A 28nm SoC with a 1.2GHz 568nJ/Prediction Sparse Deep-Neural-Network Engine with >0.1 Timing Error Rate Tolerance for IoT Applications

Paul Whatmough, S. K. Lee,
H. Lee, S. Rama, D. Brooks, G.-Y. Wei
Harvard University, Cambridge, MA

HARVARD
School of Engineering
and Applied Sciences

Motivation

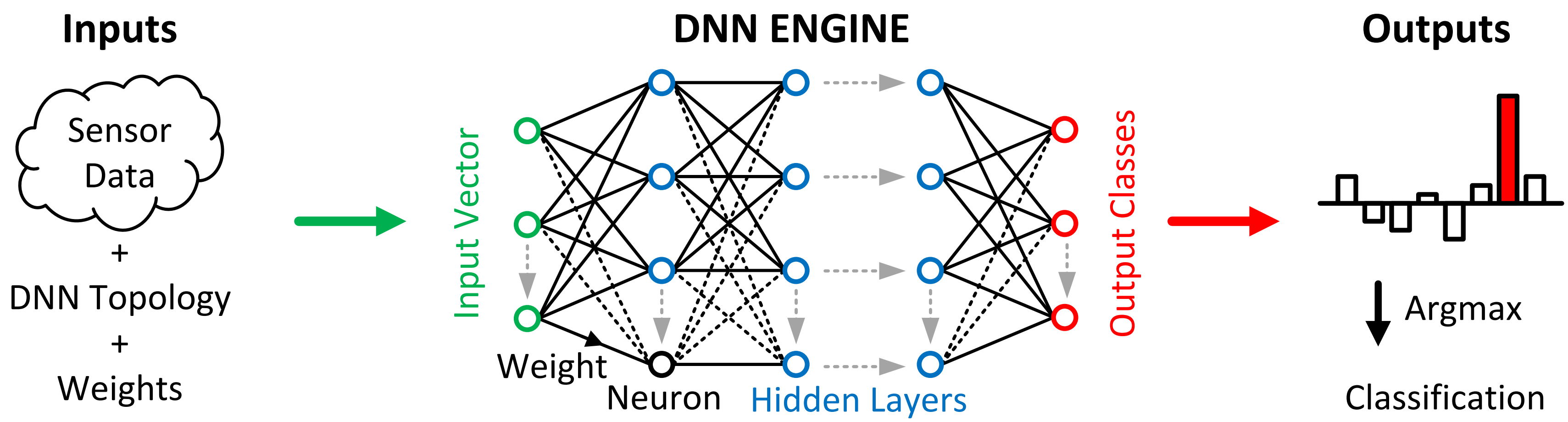
Motivation

- **Deep Neural Networks (DNNs)**
  – A universal classifier with many diverse applications
  – Interpret noisy real-world data from sensor-rich systems

Motivation

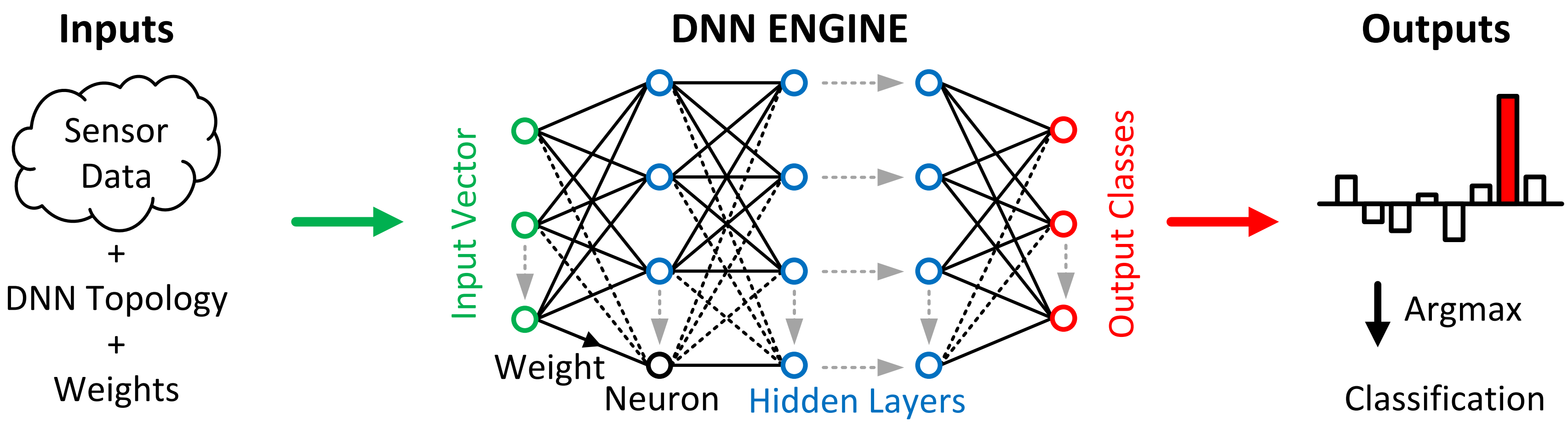
- **Deep Neural Networks (DNNs)**
  - A universal classifier with many diverse applications
  - Interpret noisy real-world data from sensor-rich systems
  - **Large memory footprint and compute load**
  - **Perform DNN inference on device in micro-Joules**

Motivation



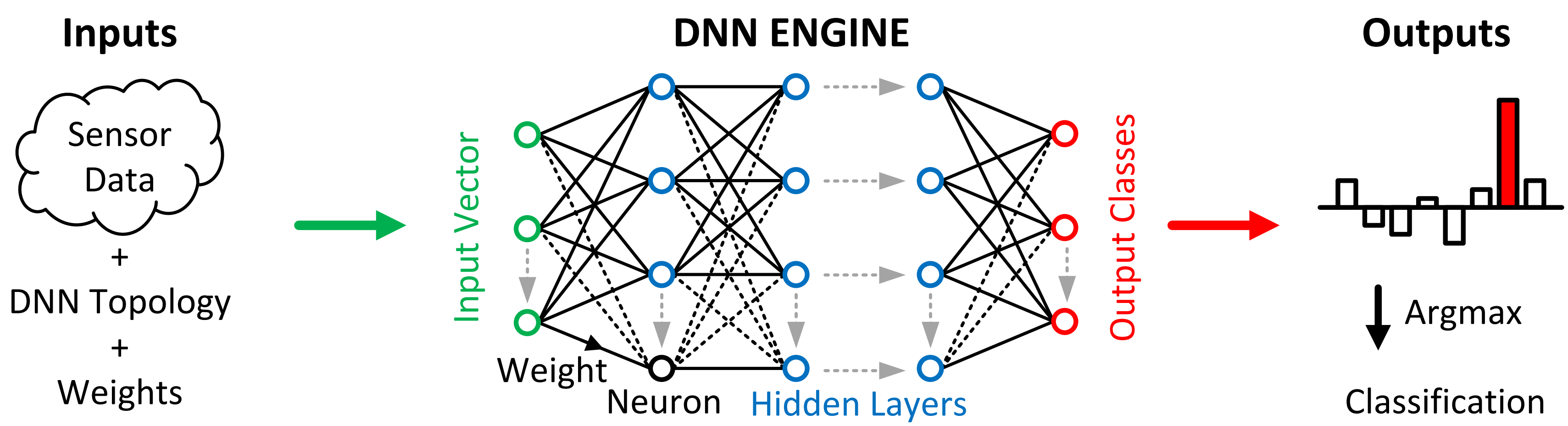
- **DNN ENGINE – A Shared Programmable Classifier**
  - **Specialized Architecture and Efficient Digital Circuits**

Motivation



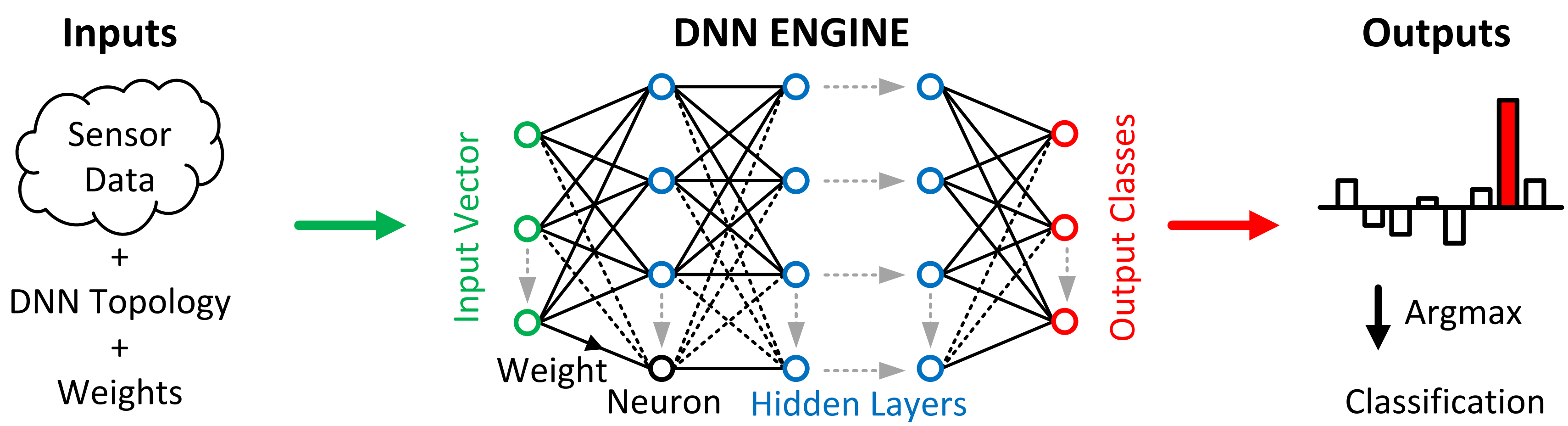
- **DNN ENGINE – A Shared Programmable Classifier**
  - **Specialized Architecture and Efficient Digital Circuits**
  - **Parallelism:** Balanced memory bandwidth and throughput

Motivation



- **DNN ENGINE – A Shared Programmable Classifier**
  - **Specialized Architecture and Efficient Digital Circuits**
  - **Parallelism:** Balanced memory bandwidth and throughput
  - **Sparsity:** HW support for sparse data and small datatypes

Motivation

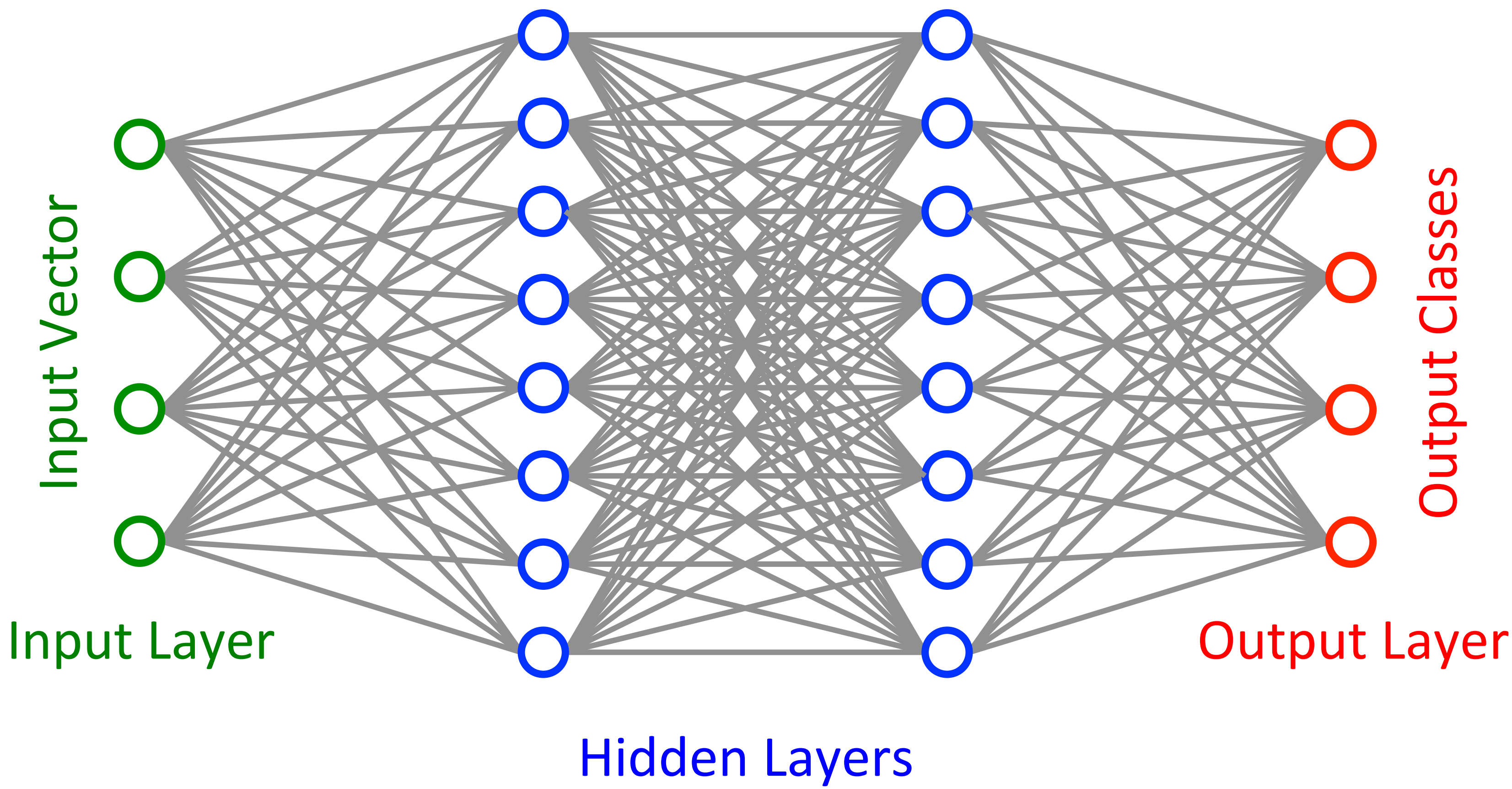


- **DNN ENGINE – A Shared Programmable Classifier**
  - **Specialized Architecture and Efficient Digital Circuits**
  - **Parallelism:** Balanced memory bandwidth and throughput
  - **Sparsity:** HW support for sparse data and small datatypes
  - **Resilience:** Razor adaptation to minimize PVT margins

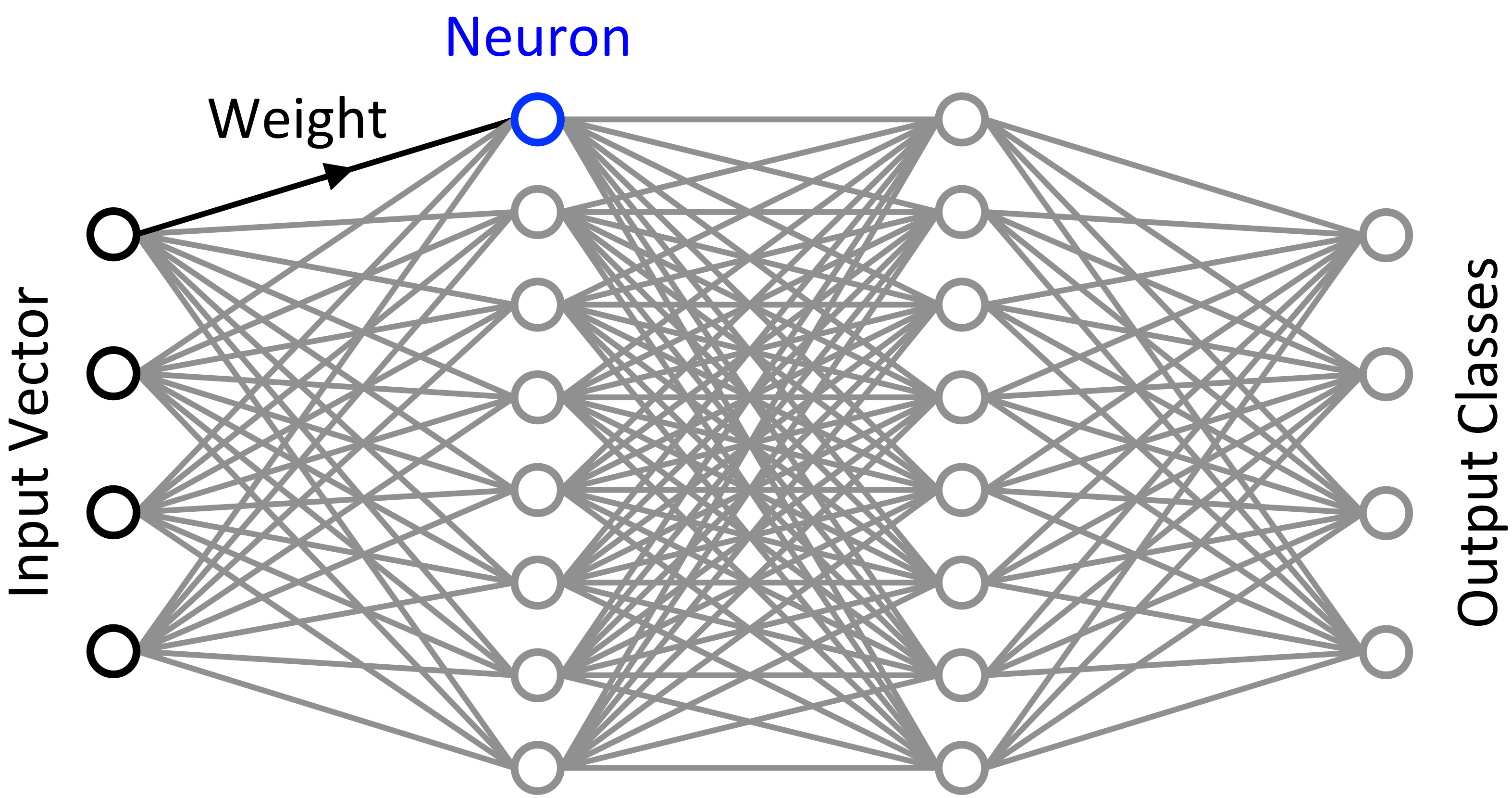
Outline

- Motivation
- DNN Engine
  - Parallelism
  - Sparsity
  - Resilience
- Measurement Results
- Summary

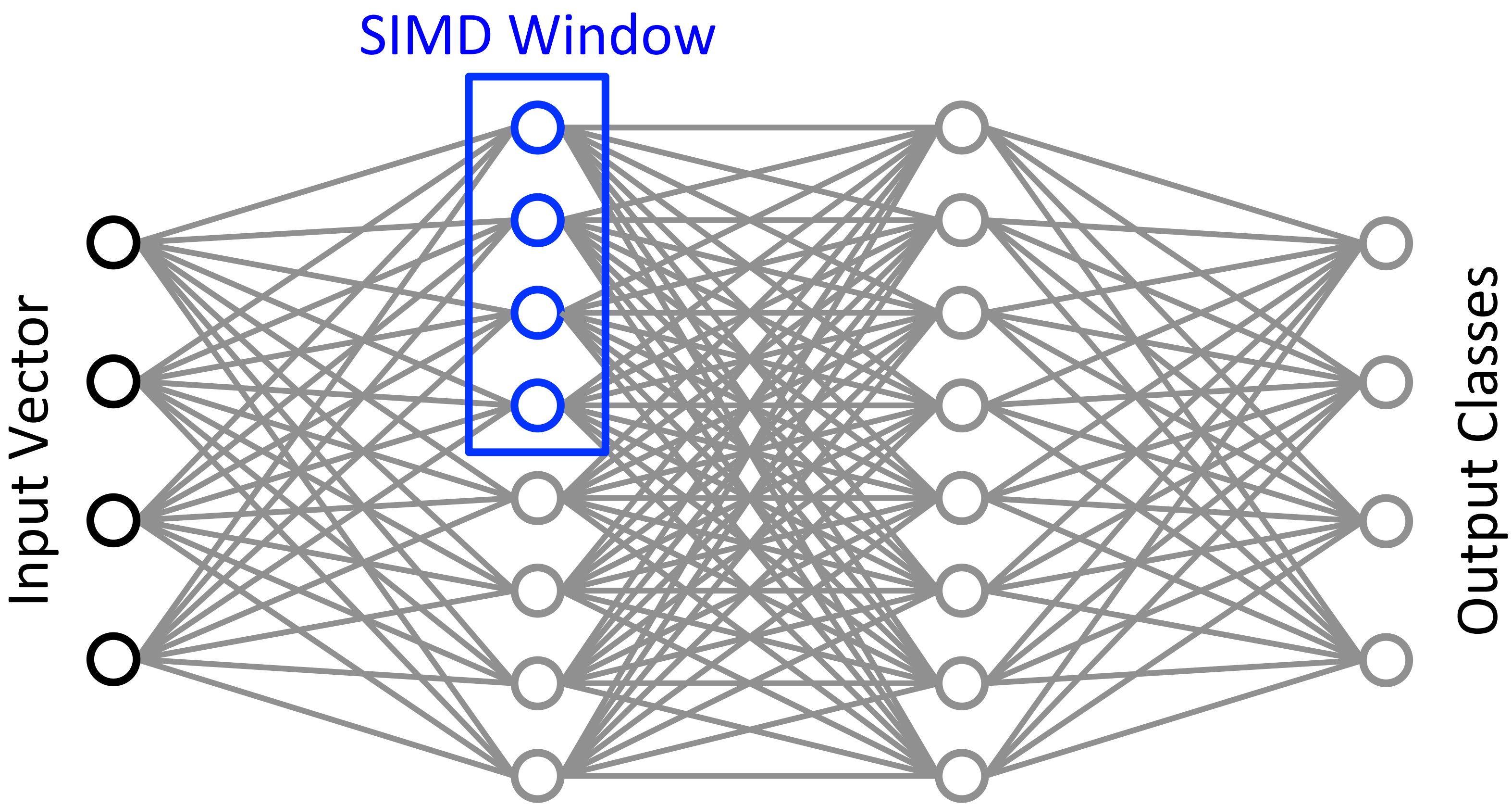
Fully-Connected DNN Graph



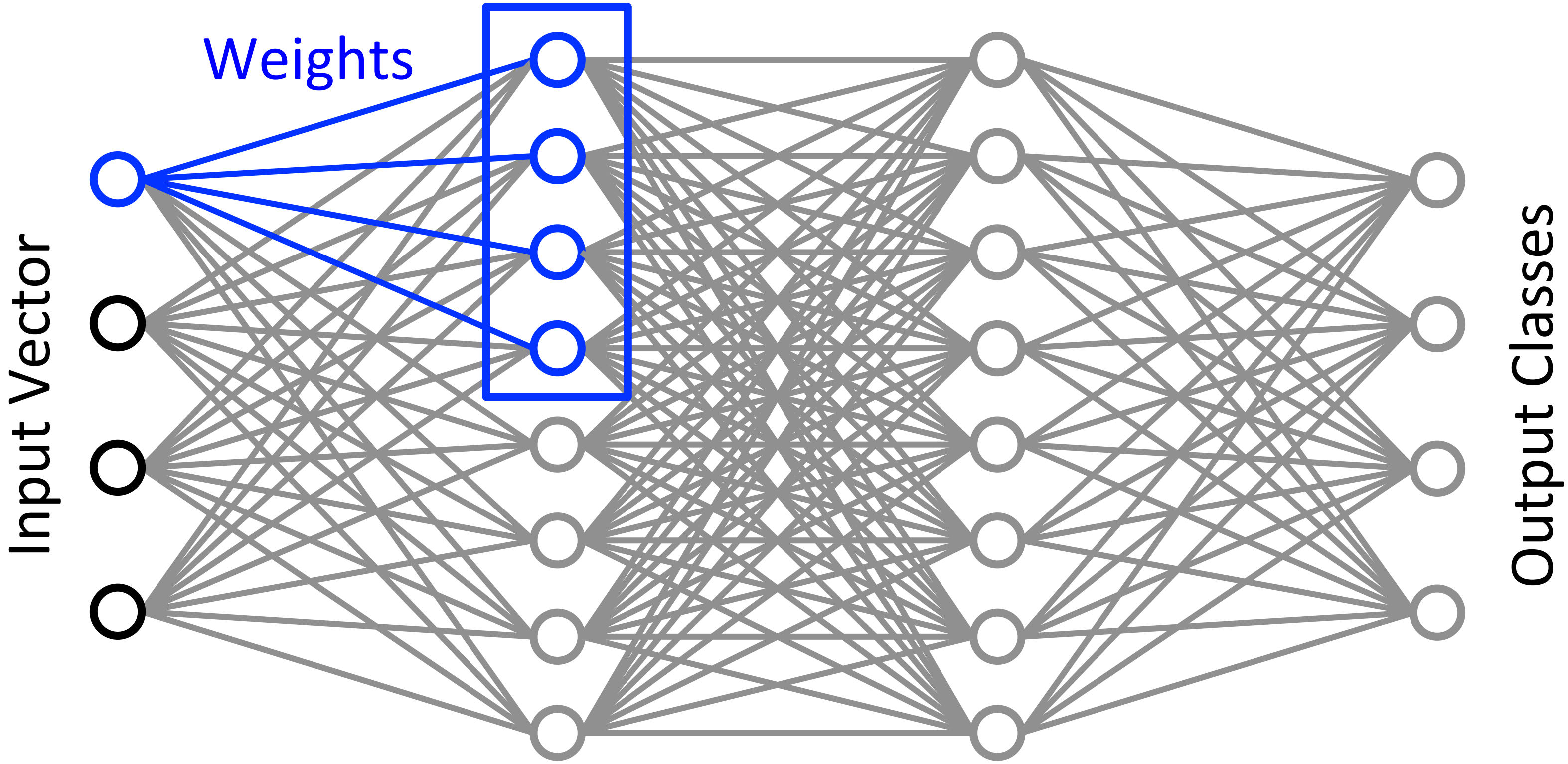
Fully-Connected DNN Graph



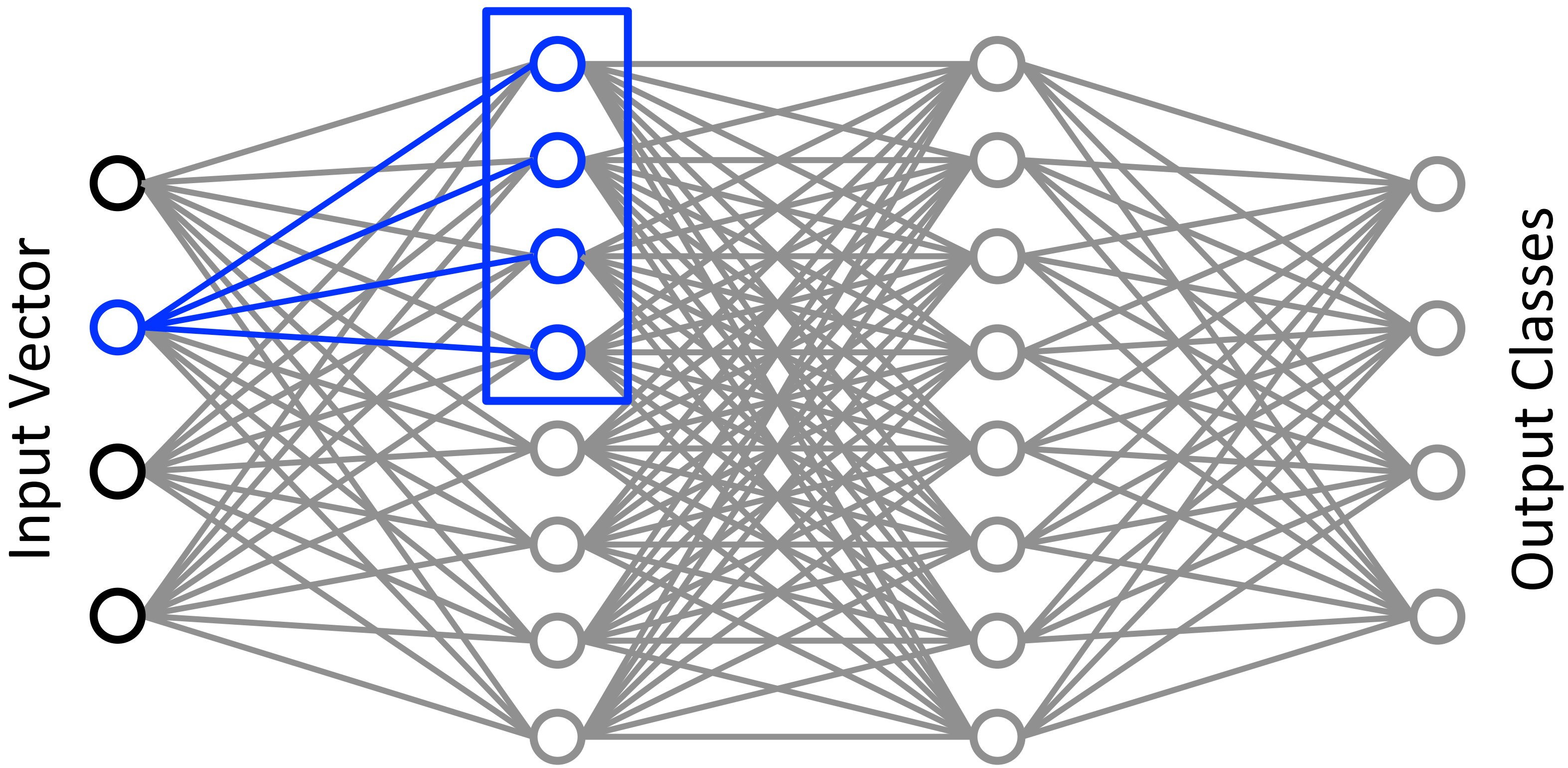
Fully-Connected DNN Graph



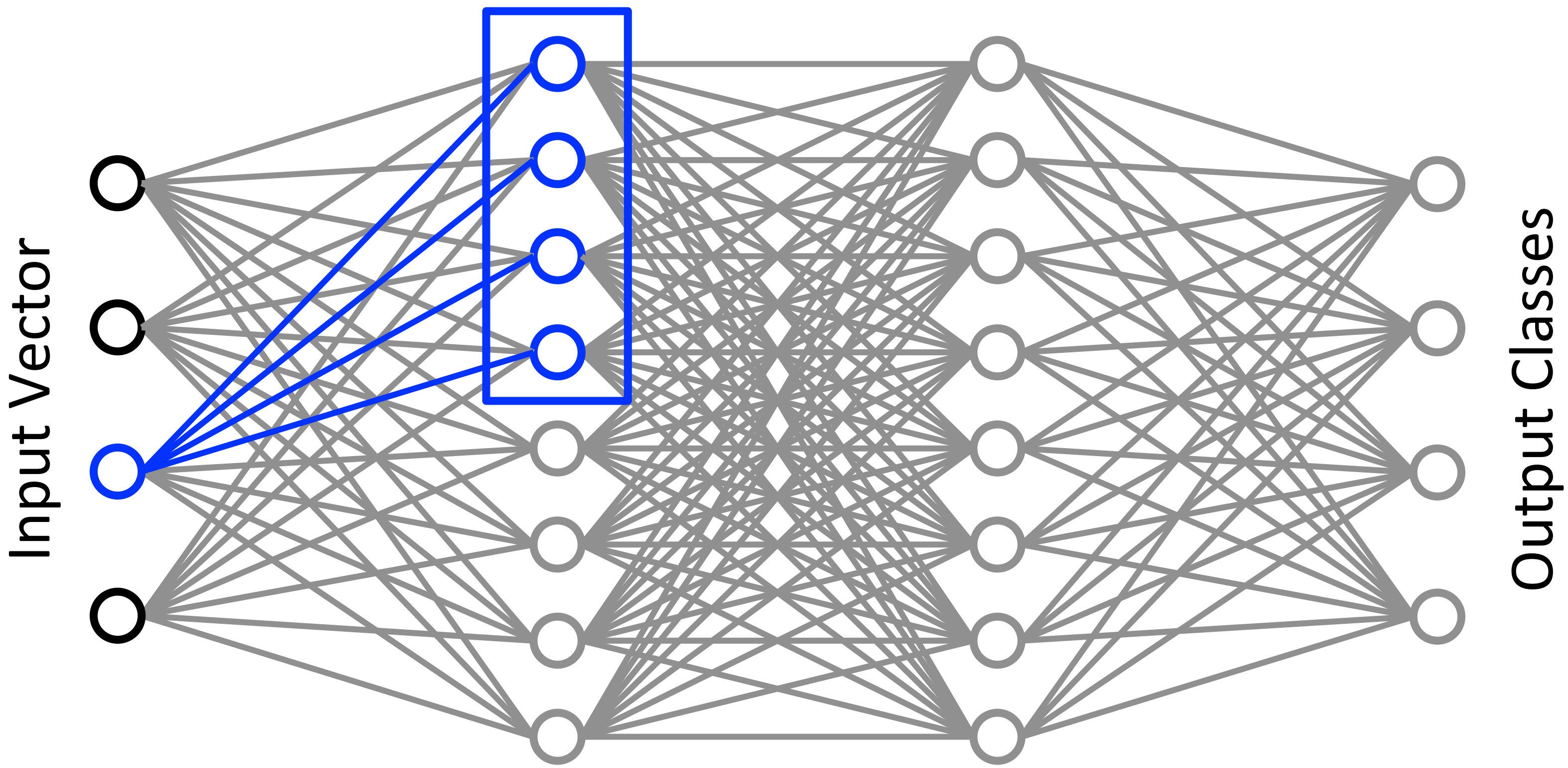
Fully-Connected DNN Graph



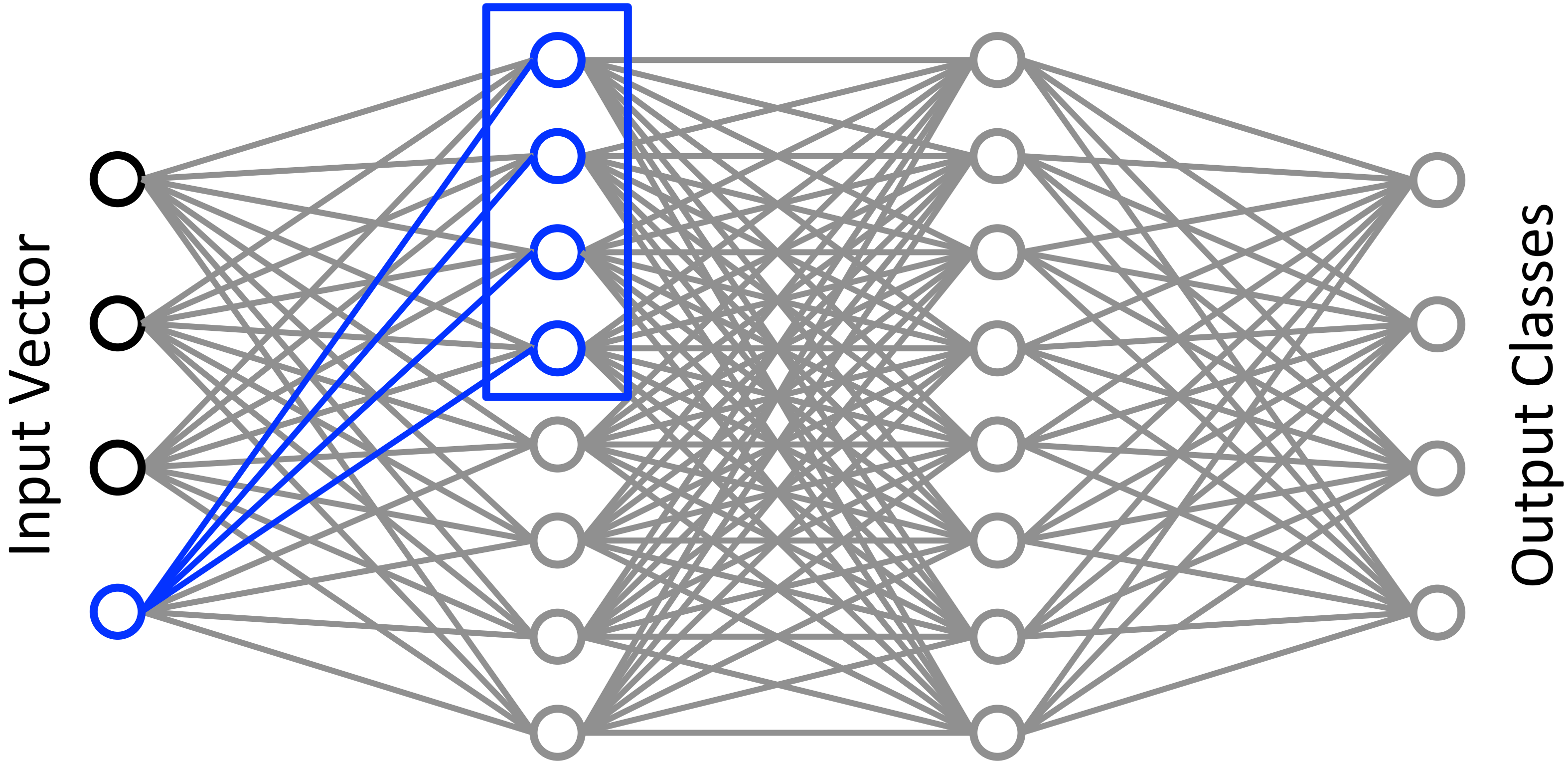
Fully-Connected DNN Graph



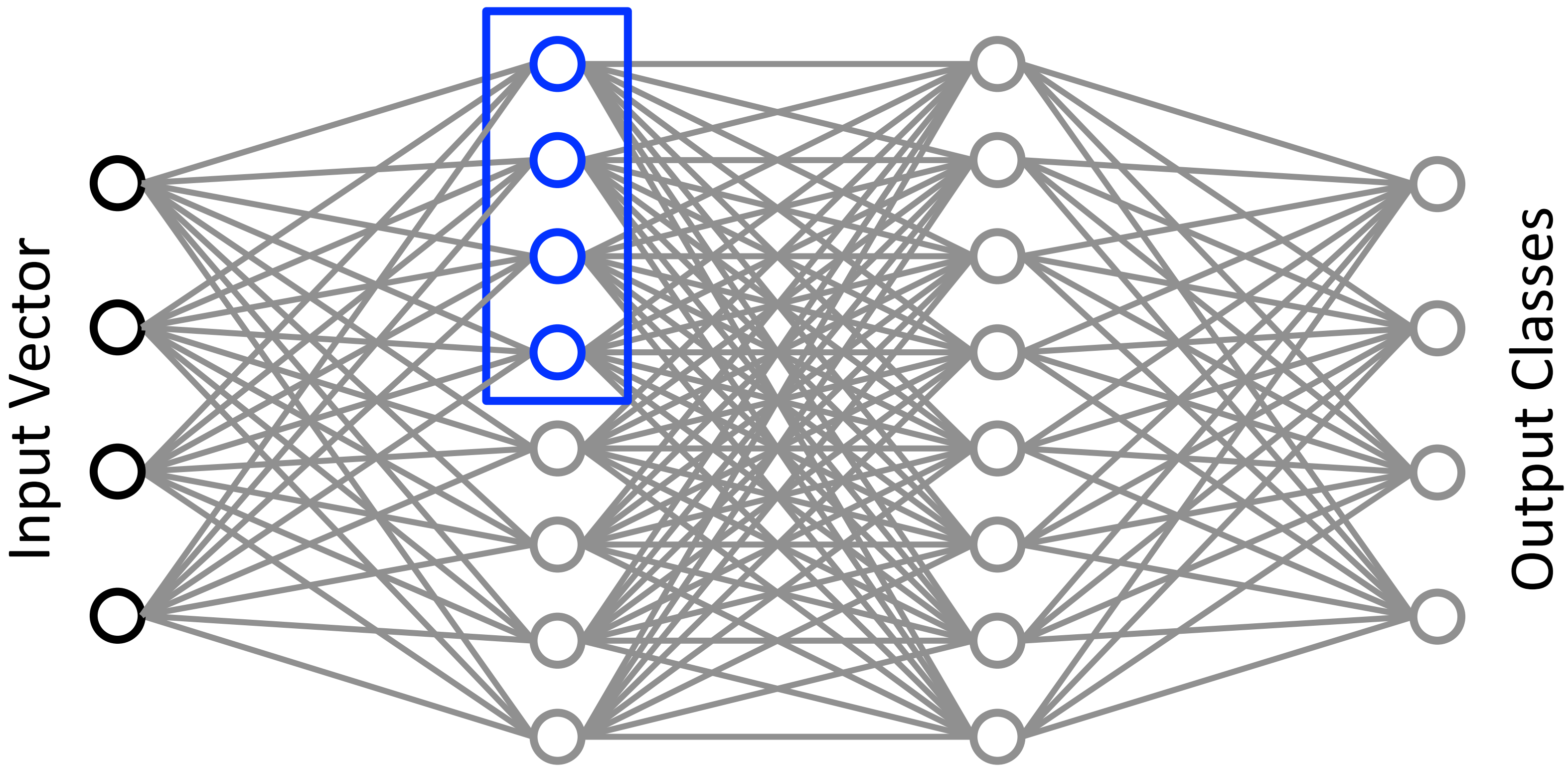
Fully-Connected DNN Graph



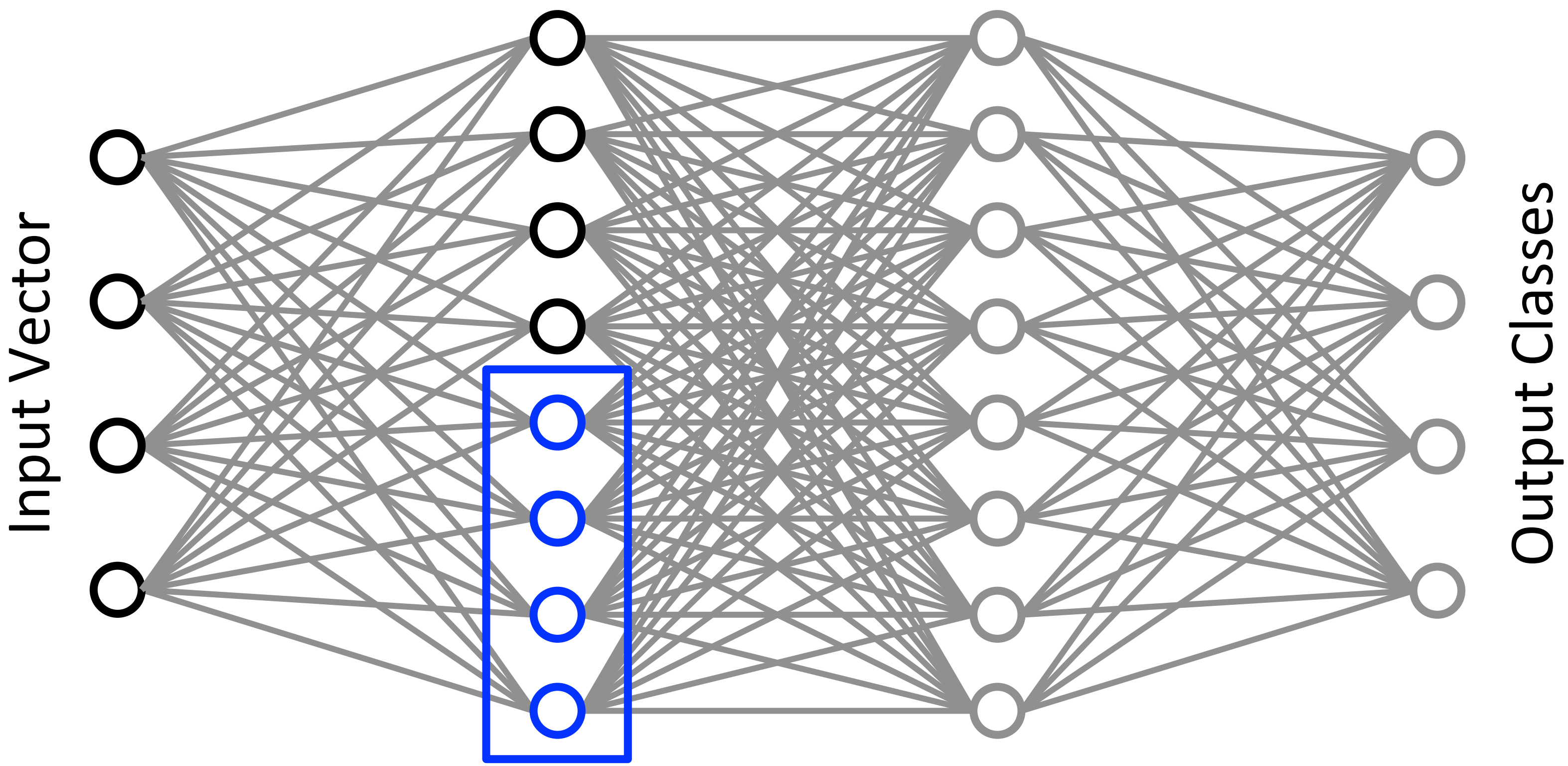
Fully-Connected DNN Graph



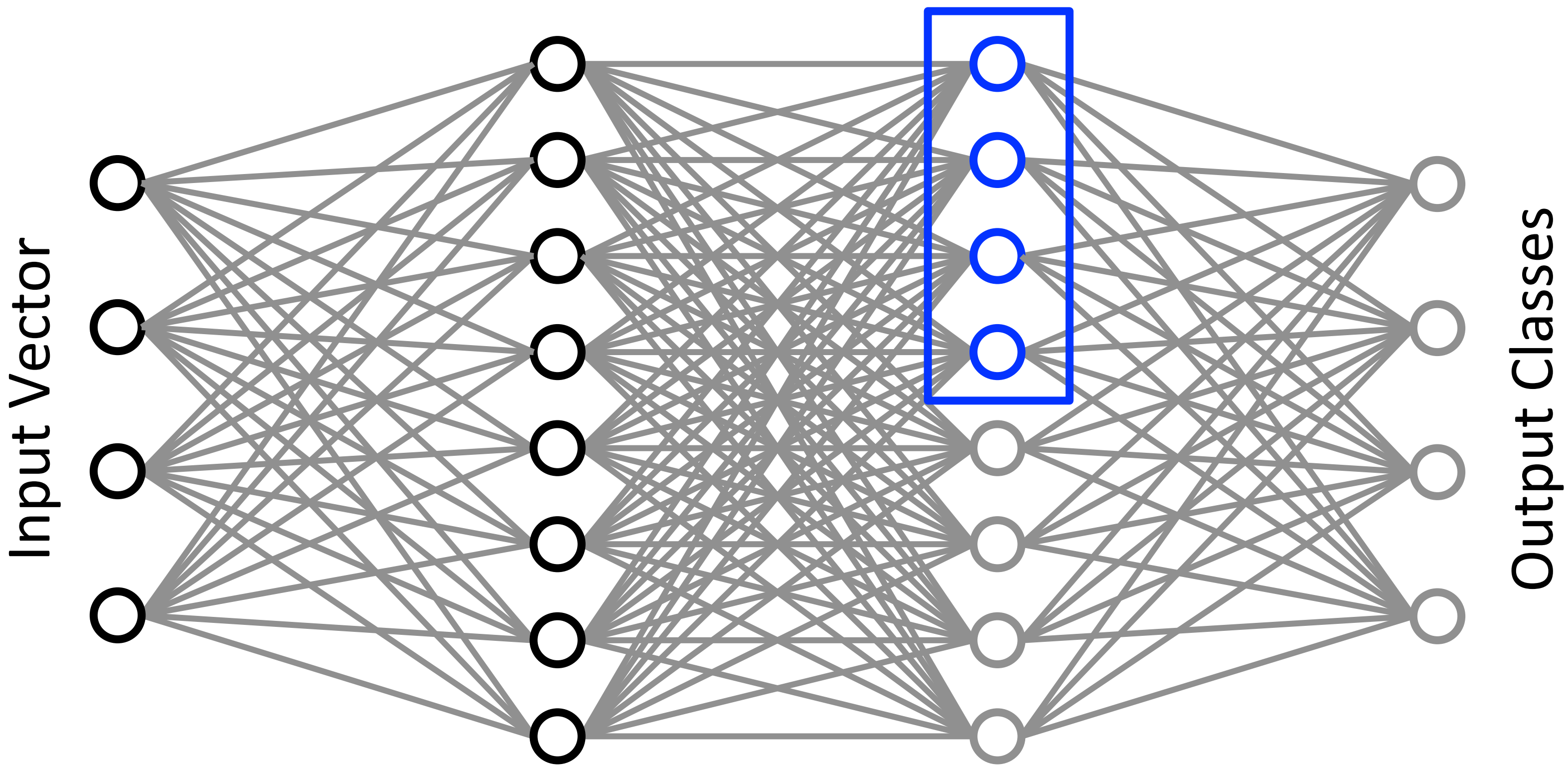
Fully-Connected DNN Graph



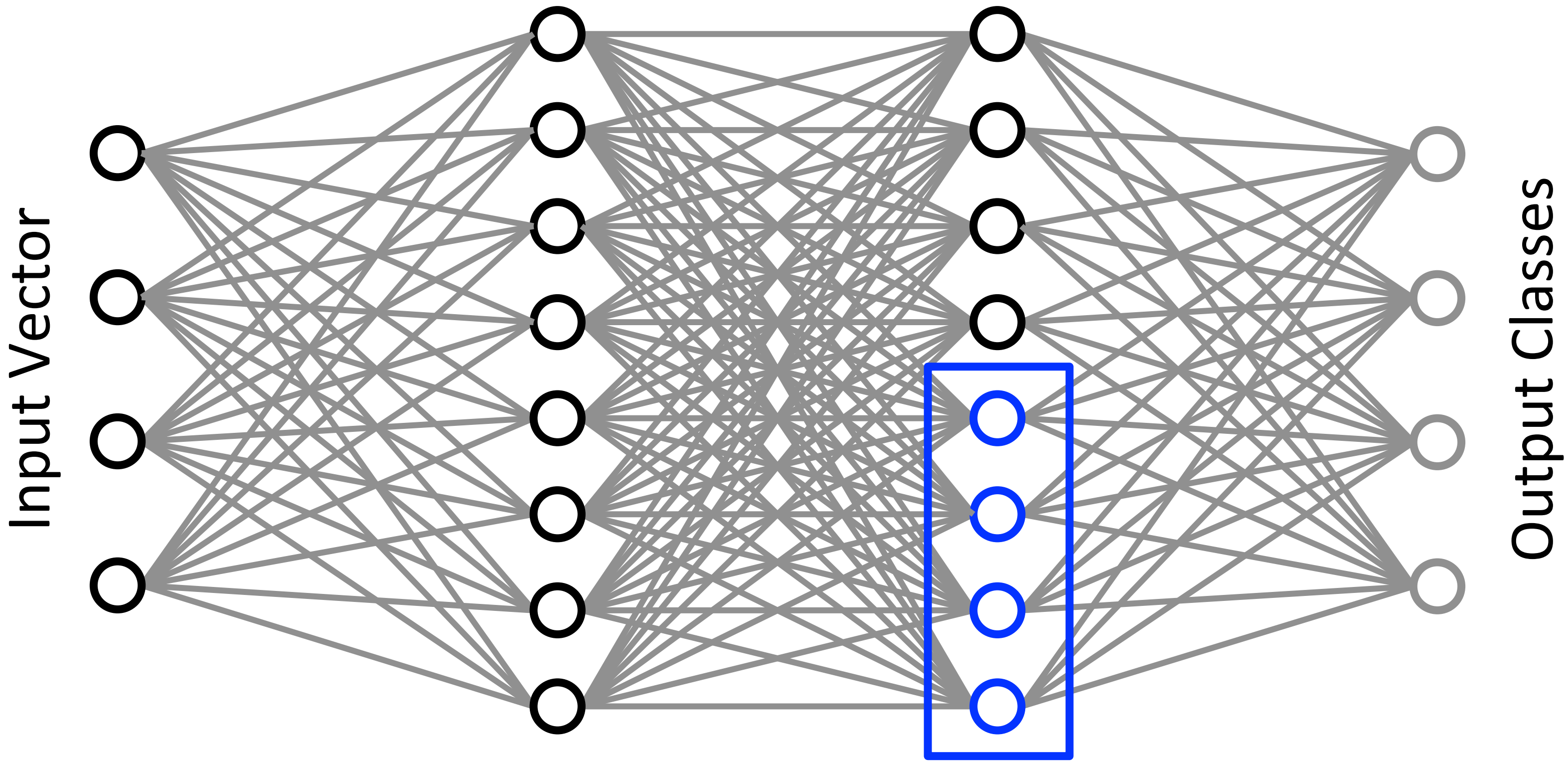
Fully-Connected DNN Graph



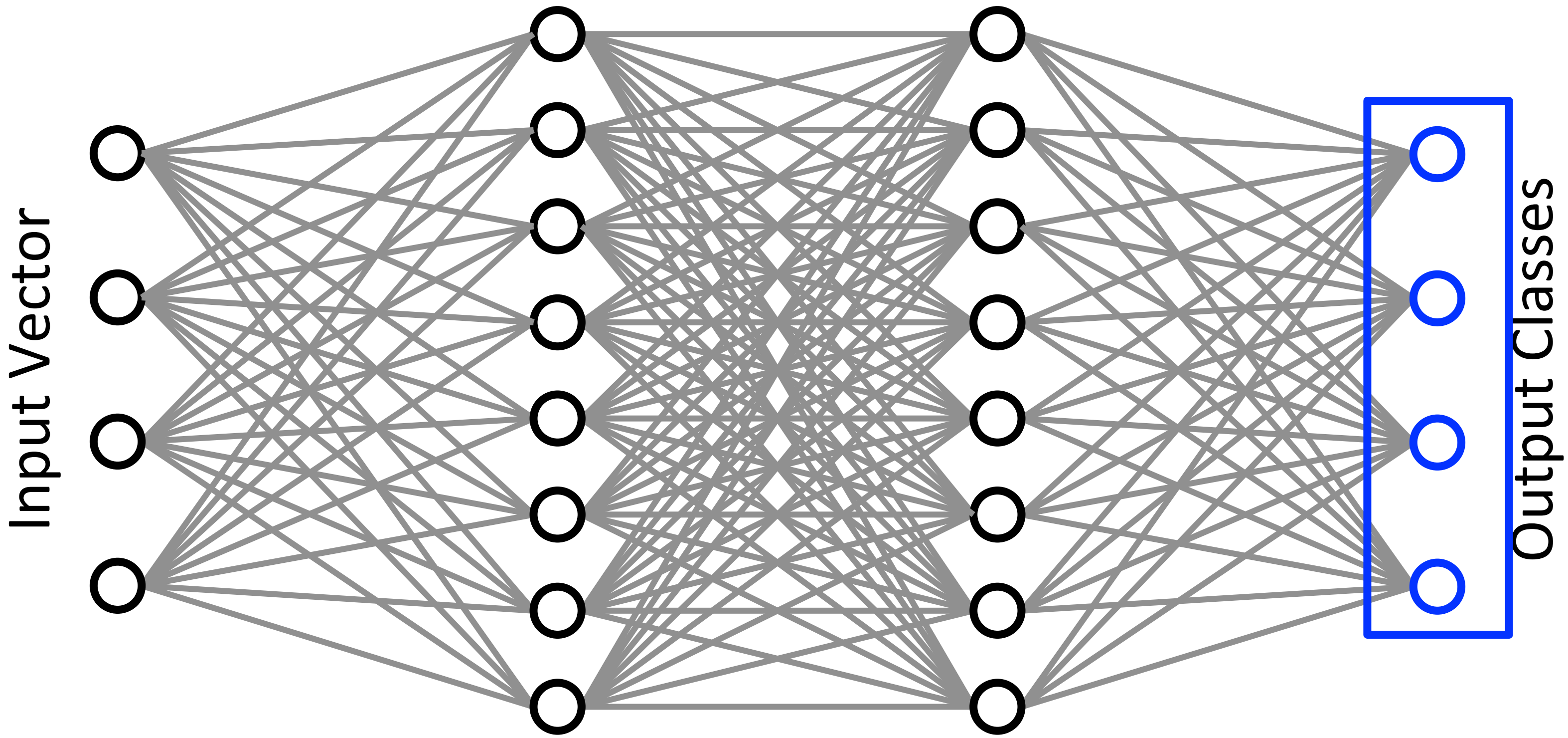
Fully-Connected DNN Graph



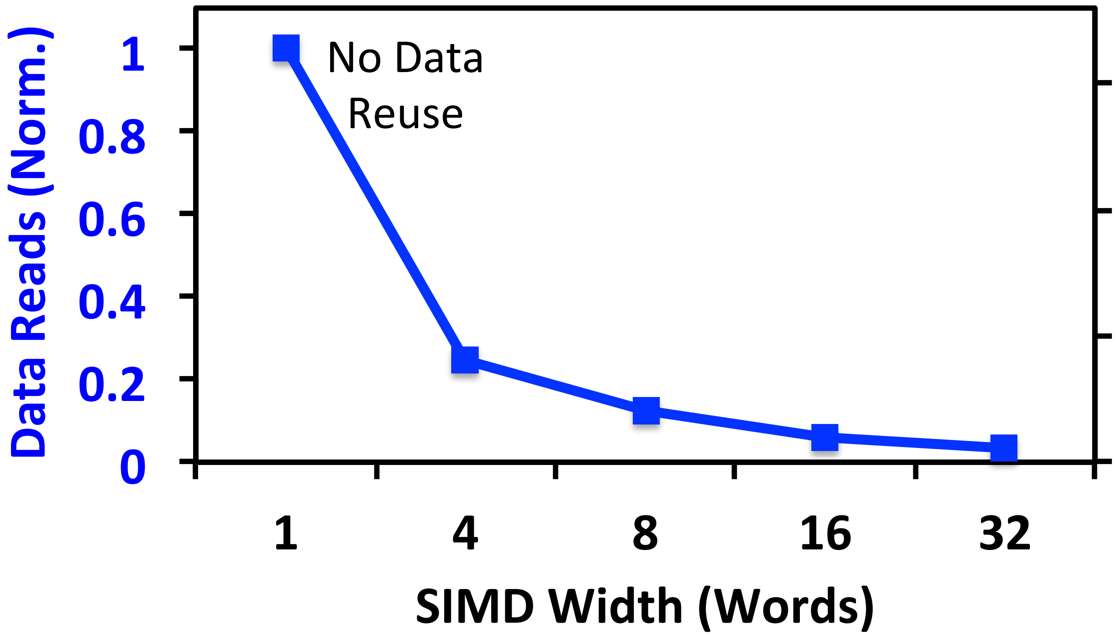
Fully-Connected DNN Graph



Fully-Connected DNN Graph

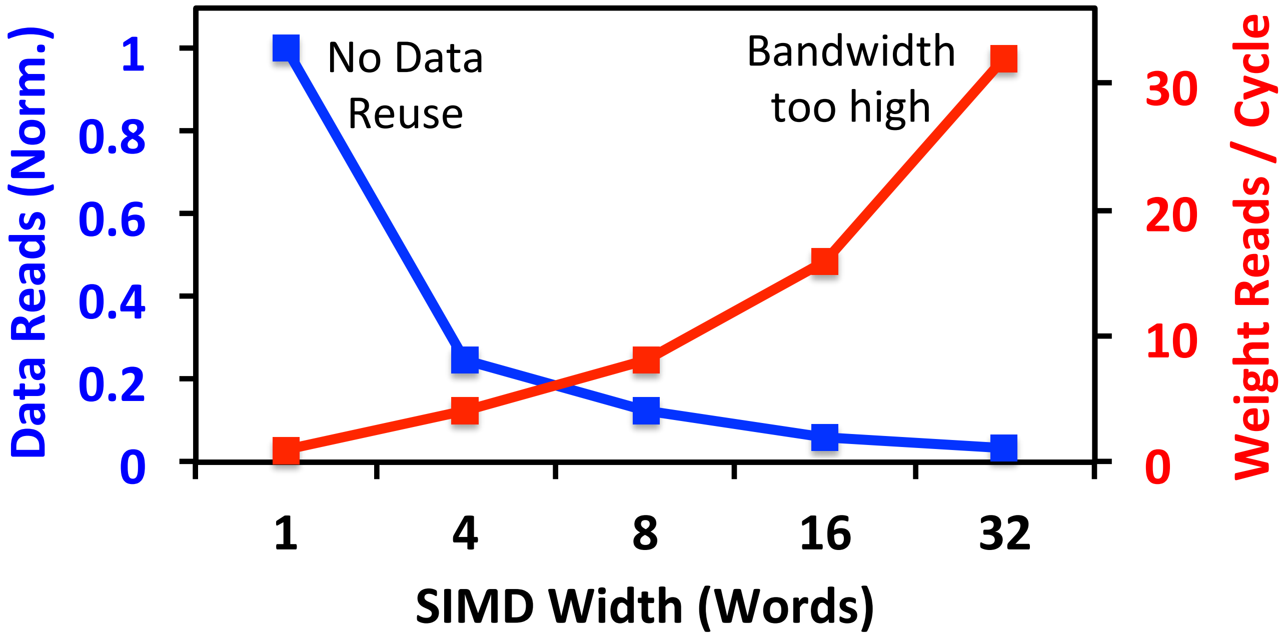


Balancing Efficiency and Bandwidth



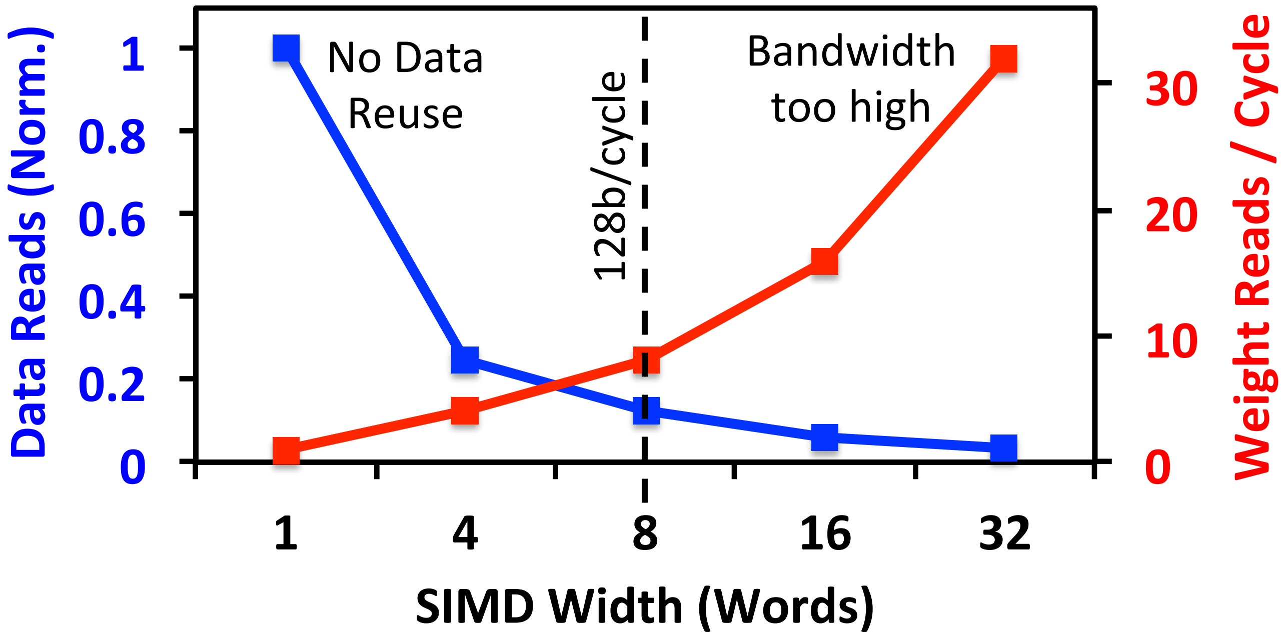
- Parallelism increases throughput and data reuse

Balancing Efficiency and Bandwidth



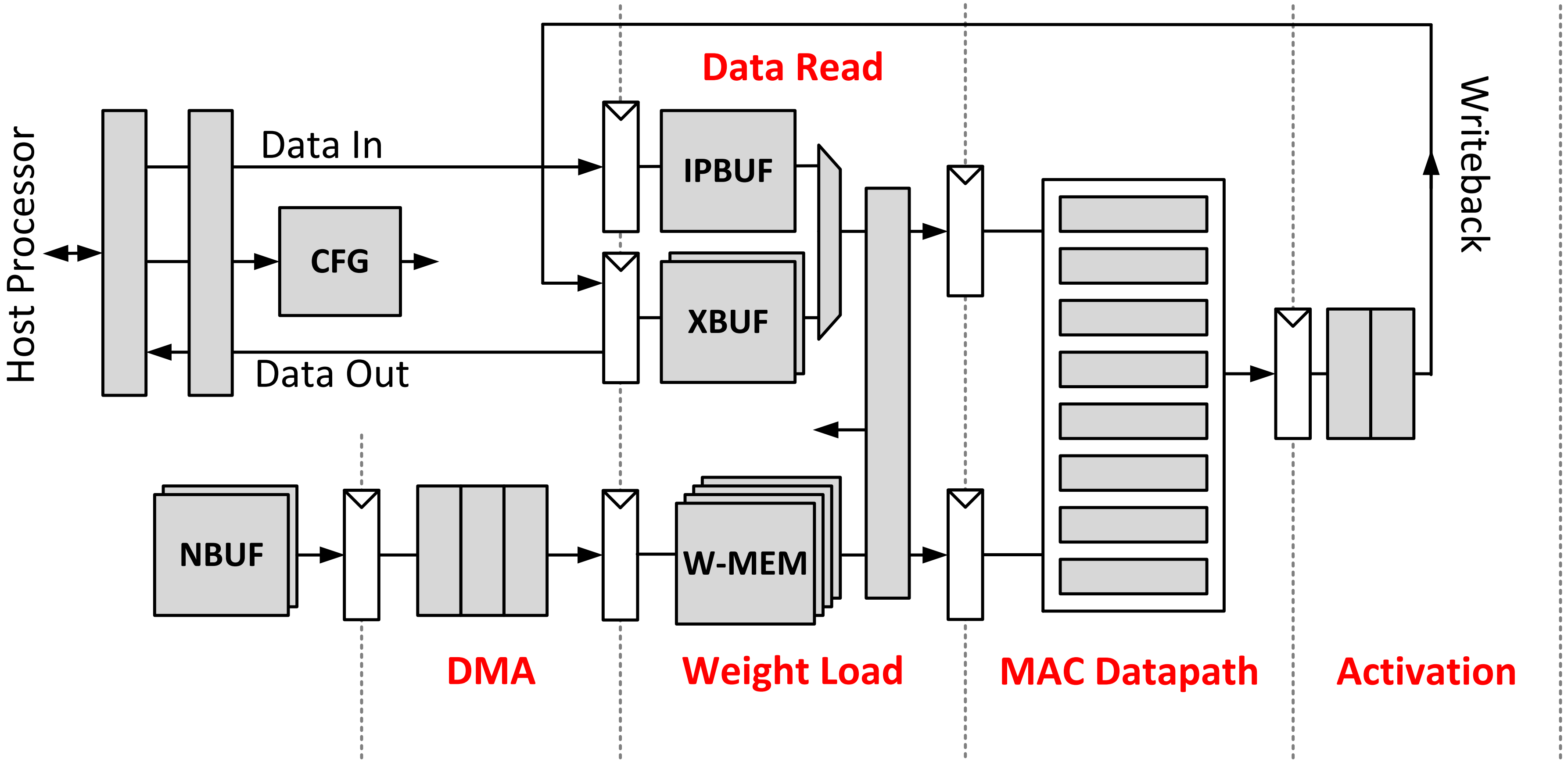
- Parallelism increases throughput and data reuse
  - But also increases Memory Bandwidth demands

Balancing Efficiency and Bandwidth

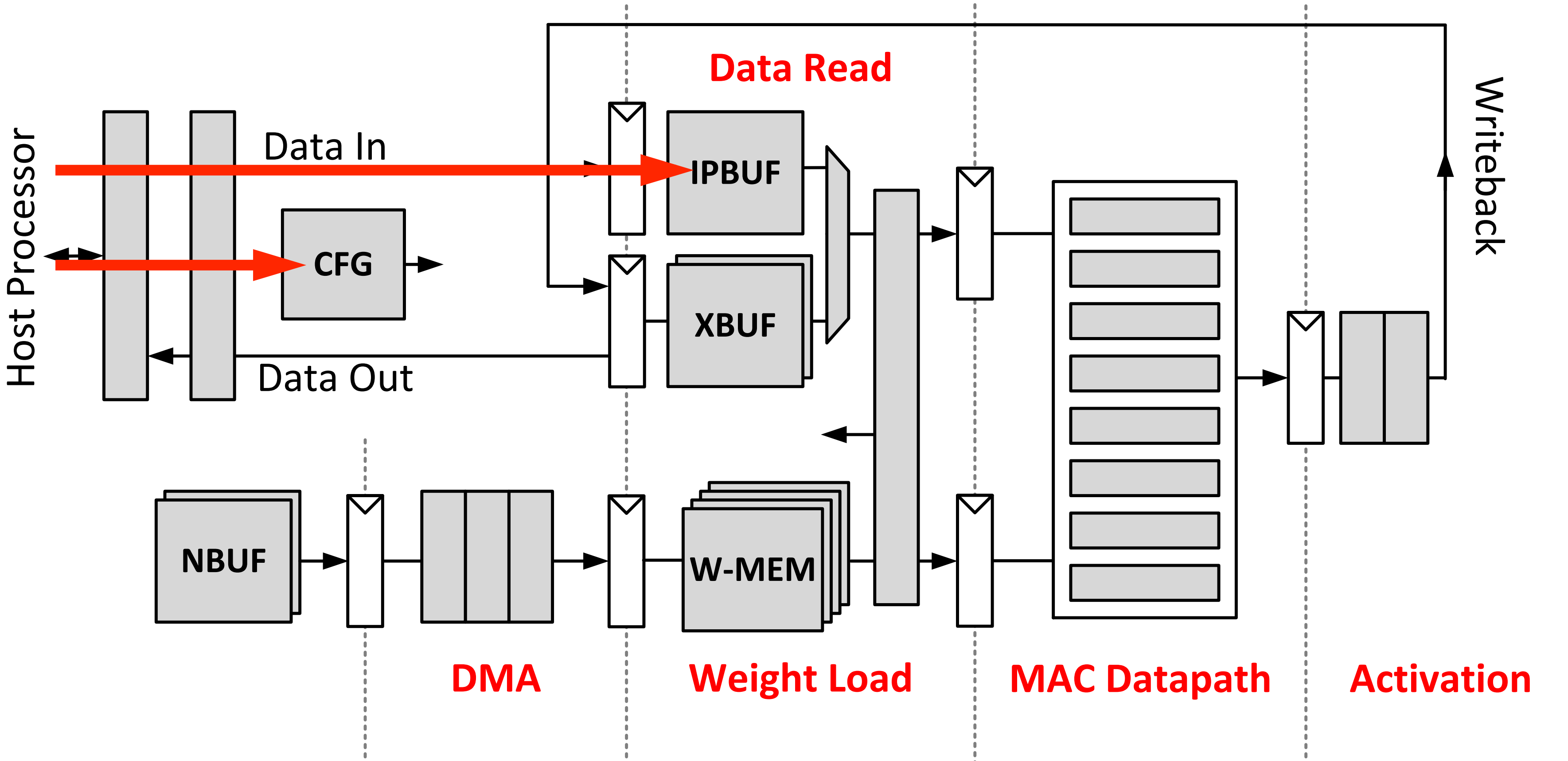


- Parallelism increases throughput and data reuse
  - But also increases Memory Bandwidth demands
- 8-Way SIMD is efficient at reasonable memory BW
  - 10x Activation reuse, with 128b AXI channel

DNN ENGINE Micro-Architecture

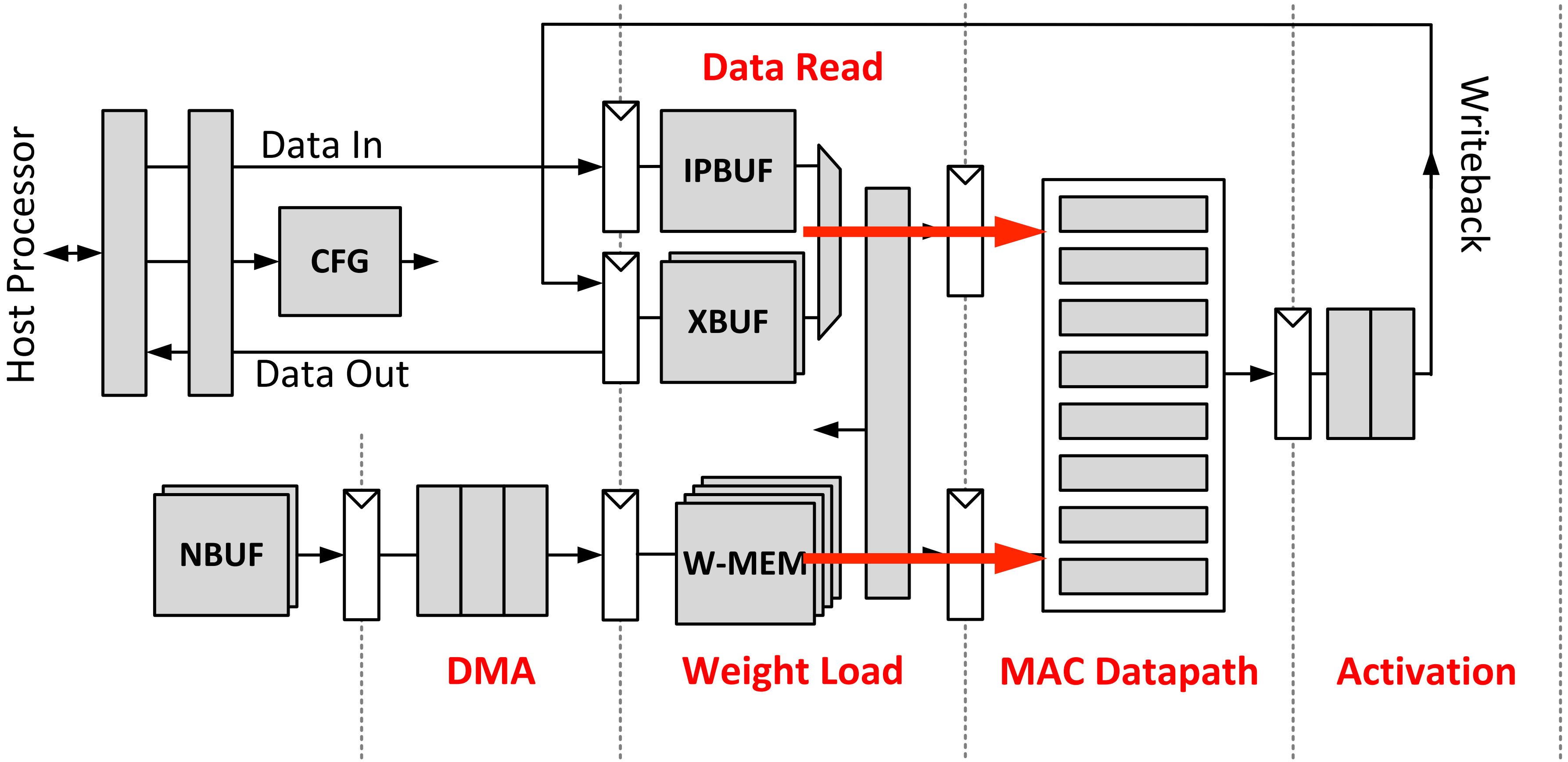


DNN ENGINE Micro-Architecture



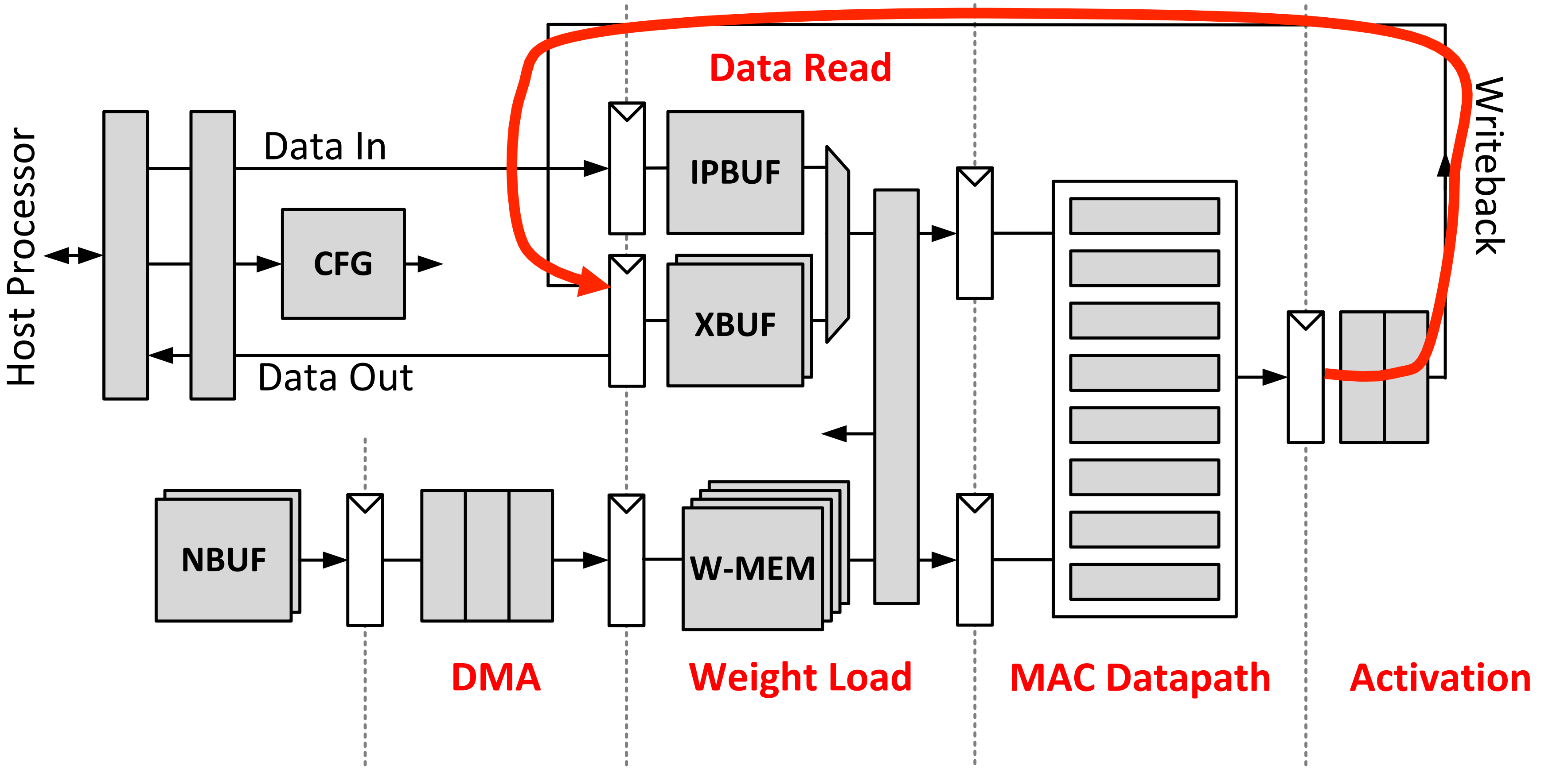
- Host Processor loads configuration and input data

DNN ENGINE Micro-Architecture



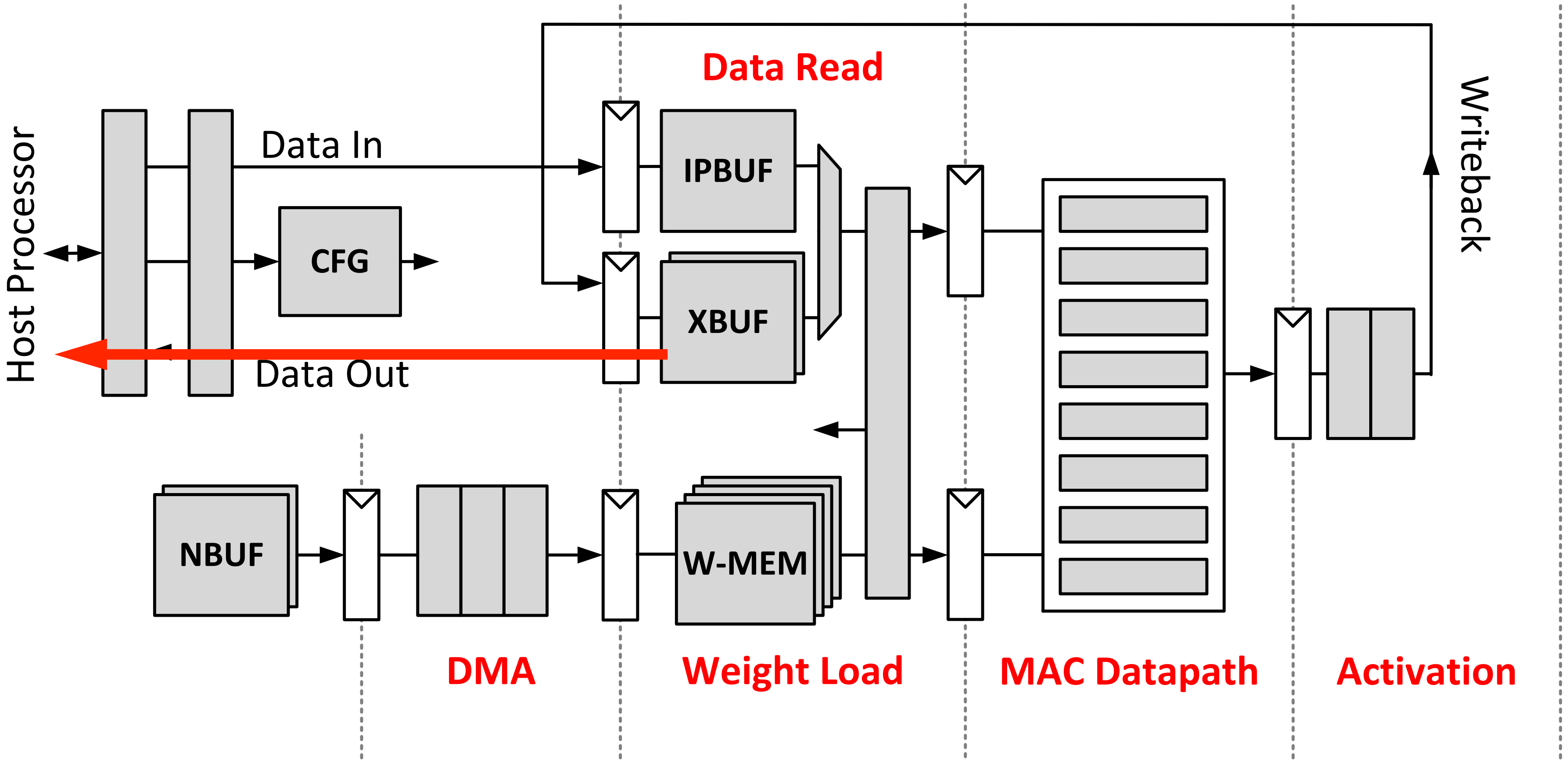
- Accumulate products of Activation and Weights

DNN ENGINE Micro-Architecture



- Add bias, apply ReLU activation and writeback

DNN ENGINE Micro-Architecture

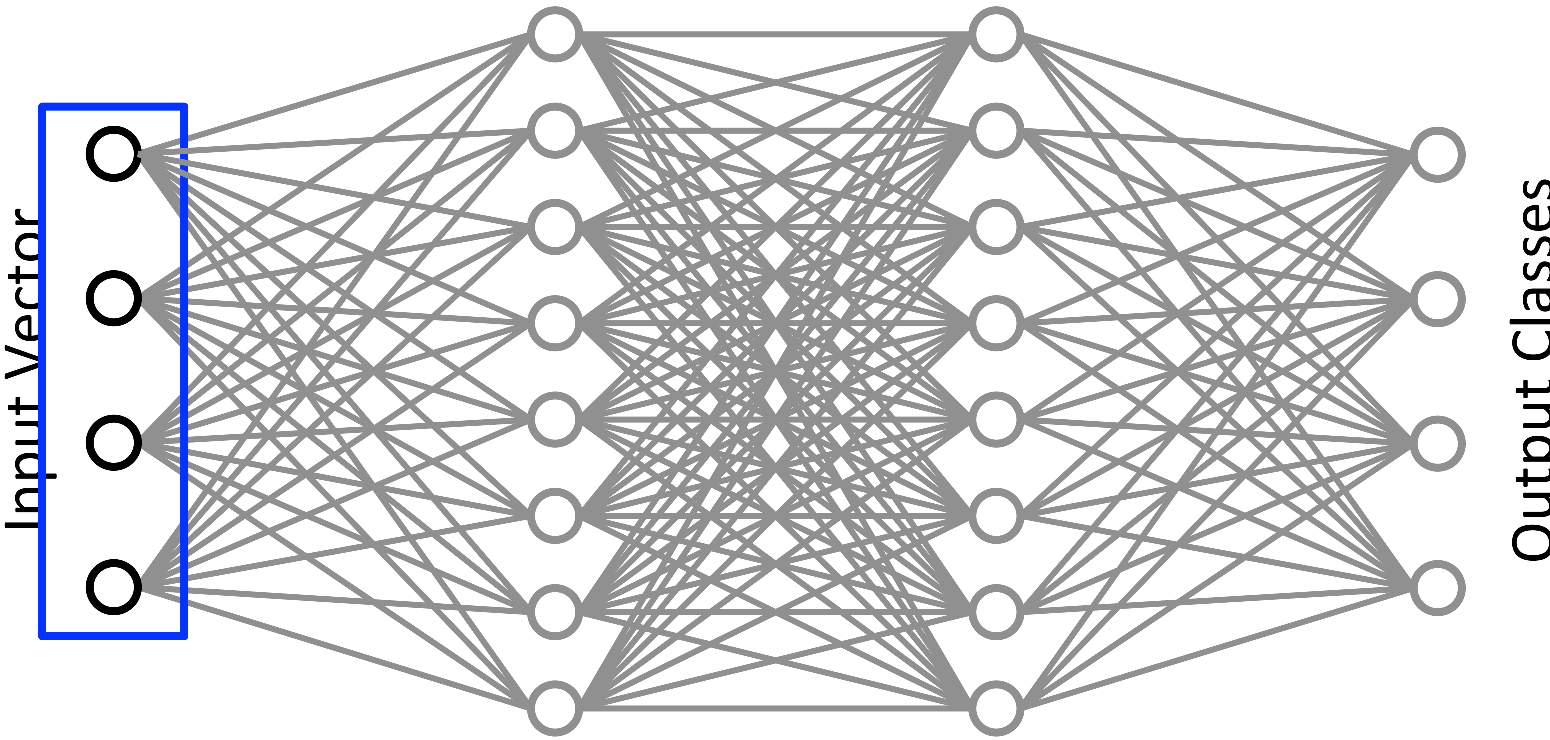


- IRQ to host, which retrieves prediction data

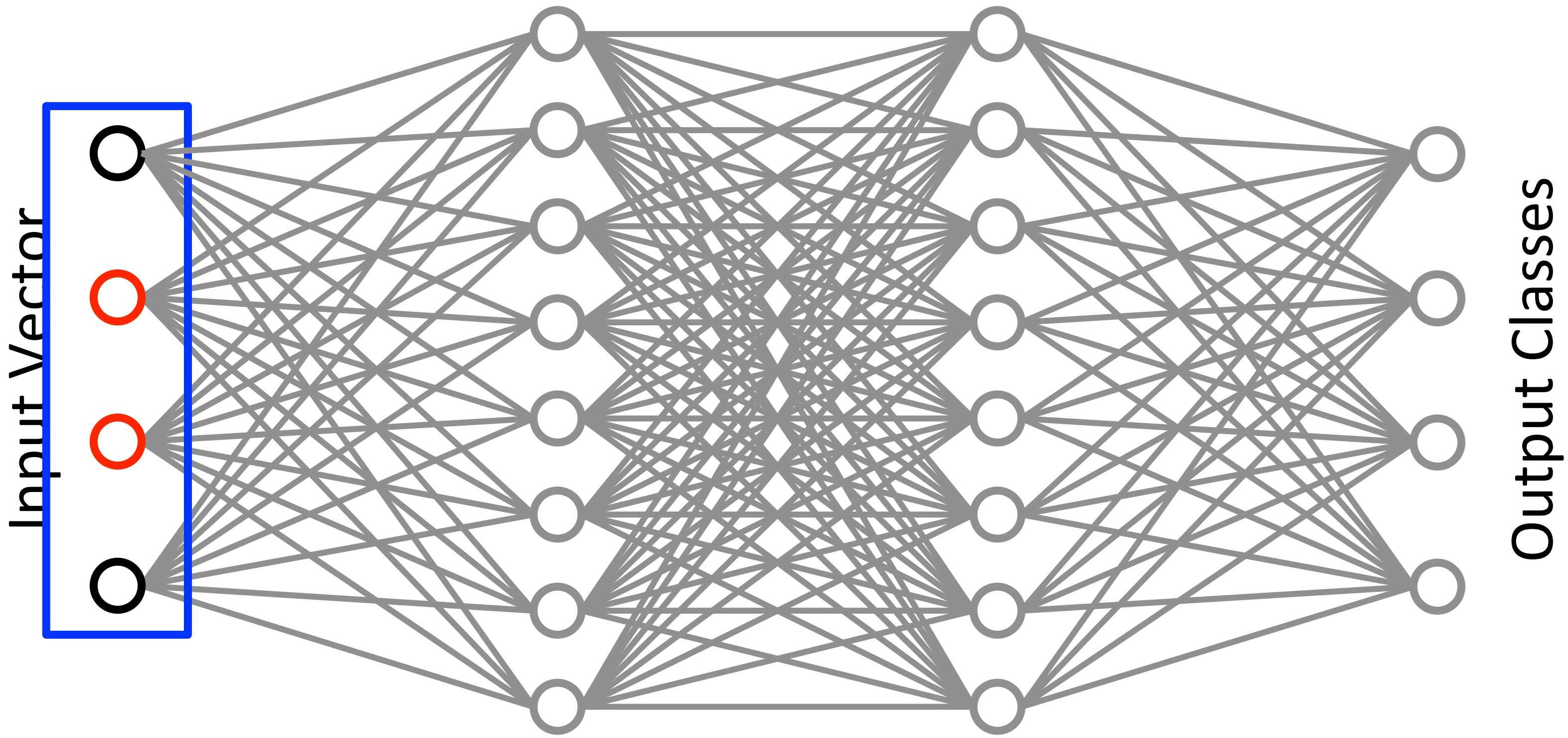
Outline

- Motivation
- DNN Engine
  - Parallelism
  - Sparsity
  - Resilience
- Measurement Results
- Summary

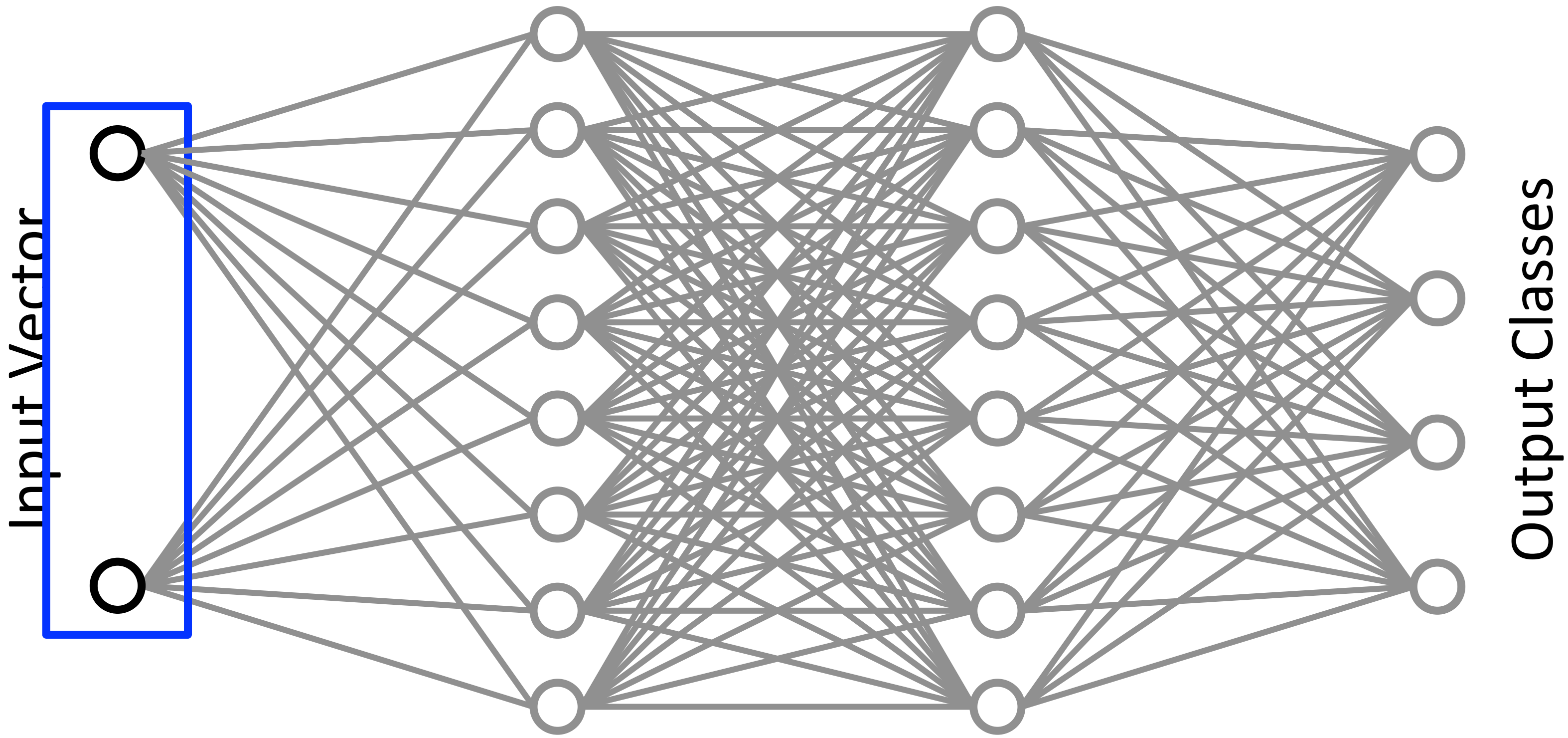
Exploiting Sparse Data in DNNs



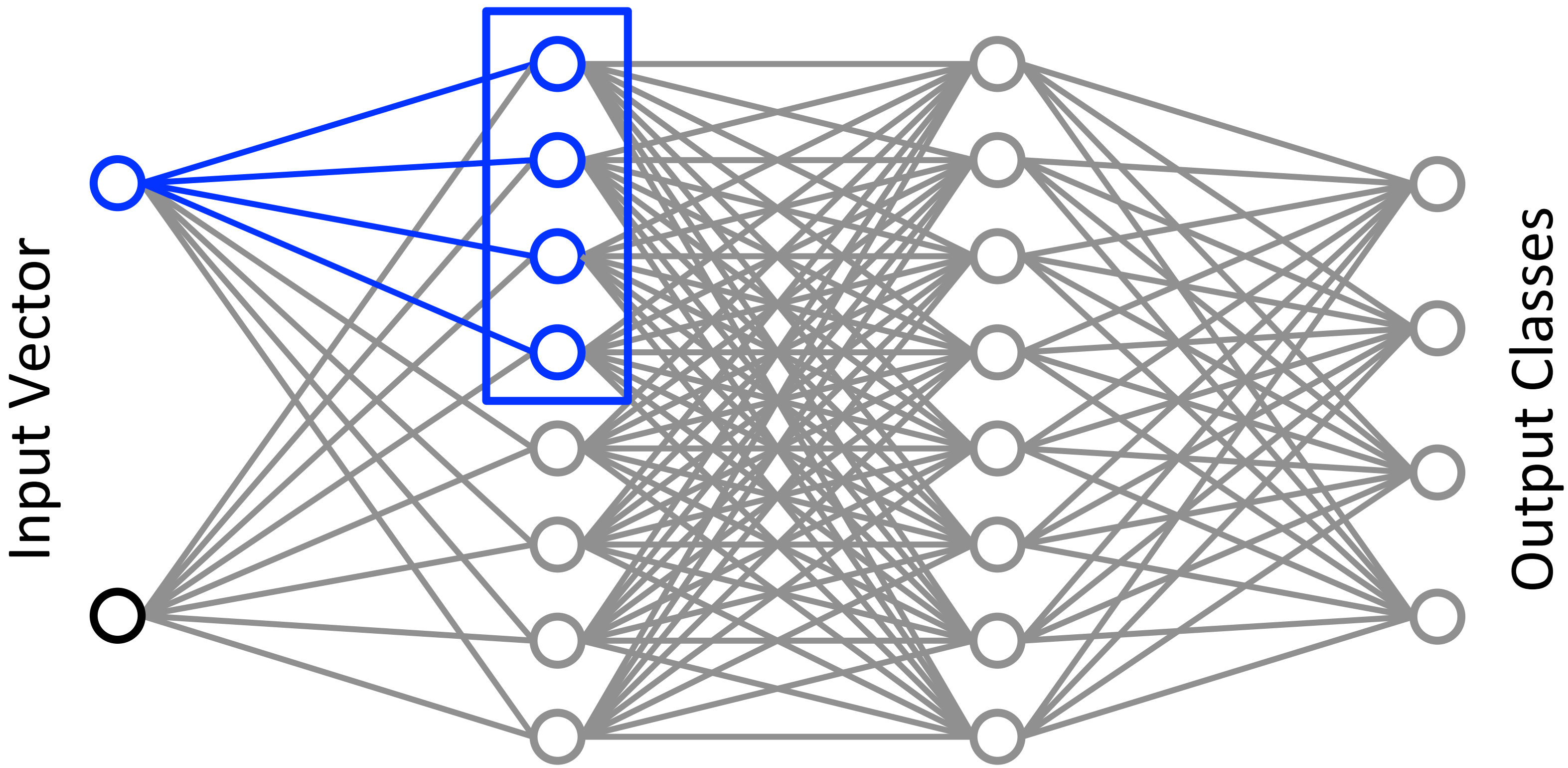
Exploiting Sparse Data in DNNs



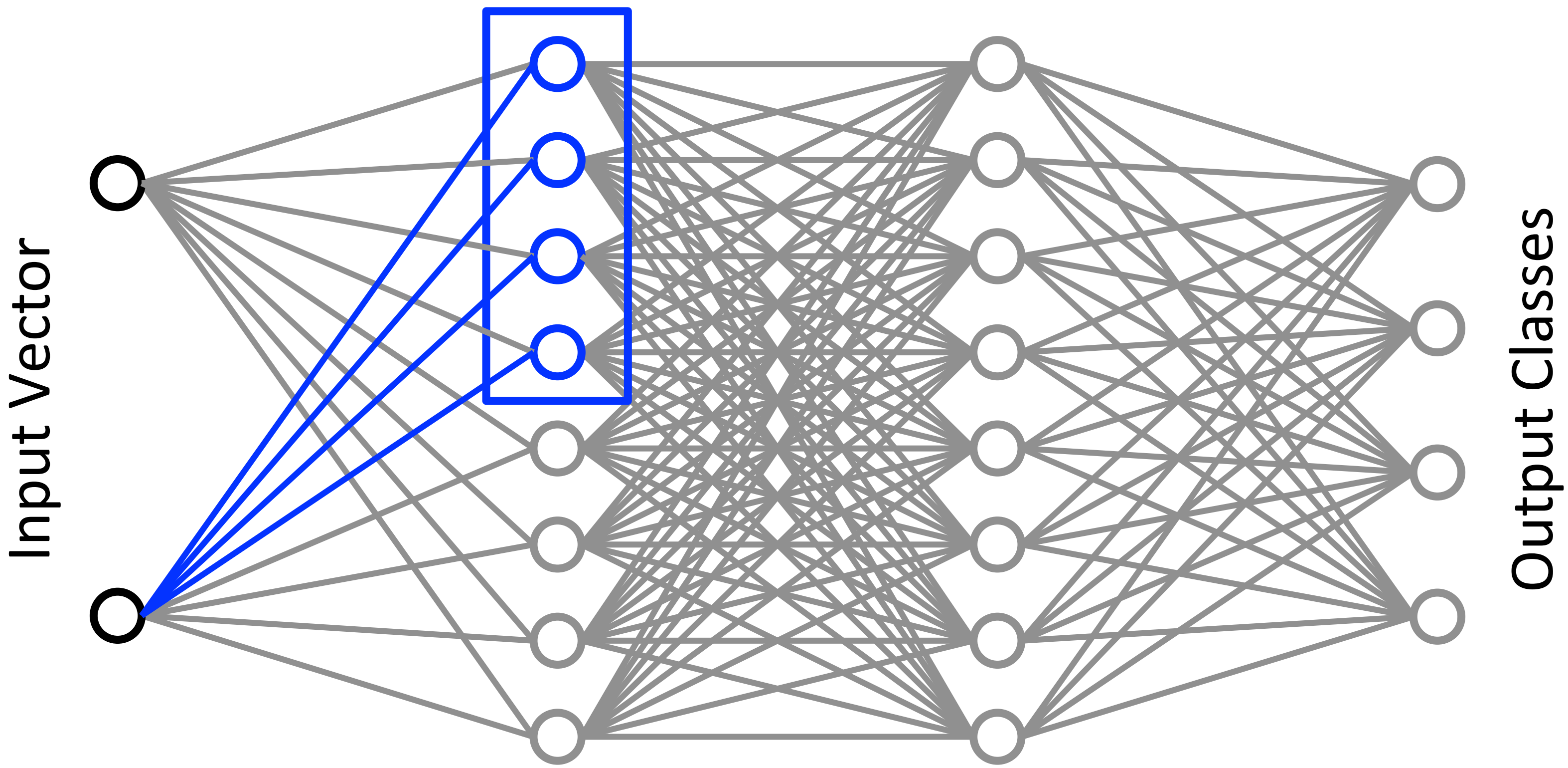
Exploiting Sparse Data in DNNs



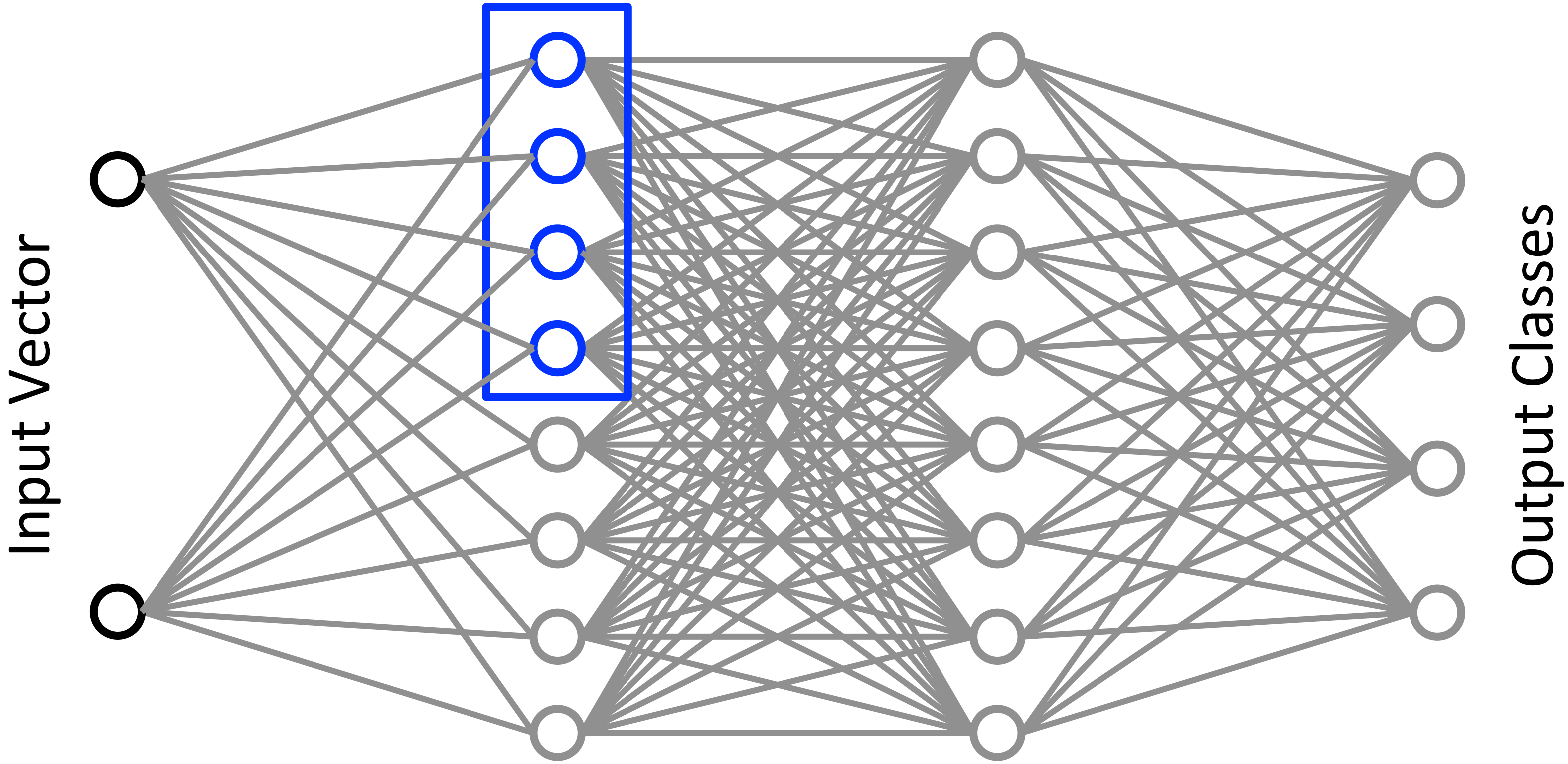
Exploiting Sparse Data in DNNs



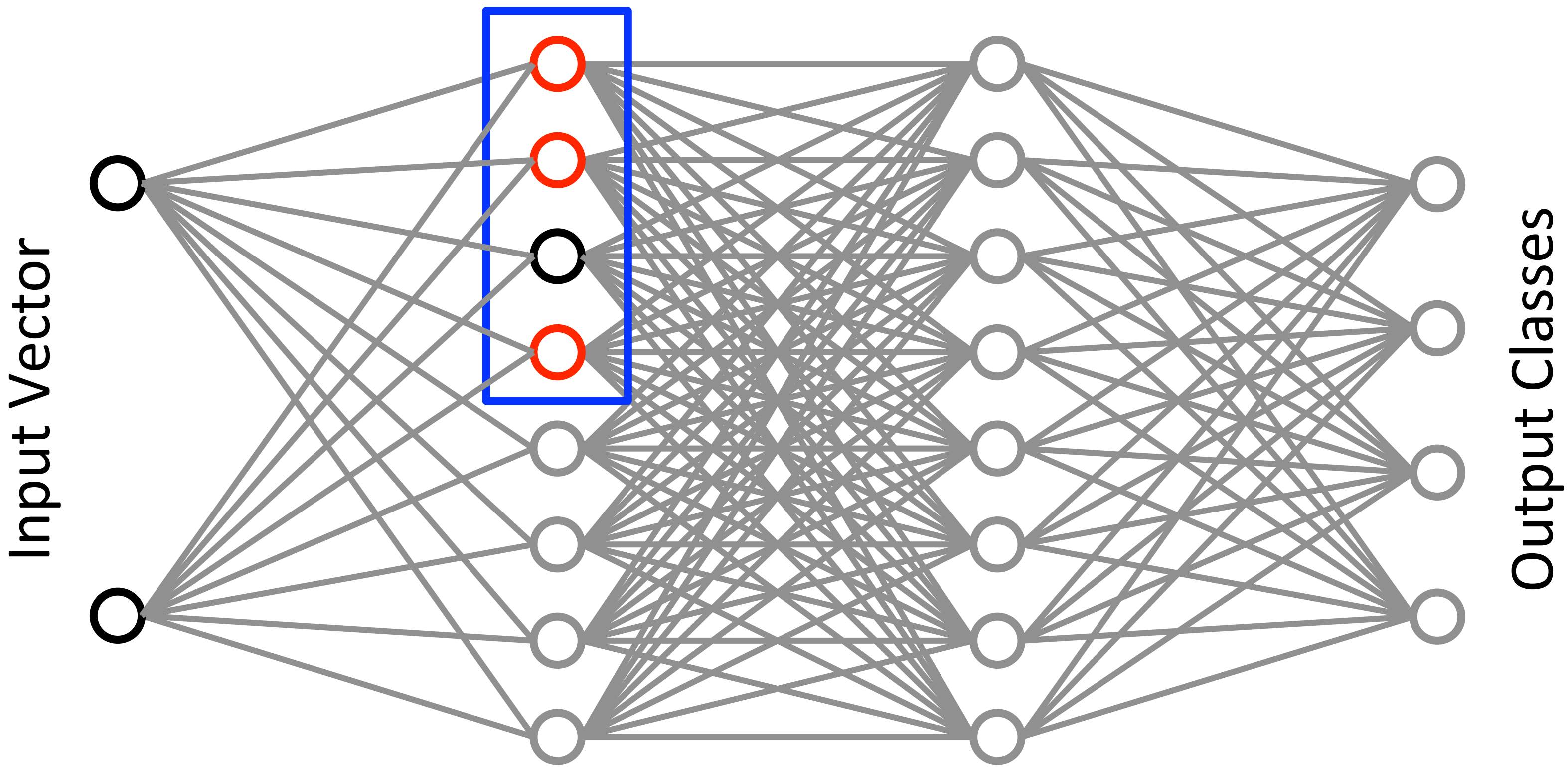
Exploiting Sparse Data in DNNs



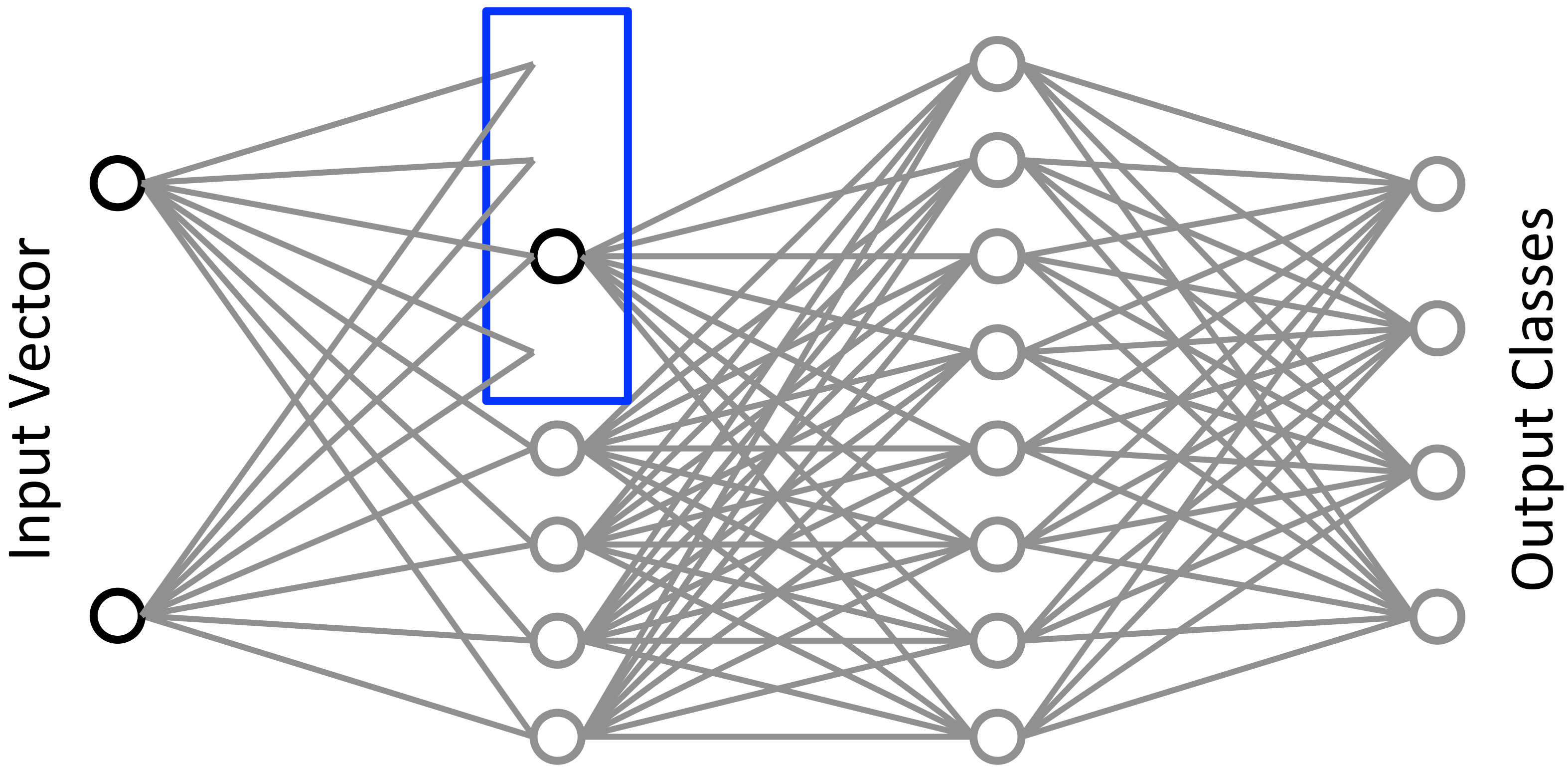
Exploiting Sparse Data in DNNs



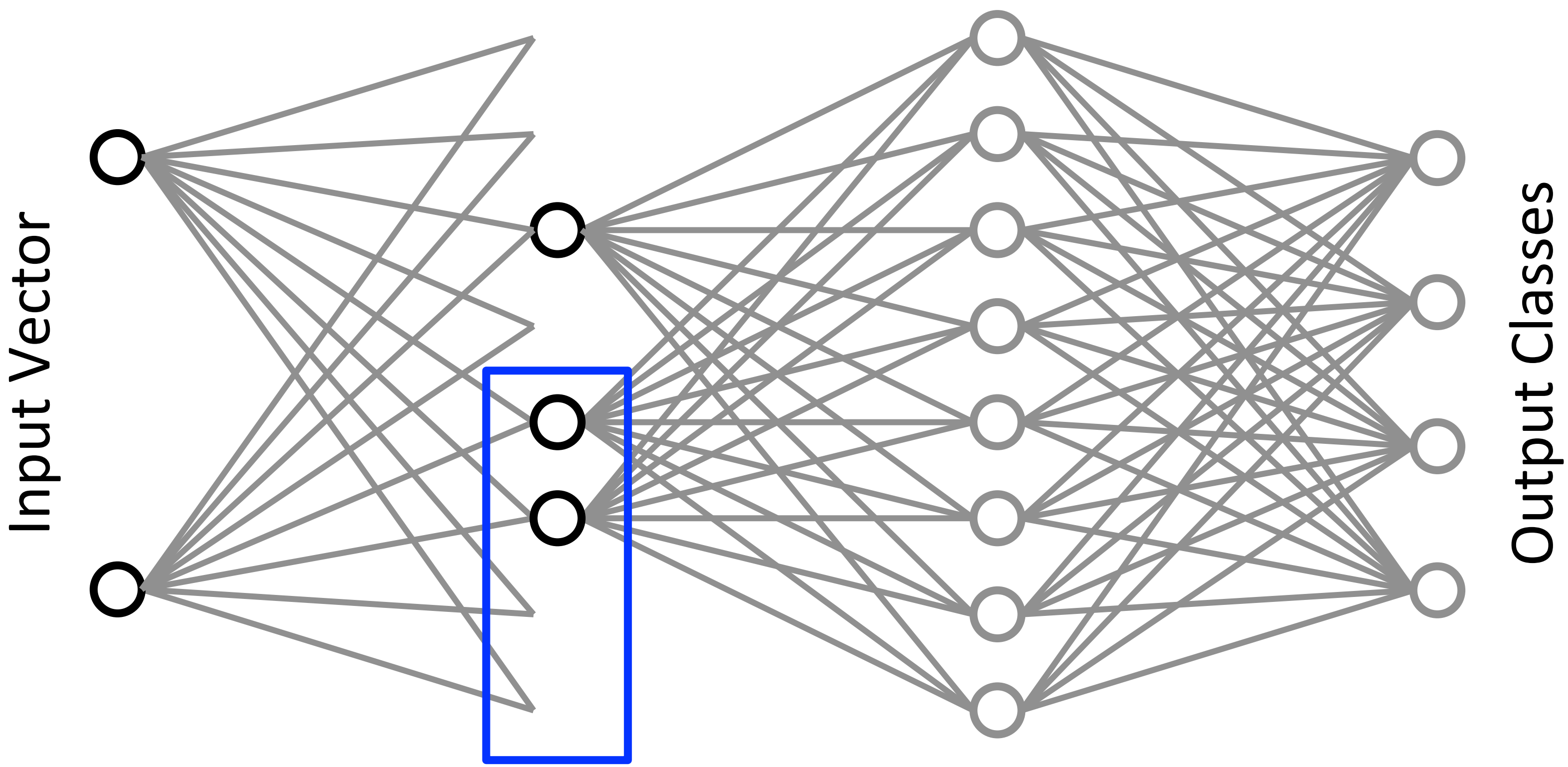
Exploiting Sparse Data in DNNs



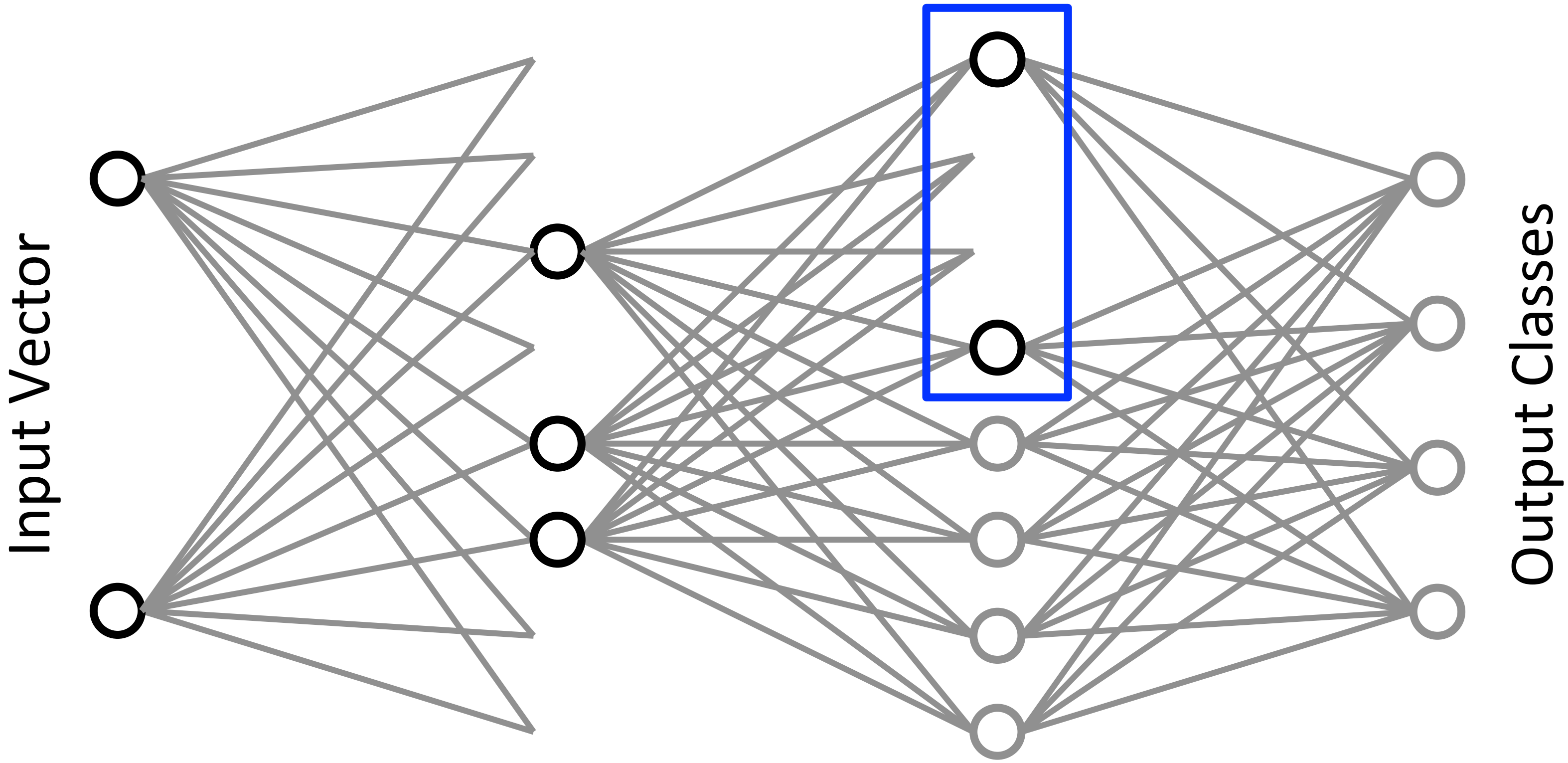
Exploiting Sparse Data in DNNs



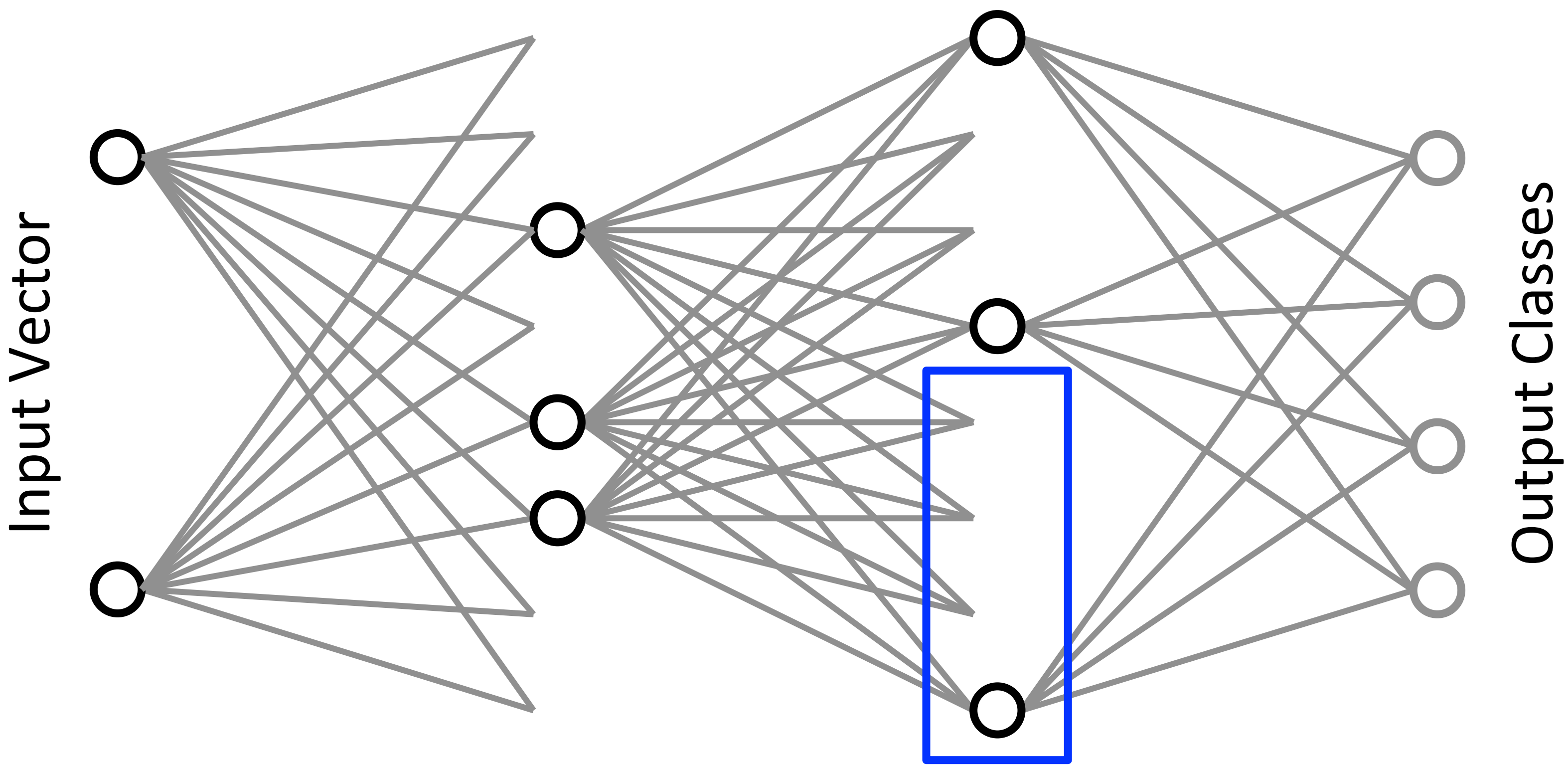
Exploiting Sparse Data in DNNs



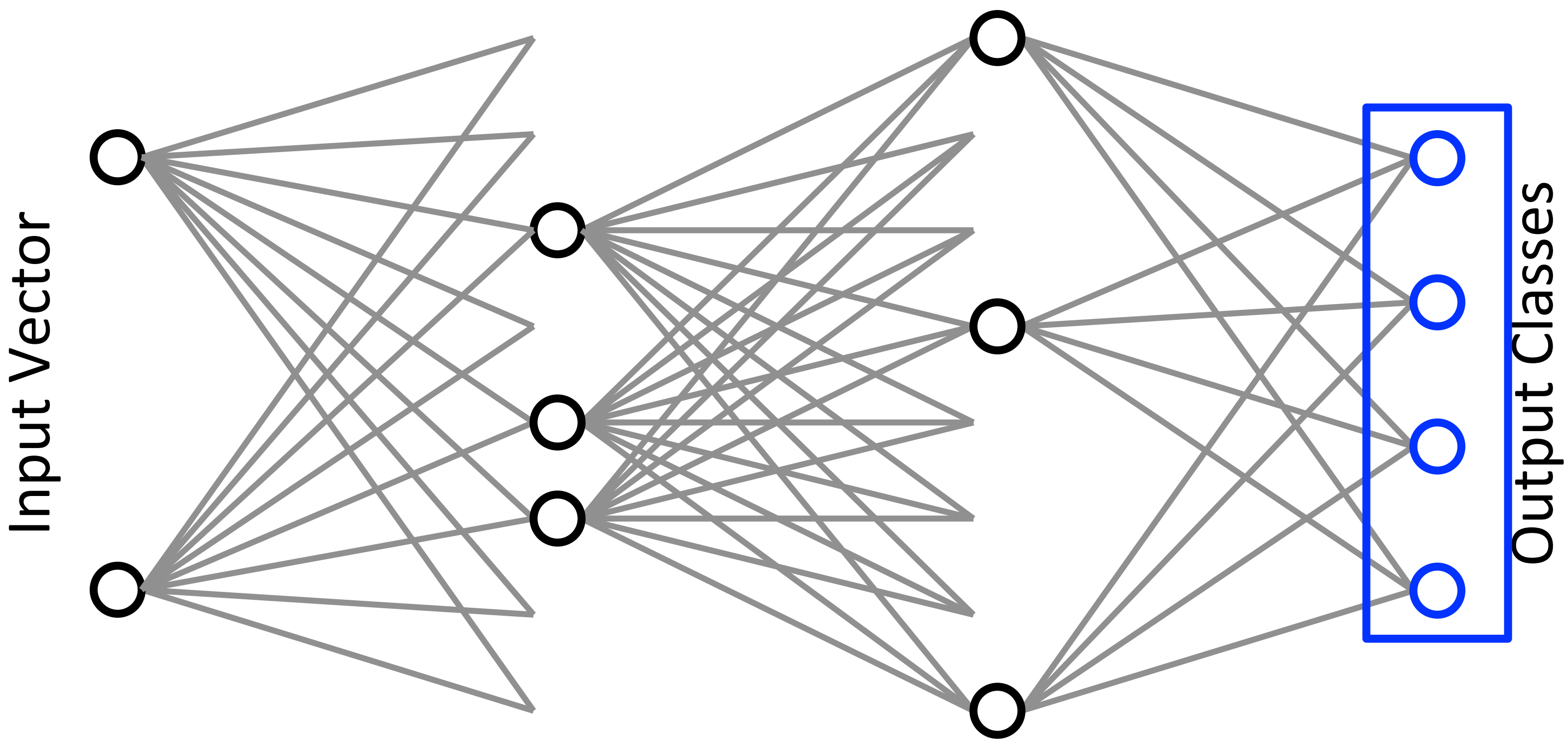
Exploiting Sparse Data in DNNs



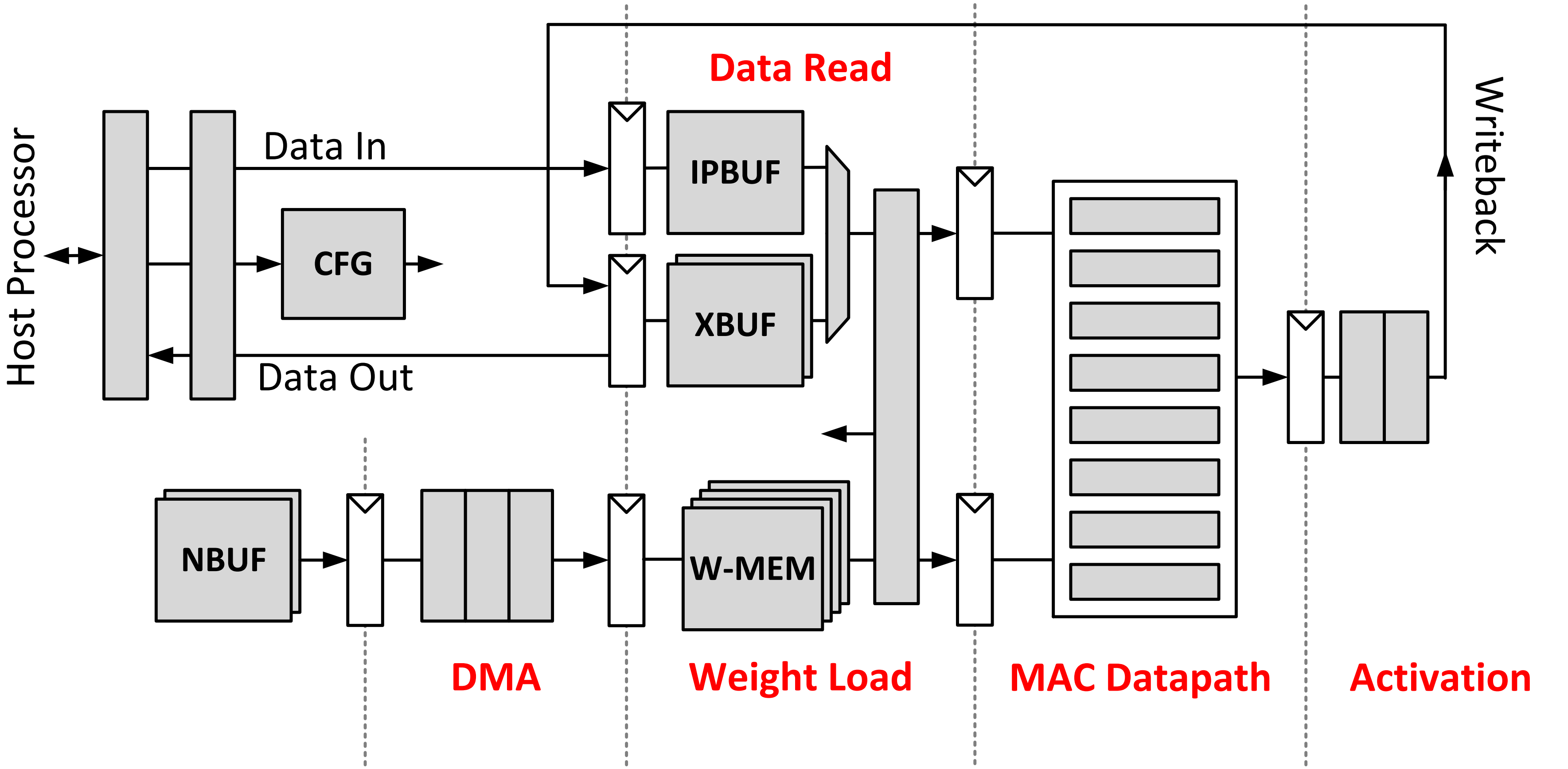
Exploiting Sparse Data in DNNs



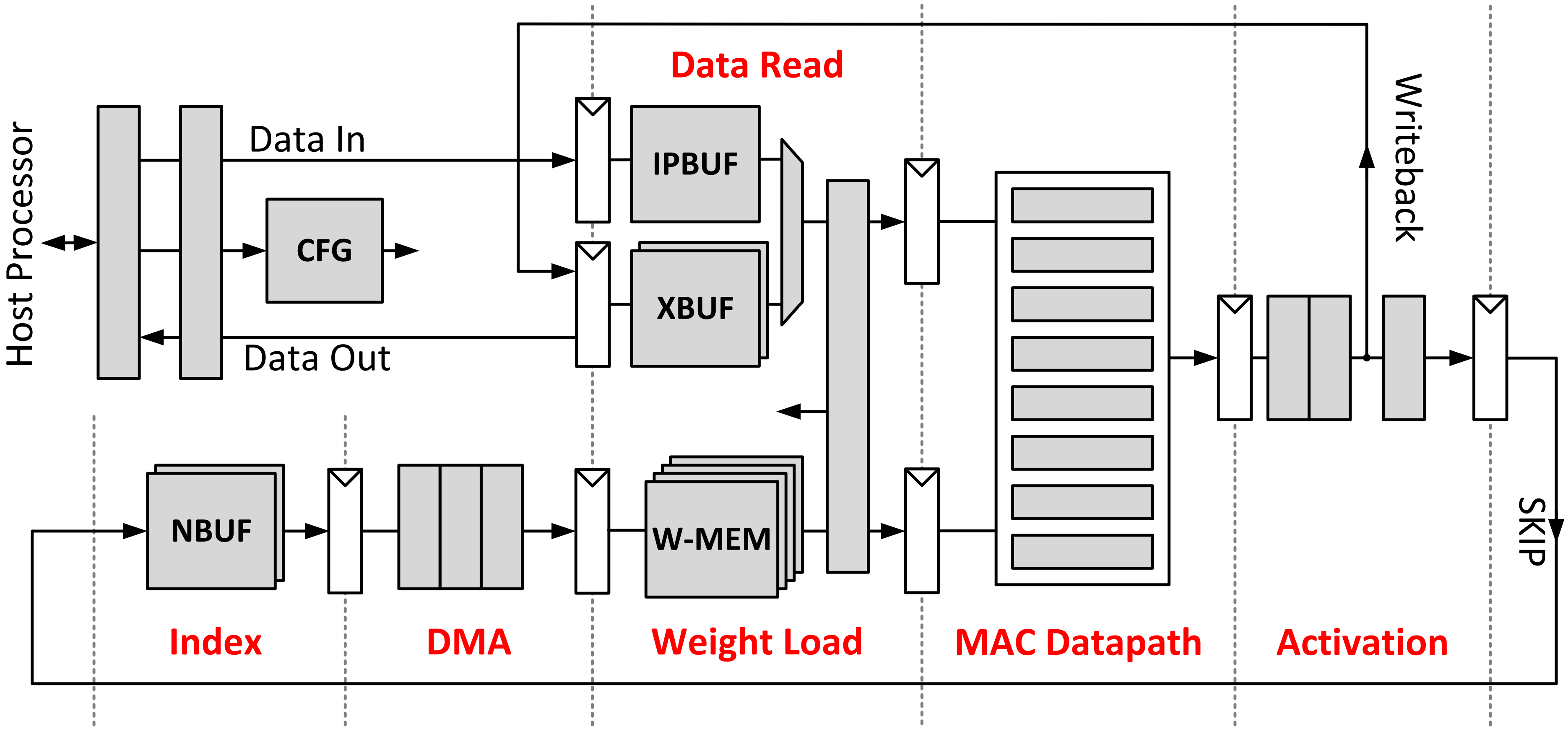
Exploiting Sparse Data in DNNs



DNN ENGINE Micro-Architecture



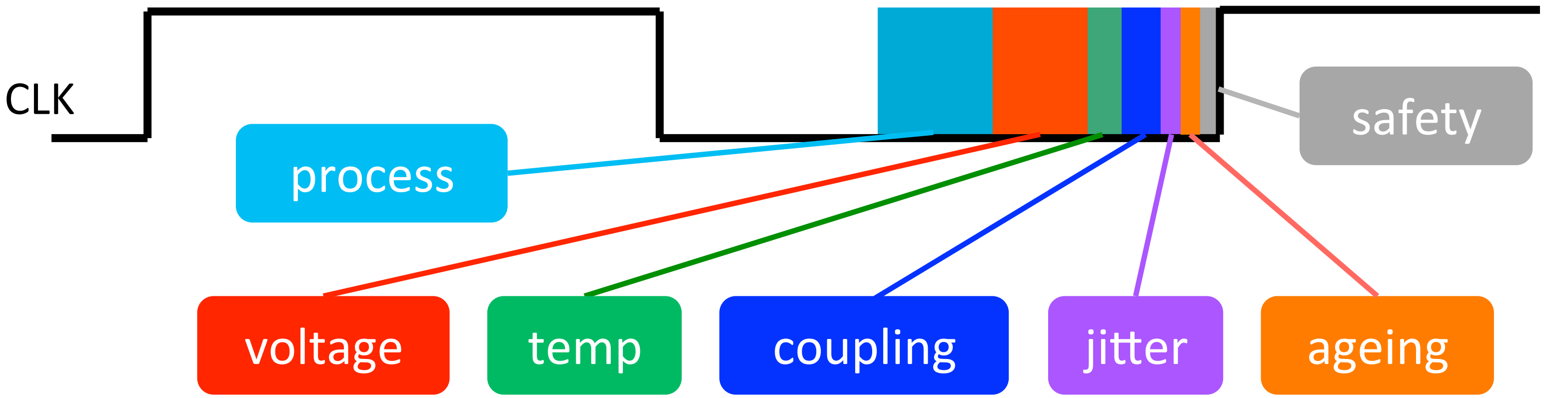
DNN ENGINE Micro-Architecture



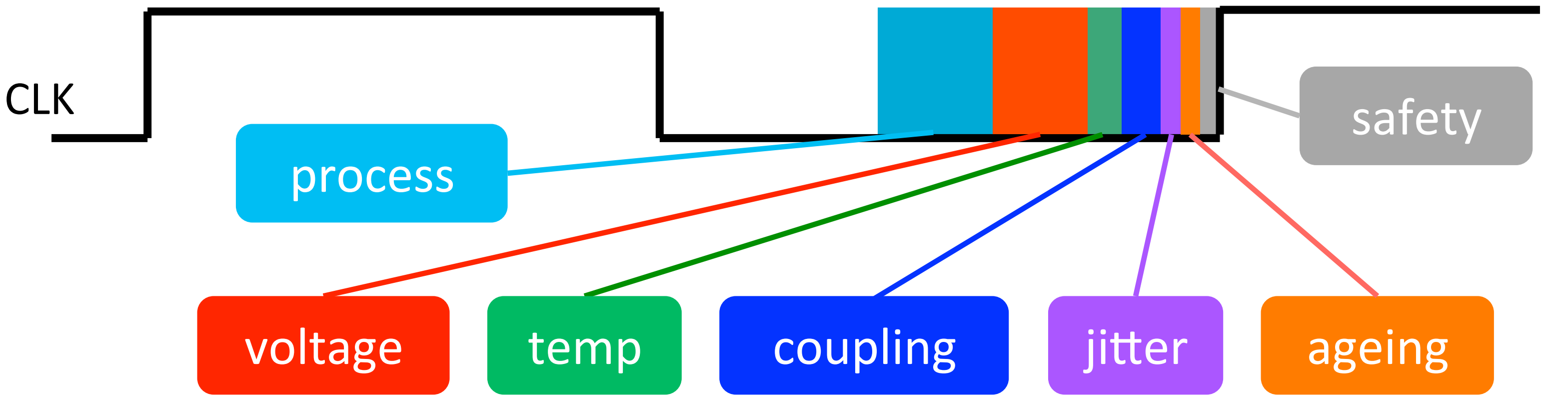
Outline

- Motivation
- DNN Engine
  - Parallelism
  - Sparsity
  - Resilience
- Measurement Results
- Summary

Razor Hardware Accelerators

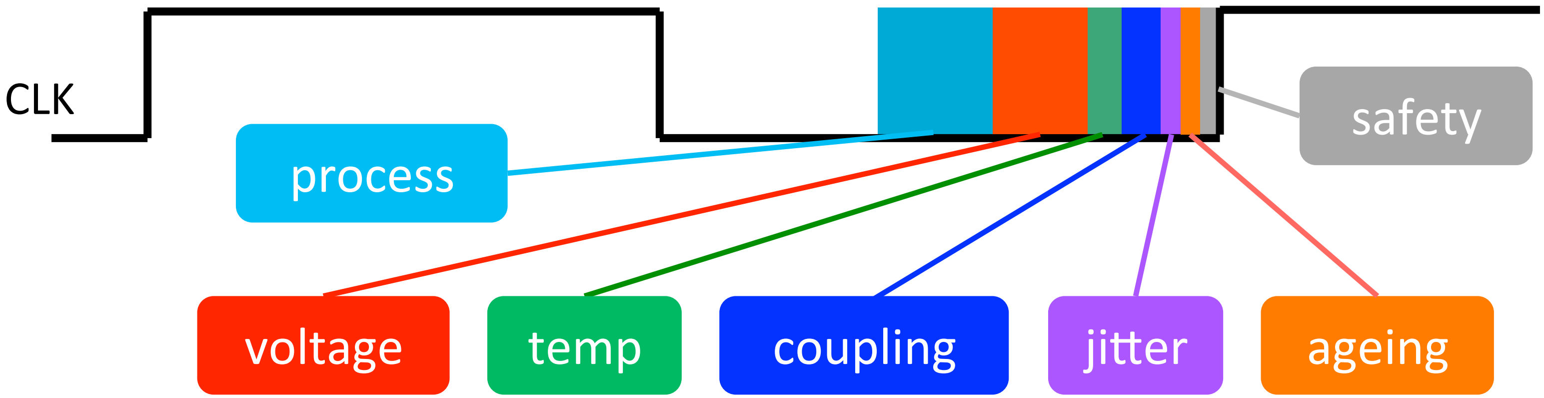


Razor Hardware Accelerators



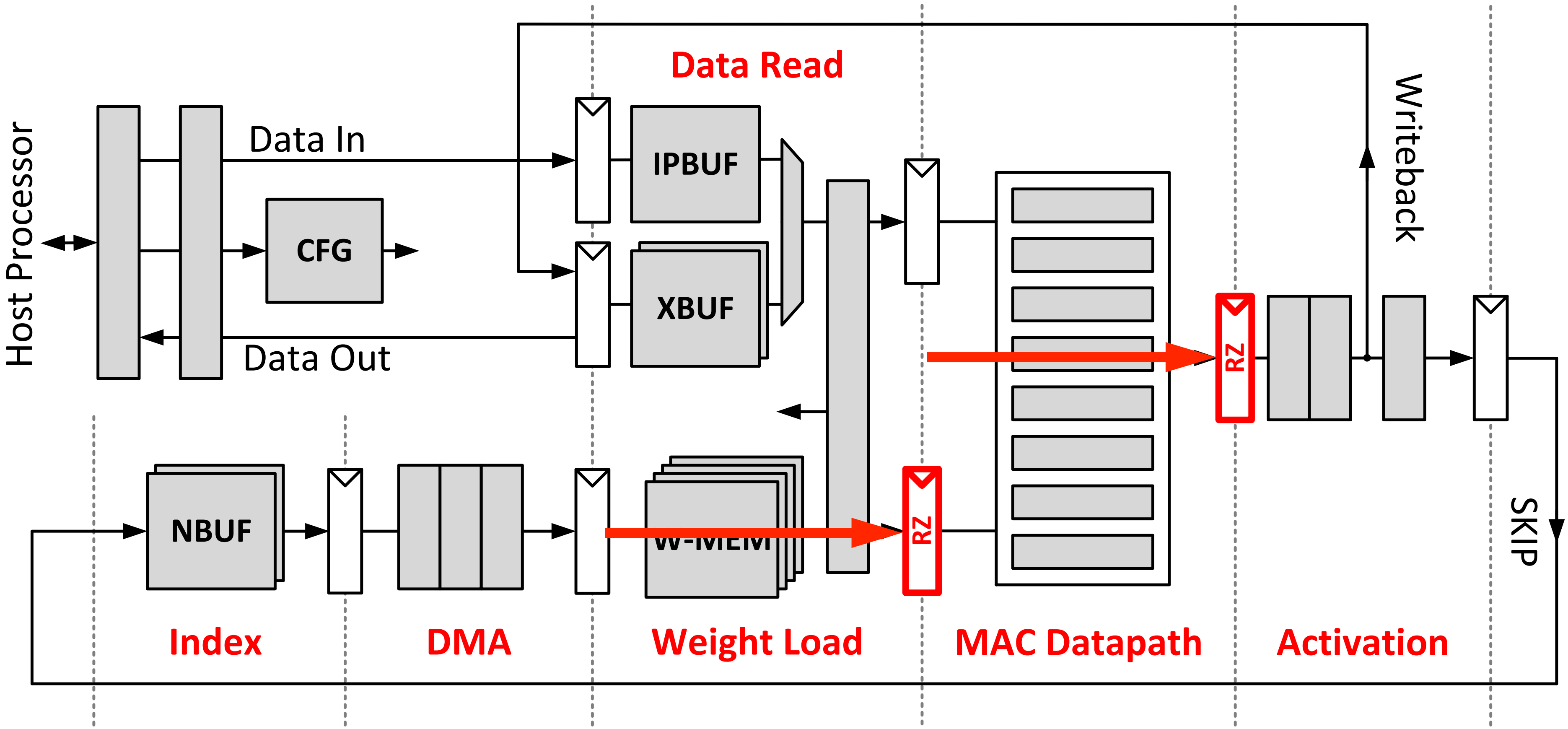
- Occasionally allow real critical paths to fail timing
  - Explicitly check for late-arriving signals
  - Adapt Vdd/Fclk to track near-zero error-rate

Razor Hardware Accelerators



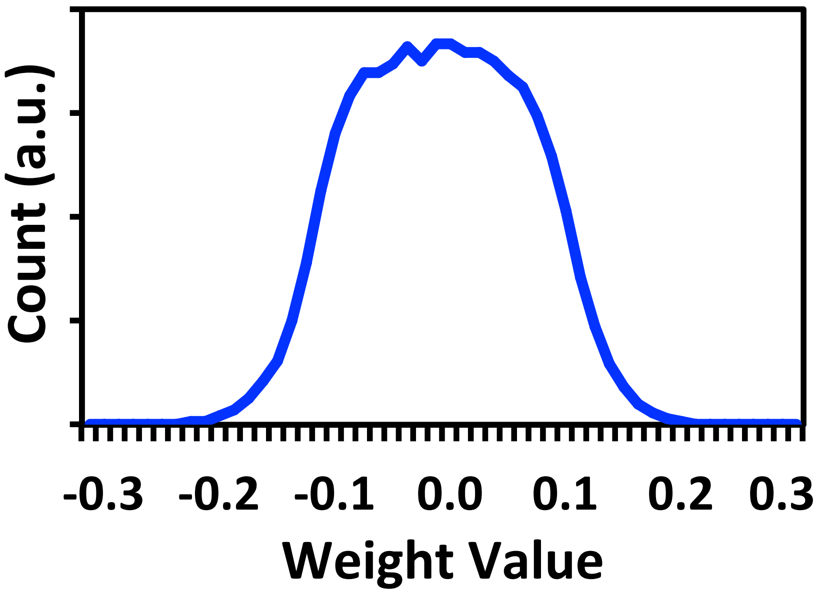
- Occasionally allow real critical paths to fail timing
  - Explicitly check for late-arriving signals
  - Adapt Vdd/Fclk to track near-zero error-rate
- Must ensure timing errors do not affect system
  - Roll-back and replay [Das et al., CICC'13]
  - Algorithmic error tolerance [Whatmough et al., ISSCC'13]

DNN Engine Micro-Architecture



- Two Razor stages: W Memory and MAC Datapath

Memory: Algorithm-Level Tolerance

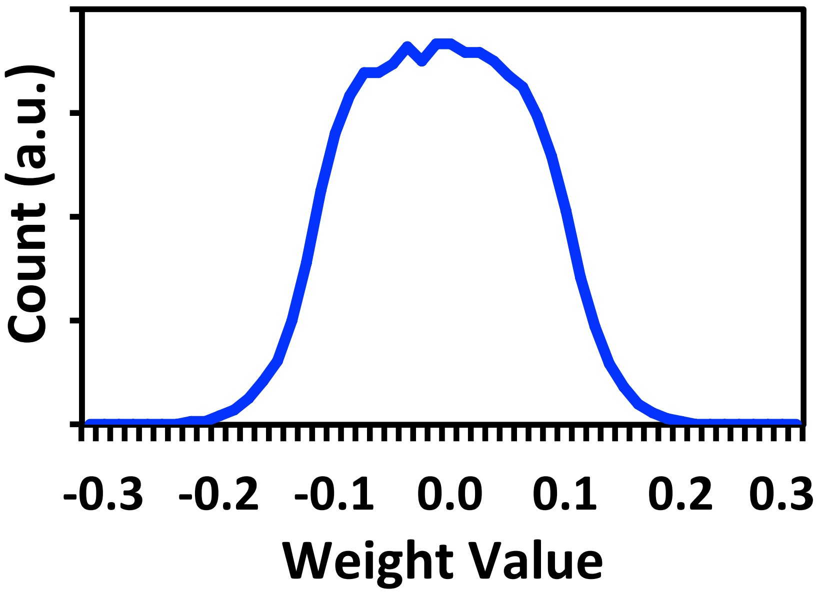


Two's Compliment (TC)

```
-1:  11111111
+1:  00000001
```

- Weights are zero-mean Gaussians (regularization)
  - Small magnitude, random sign (worst-case in TC)

Memory: Algorithm-Level Tolerance



Two's Compliment (TC)

−1: 11111111

+1: 00000001

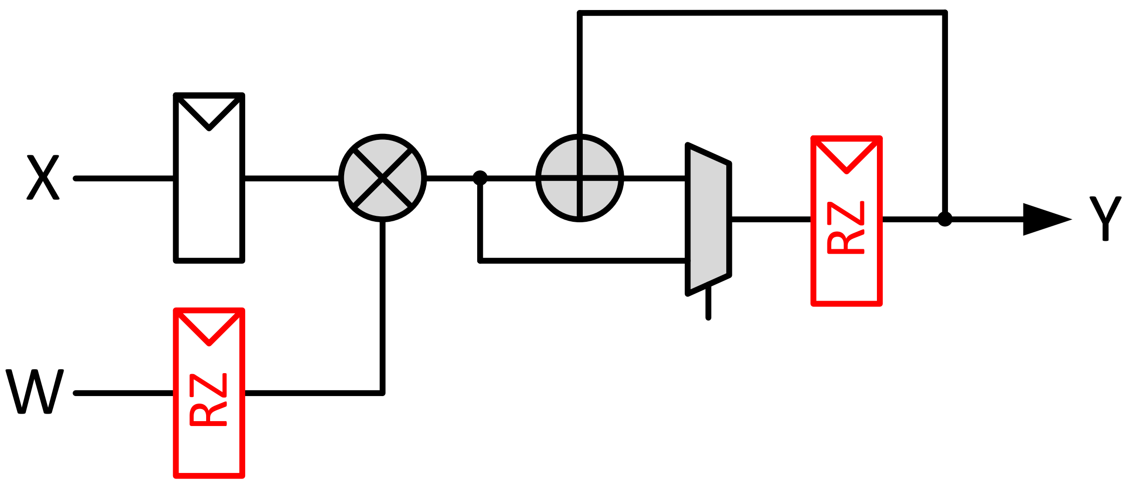
Sign-Magnitude (SM)

−1: 10000001

+1: 00000001

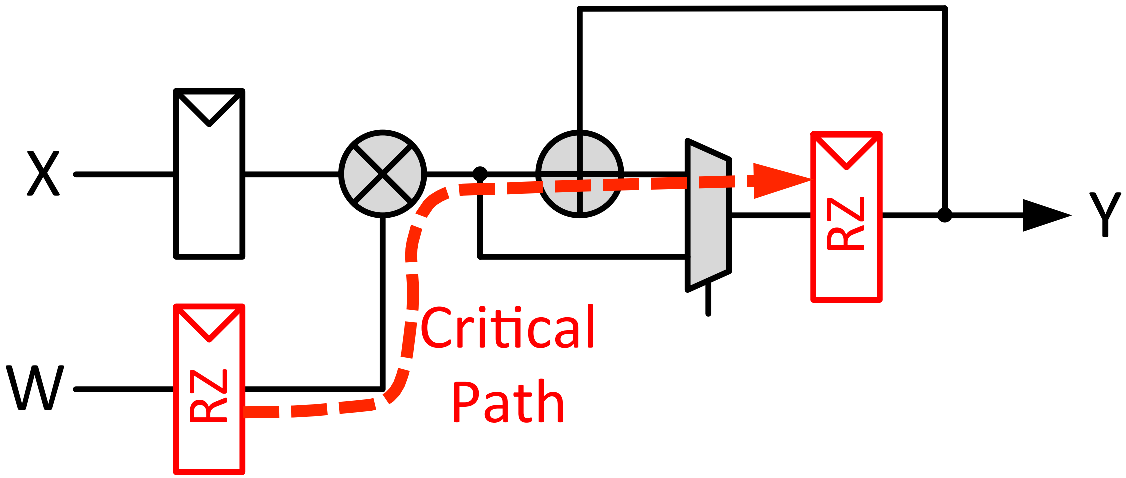
- Weights are zero-mean Gaussians (regularization)
  - Small magnitude, random sign (worst-case in TC)
- Sign-Magnitude (SM) reduces switching in weights
  - Lowers switched cap (power) and bit error rate

Datapath: Circuit-Level Tolerance



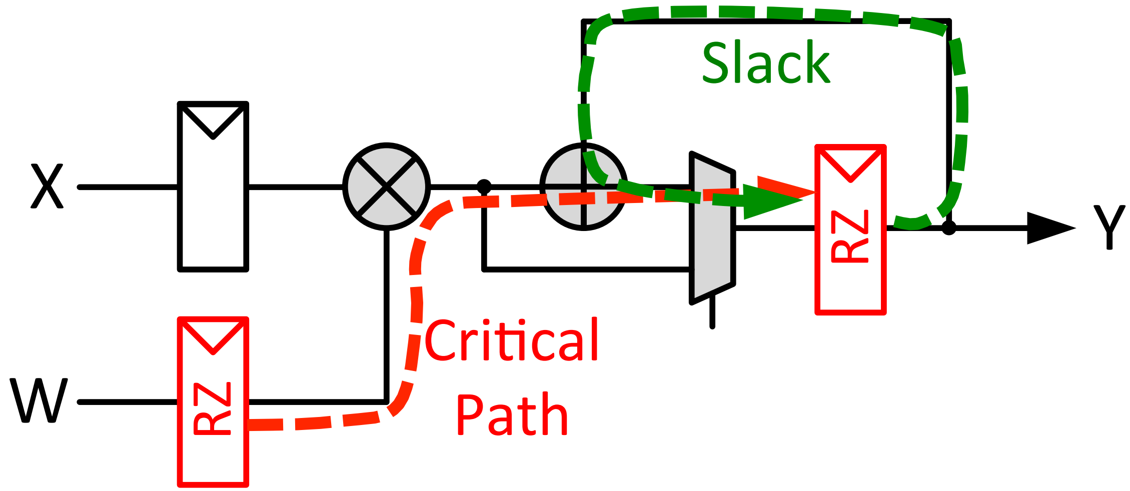
- Algorithmic error tolerance in datapath is poor
  - Errors are persistent in accumulator
- Circuit-level error tolerance using time borrowing
  - Single-sided critical path at accumulator register

Datapath: Circuit-Level Tolerance



- Algorithmic error tolerance in datapath is poor
  - Errors are persistent in accumulator
- Circuit-level error tolerance using time borrowing
  - Single-sided critical path at accumulator register

Datapath: Circuit-Level Tolerance

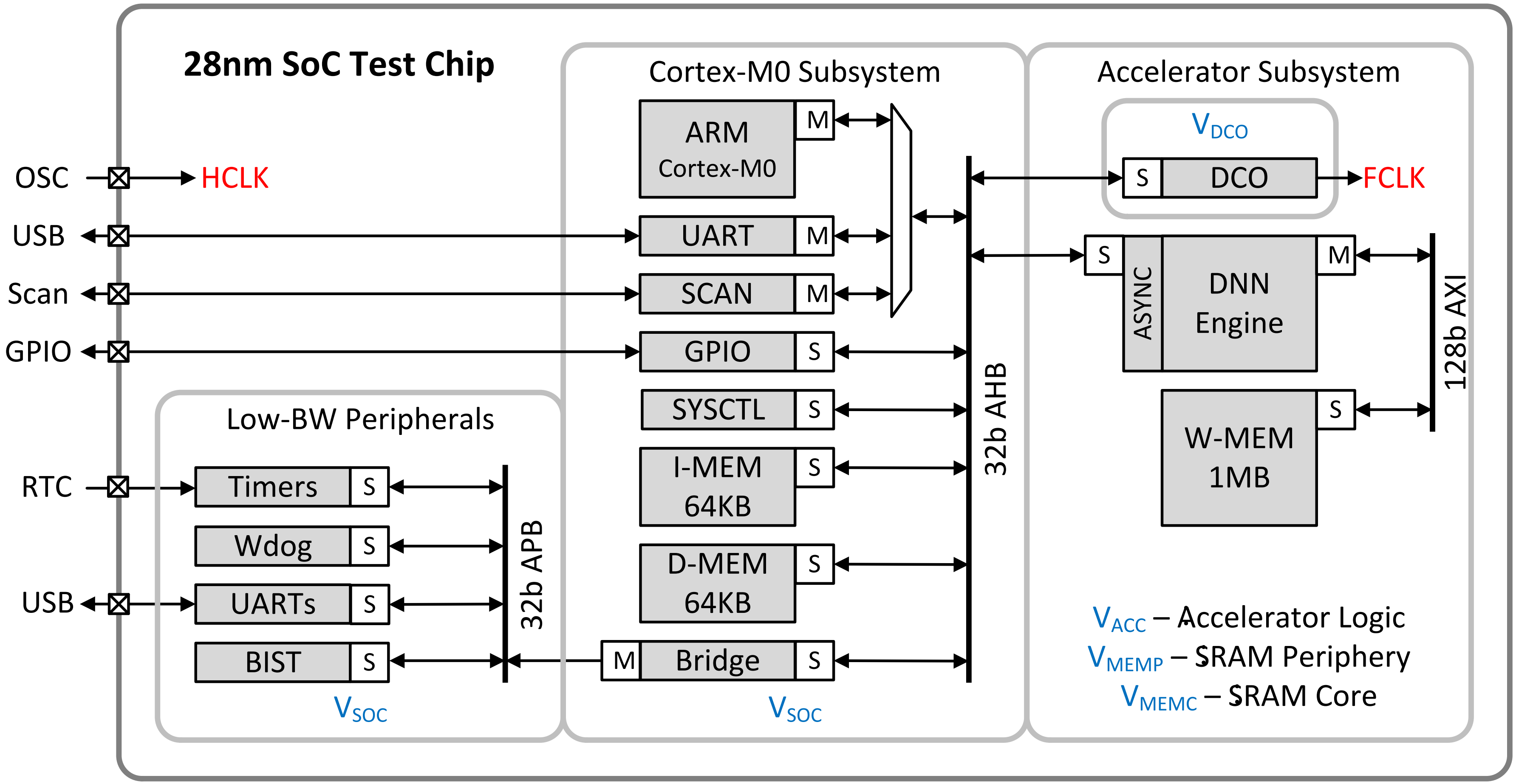


- Algorithmic error tolerance in datapath is poor
  - Errors are persistent in accumulator
- Circuit-level error tolerance using time borrowing
  - Single-sided critical path at accumulator register

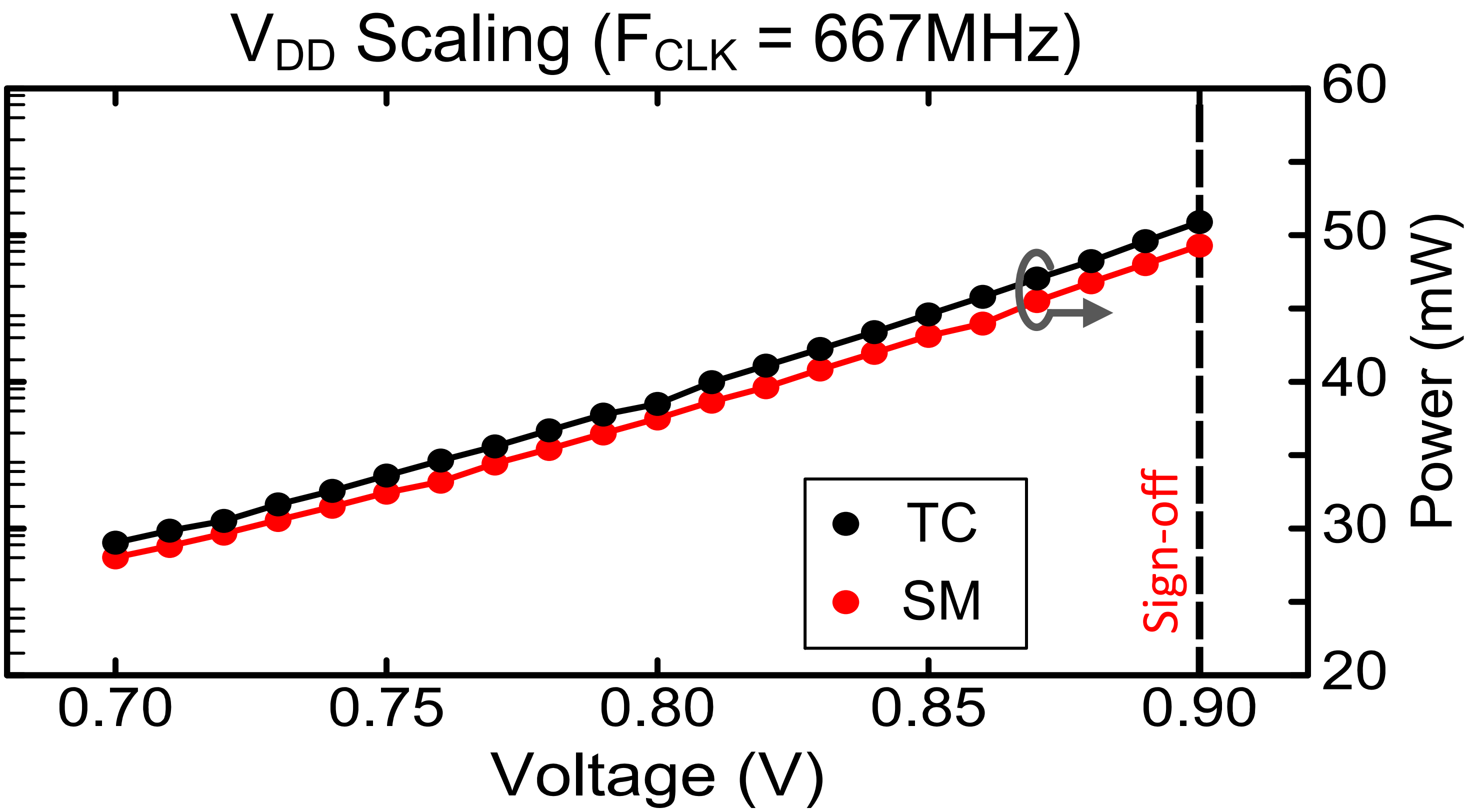
Outline

- Motivation
- DNN Engine
  - Parallelism
  - Sparsity
  - Resilience
- Measurement Results
- Summary

28nm SoC for IoT Applications

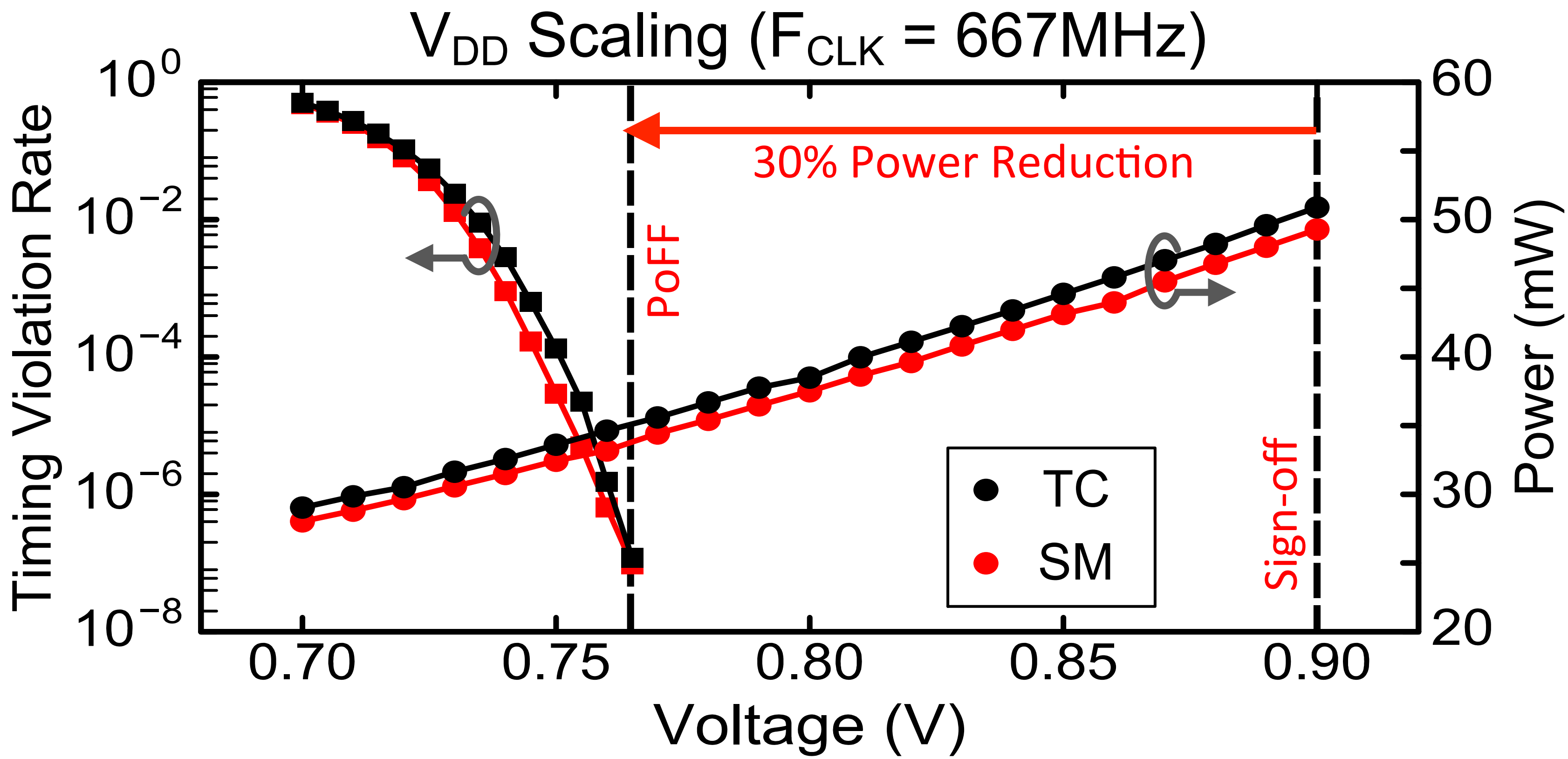


Razor Voltage Scaling



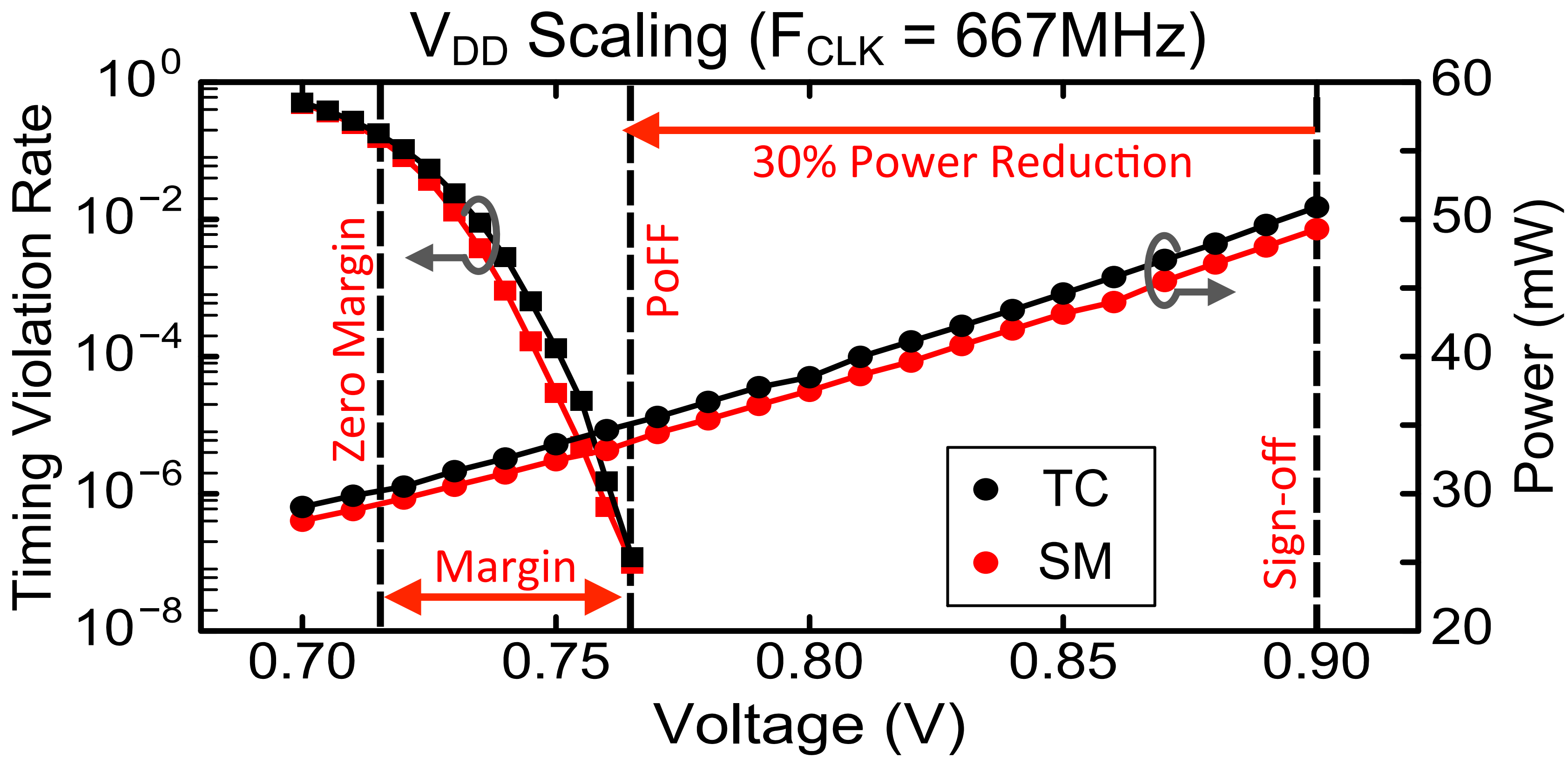
- FC-DNN 784-256-256-256-10 (98.36% MNIST)

Razor Voltage Scaling



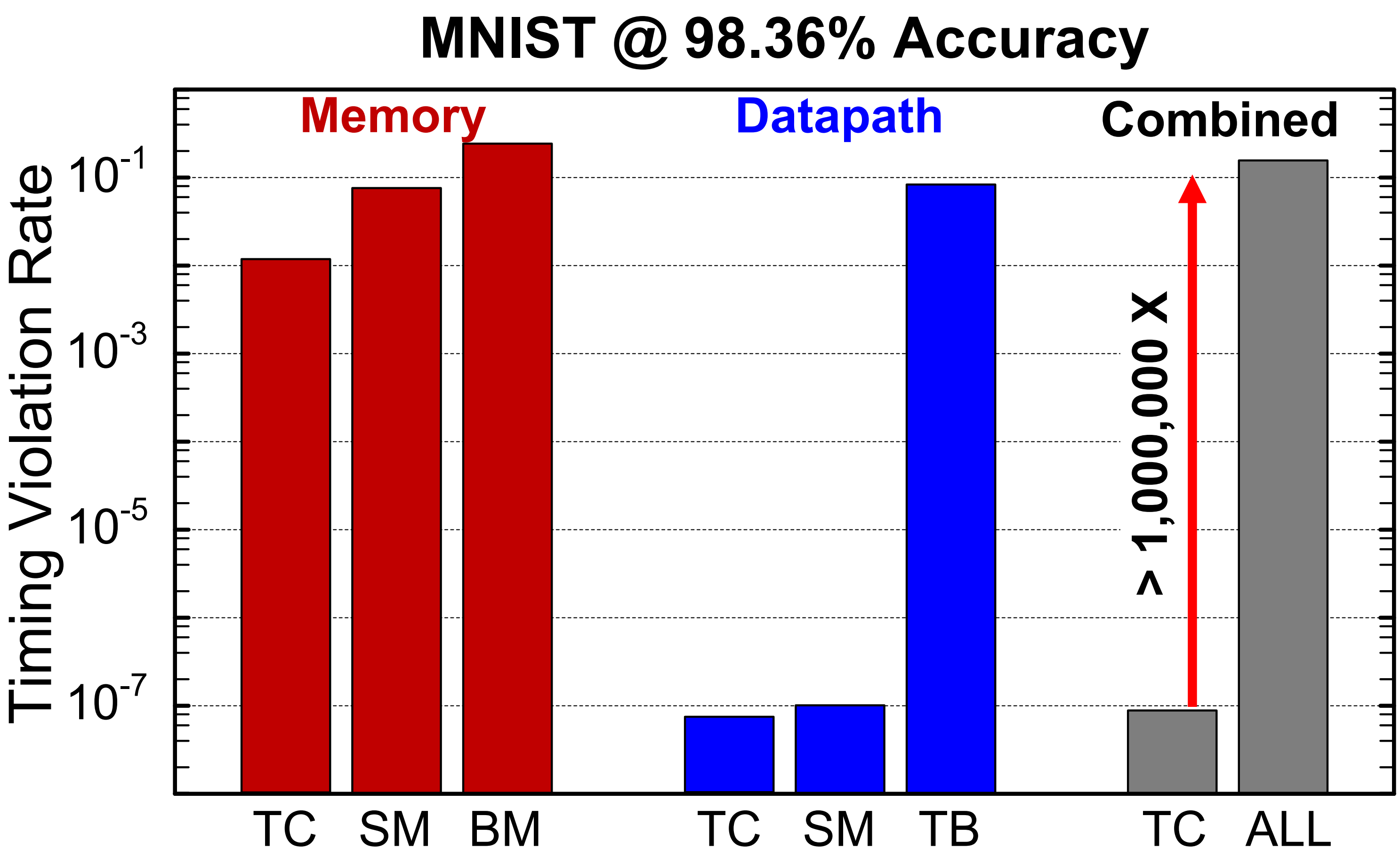
- Operation at PoFF (770mV) lowers power by 30%

Razor Voltage Scaling



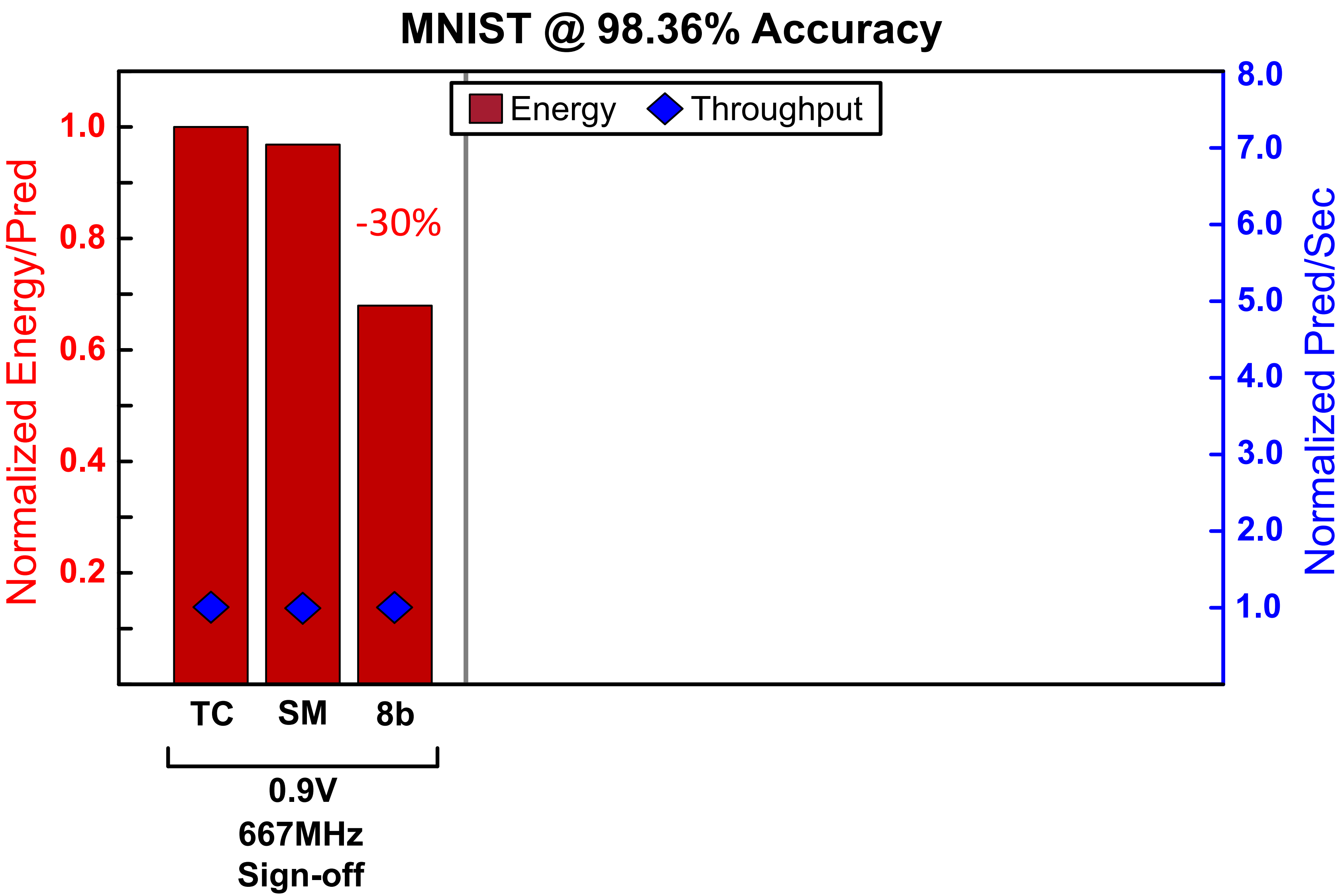
- Margin for fast variations down to 715mV

Timing Error Tolerance Summary

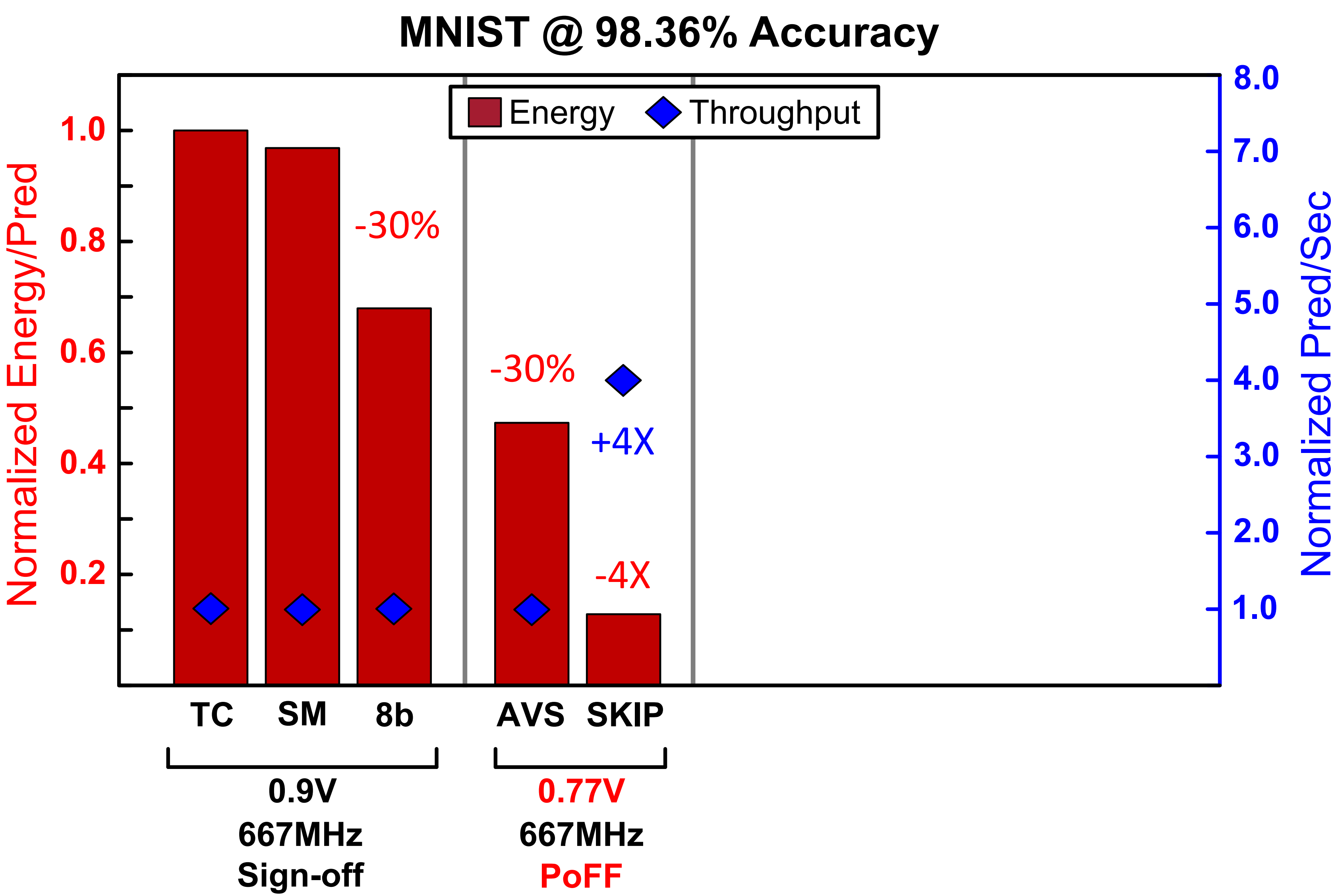


- Aggregate timing violation rate greater than 10^{-1}

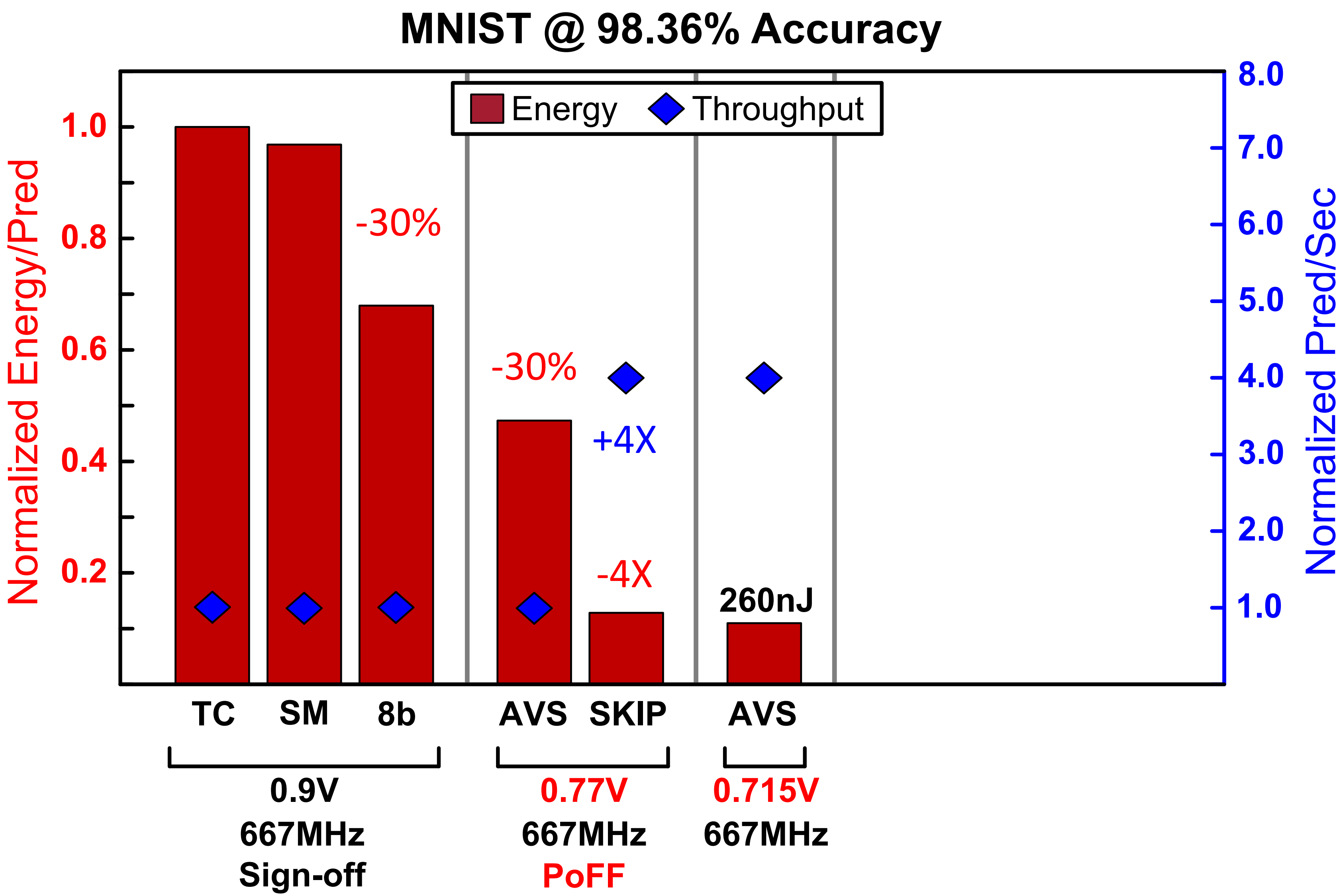
Energy/Throughput Summary



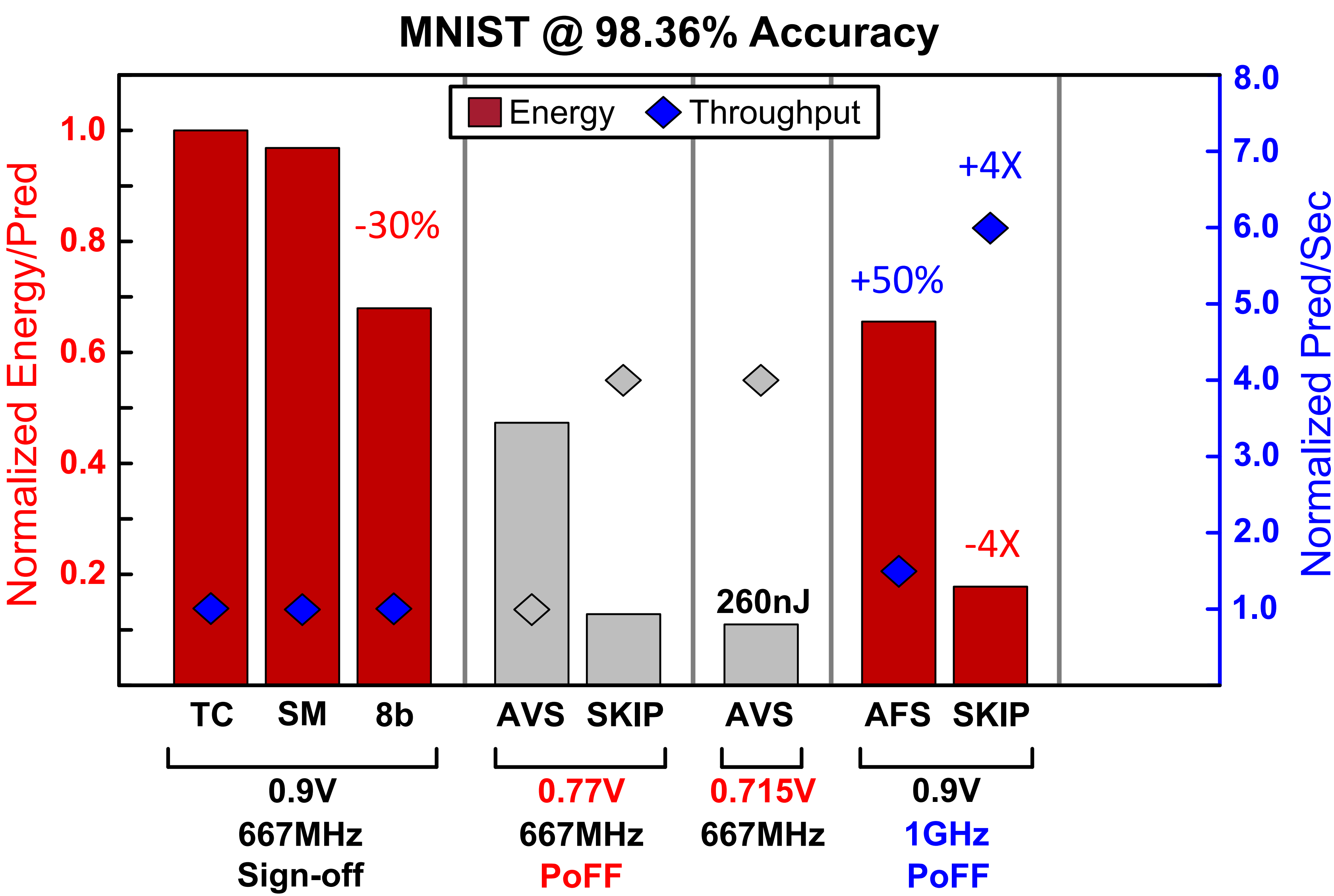
Energy/Throughput Summary



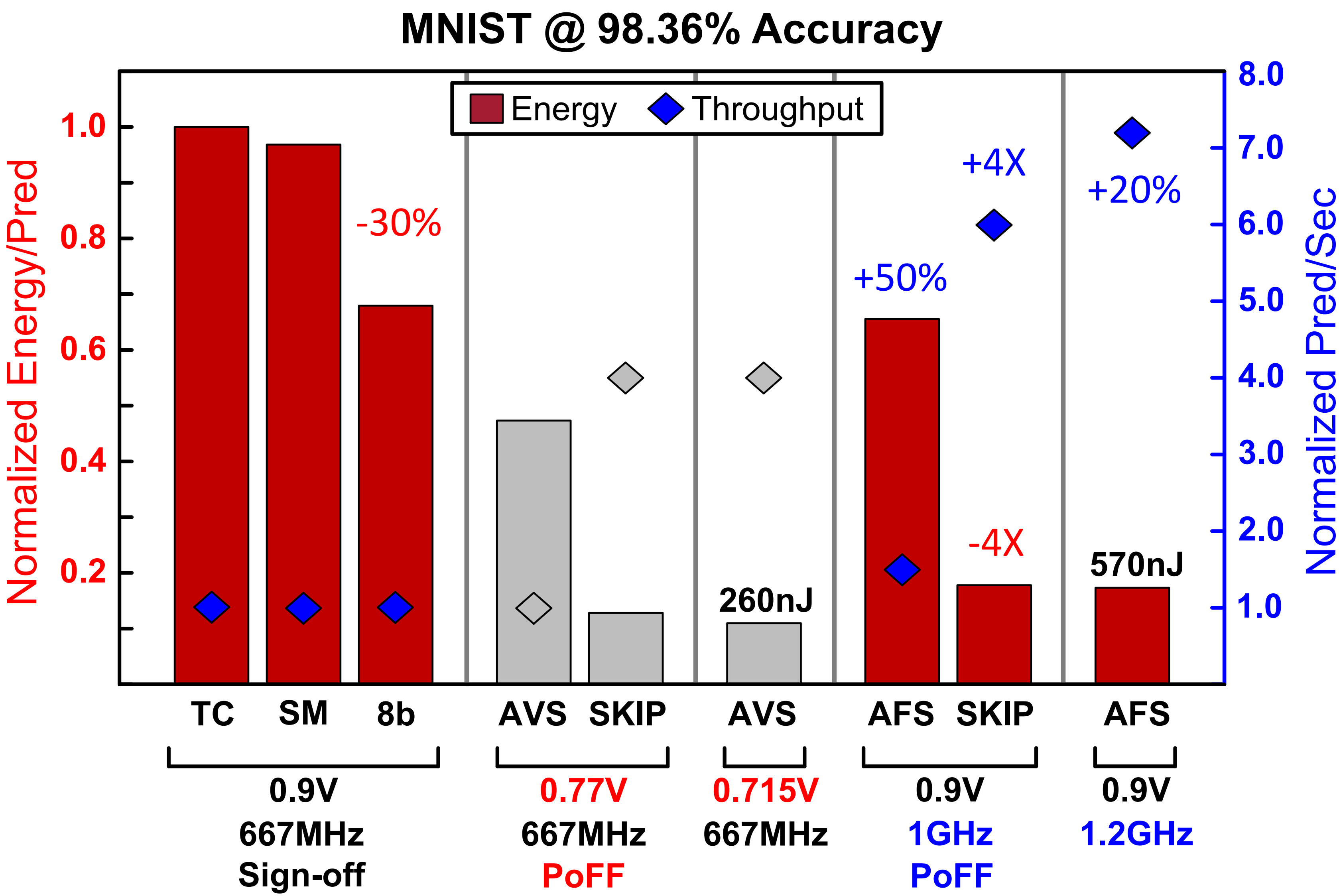
Energy/Throughput Summary



Energy/Throughput Summary

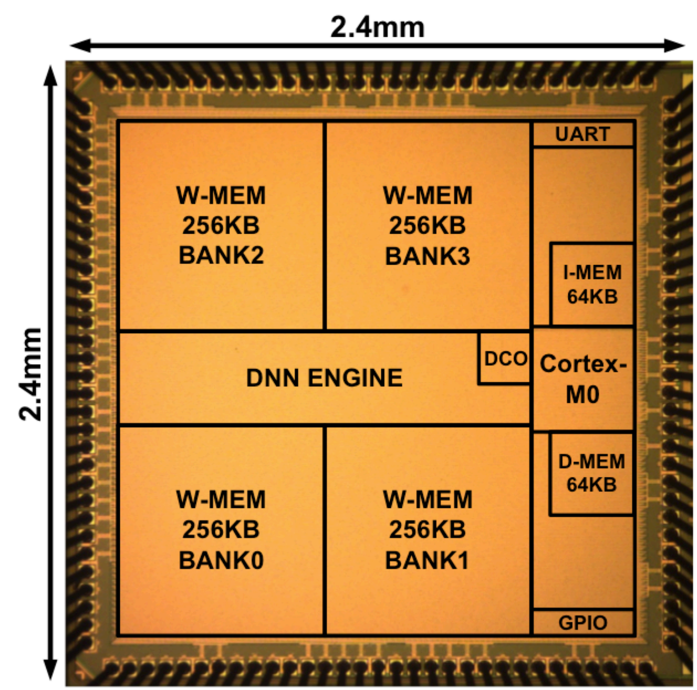


Energy/Throughput Summary



Summary and Conclusion

- **DNN Engine – A Shared Programmable Classifier**
  - **Parallelism:** 10X Data Reuse @ 128b/cycle BW
  - **Sparsity:** +4X Throughput, -4X Energy
  - **Resilience:** +50% Throughput or -30% Energy w/Razor
- Energy Efficient DNN
  - 568 nJ/pred @ 1.2 GHz
- Timing Error Tolerance
  - Pe >10^{-1} @ 98.36% (MNIST)



INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

ISSCC®

UNIVERSITY OF PENNSYLVANIA