G-CoS: GNN-Accelerator Co-Search Towards Both Better Accuracy and Efficiency

Yongan Zhang, Haoran You, Yonggan Fu, Tong Geng, Ang Li, Yingyan Lin

Rice University, Georgia Institute of Technology and PNNL

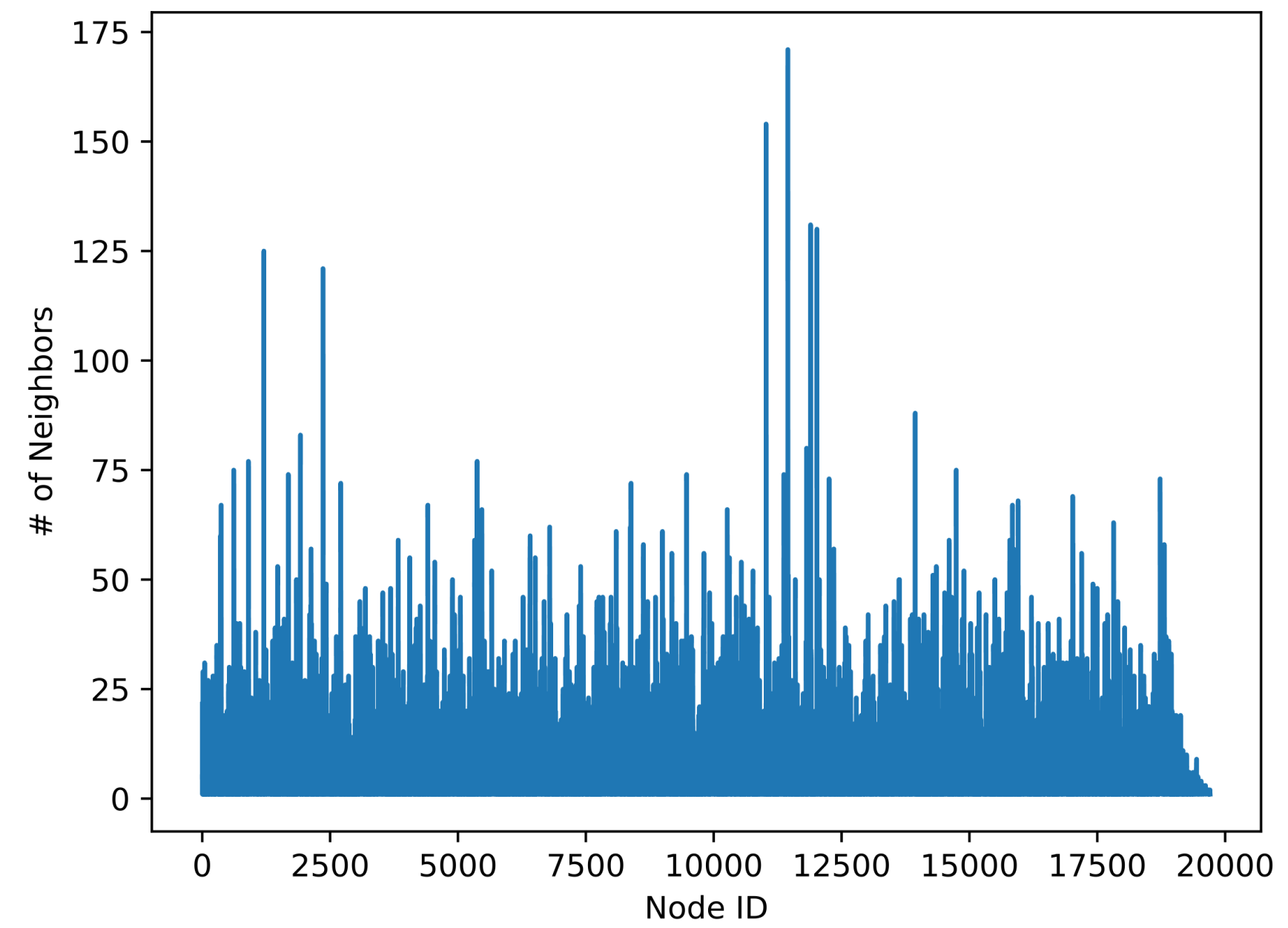
EFFICIENT AND INTELLIGENT COMPUTING

Georgia Tech College of Computing
Center for Research into Novel Computing Hierarchies

Accepted by ICCAD 2021

Background & Motivation

- Prohibitively **large number of nodes** and complex connections
  - Reddit post dataset: **232,965** nodes and ~**50** neighbors per node
- **Unbalanced** and **irregular** connections among the nodes

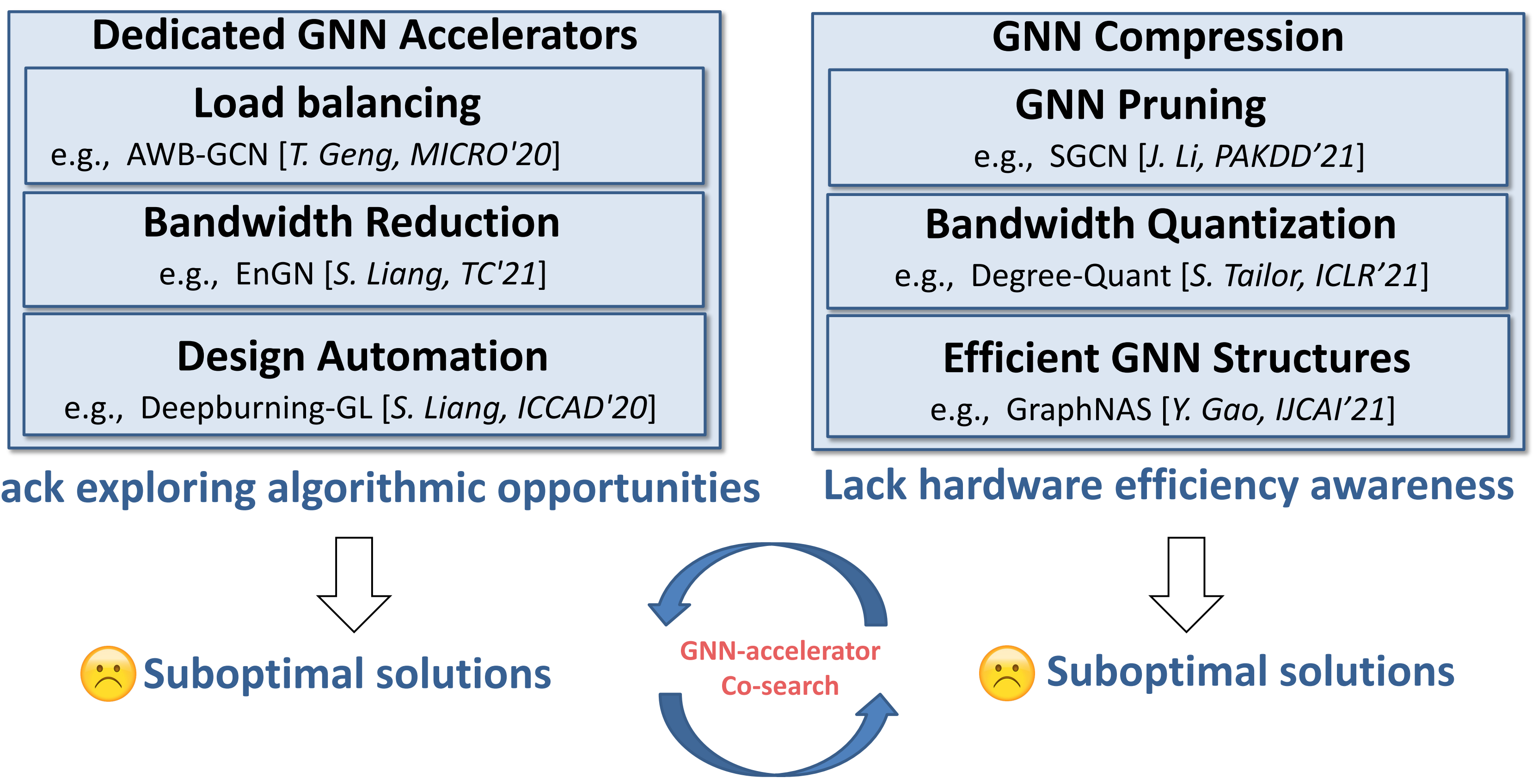


Distinct number of neighbors for each node in Pubmed dataset

- **High dimension** of GNNs' node feature vectors
  - CiteSeer dataset: **3703** features for each node

How to efficiently execute the GNN workloads?

Previous Works and Limitations



Dedicated GNN Accelerators

Load balancing
e.g., AWB-GCN [*T. Geng, MICRO'20*]

Bandwidth Reduction
e.g., EnGN [*S. Liang, TC'21*]

Design Automation
e.g., Deepburning-GL [*S. Liang, ICCAD'20*]

Lack exploring algorithmic opportunities

☹ **Suboptimal solutions**

GNN Compression

GNN Pruning
e.g., SGCN [*J. Li, PAKDD'21*]

Bandwidth Quantization
e.g., Degree-Quant [*S. Tailor, ICLR'21*]

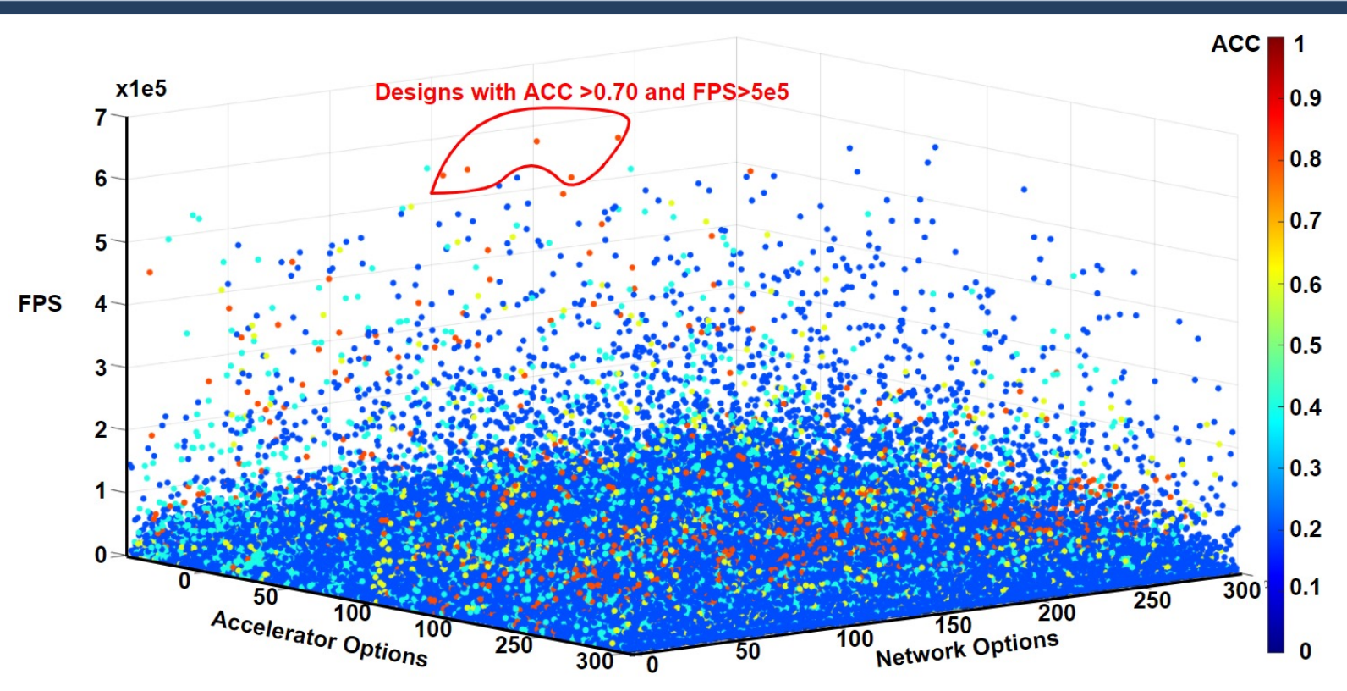
Efficient GNN Structures
e.g., GraphNAS [*Y. Gao, IJCAI'21*]

Lack hardware efficiency awareness

☹ **Suboptimal solutions**

GNN-accelerator Co-search

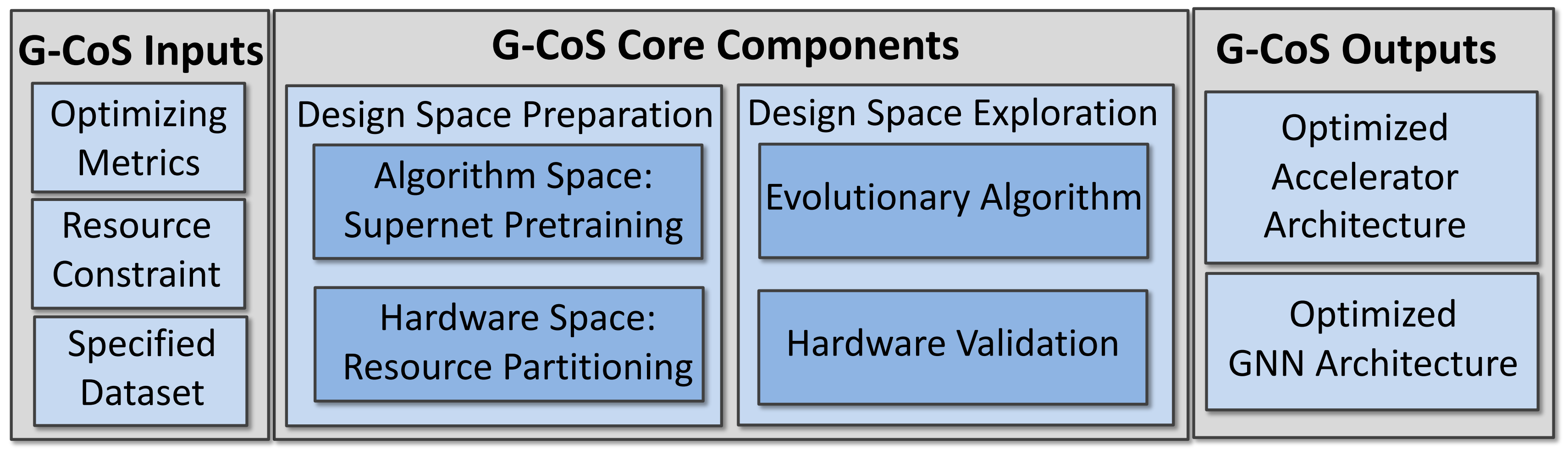
Co-search Challenges



An overview of the design space

- ➢***Joint***: Prohibitively large and sparse joint space
- ➢***Algorithm***: Excessive re-training cost during search
- ➢***Accelerator***: lack of generic accelerator space dedicated for GNNs

G-CoS: Overview



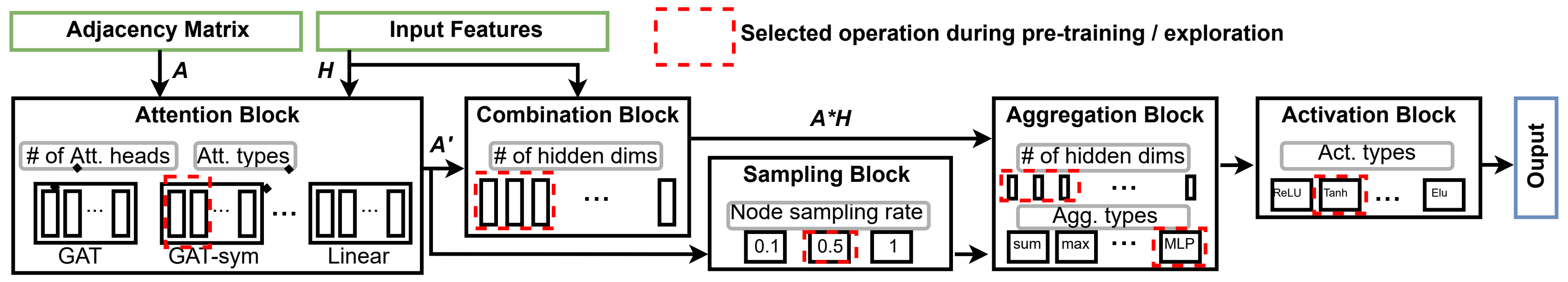
G-CoS Inputs: Optimizing Metrics; Resource Constraint; Specified Dataset

G-CoS Core Components: Design Space Preparation — Algorithm Space: Supernet Pretraining; Hardware Space: Resource Partitioning. Design Space Exploration — Evolutionary Algorithm; Hardware Validation

G-CoS Outputs: Optimized Accelerator Architecture; Optimized GNN Architecture

- ➢**G-CoS**: a **G**NN and accelerator **co-s**earch framework
  - ➢The first to **jointly** search for the matched GNN structures and accelerators
  - ➢Optimize both **task accuracy** and **acceleration efficiency**
- ➢**Enabler 1**: One-shot GNN and accelerator co-search algorithm
  - ➢Simultaneous and efficient search for both networks and accelerators
- ➢**Enabler 2**: Generic GNN structure and hardware accelerator space
  - ➢Great potential in boosting the accuracy and hardware efficiency

G-CoS: Algorithm Design Space

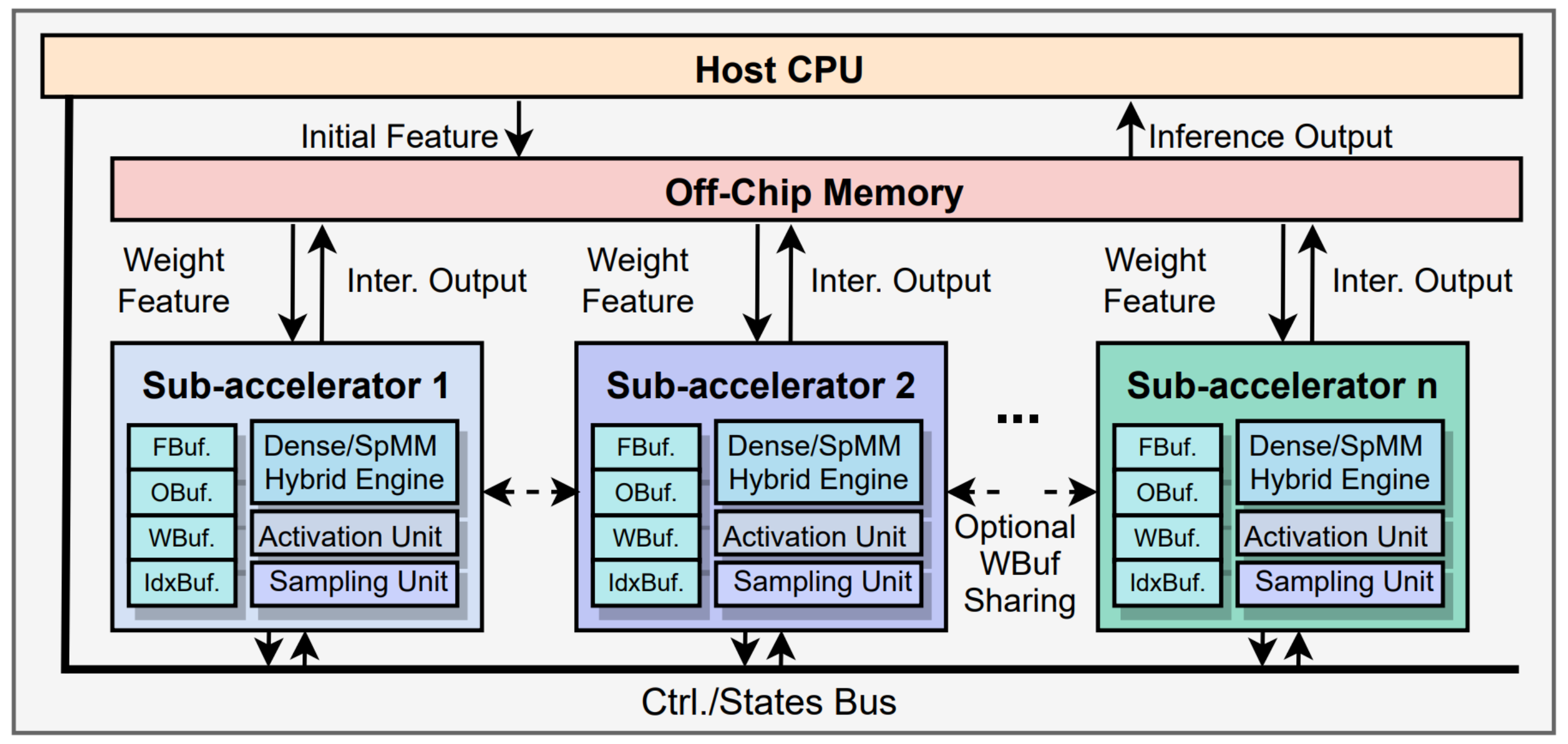


Attention types	[skip, GCN, GAT, GAT-sym, COS, Linear, Gene-Linear]
Aggregation types	[sum, mean, max]
Activation types	[skip, Sigmoid, Tanh, ReLu, Linear, Softplus, Leaky ReLU]
# of hidden dimensions	[4, 8 ,16, 32, 64, 128 ,256]
# of Attention heads	[1, 2, 4, 6, 8, 16]
Node sampling rate	[0.1, 0.5, 1]

- ➢Comprehensively cover the commonly used GNN structures
- ➢Candidate networks are sampled by choosing an option for each parameter
- ➢The space comprises more than 10^{19} network choices
  - ➢Leads to larger application **versatility and accuracy potential**

- ➢**Supernet pretraining** → **Better proxy accuracy**
  - ➢Uniform sampling + single path activation + Weight
- ➢**Evolutionary search algorithm** → **Improved search efficiency**
  - ➢One-shot network and accelerator search

G-CoS: Accelerator Design Space



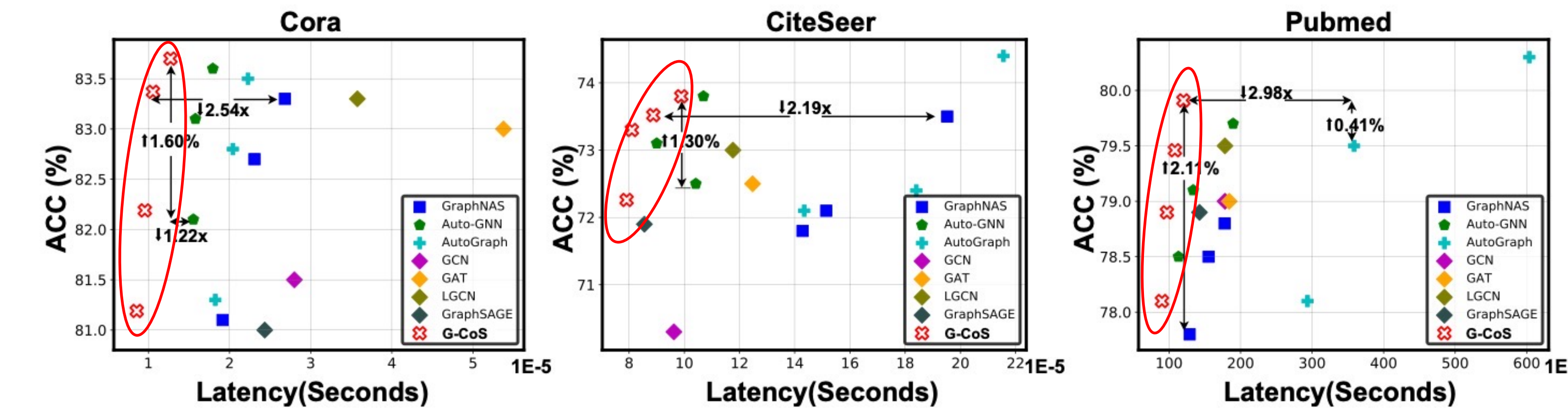
- ➢Multiple individually customizable sub-accelerators
  - ➢Both **latency and utilization friendly**
- ➢Different weights'/features' regions → Different workloads
- ➢Resource partition ← → Workload sizes
  - ➢Sparsity is considered by analyzing the pretrained supernet

	Tiling Mode	Kernel Mode	Buffer Re-purposing	Wbuf Sharing	Tiling Size
Format	3	4	2	2	N (~10-100)
# of Choices	[0,1,2]	[0,1,2,3]	[0,1]	[0,1]	[0,….n-1]

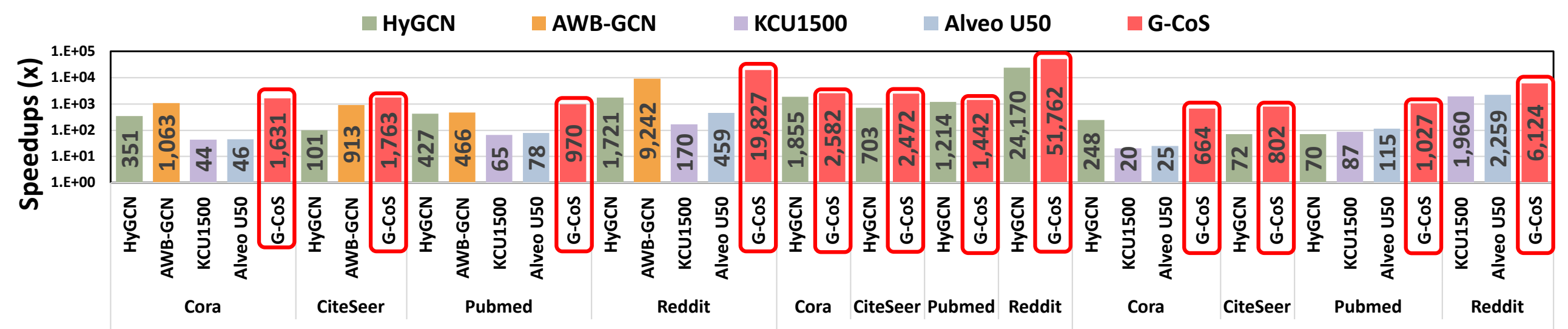
- ➢The proposed space comprises of 10^{10} ~ 10^{15} design choices
  - ➢Up to ~10^{34} design choices if combined with the GNN structure space
- ➢Reflect different tiling and scheduling configuration to cover
  - ➢**Different reuse strategies**
  - ➢**Different tradeoff** among parallelism, memory and bandwidth usage

G-CoS: Evaluation

Accuracy and latency tradeoff



Hardware search ablation with fixed GNN models



- ➢Comparing with **SOTA GNAS works** and **manually designed GNN accelerators**
  - ➢G-CoS consistently leads to a **better accuracy vs. latency frontier**
  - ➢G-CoS can **flexibly and efficiently tradeoff** between accuracy and latency