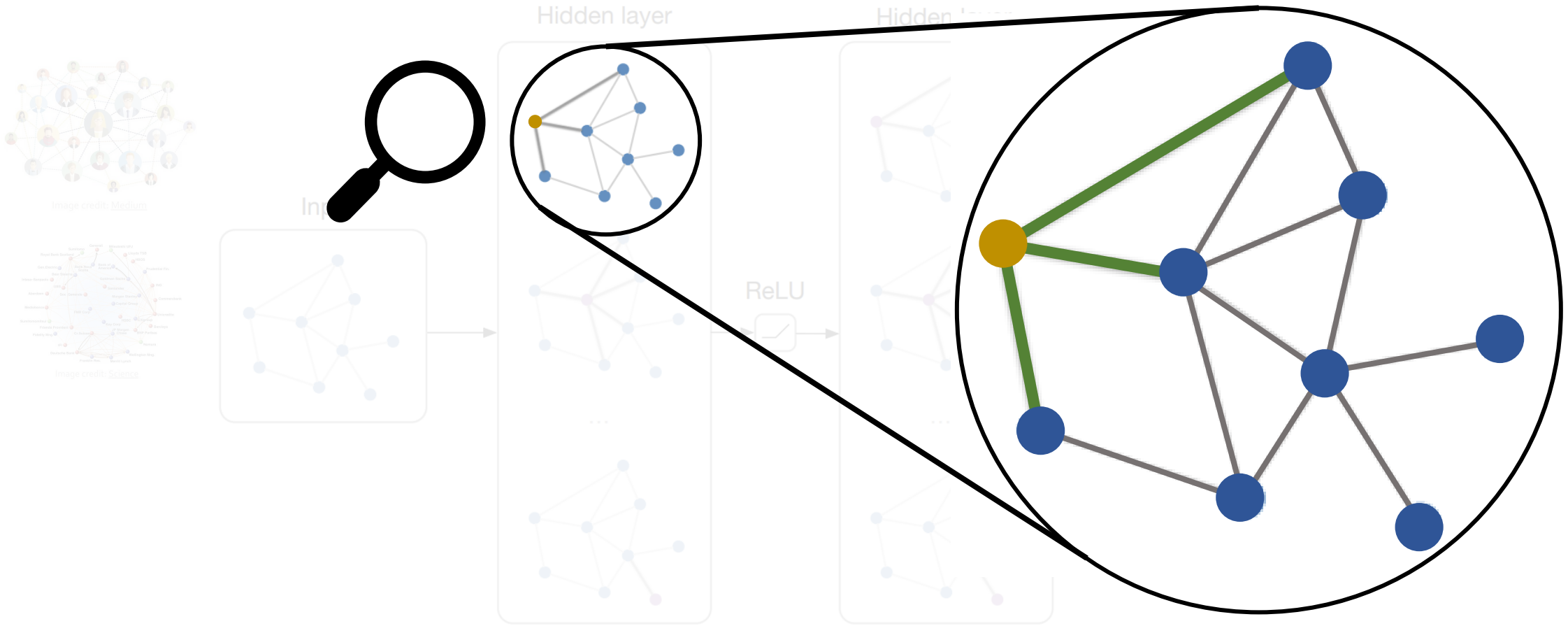




[Slide credit: Structured deep models: Deep learning on graphs and beyond]

What are Graph Neural Networks (GNNs)?

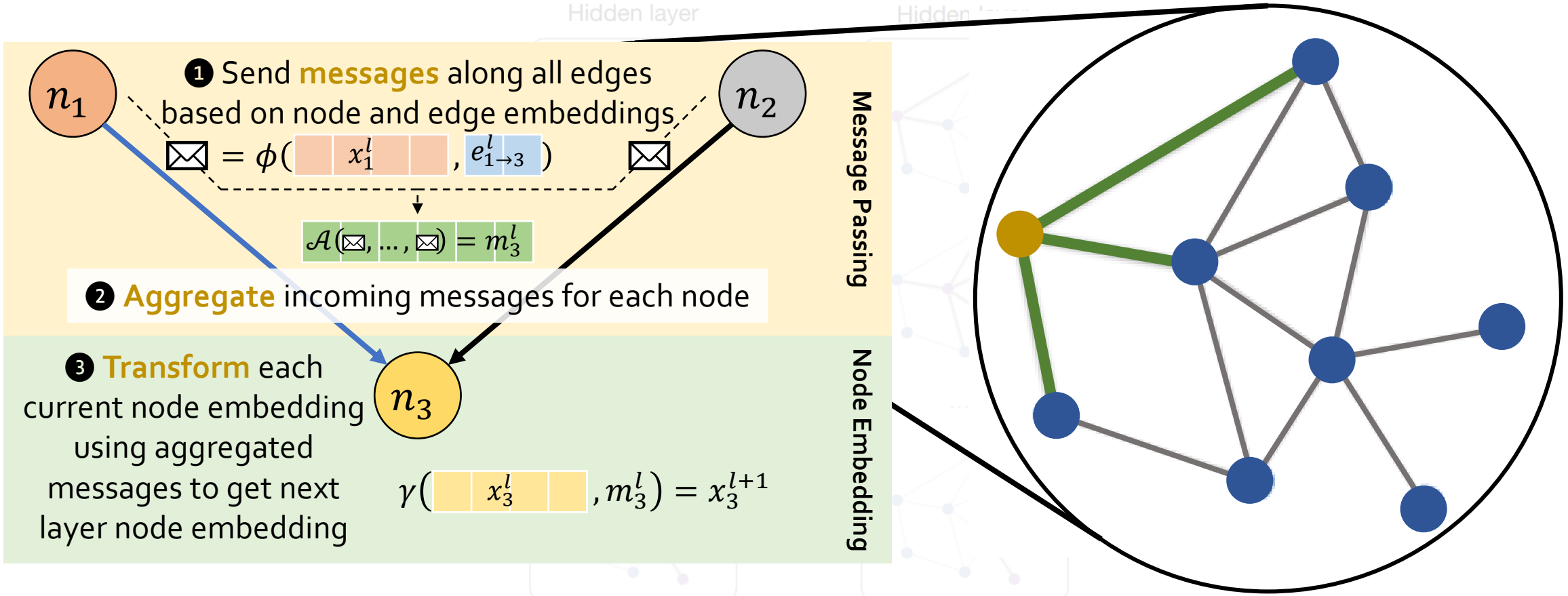
- **Mainstream:** Pass **messages** between pairs of nodes, **aggregate**, and **transform**



[Slide credit: Structured deep models: Deep learning on graphs and beyond]

What are Graph Neural Networks (GNNs)?

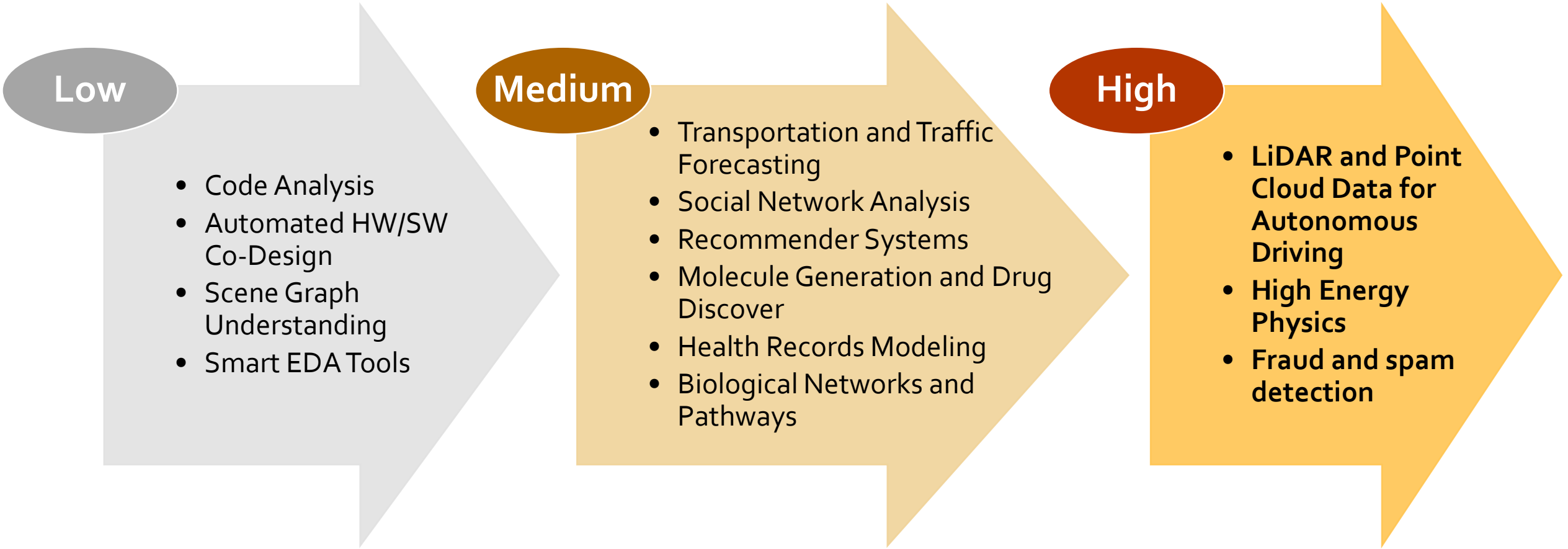
- **Mainstream:** Pass **messages** between pairs of nodes, **aggregate**, and **transform**



[Slide credit: Structured deep models: Deep learning on graphs and beyond]

Why Accelerate GNNs?

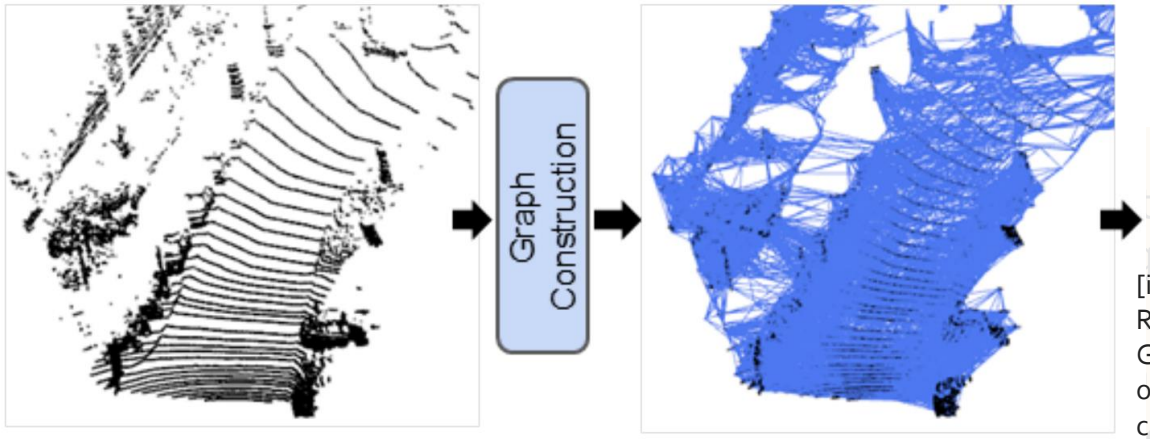
- **Numerous applications; many require real-time processing**



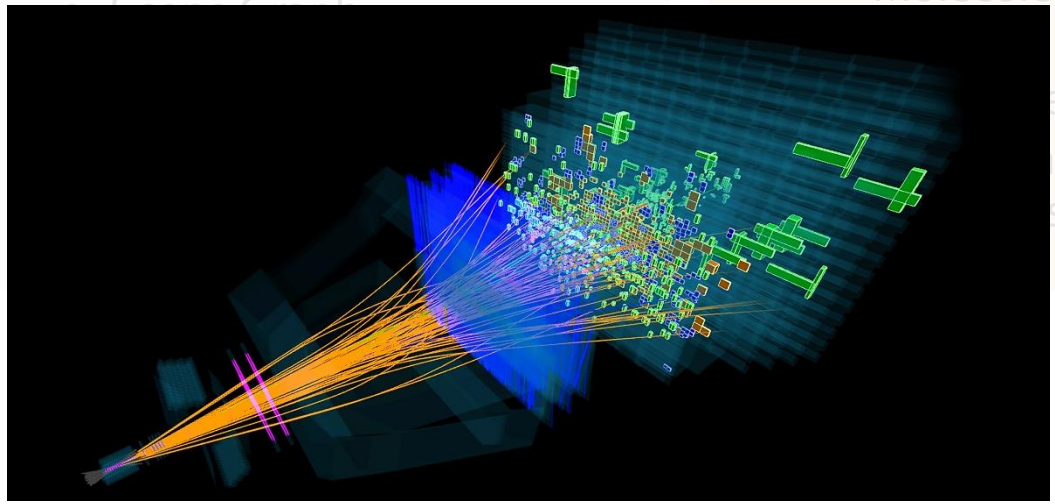
Zhou, Jie, et al. "Graph neural networks: A review of methods and applications." *AI Open*, 2020

Why Accelerate GNNs?

- **Numerous applications; many require real-time processing**



[image source] Shi, Weijing, and Raj Rajkumar. "Point-gnn: Graph neural network for 3d object detection in a point cloud." CVPR 2020



[image source] https://www.quantamagazine.org/growing-anomalies-at-the-large-hadron-collider-hint-at-new-particles-20200526/

High

- **LiDAR and Point Cloud Data for Autonomous Driving**
- **High Energy Physics**
- **Fraud and spam detection**

Outline

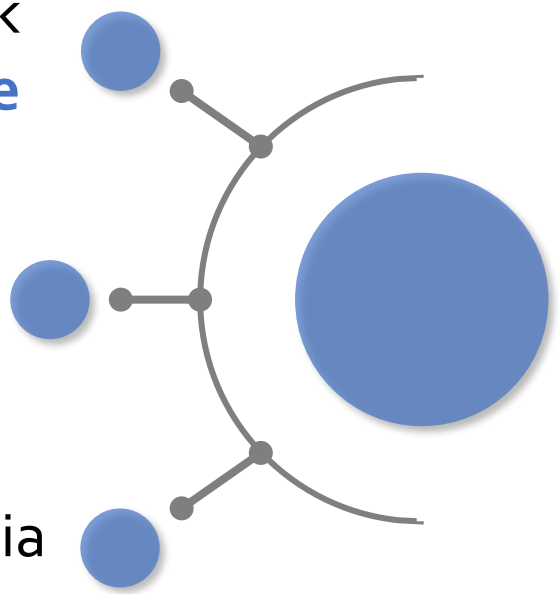
- Background and Motivation
  - What are Graph Neural Networks (GNNs)?
  - Why accelerate GNNs?
- **Existing Accelerator Limitations**
  - Not generic, not real-time
- Generic and Parallelized GNN Accelerator
  - Generic message passing framework — GenGNN
  - Node-level and edge-level parallelism
- Evaluation: CPU/GPU

Existing Work

Most focus on Graph Convolution Network (GCN): A **limited type**

Heavy **pre-processing**: Not suitable for real-time

Most on ASIC via **simulation**: not end-to-end, far from practical

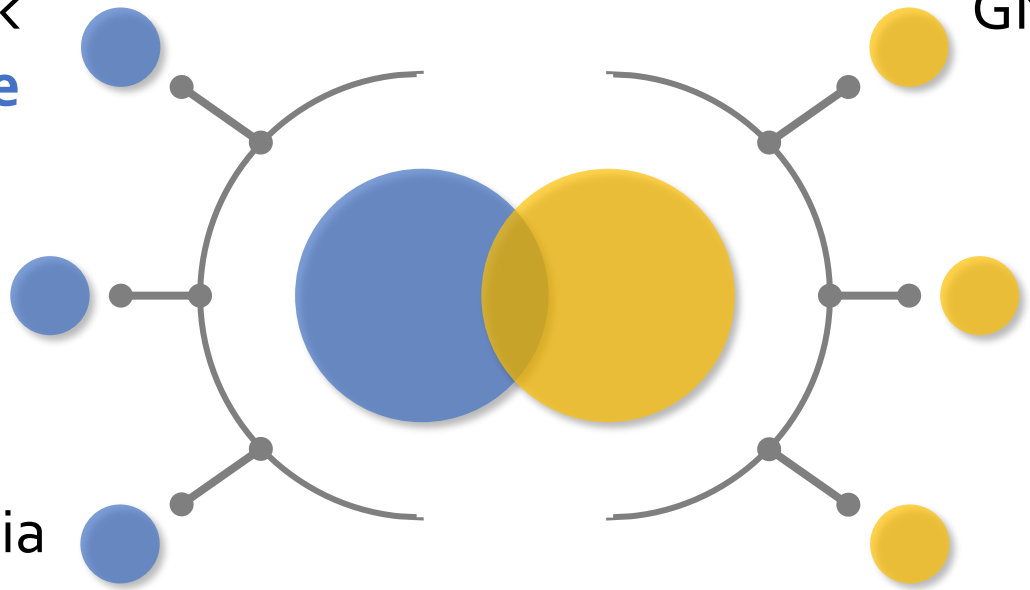


Existing Work vs. Ours

Most focus on Graph Convolution Network (GCN): A **limited type**

Heavy **pre-processing**: Not suitable for real-time

Most on ASIC via **simulation**: not end-to-end, far from practical



Support a wide range of GNNs: **Generic**

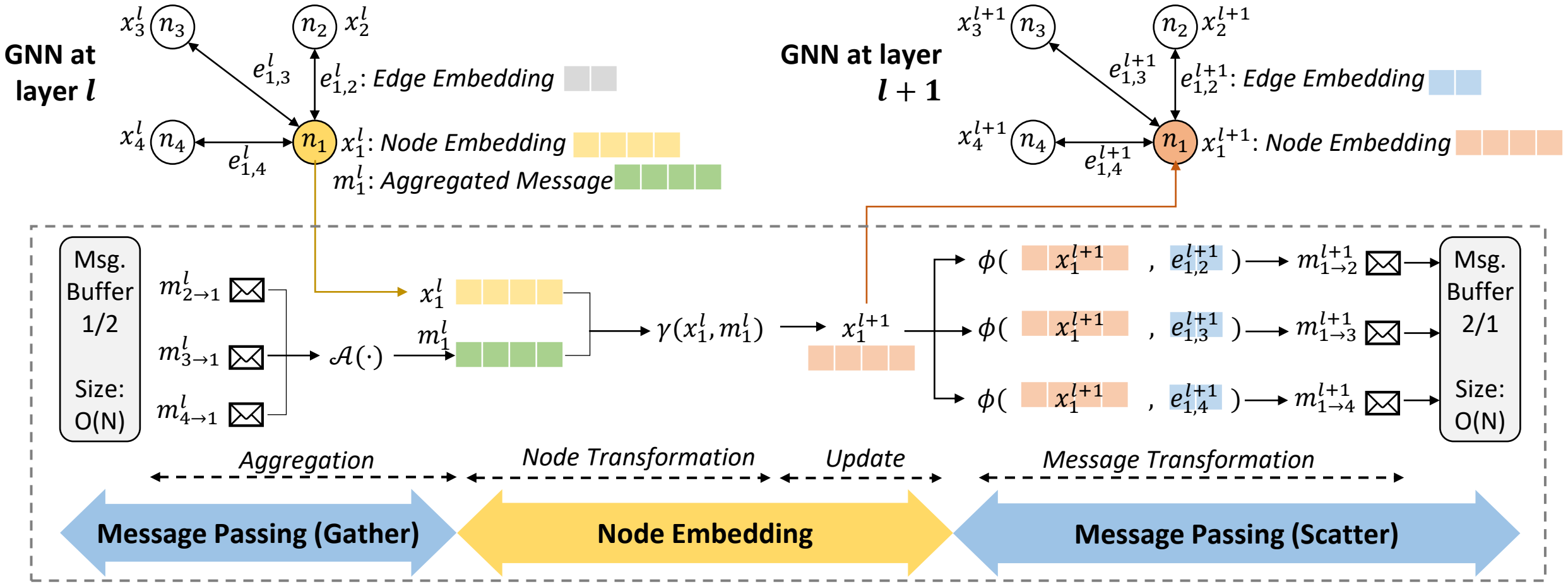
No pre-processing: **Real-time** oriented

Evaluated **end-to-end** on FPGA

Outline

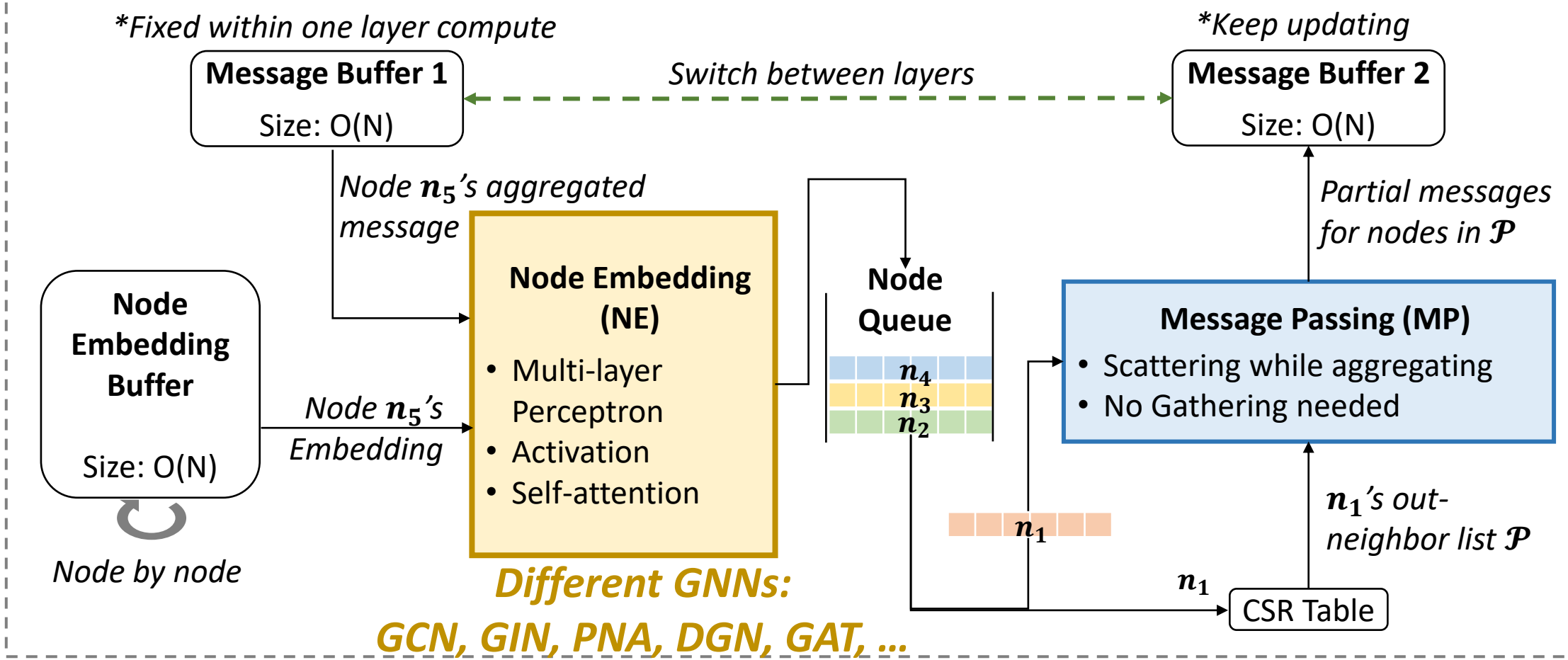
- Background and Motivation
  - What are Graph Neural Networks (GNNs)?
  - Why accelerate GNNs?
- Existing Accelerator Limitations
  - Not generic, not real-time
- **Generic and Parallelized GNN Accelerator**
  - Generic message passing framework — GenGNN
  - Node-level and edge-level parallelism
- Evaluation: CPU/GPU

Message Passing in GNNs

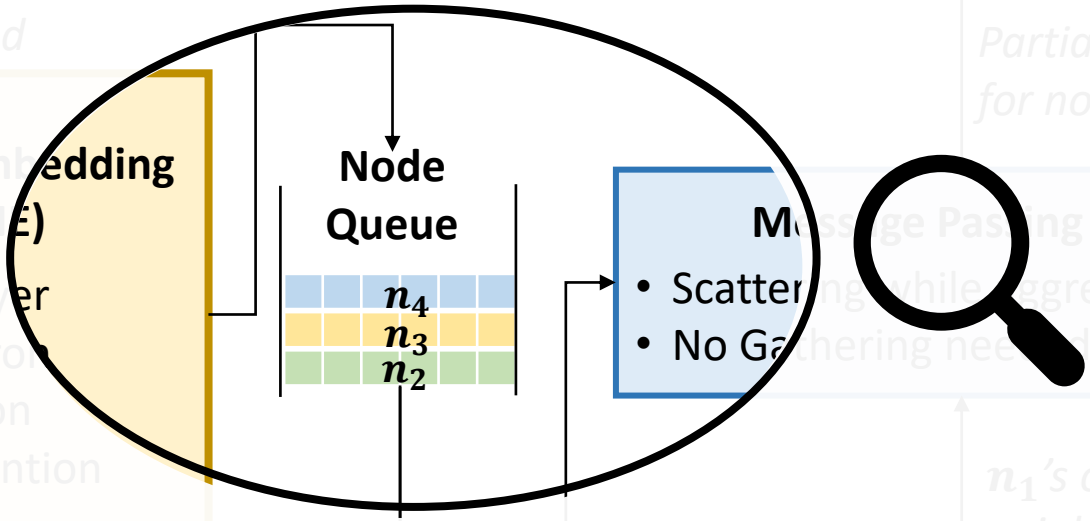


Architecture to Support Message Passing

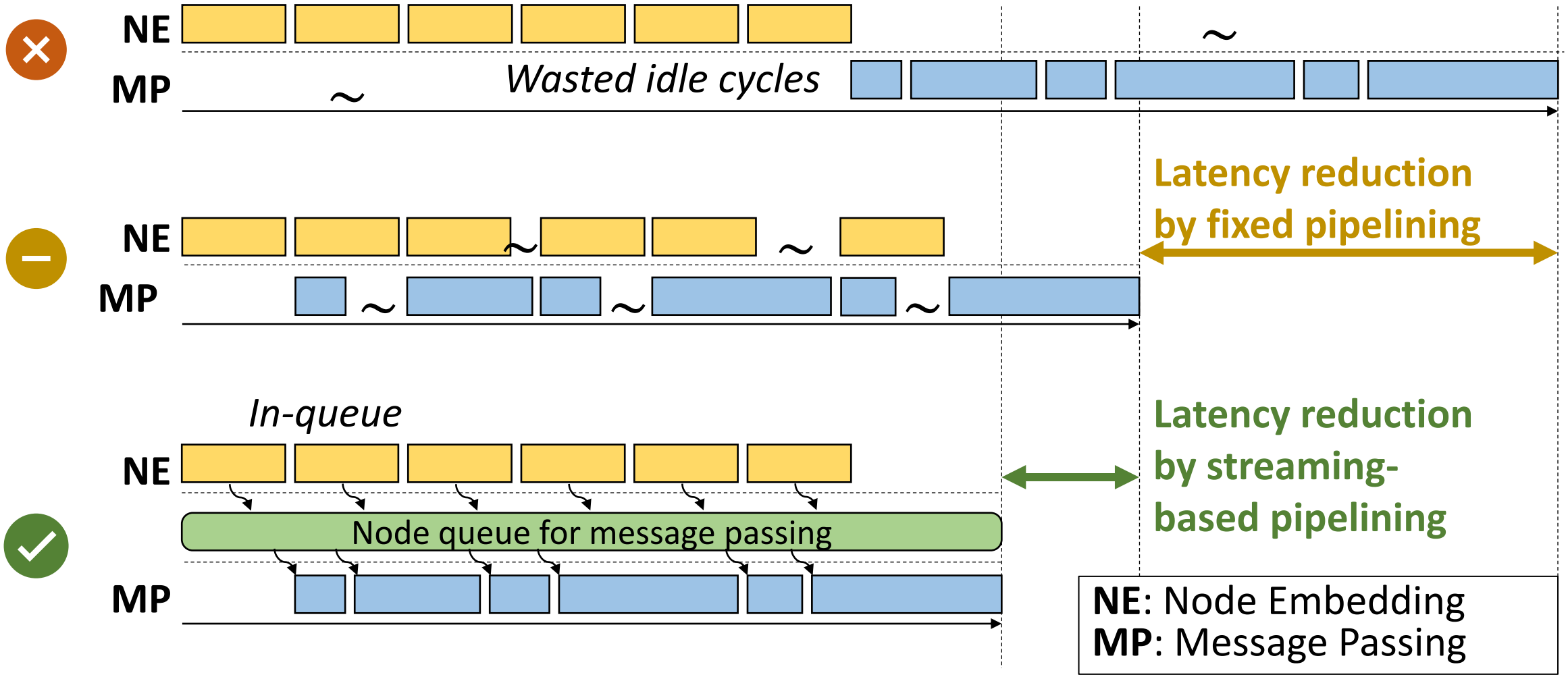
Overall Architecture of GenGNN (layer-recursive)



Streaming-based Pipelining



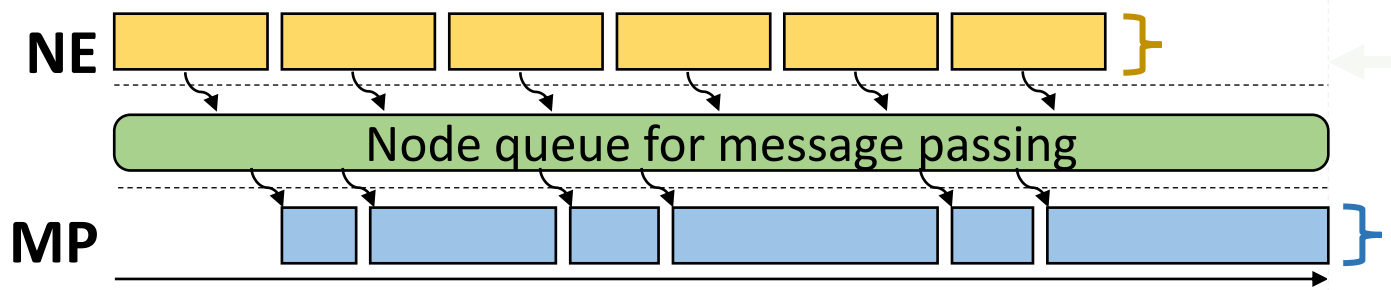
Streaming-based Pipelining



GenGNN Limitation

NE stage processes nodes **one at a time**

MP stage processes edges **one at a time**

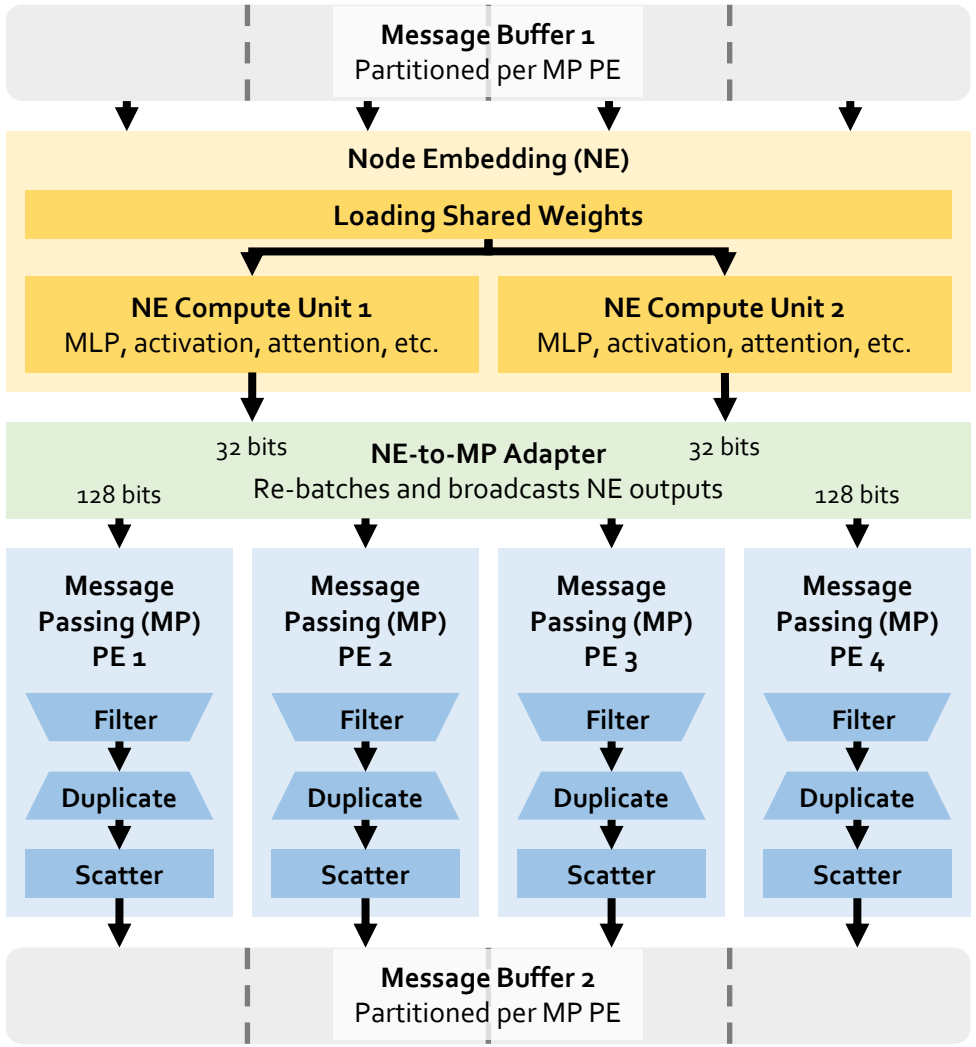


NE: Node Embedding
MP: Message Passing

Outline

- Background and Motivation
  - What are Graph Neural Networks (GNNs)?
  - Why accelerate GNNs?
- Existing Accelerator Limitations
  - Not generic, not real-time
- **Generic and Parallelized GNN Accelerator**
  - Generic message passing framework — GenGNN
  - Node-level and edge-level parallelism
- Evaluation: CPU/GPU

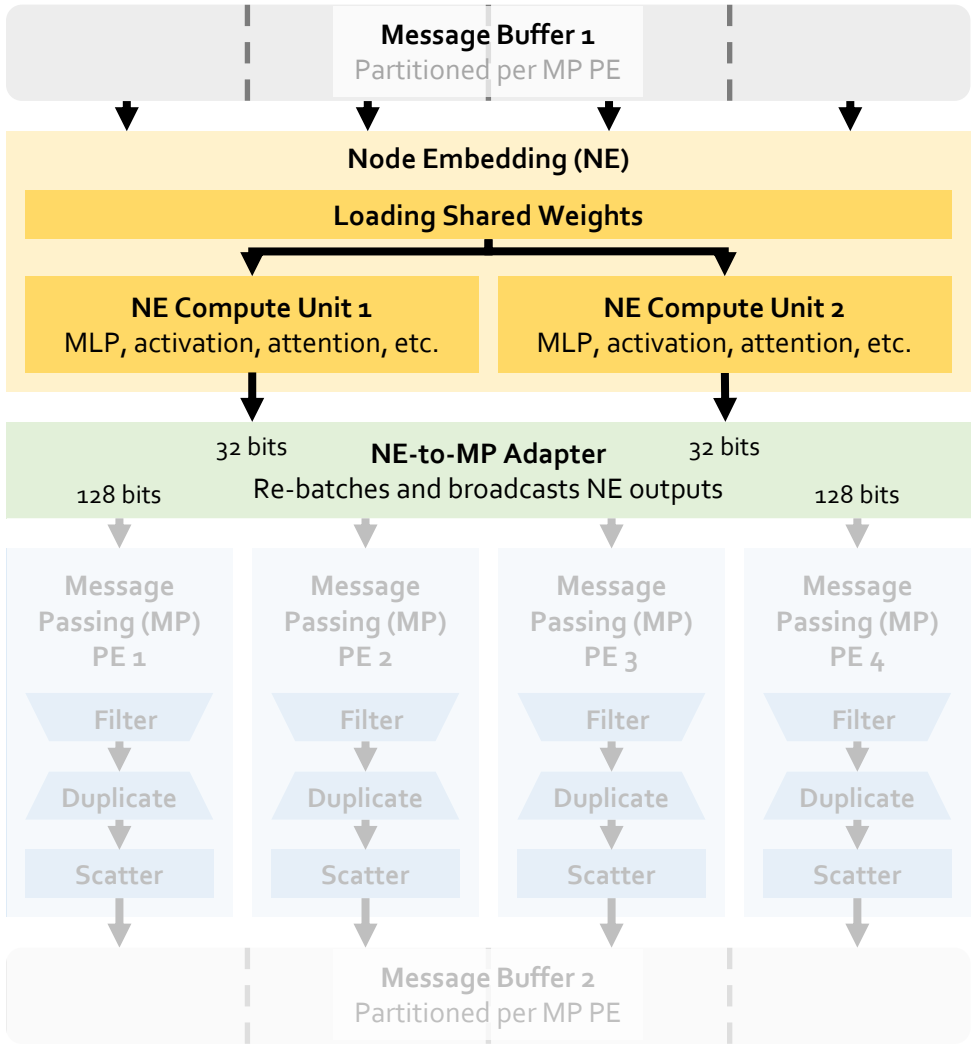
Parallelized Architecture



Four key components:

- **Partitioned** message buffers
- **Multi-node** NE PE
- NE-to-MP **adapter**
- **Independent** MP PEs

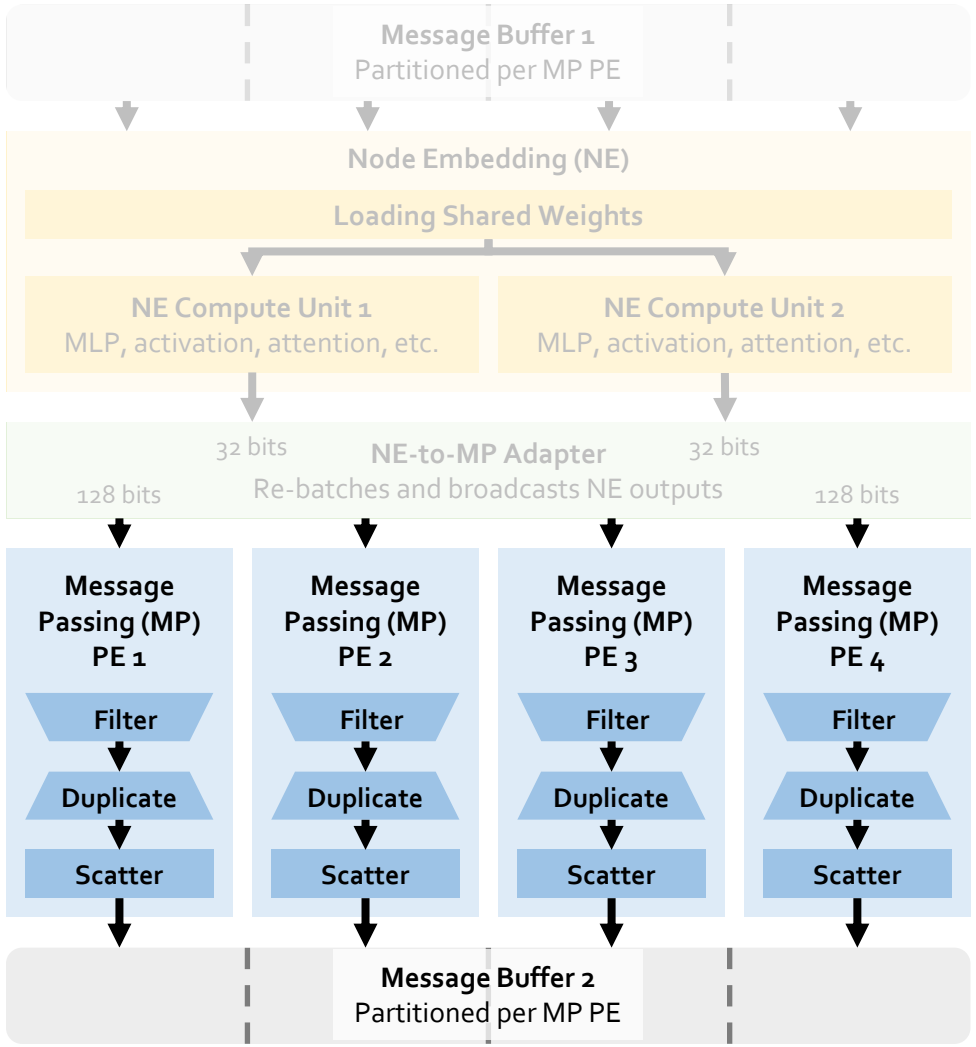
Node-Level Parallelism



Four key components:

- Partitioned message buffers
- **Multi-node** NE PE
  - Shares weights between multiple nodes
- NE-to-MP **adapter**
  - Broadcasts node embeddings
- Independent MP PEs

Edge-Level Parallelism

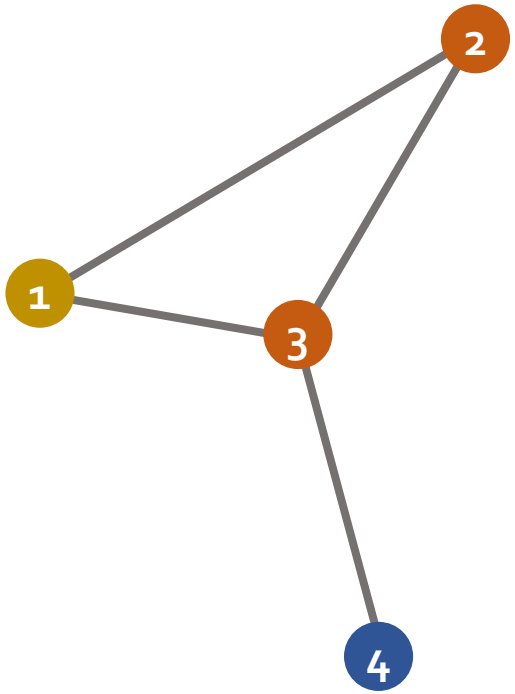


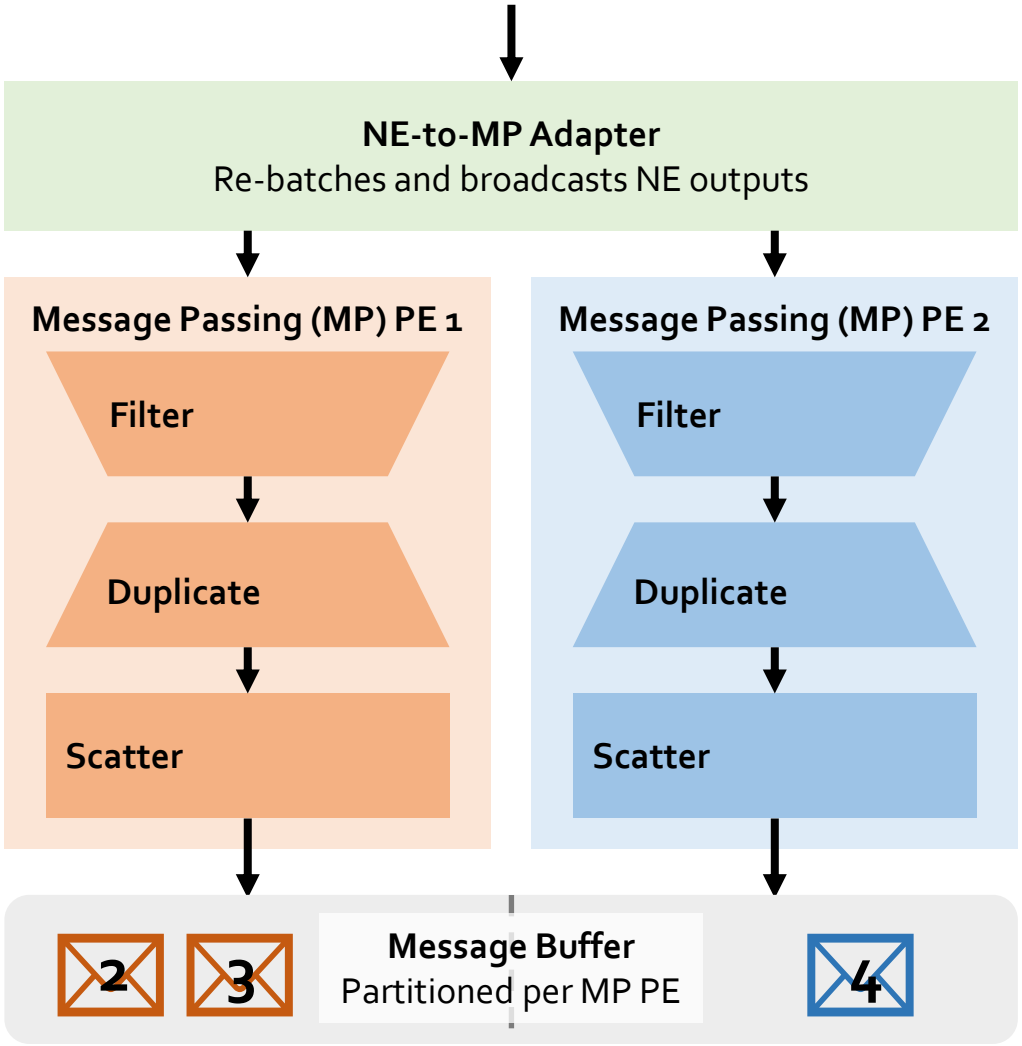
Four key components:

- **Partitioned** message buffers
- **Multi-node** NE PE
- NE-to-MP **adapter**
- **Independent** MP PEs
  - Each handles edges to a subset of nodes
    - PEs never need to access same partition of message buffer
  - Filter → duplicate → scatter dataflow makes loops predictable, minimizing overhead

Filter → Duplicate → Scatter Dataflow

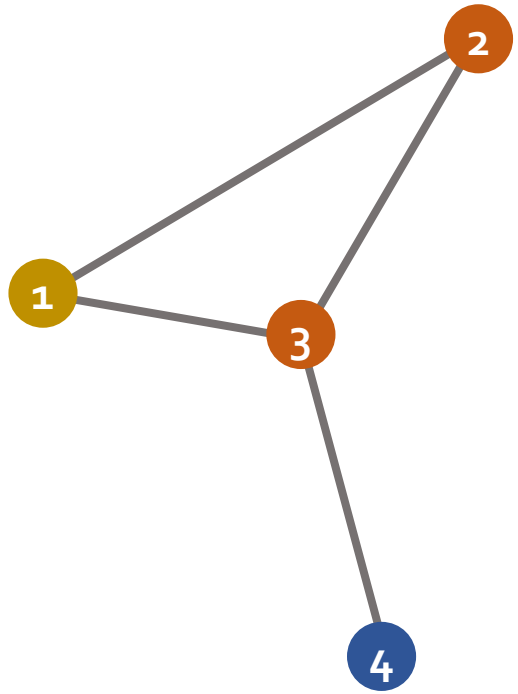
- **Example: two message passing PEs**
  - PE 1 handles red nodes
  - PE 2 handles blue node
- **We'll follow node 1's embedding**

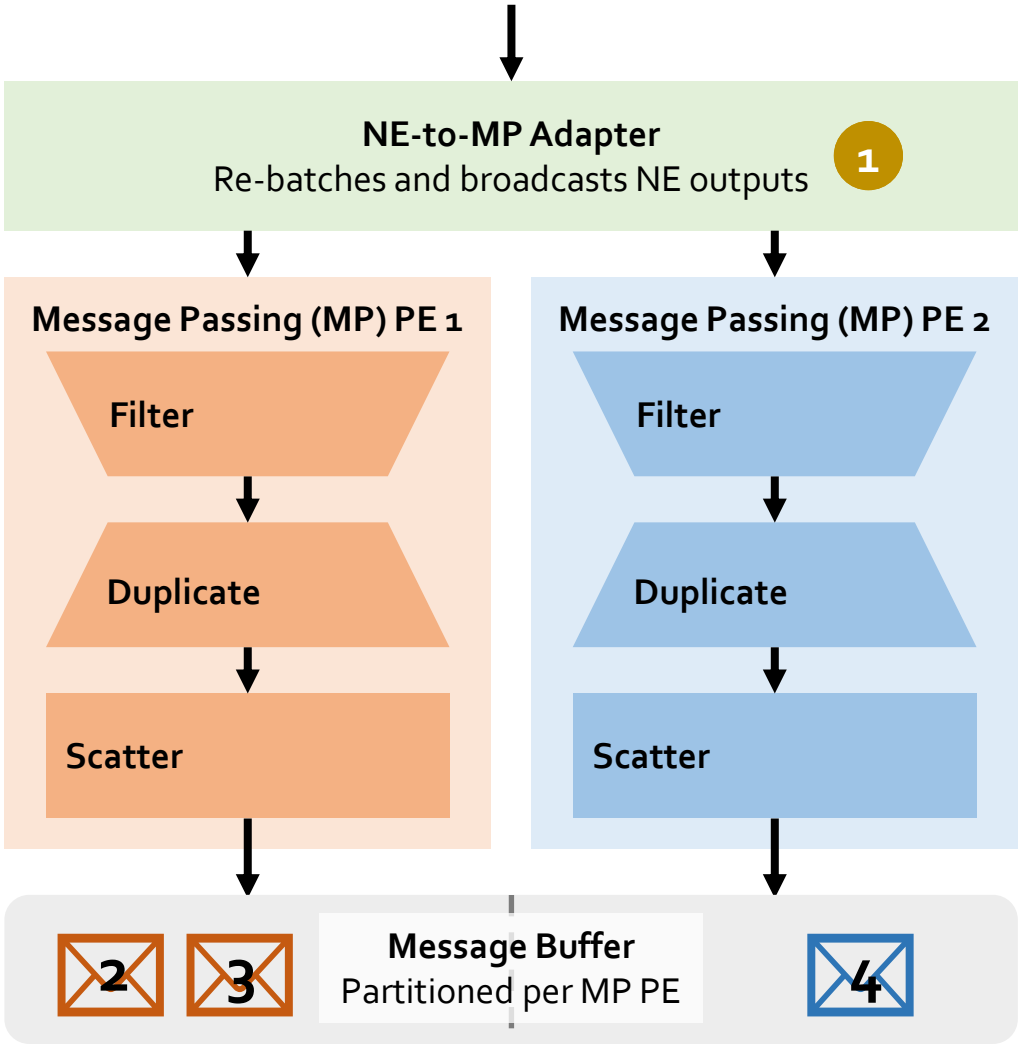




Filter → Duplicate → Scatter Dataflow

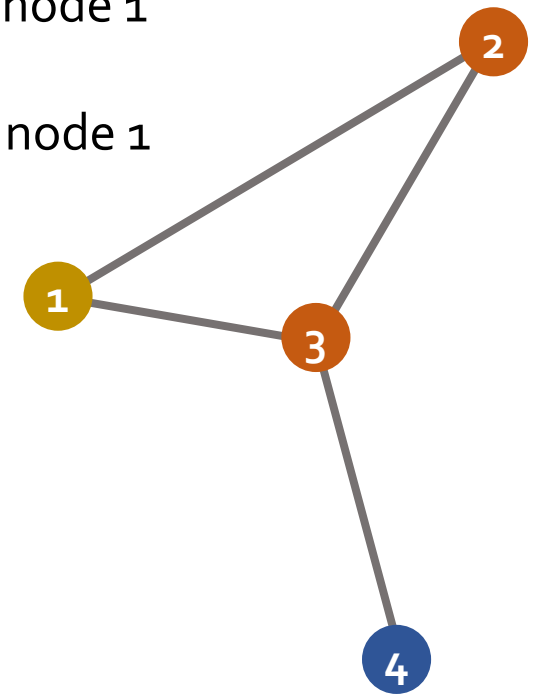
- Adapter accepts node embedding and broadcasts to all MP PEs

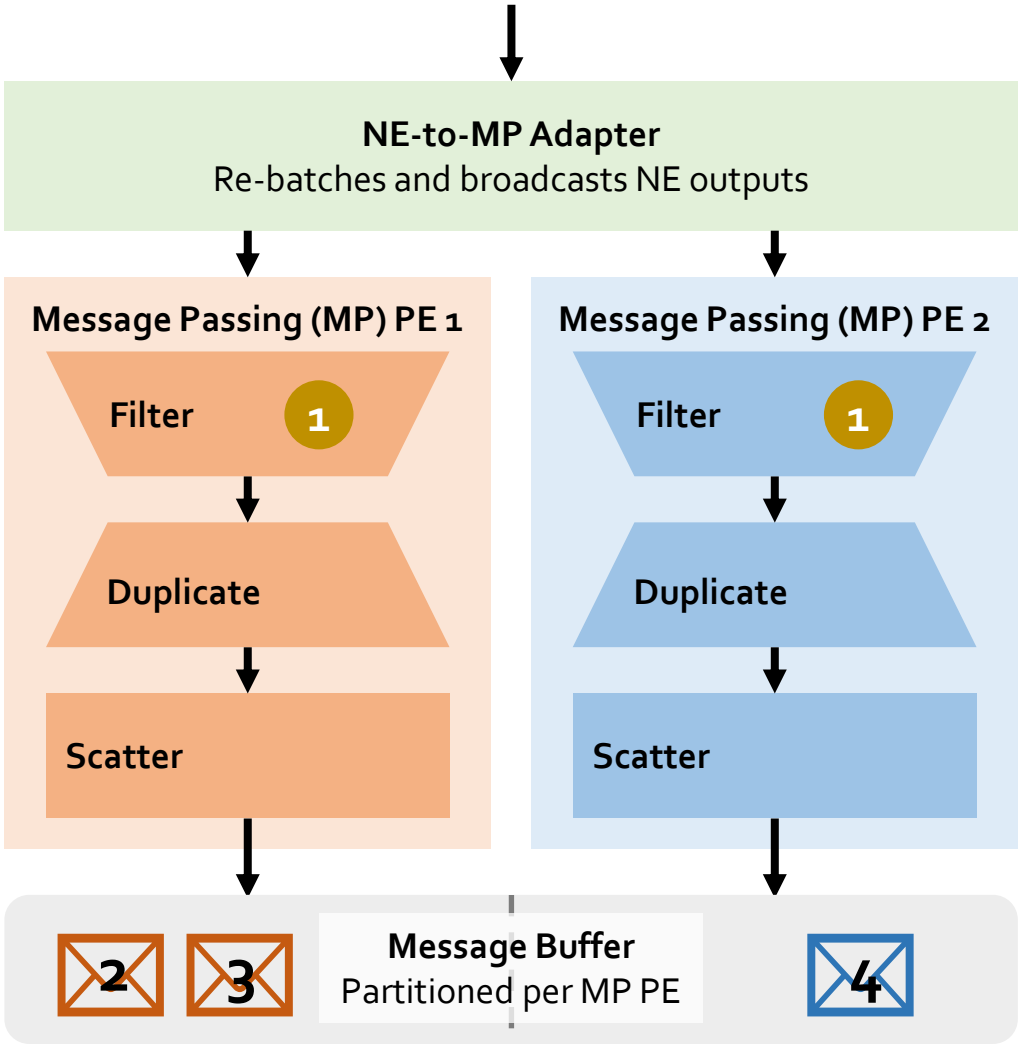




Filter → Duplicate → Scatter Dataflow

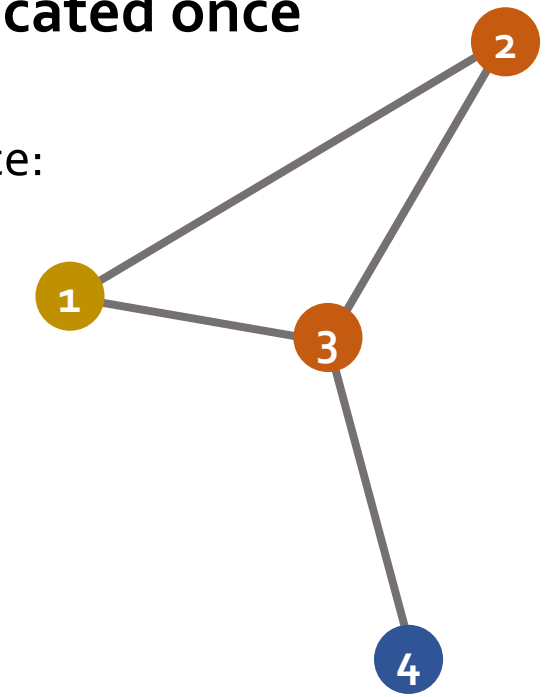
- **Adapter accepts node embedding and broadcasts to all MP PEs**
- **Each PE checks if it needs this node embedding**
  - PE 1 checks for edges from node 1 to a red node
  - PE 2 checks for edges from node 1 to a blue node

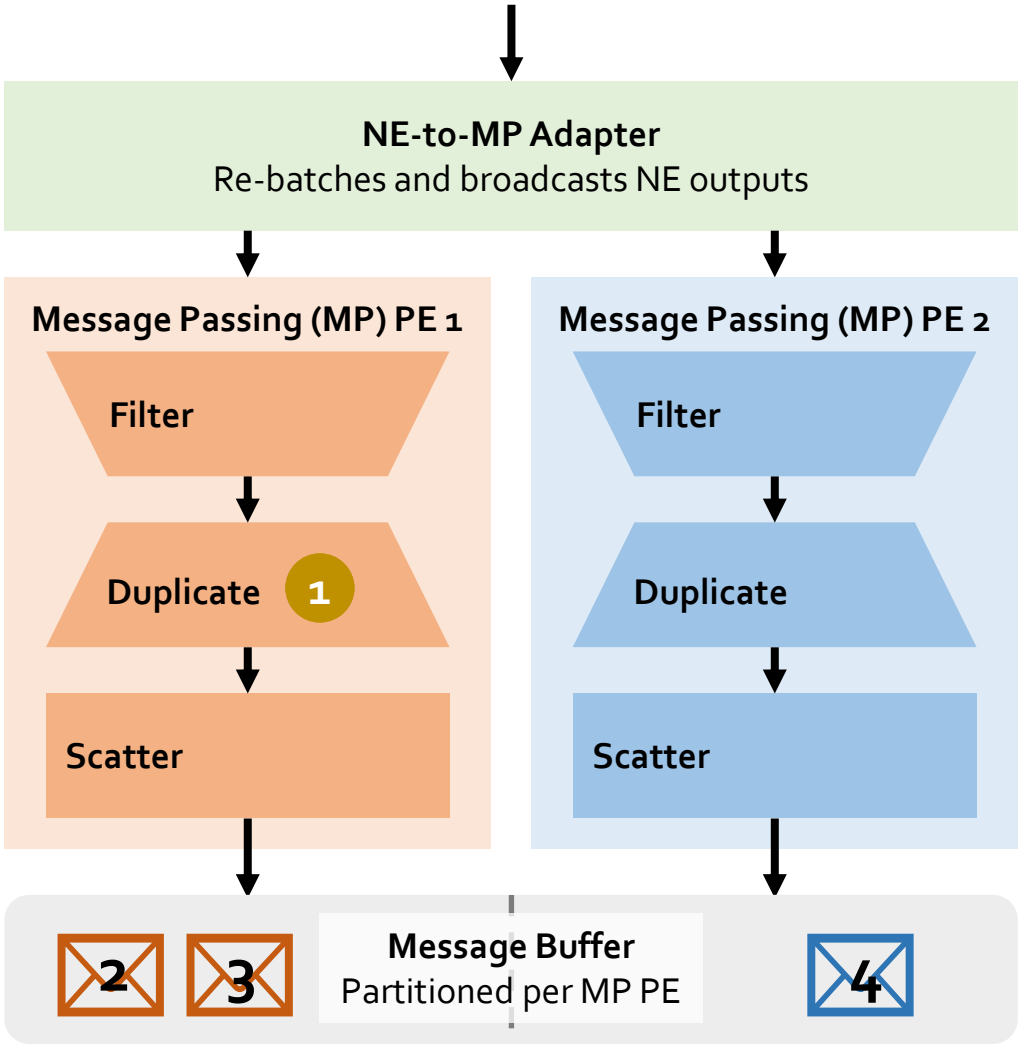




Filter → Duplicate → Scatter Dataflow

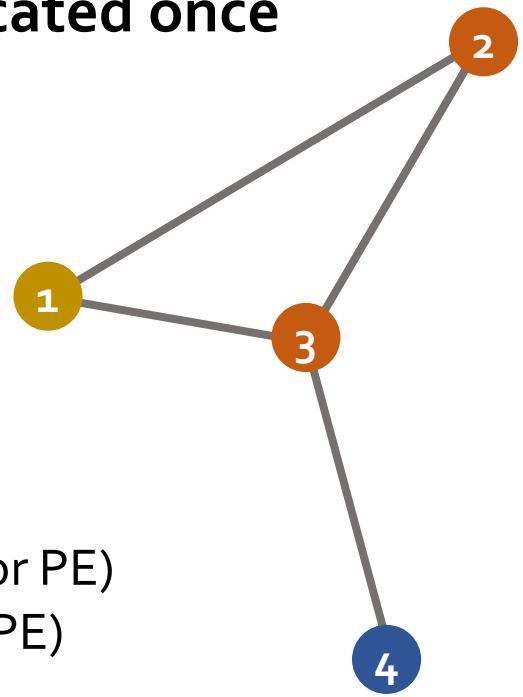
- **Adapter accepts node embedding and broadcasts to all MP PEs**
- **Each PE checks if it needs this node embedding**
- **Node embedding is duplicated once for each edge to scatter**
  - PE 1 duplicates node 1 twice:
    - Once for 1 → 2
    - Once for 1 → 3





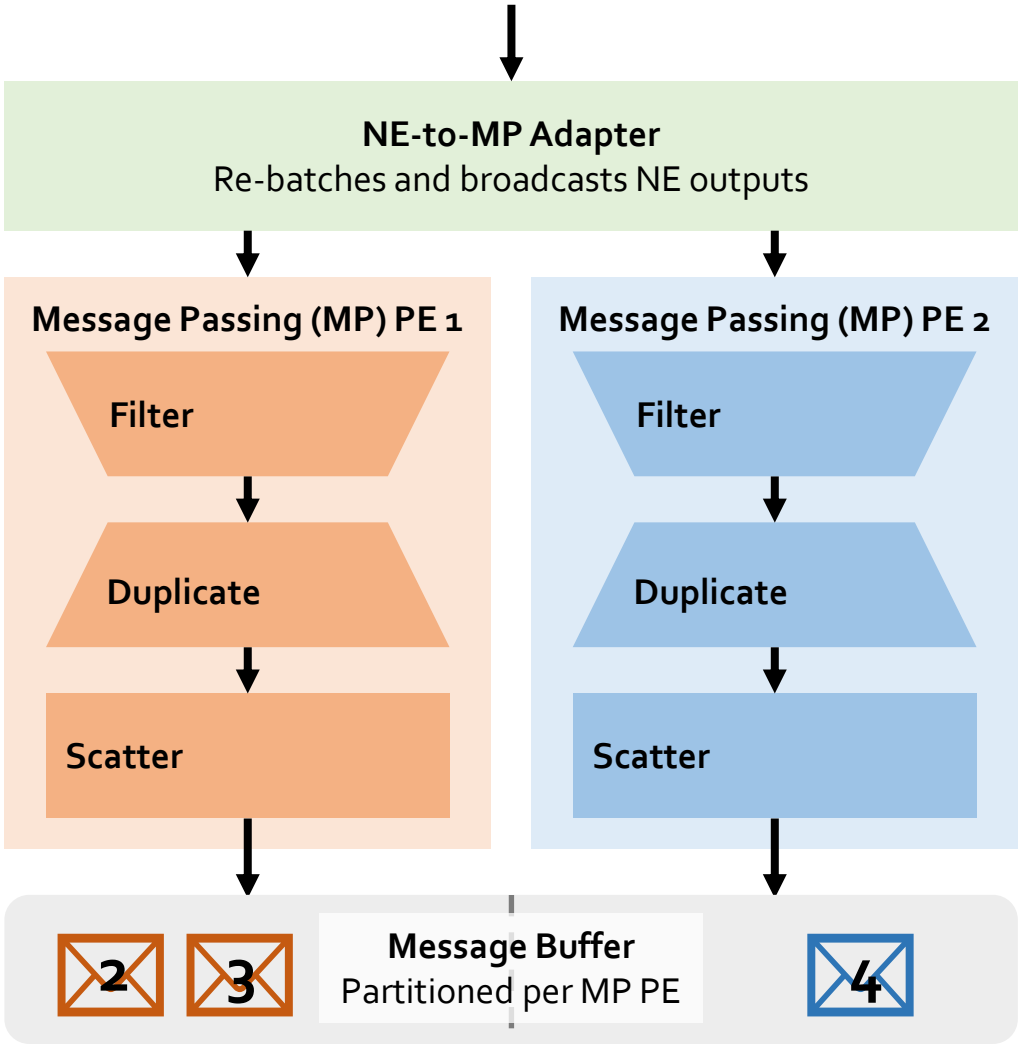
Filter → Duplicate → Scatter Dataflow

- Adapter accepts node embedding and broadcasts to all MP PEs
- Each PE checks if it needs this node embedding
- Node embedding is duplicated once for each edge to scatter



Predictable loop behavior
→ minimal loop overhead

- Filter: once per node
- Duplicate: once per edge (for PE)
- Scatter: once per edge (for PE)



Outline

- Background and Motivation
  - What are Graph Neural Networks (GNNs)?
  - Why accelerating GNNs?
- Existing Accelerator Limitations
  - Not generic, not real-time
- Ours: Generic GNN Accelerator – GenGNN
  - Generic message passing framework
  - Model-specific components
- **Evaluation: CPU/GPU**

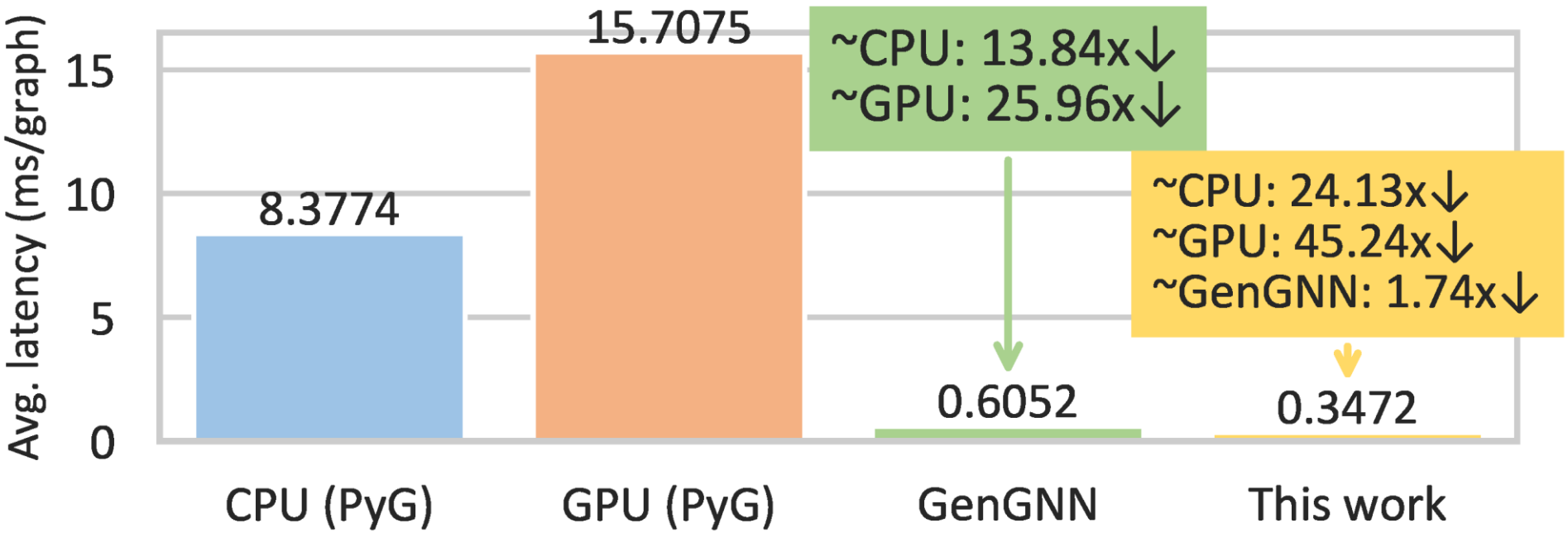
Experiment Setup

- **Representative model: Directional Graph Network (DGN)**
- **Xilinx Alveo U50 Acceleration Board**
- **Dataset: MolHIV (molecule classification)**; 4k graphs

Compute Resources	
Look-up Tables (LUTs)	872K
Registers	1,743K
DSP Slices	5,952

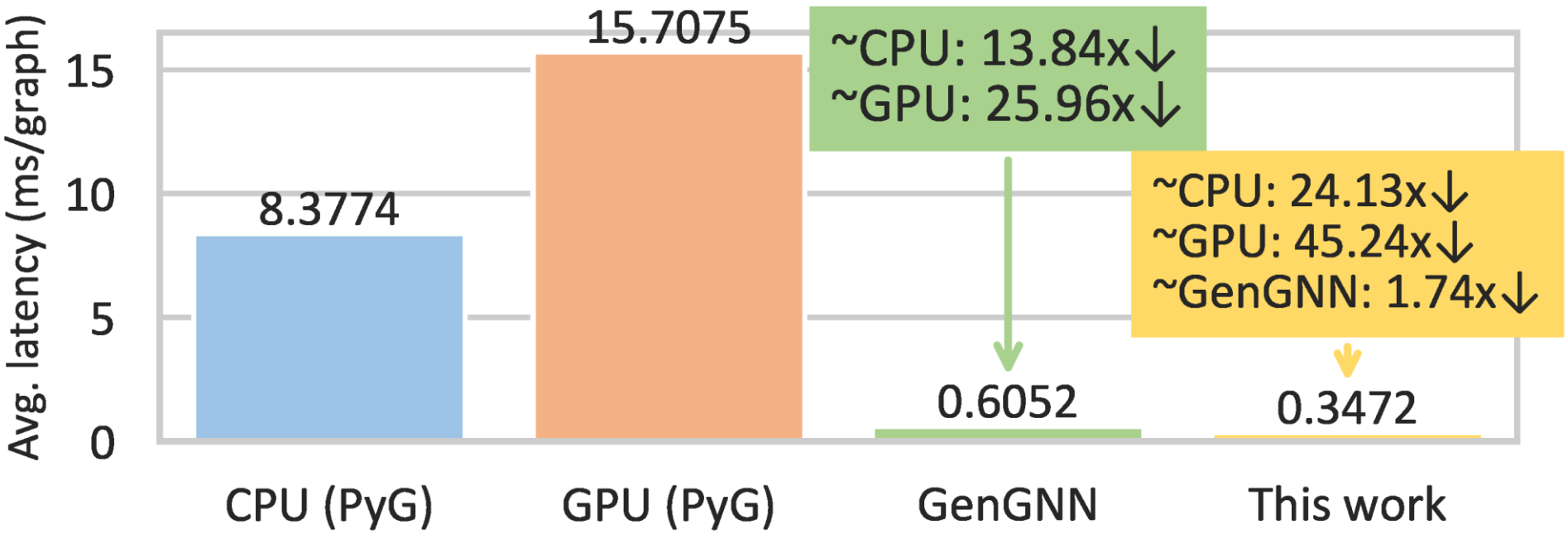
Evaluation

- **Baseline: CPU/GPU (batchsize = 1)**



Evaluation

- **Baseline: CPU/GPU (batchsize = 1)**



- **Primarily limited by kernel start/stop overhead (93% of inference latency)**
  - Comparing only time spent on computation, this work achieves nearly **12x** speedup over GenGNN

Future Directions

- **Reducing overhead for higher throughput**
  - Opportunity for up to 14x speedup
- **Design automation and design space exploration**
  - E.g., automatically determine best number of PEs for a dataset
- **Optimization for large graphs**
  - Enable ultra-fast inference for graphs that do not fit on-chip

Summary & Thanks!

- **Graph Neural Networks (GNNs) require acceleration and real-time processing**
  - Not all GNNs are sparse matrix multiplications (only few of them are)
- **Our contribution: a generic and parallelized GNN acceleration framework on FPGA**
  - Generic: supports a wide range of GNN models
  - Real-time: no pre-processing
  - Verified: evaluated end-to-end on FPGA
- **Beats GPU, CPU, and our previous work, GenGNN**
  - 24x speedup over CPU
  - 45x speedup over GPU
  - 1.7x speedup over GenGNN

Contact:
rishov.sarkar@gatech.edu
Sharc Lab @ Georgia Tech
(https://sharclab.ece.gatech.edu/)