Enhancing Programmable Accelerators for Sparsity

Vidushi Dadu*, Jian Weng*,

Sihao Liu*, Tony Nowatzki*

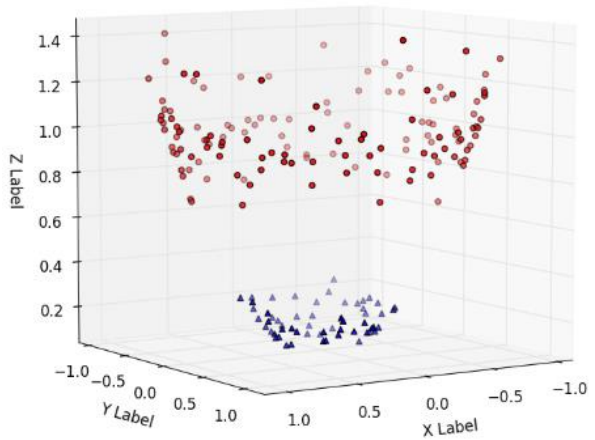
*UCLA

Yarch, HPCA 2019, Feb 17

UCLA

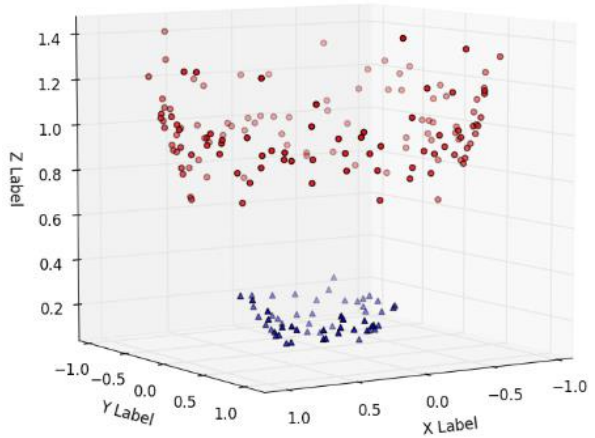
POLY ARCH

Irregular workloads are ubiquitous

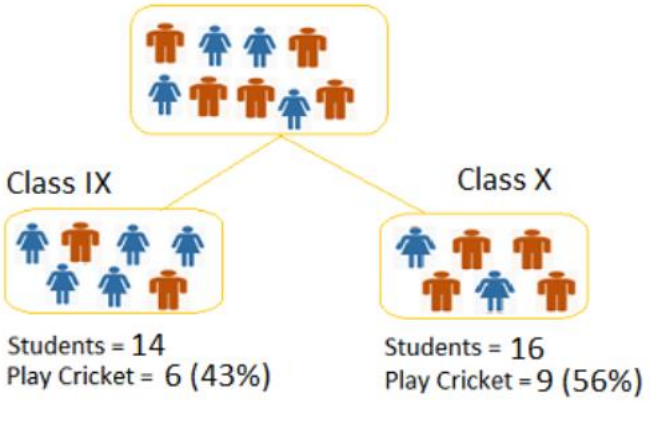


Kernel-SVM on High-dim Data

Irregular workloads are ubiquitous

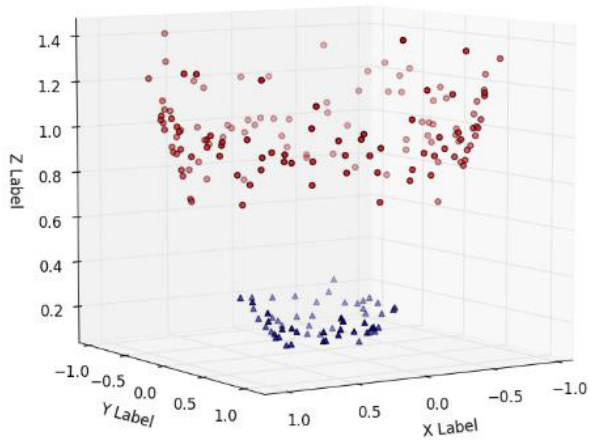


Kernel-SVM on High-dim Data

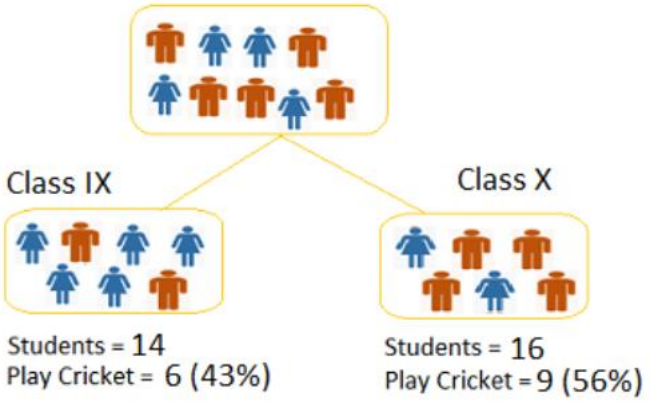


Decision Tree training

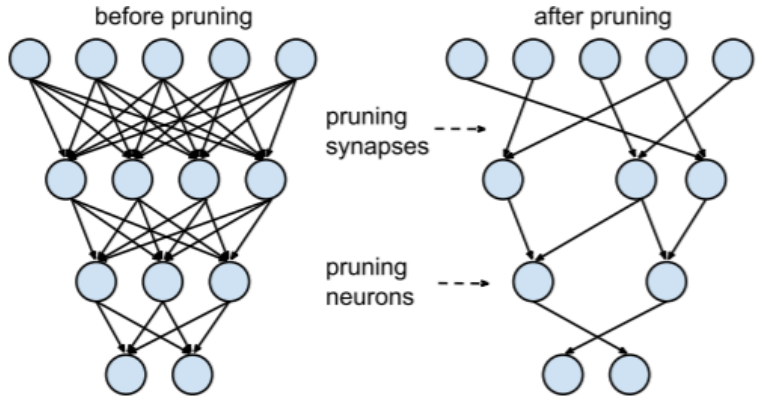
Irregular workloads are ubiquitous



Kernel-SVM on High-dim Data

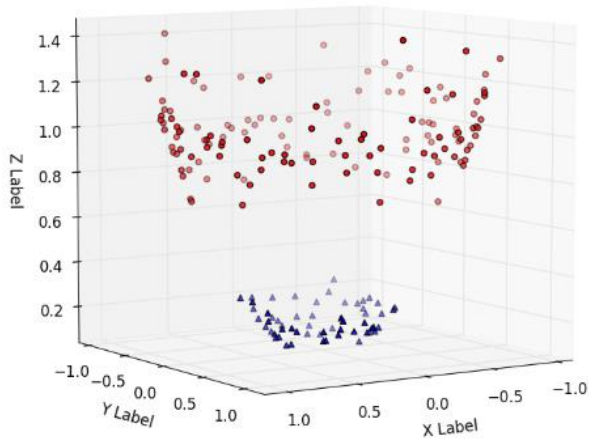


Decision Tree training

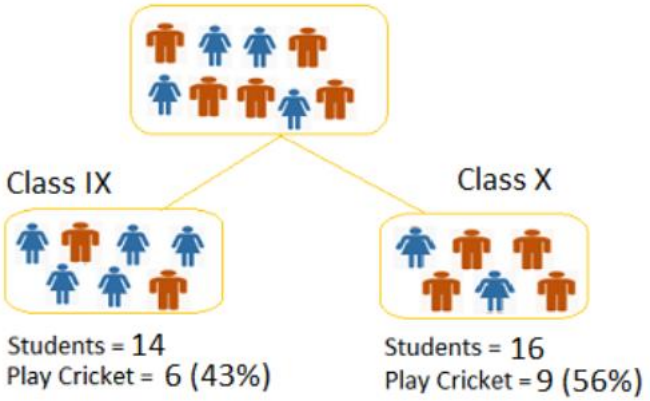


Pruned Deep Neural Networks

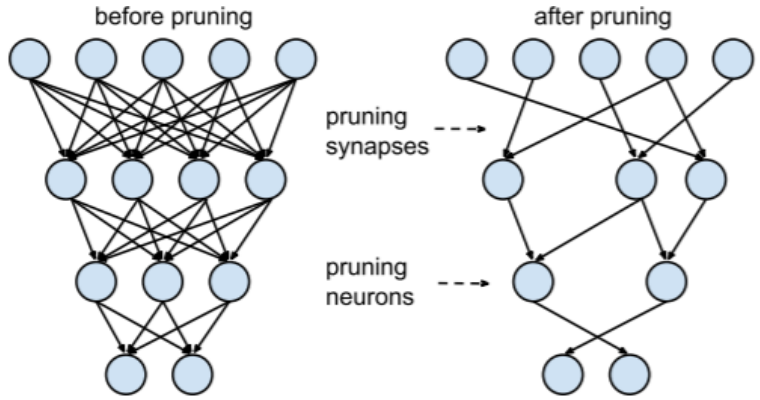
Irregular workloads are ubiquitous



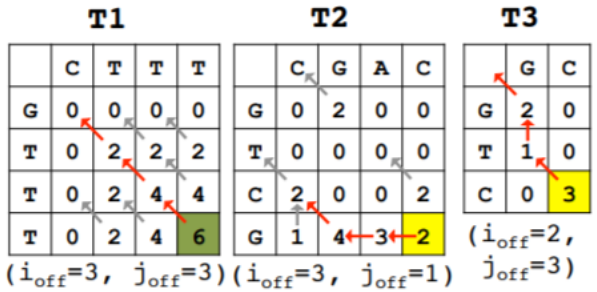
Kernel-SVM on High-dim Data



Decision Tree training

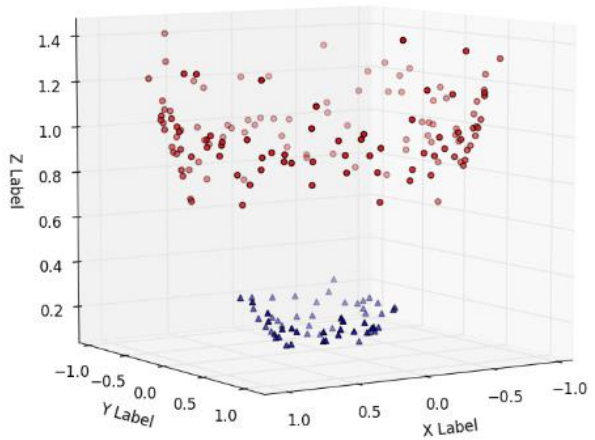


Pruned Deep Neural Networks

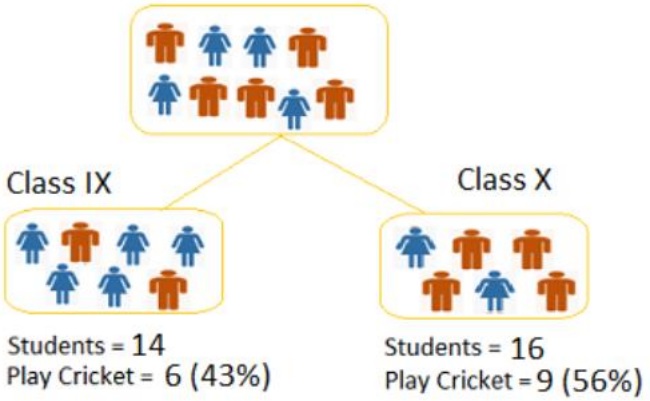


Genomics

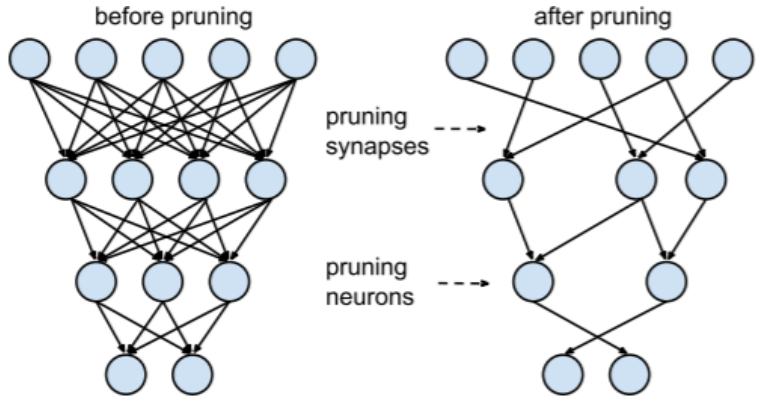
Irregular workloads are ubiquitous



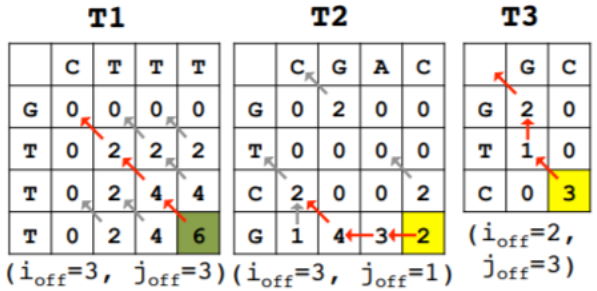
Kernel-SVM on High-dim Data

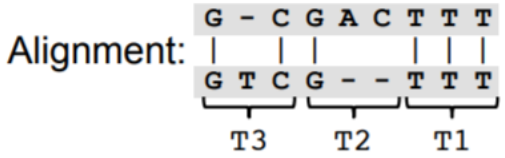


Decision Tree training

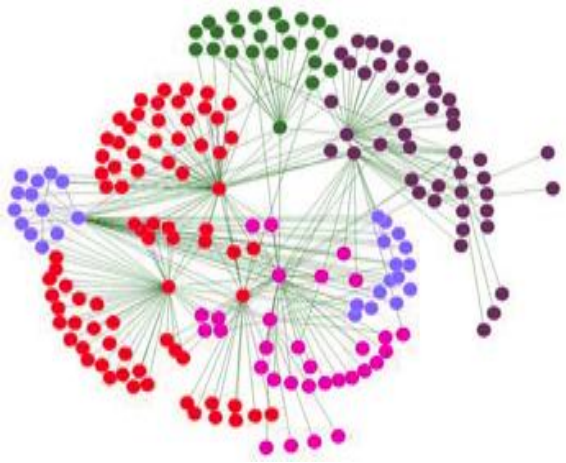


Pruned Deep Neural Networks



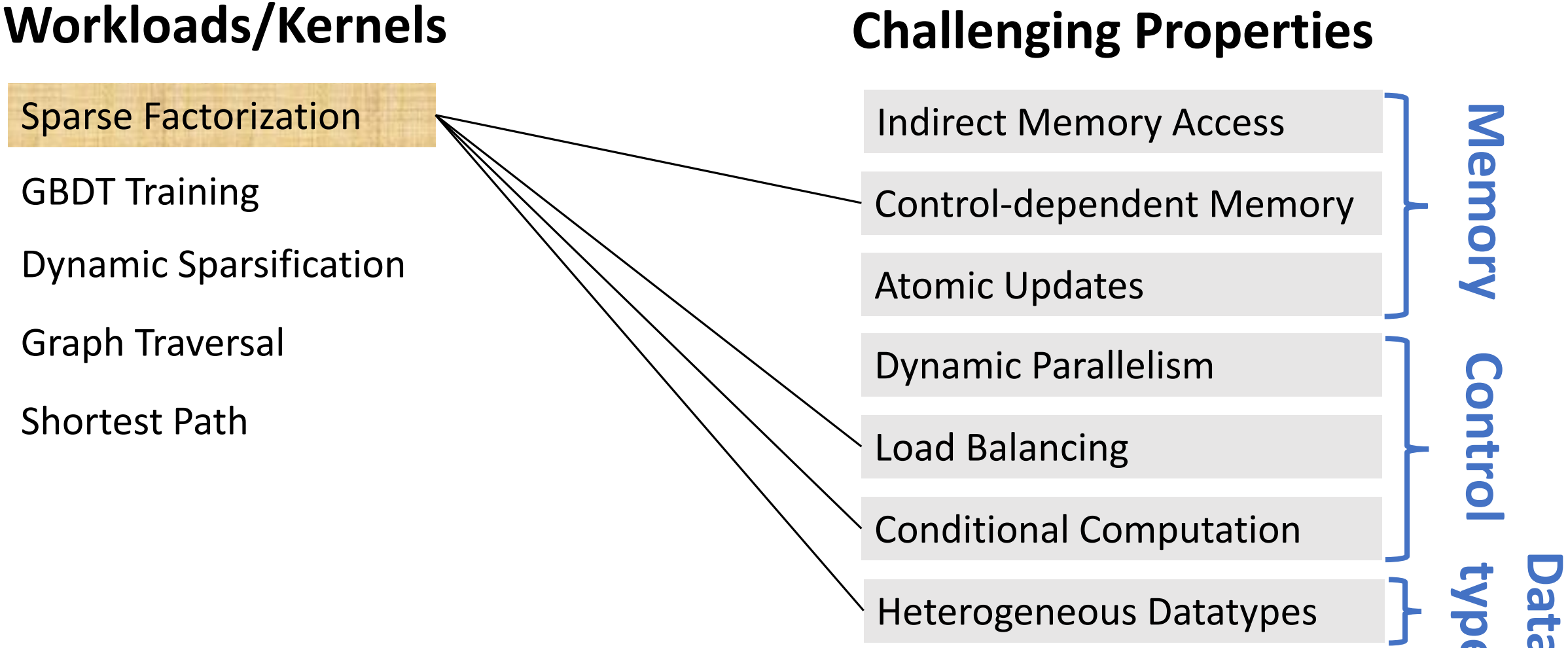


Genomics

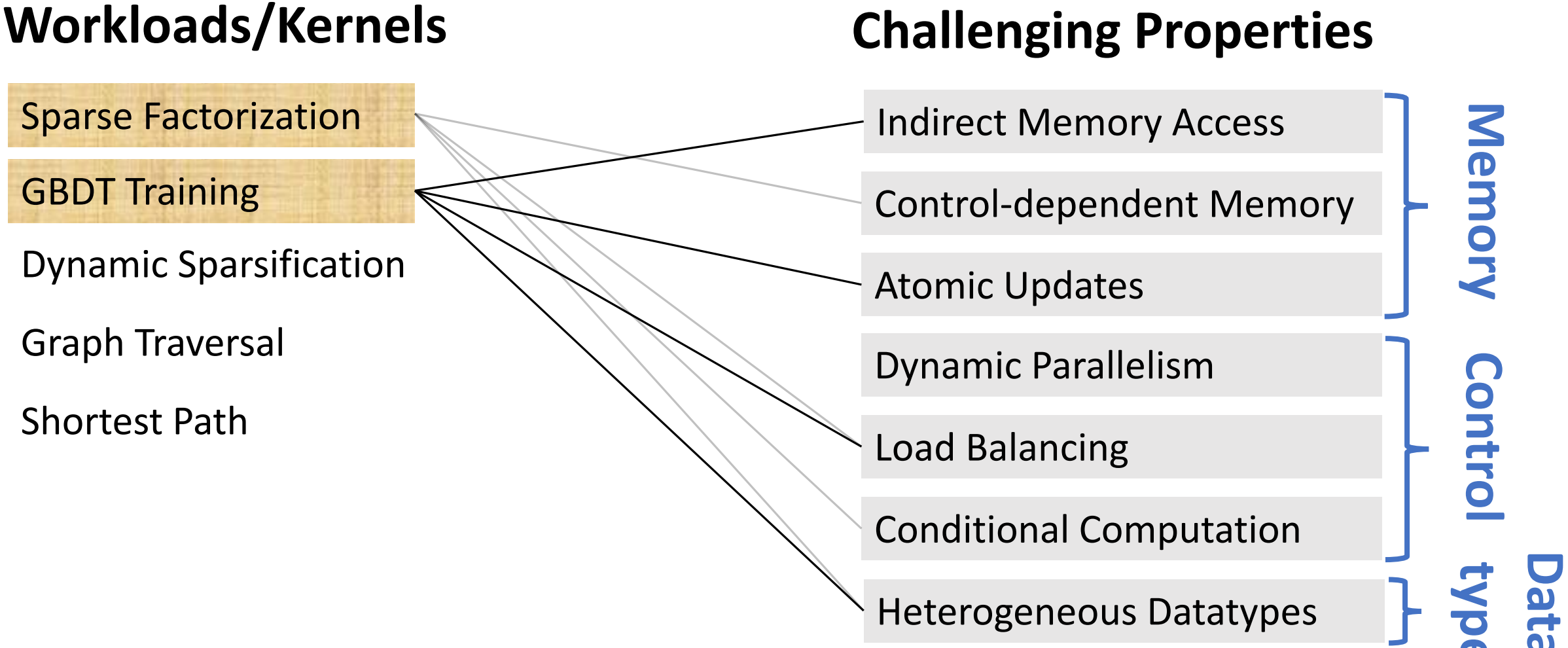


Graph Processing

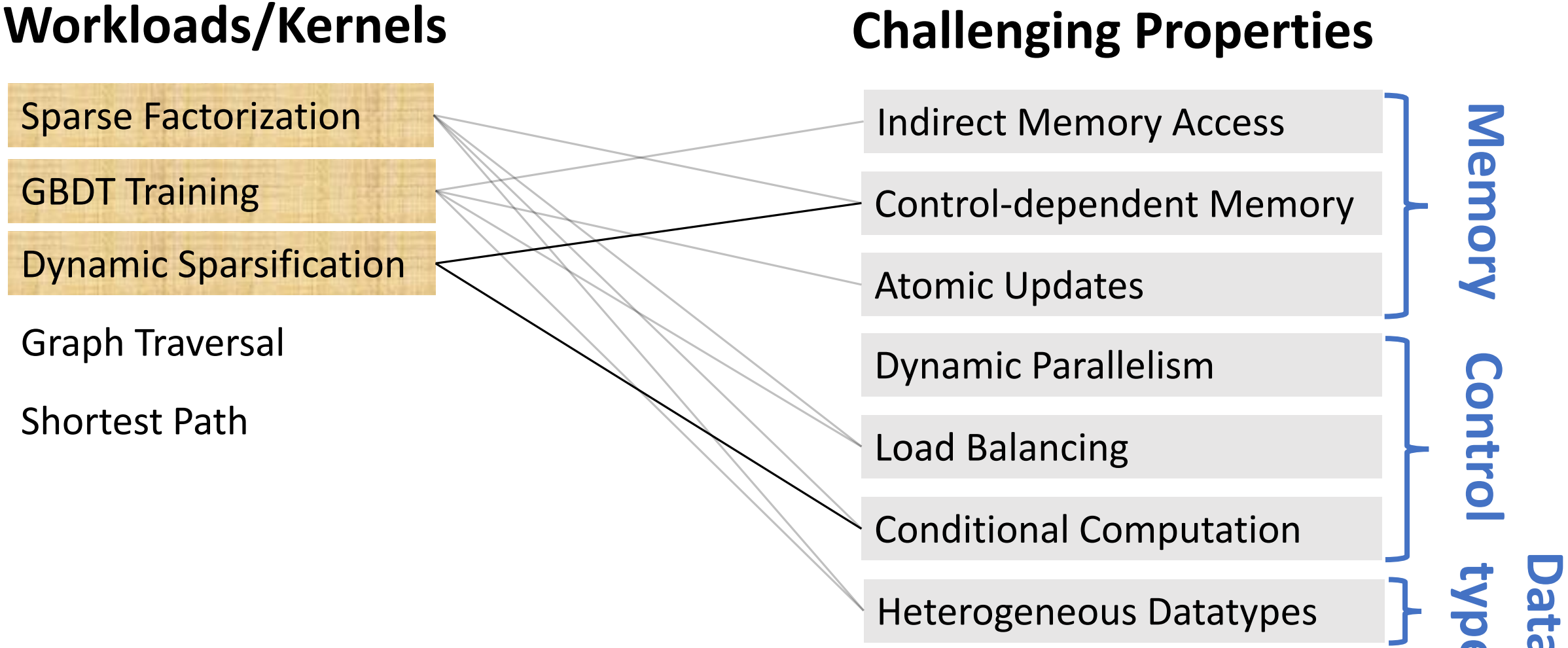
Sparsity in Workloads Requires Complex Architecture Mechanisms



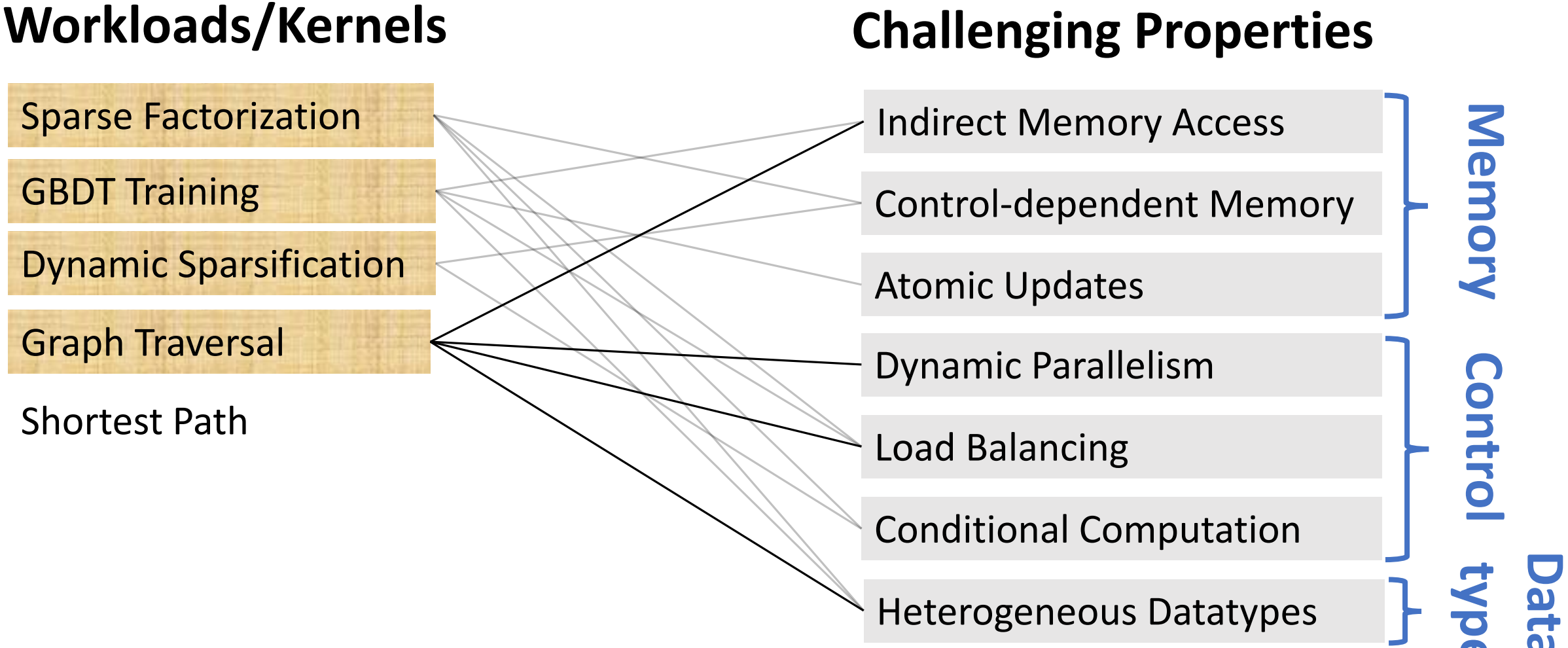
Sparsity in Workloads Requires Complex Architecture Mechanisms



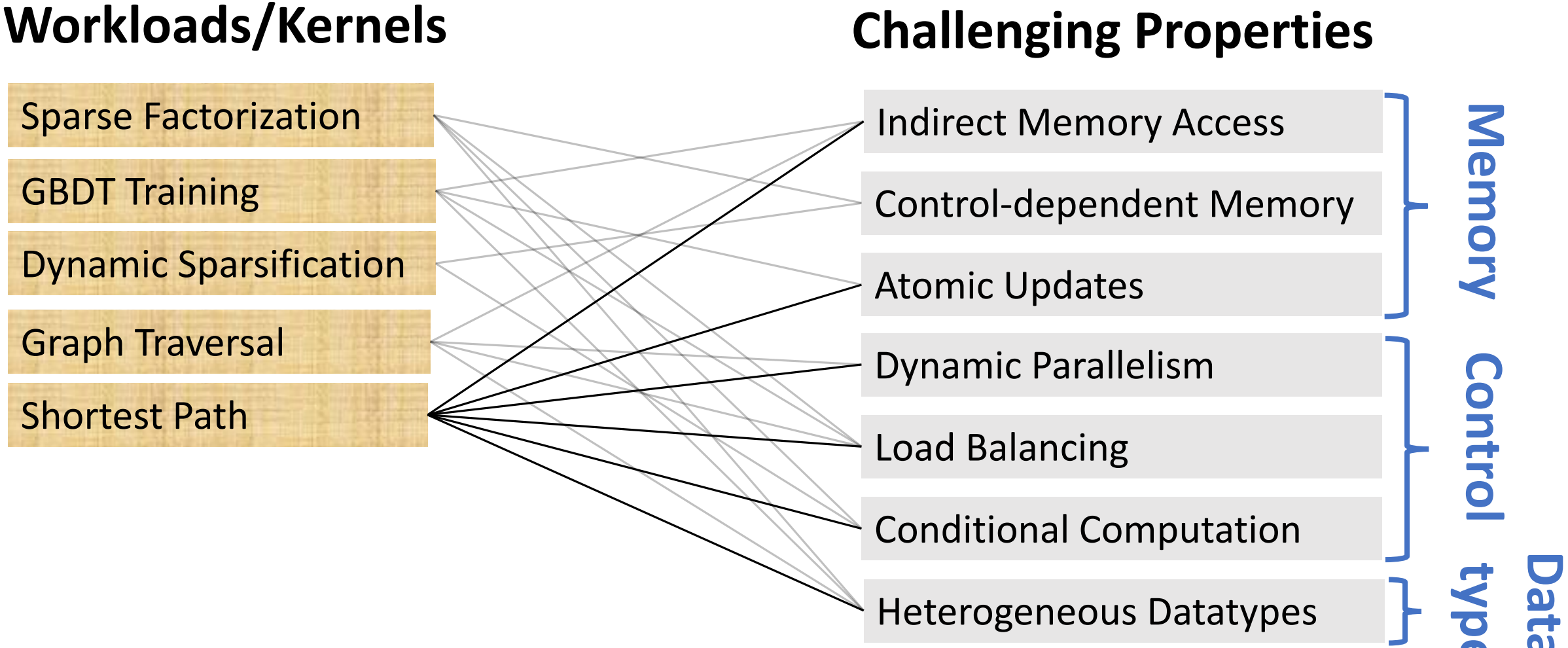
Sparsity in Workloads Requires Complex Architecture Mechanisms



Sparsity in Workloads Requires Complex Architecture Mechanisms



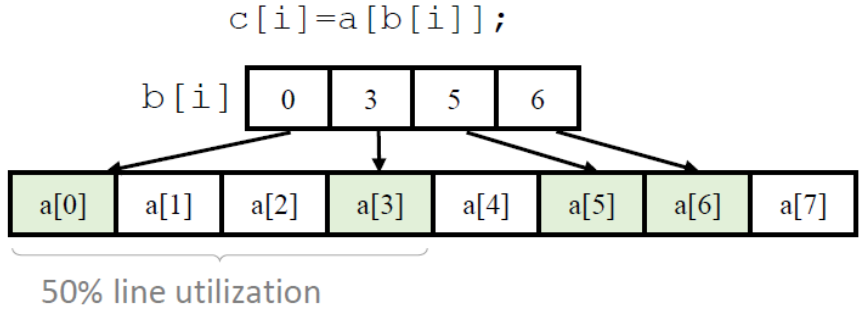
Sparsity in Workloads Requires Complex Architecture Mechanisms



GPUs are insufficient for irregularity

Memory Irregularity (Wide mem)

Accesses to random memory location causes inefficient cache line utilization.



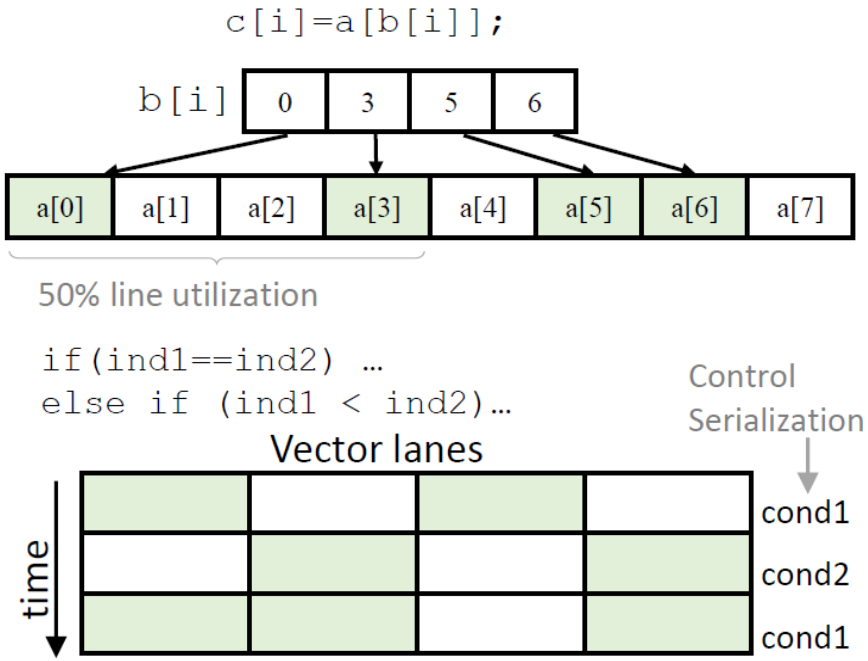
GPUs are insufficient for irregularity

Memory Irregularity (Wide mem)

Accesses to random memory location causes inefficient cache line utilization.

Control Irregularity (SIMD)

Data-dependent compute leads to masking in vector architectures.



GPUs are insufficient for irregularity

Memory Irregularity (Wide mem)

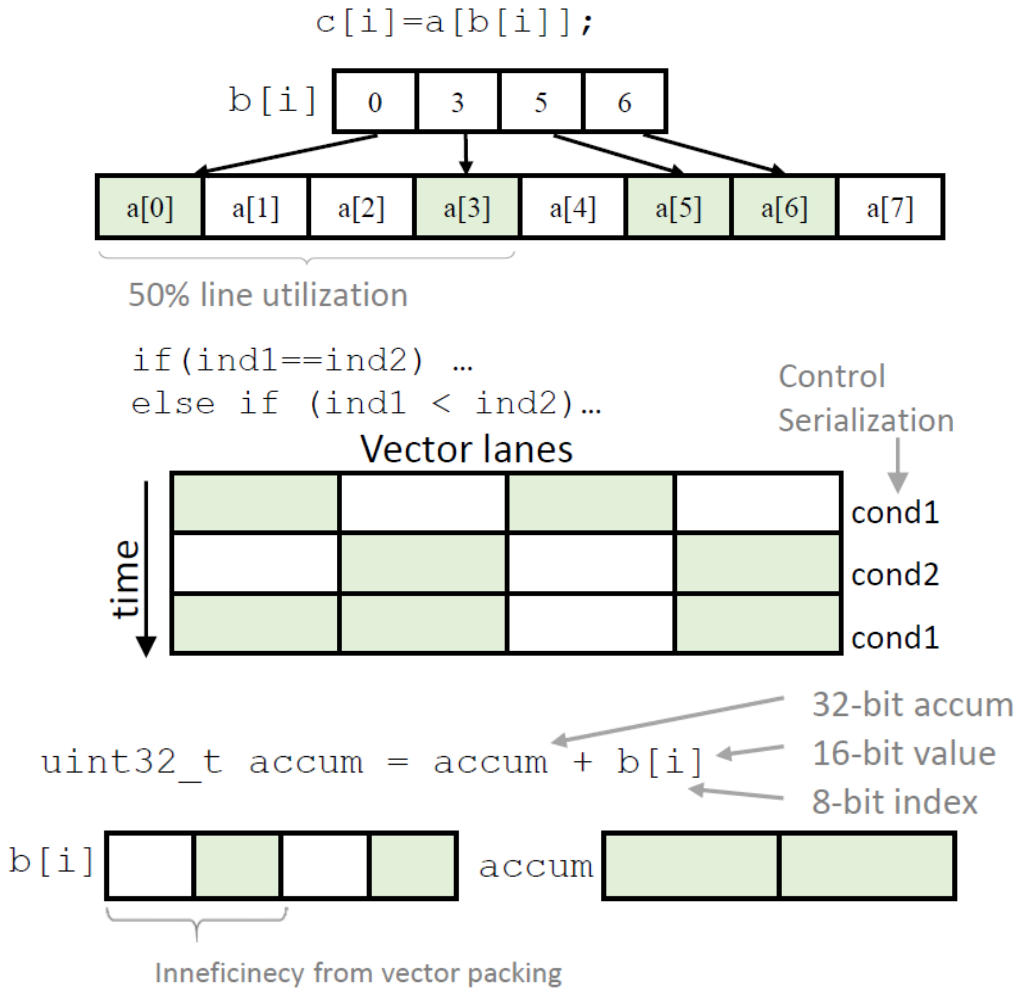
Accesses to random memory location causes inefficient cache line utilization.

Control Irregularity (SIMD)

Data-dependent compute leads to masking in vector architectures.

Datatype Irregularity (Fixed vector width)

Irregular algorithms doesn't allow efficient packing of lower datatypes.



Is Massive Scalar Processor Sufficient?

Is Massive Scalar Processor Sufficient?

There are a couple of problems:

- **General purpose overheads:** Maintaining the program counter and precise state limits performance.

Is Massive Scalar Processor Sufficient?

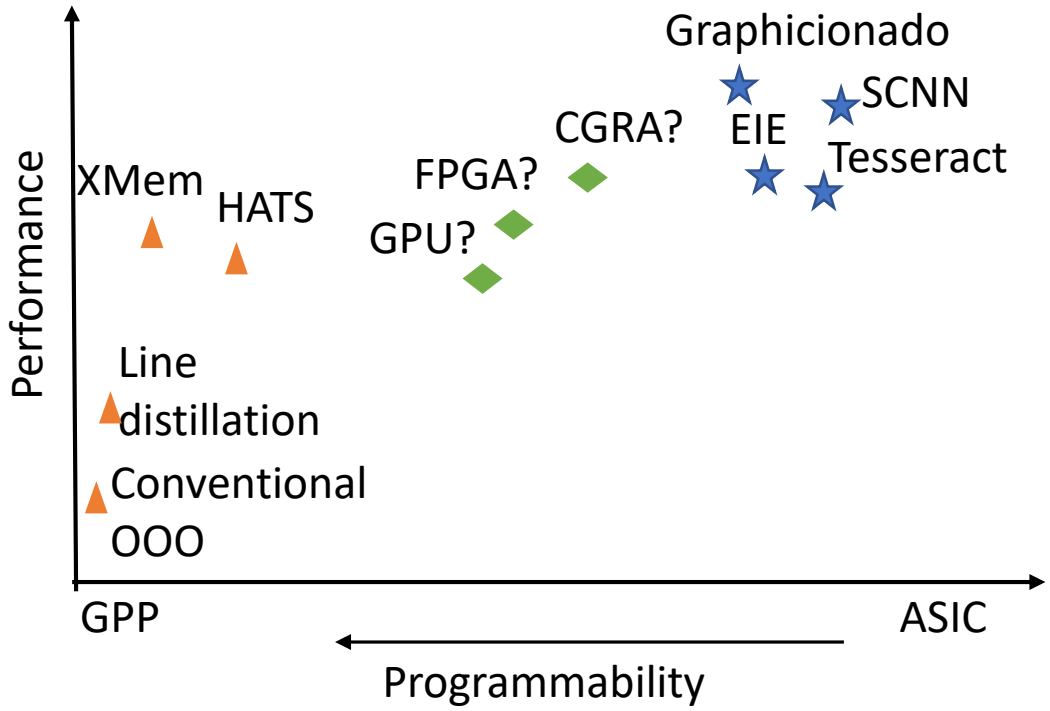
There are a couple of problems:

- **General purpose overheads:** Maintaining the program counter and precise state limits performance.

- **Programmability:** Such architecture hurts the performance of regular algorithms which have high acceleration potential.

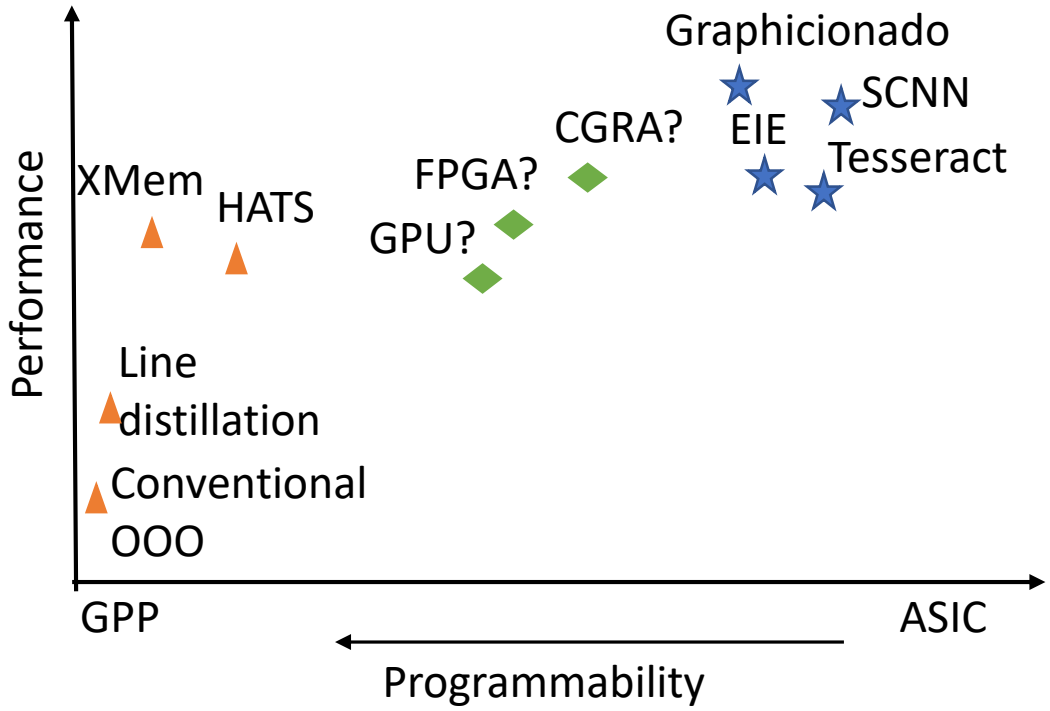
Hope: Sparse Accelerators Have Been Successful

1. **SCNN:** for sparse conv. layer
2. **EIE:** for sparse FC layers
3. **Graphicionado:** graph processor
4. **HATS:** Locality-aware scheduling for graph processing
5. **XMem:** Programmer hints to prefetchers, caching policies



Hope: Sparse Accelerators Have Been Successful

1. **SCNN:** for sparse conv. layer
2. **EIE:** for sparse FC layers
3. **Graphicionado:** graph processor
4. **HATS:** Locality-aware scheduling for graph processing
5. **XMem:** Programmer hints to prefetchers, caching policies



It will be useful if we have an architecture which gives high performance on the set of workloads we care about most.

Goal: Is there an accelerator paradigm which performs well on most irregular workloads?

Goal: Is there an accelerator paradigm which performs well on most irregular workloads?

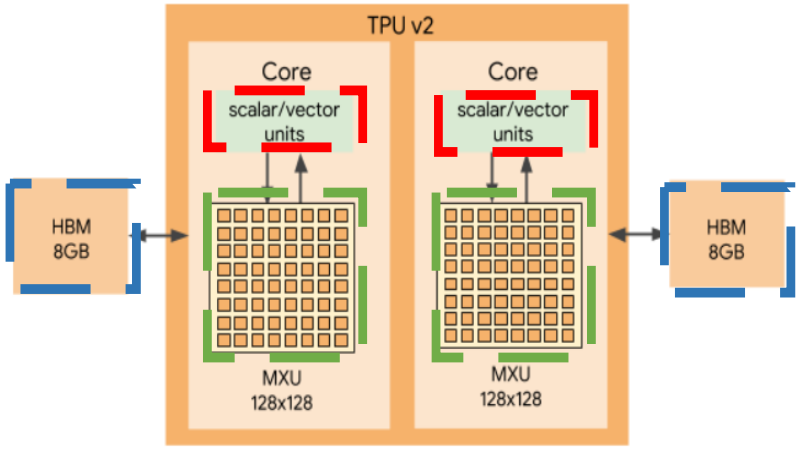
In this talk, I will focus on **Irregular Machine Leaning.**

In machine learning domain, both dense and sparse scenarios are pretty common.

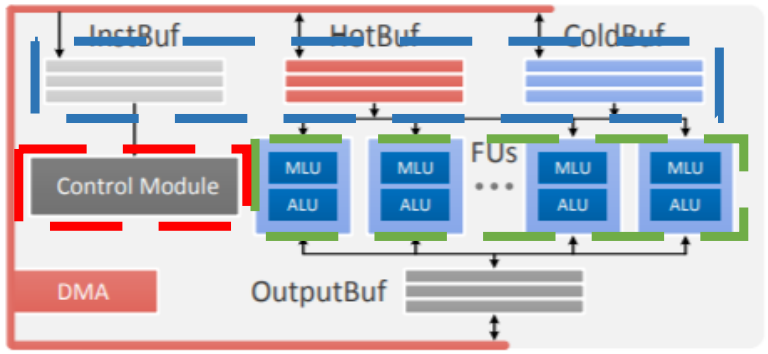
Sub-goal: Design a programmable sparse accelerator while maintaining efficiency for the dense workloads.

Our Approach: Start with a Dense Programmable Accelerator

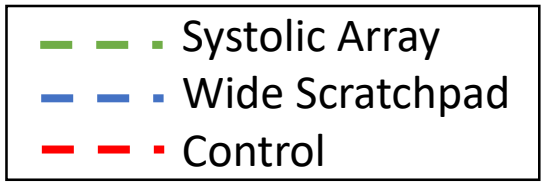
Our Approach: Start with a Dense Programmable Accelerator



Google TPU v2



PuDianNao (ASPLOS'15)



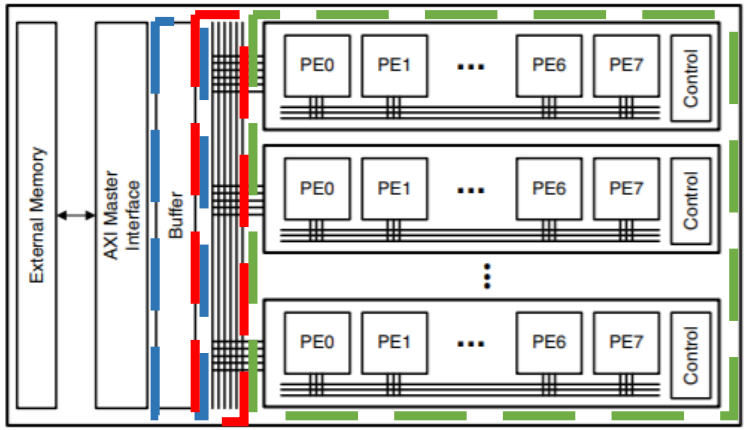
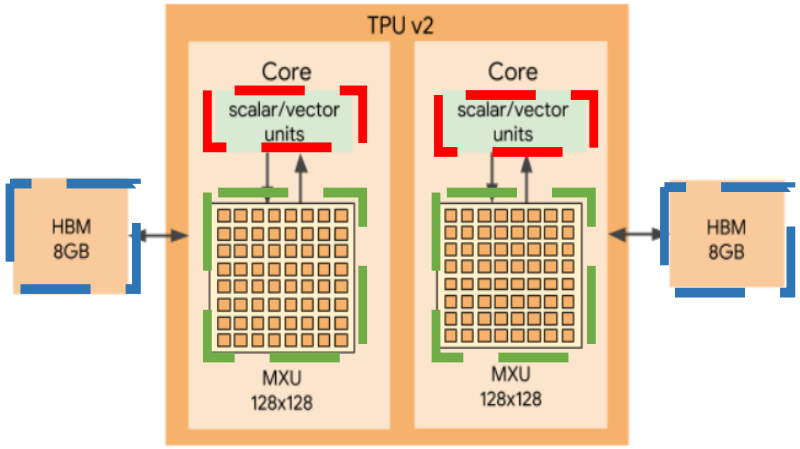
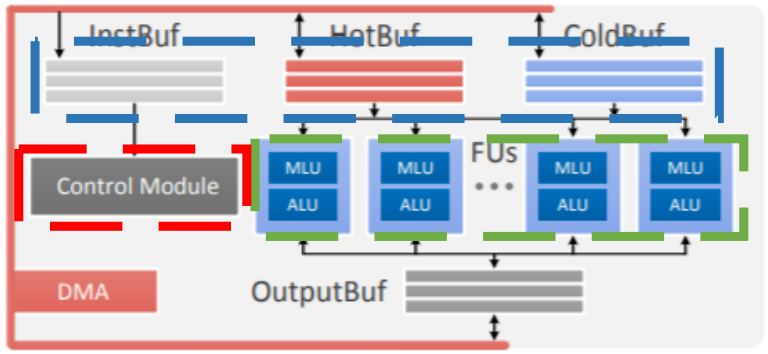


Tabla (HPCA'16)

Our Approach: Start with a Dense Programmable Accelerator



Google TPU v2



PuDianNao (ASPLOS'15)

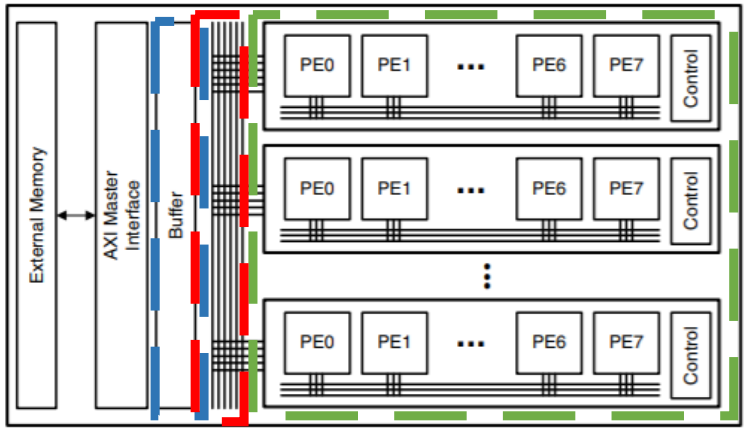
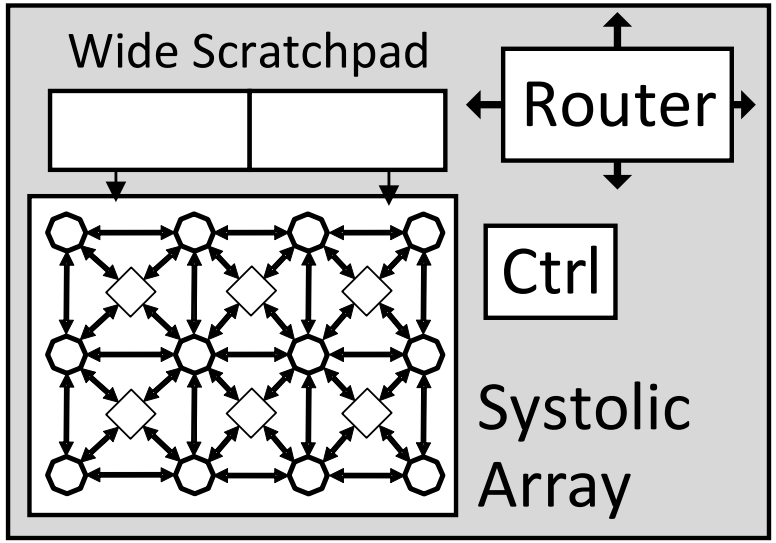


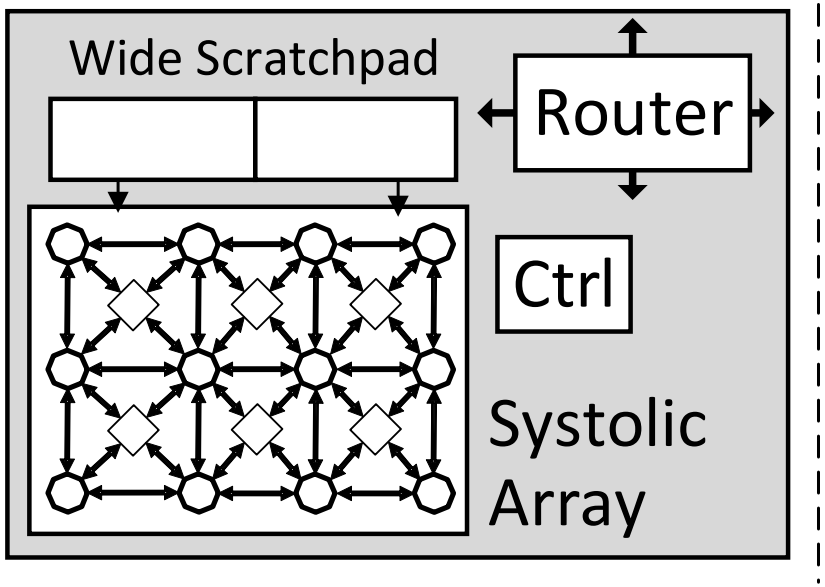
Tabla (HPCA'16)



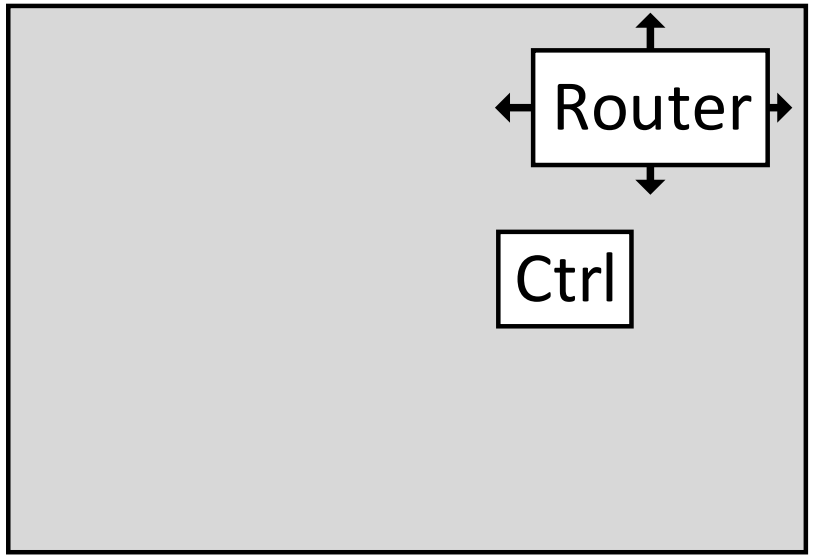
Stereotypical Dense Accelerator Core

Systolic Array
Wide Scratchpad
Control

Overview of Sparsity-Enabled SPU core

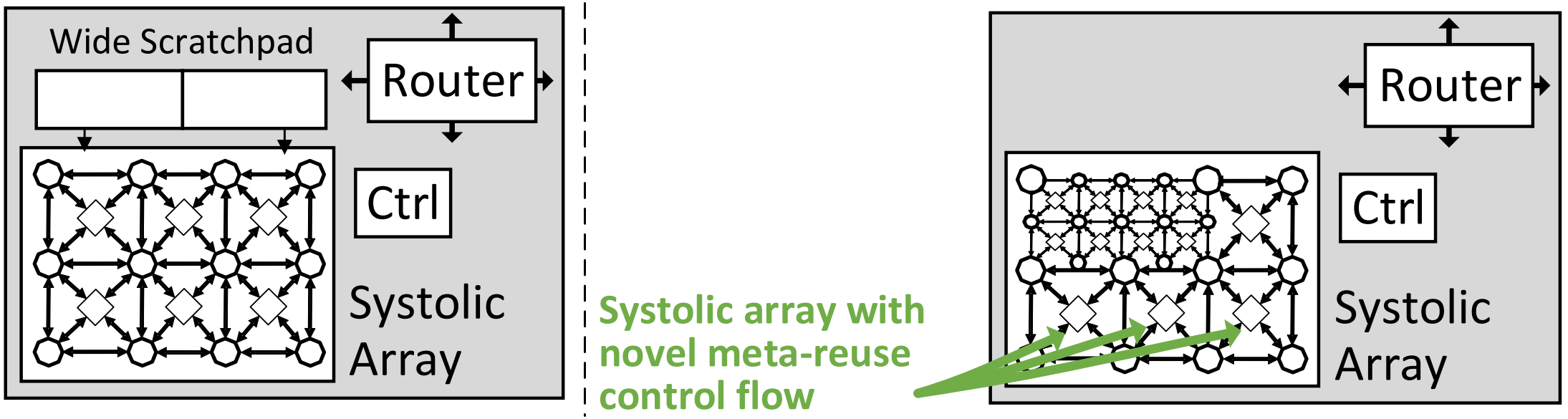


(a) Stereotypical Dense Accelerator Core



(b) Sparsity Enabled Accelerator (SPU Core)

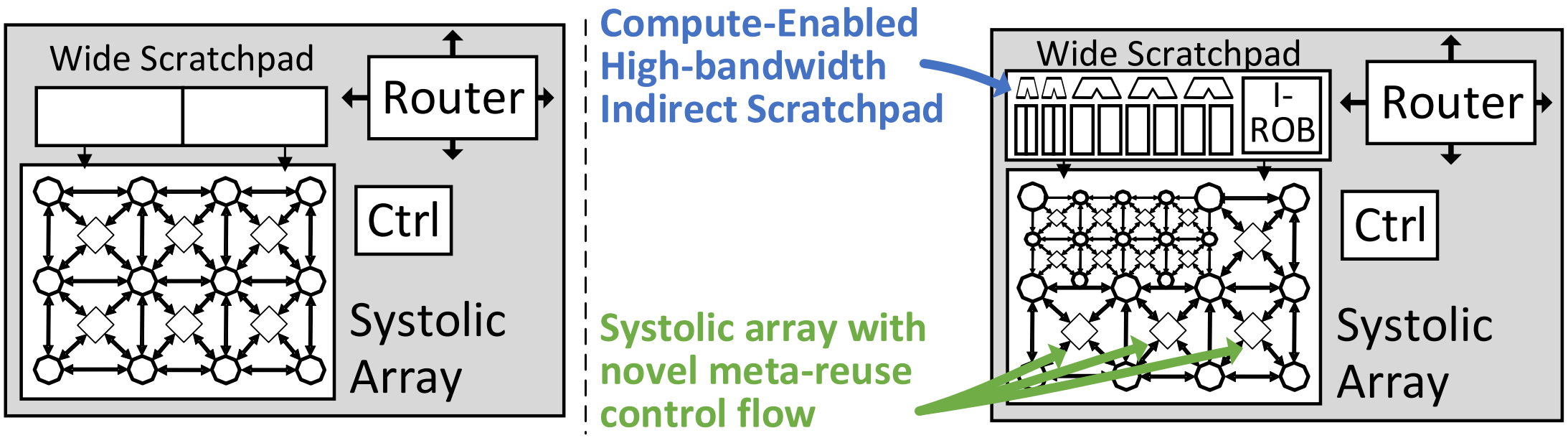
Overview of Sparsity-Enabled SPU core



(a) Stereotypical Dense Accelerator Core

(b) Sparsity Enabled Accelerator (SPU Core)

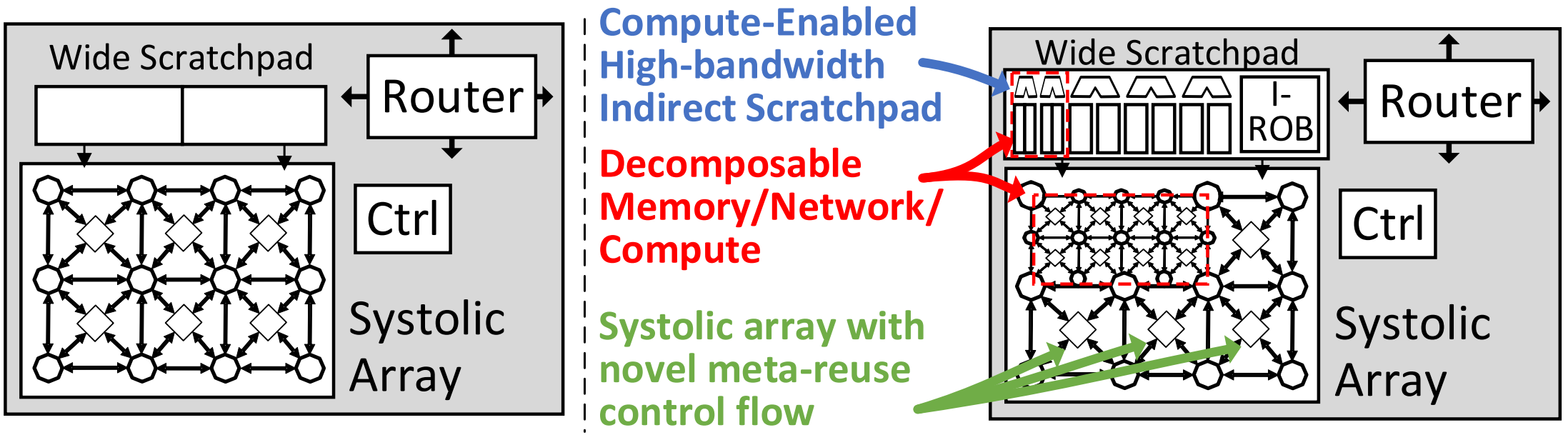
Overview of Sparsity-Enabled SPU core



(a) Stereotypical Dense Accelerator Core

(b) Sparsity Enabled Accelerator (SPU Core)

Overview of Sparsity-Enabled SPU core



(a) Stereotypical Dense Accelerator Core

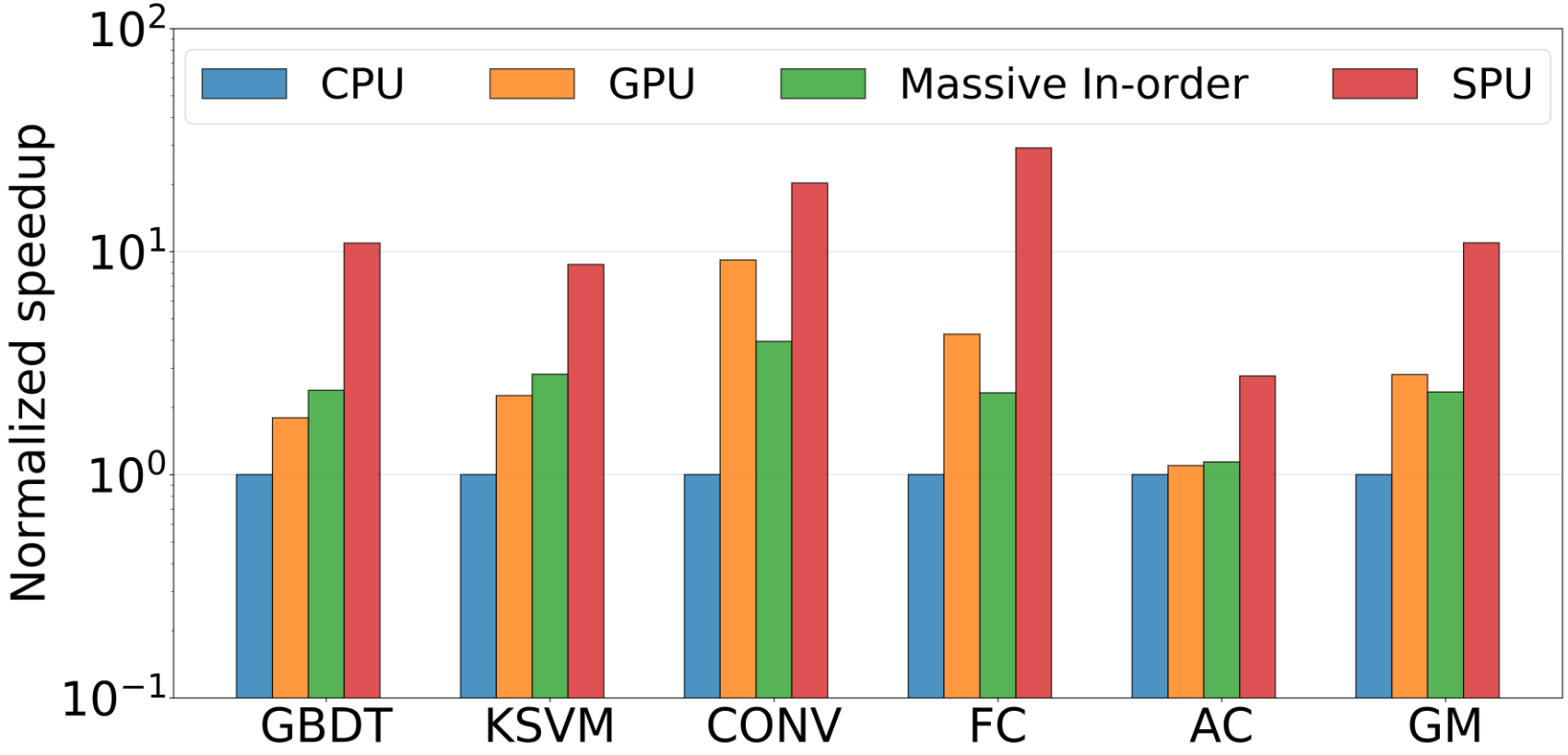
(b) Sparsity Enabled Accelerator (SPU Core)

Evaluation Methodology

- gem5 cycle level simulator
- Dataflow compiler
- Benchmarks (top-5 ML algorithms using by Facebook in 2018*):
  - KSVM, Conv layer, FC layer, GBDT, Arithmetic Circuits
- Datasets: Open-source practical datasets
- Baselines:
  - CPU: 4-core CPU (reference only)
  - GPU: Nvidia P4000
  - Massive In-order: SPU but with 512 inorder cores
  - SPU: Sparse Proc. Unit (64 Core)

*Kim Hazelwood, et al. Applied machine learning at facebook: A datacenter infrastructure perspective. In High Performance Computer Architecture (HPCA), 2018

SPU achieves 2.2-6.8x performance over GPU



Ongoing/Future Work

- Compiler for streaming-dataflow architecture.
- Specialized cache hierarchy
- Hardware-software (ISA) primitives for other irregular domains:
  - Graph processing
  - Genomics
  - Compression/Decompression