ScRRAMBLe
Block Sparse Neural Network Architecture for Analog Compute-in Memory Accelerators

Vikrant Jaltare
Department of Bioengineering &
Institute for Neural Computation
UC San Diego

10th International Conference on Rebooting Computing 2025

INTEGRATED SYSTEMS NEUROENGINEERING

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Shu Chien-Gene Lay Department of Bioengineering

INC

Case for RRAM* based Compute-in Memory (CIM) Technologies

**valid for conductance-based memristive technologies*

Why a decades old architecture decision is impeding the power of AI computing

Most computers are based on the von Neu
which separates compute and memory. Th
been perfect for conventional computing,
traffic jam in AI computing.

IBM (2025)

1.1 **Computing's Energy Problem (and what we can do about it)**

Mark Horowitz

8. Conclusion

In summary, our challenge is clear: The drive for performance and the end of voltage scaling have made power, and not the number of transistors, the prin-
ents in computing performance.
vill require the creation and effective
and will require the participation of
play our cards right, and develop the
part of the design process, we will
nt computing devices.

Google Scholar | memristors and crossbar arrays

Articles | About 12,000 results (0.11 sec)

Any time
Since 2025
Since 2024
Since 2021
Custom range...

Sort by relevance
Sort by date

Any type
Review articles

Sodium-doped titania self-rectifying memristors for crossbar array neuromorphic architectures
SE Kim, JG Lee, L Ling, SE Liu, HK Lim... - Advanced ..., 2022 - Wiley Online Library
... conductance of **memristors**, these **arrays** can massively ... **memristor crossbars** in this context, we conducted simulations of a neural network composed of selectorless **crossbar arrays**. In ...
☆ Save 99 Cite Cited by 81 Related articles All 8 versions Import into BibTeX

Memristive crossbar arrays for storage and computing applications
H Li, S Wang, X Zhang, W Wang... - Advanced Intelligent ..., 2021 - Wiley Online Library
... The emergence of **memristors** with potential applications in data storage and artificial intelligence has attracted wide attentions. **Memristors** are assembled in **crossbar arrays** with data ...
☆ Save 99 Cite Cited by 250 Related articles All 11 versions Import into BibTeX

Instead of re-enginee
improvements in parall
concept, in-memory cor
Neumann architecture b
where the data are located[15]. This approach is similar to the computing scheme in the human brain, where information is processed in sparse networks of neurons and synapses, without any physical separation between computation and memory[16]. In-memory com-

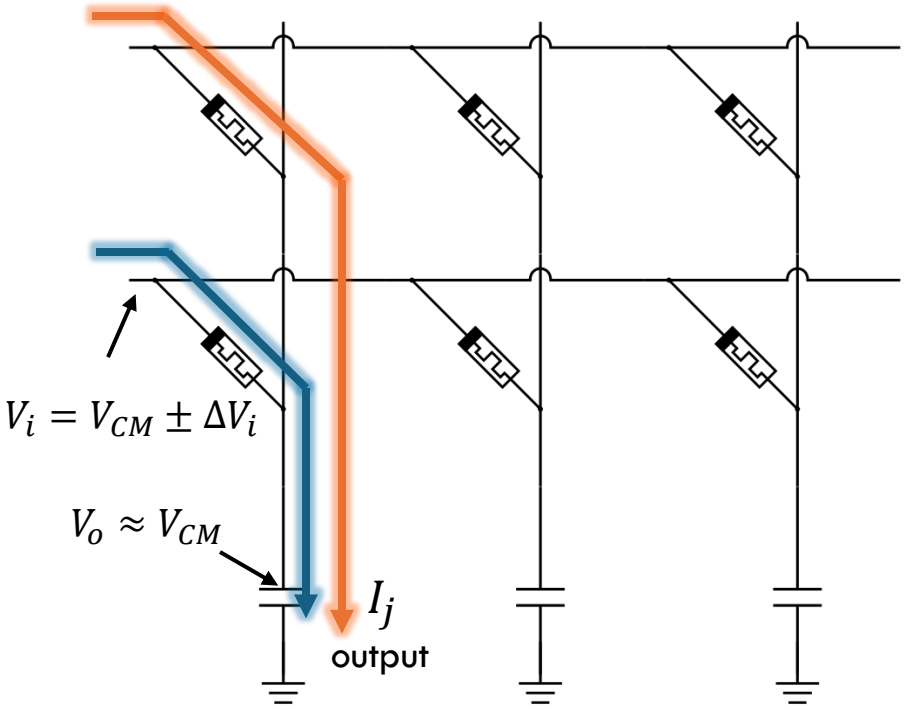
Ielmini and Wong, *Nature Electronics (2018)*

phic Computing Market Size, rowth

morphic Computing Market was valued at USD 28.5 million in 2024 and is projected to grow from USD 47.8 million in 2025 to USD 1,325.2 millionn by 2030, at a CAGR of 89.7% during the forecast period. This remarkable growth is primarily driven by the escalating demand for AI-based applications that mimic the brain's neural architecture, the increasing integration of neuromorphic computing in autonomous vehicles, robotics, and edge computing devices, and the convergence of quantum computing with neuromorphic systems. Leading industry players such as Intel, IBM, and BrainChip are pioneering advancements in neuromorphic processors, further propelling market expansion.

M&M (2025)

RRAM Crossbar Arrays for Neural Networks



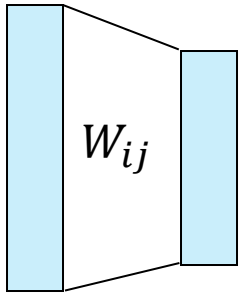
Matrix Vector Multiplication (MVM) → **Ohm's Law** $I_j = \sum_i G_{ij}\,\Delta V_i$

Neural Network Requirements	CIM Characteristics
Efficient MVM ($\mathcal{O}(n^2)$ or less)	Fast MVM ($\mathcal{O}(1)$) ✓
Large number of parameters	High Density (but area constraint) ✓
Parameter updates → learning rules → outer product	Non-volatile storage → incremental outer product updates ✓
Mid to low precision weights	Analog multi-level storage ✓

Good match! But…

Challenges with using RRAM Accelerators

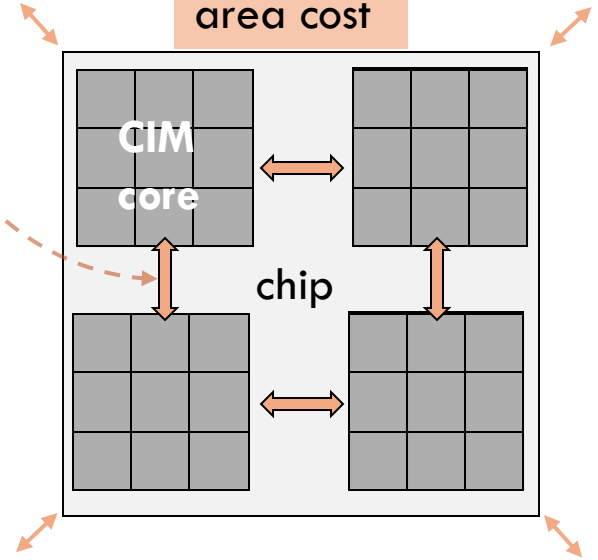
Scaling



$W_{ij} >$ core size

core-to-core communication

area cost



Signed Weights

$g_{ij} \in [g_{min}, g_{max}]$

Positive and limited dynamic range

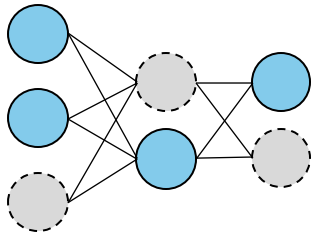
Differential encoding?

⇓

> 1 memory element per weight.

Mismatch issues

Sparsity

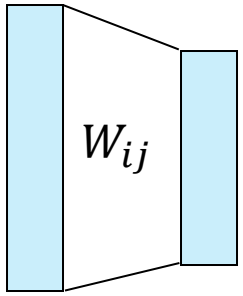


Dropout → fine-grained sparsity

Requires off-chip digital controller → Latency

Challenges with using RRAM Accelerators

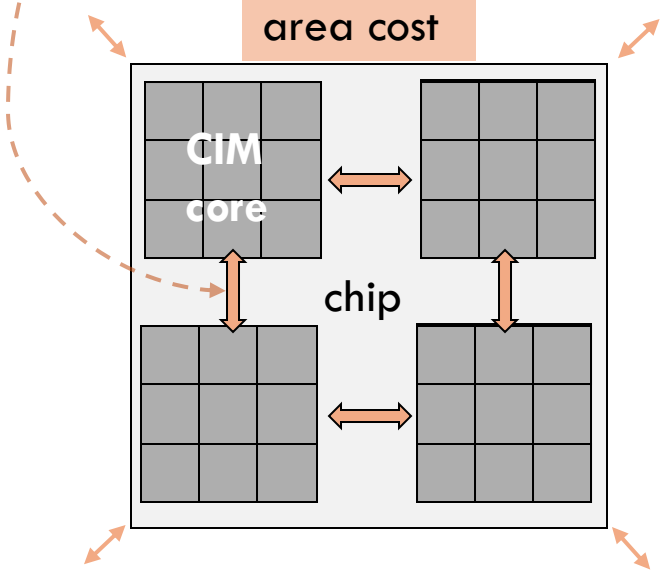
Scaling



$W_{ij} >$ core size

core-to-core communication

area cost



Signed Weights

$g_{ij} \in [g_{min}, g_{max}]$

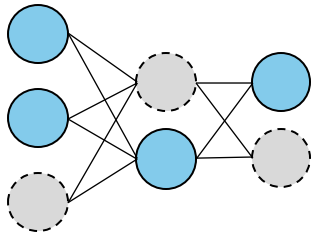
Positive and limited

weight.

Mismatch issues

ScRRAMBLe → Framework to leverage these constraints as features for neural network design?

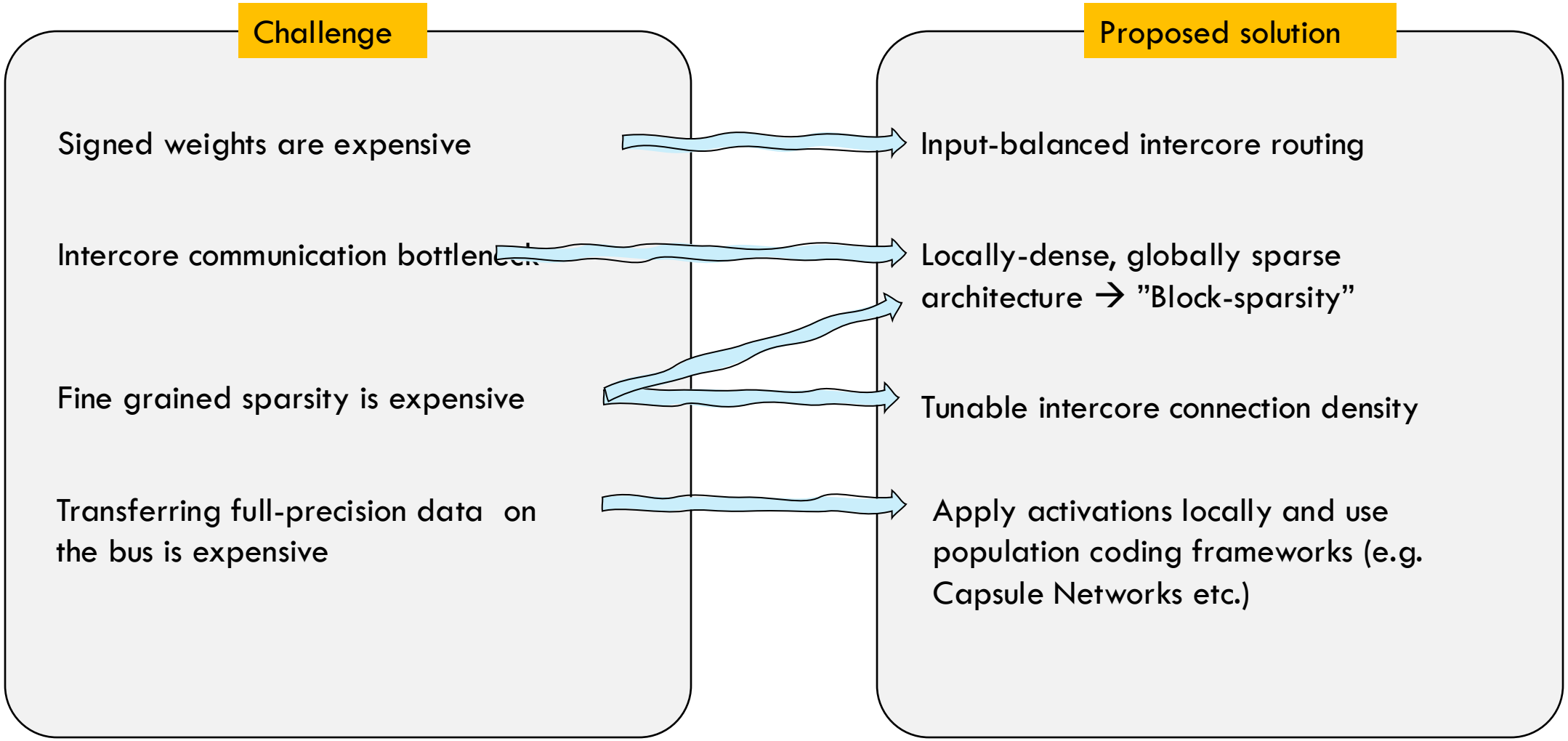
Sparsity



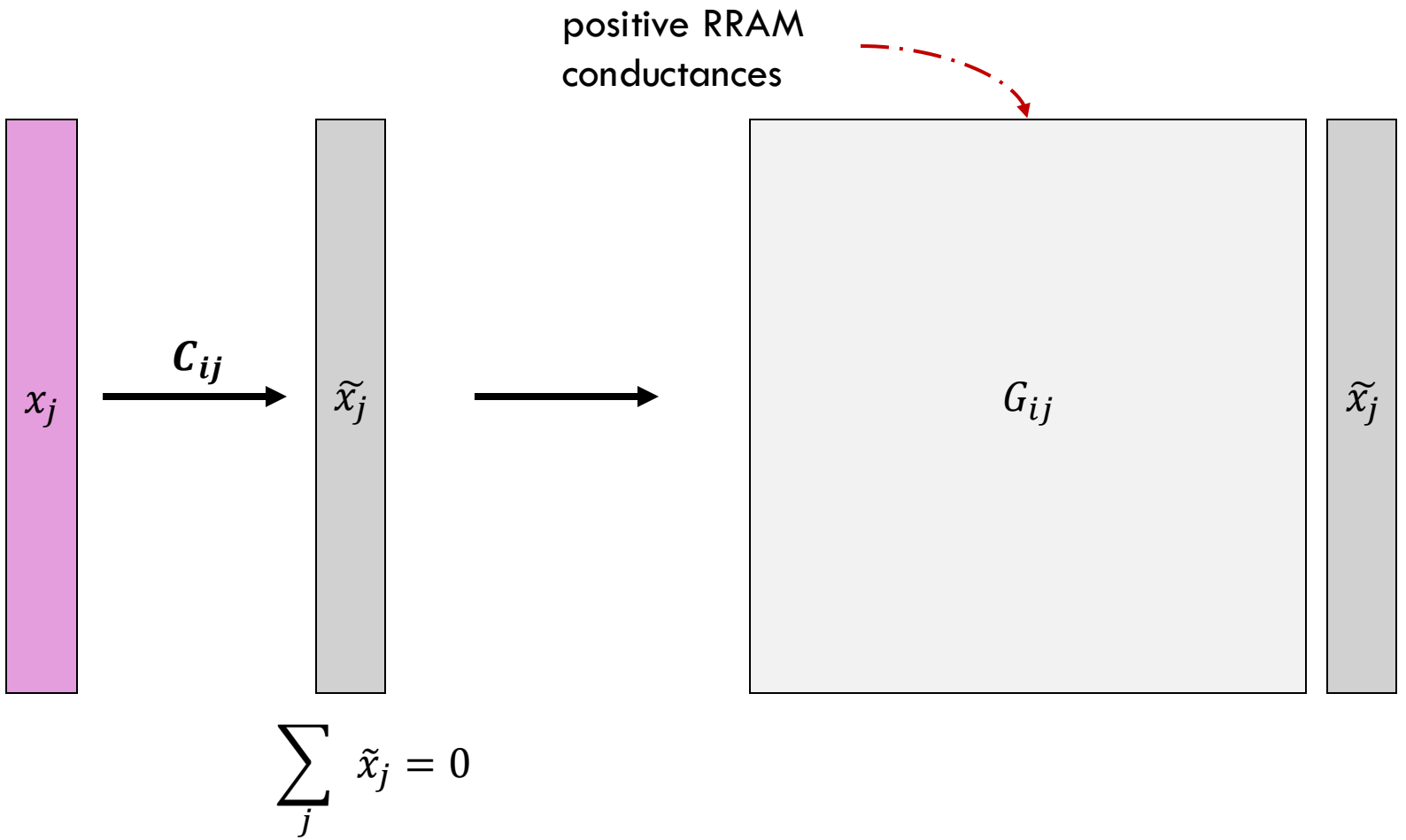
Dropout → fine-grained sparsity

Requires off-chip digital controller → Latency

Bird’s eye view of ScRRAMBLe



Signed Weights with Input Balancing



Offset canceling

$$y = \sum_j G_{ij}\tilde{x}_j$$

signed neural net weights

$$\Updownarrow$$

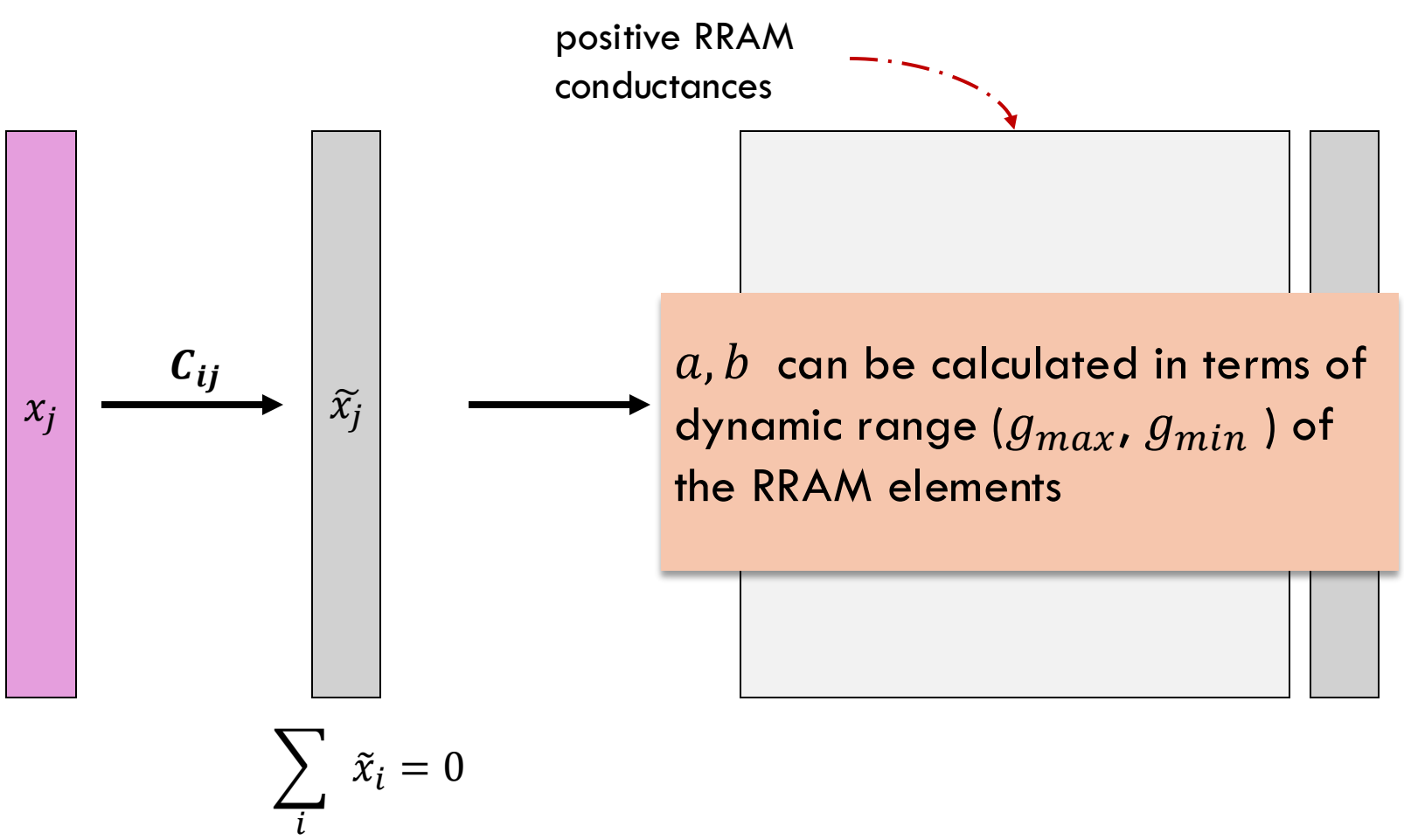
$$y = \sum_j aW_{ij}\tilde{x}_j$$

If $a, b \in \mathbb{R}, W_{ij}$ is signed and,

$$G_{ij} = a(W_{ij} + b)$$

$$\because ab \sum_j \tilde{x}_j = 0$$

Signed Weights with Input Balancing



Offset canceling

$$y = \sum_j G_{ij}\tilde{x}_j$$

signed neural net weights

$$\Updownarrow$$

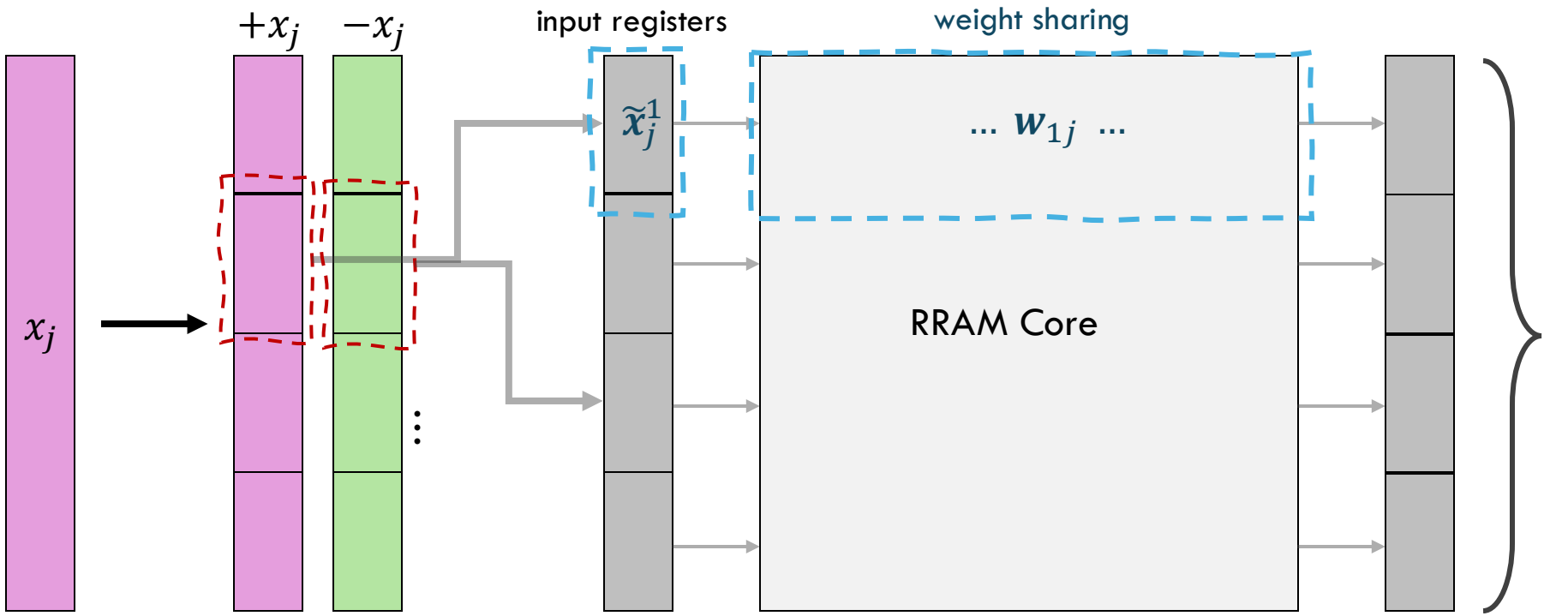
$$y = \sum_j aW_{ij}\tilde{x}_j$$

If $a, b \in \mathbb{R}, W_{ij}$ is signed and,

$$G_{ij} = a(W_{ij} + b)$$

$$\because ab\sum_j \tilde{x}_j = 0$$

Implementing Input-Balanced Routing

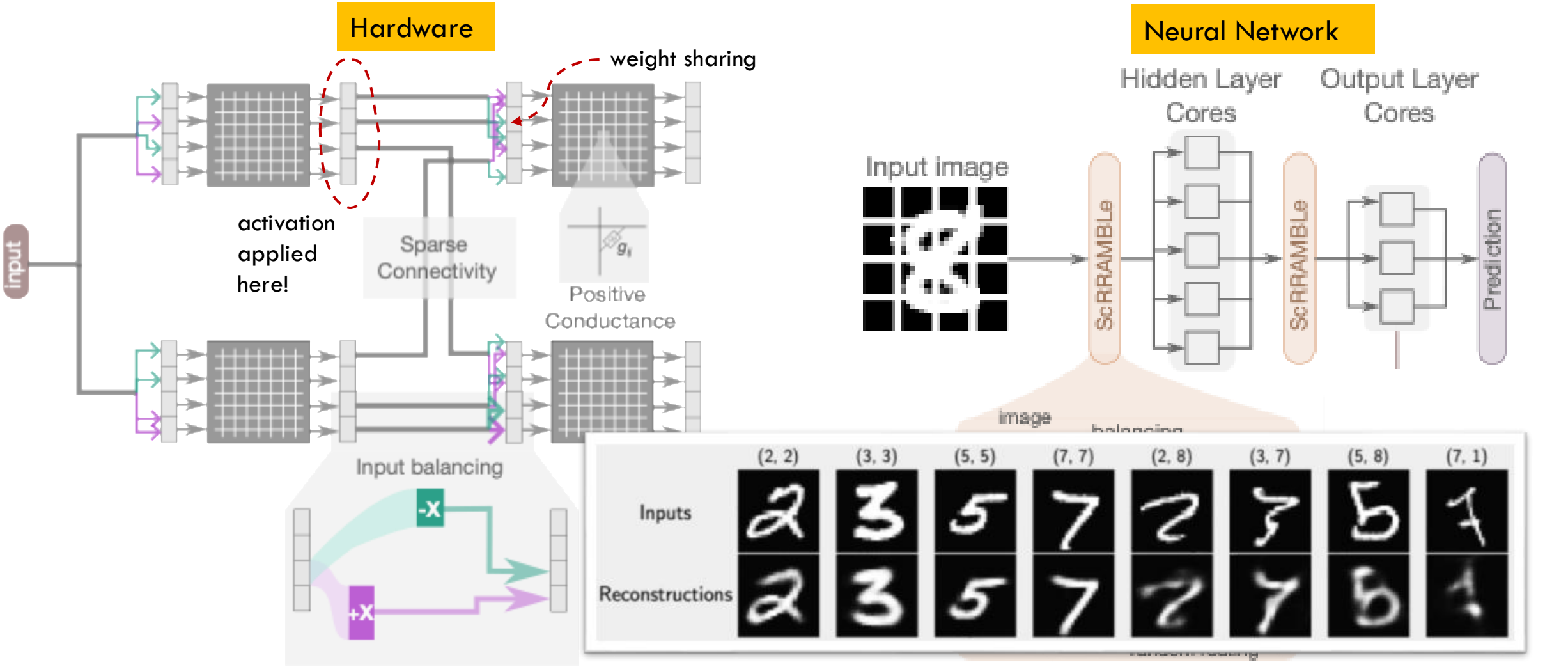


$$\sum_j \tilde{x}_j = 0$$

Can be implemented using LUTs in hardware synthesis

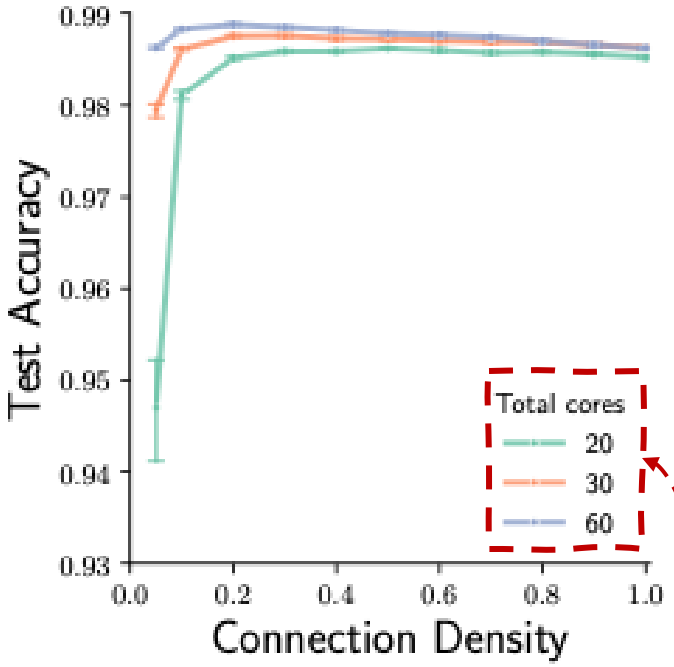
Signed weights + No need for having off-chip controller for activations!

ScRRAMBLe Architecture

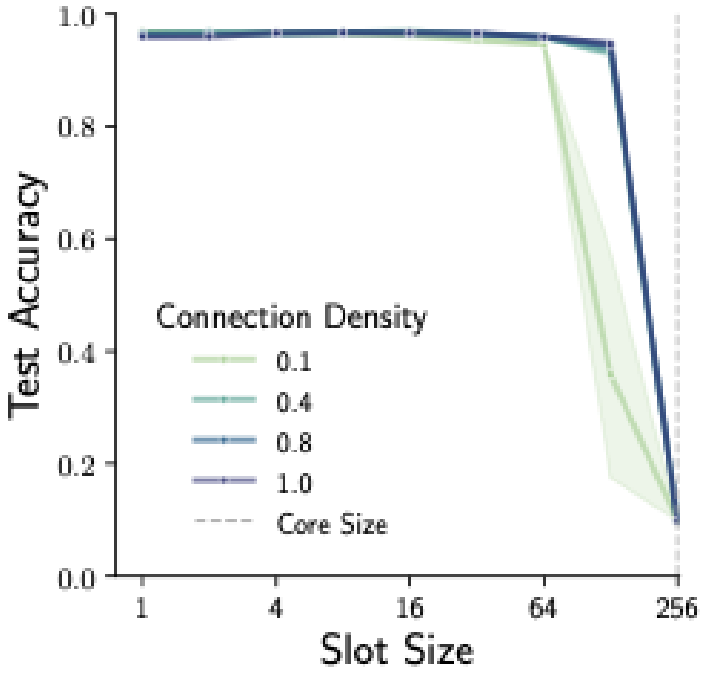


Performance of block-sparse networks

Optimal Sparsity

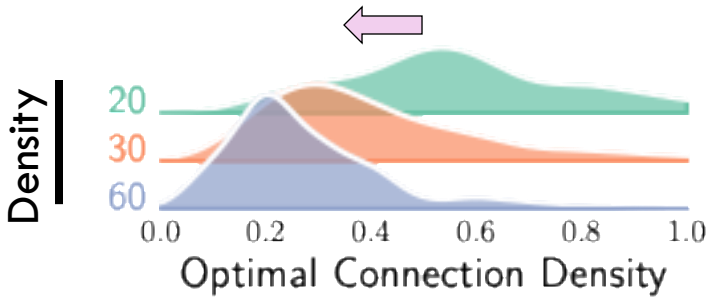


Slot/Chunk size



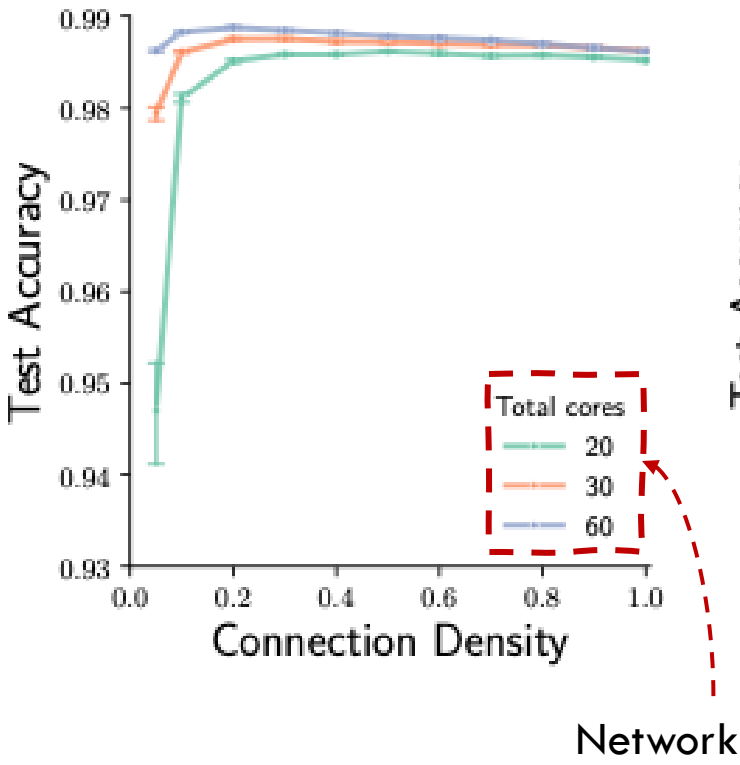
Network size

Small-world optimality



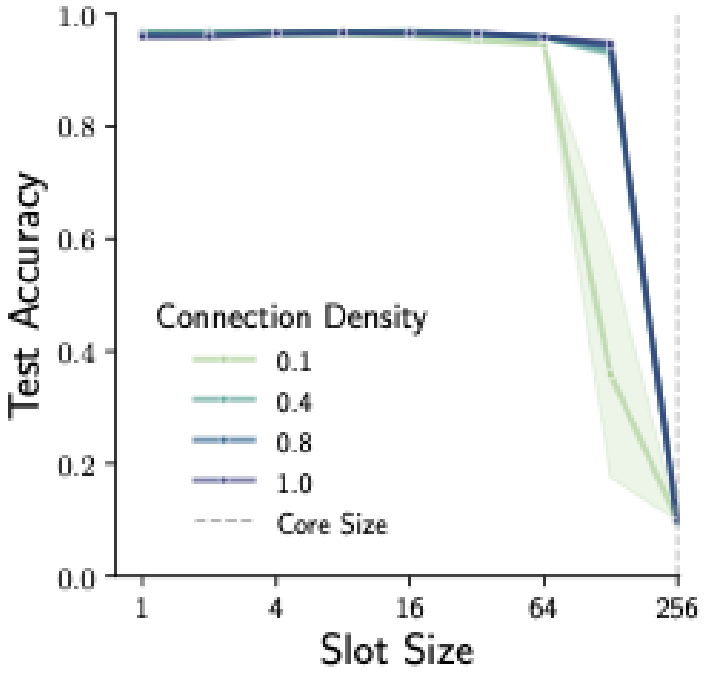
Performance of block-sparse networks

Optimal Sparsity

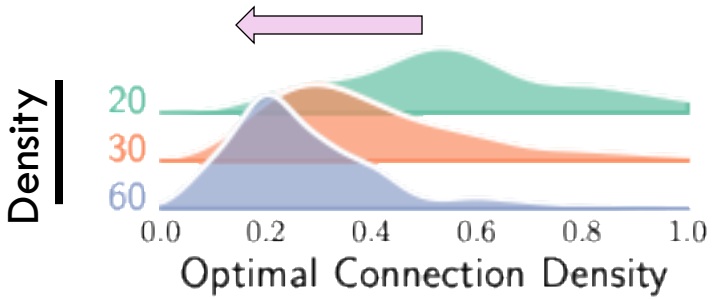


Network size

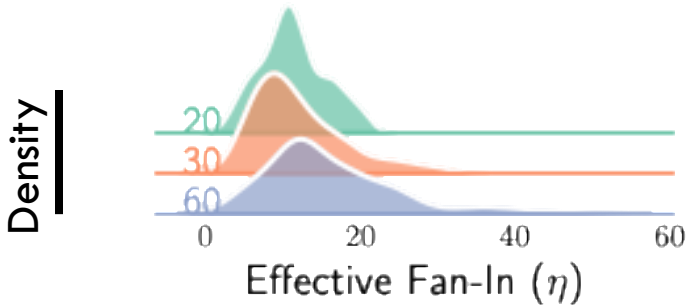
Slot/Chunk size



Small-world optimality

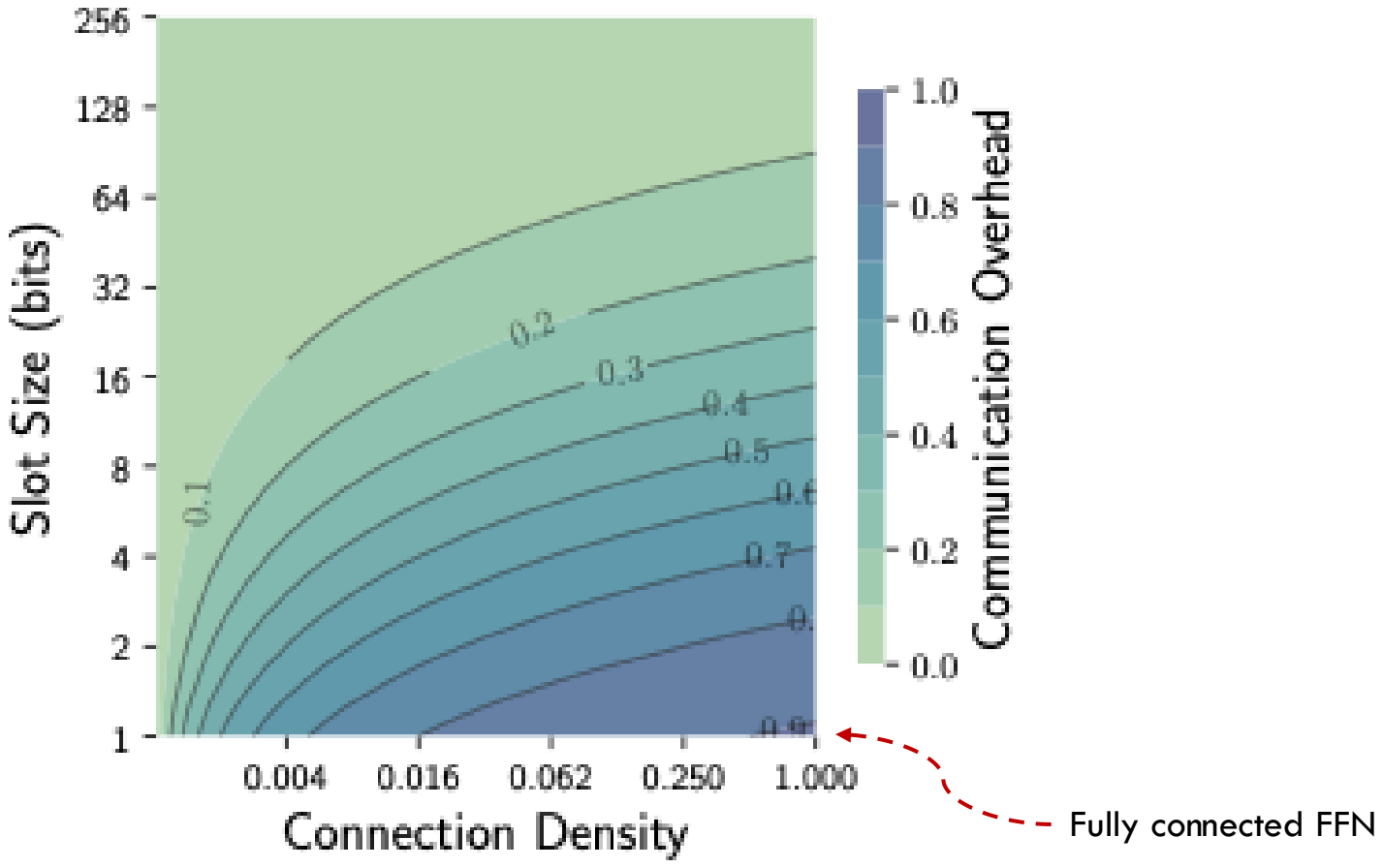


$\eta = N_c p$ → Effective Fan-in



η is constant over network size ⟹ some form of small-worldness

Overcoming Communication Bottlenecks



Assumptions

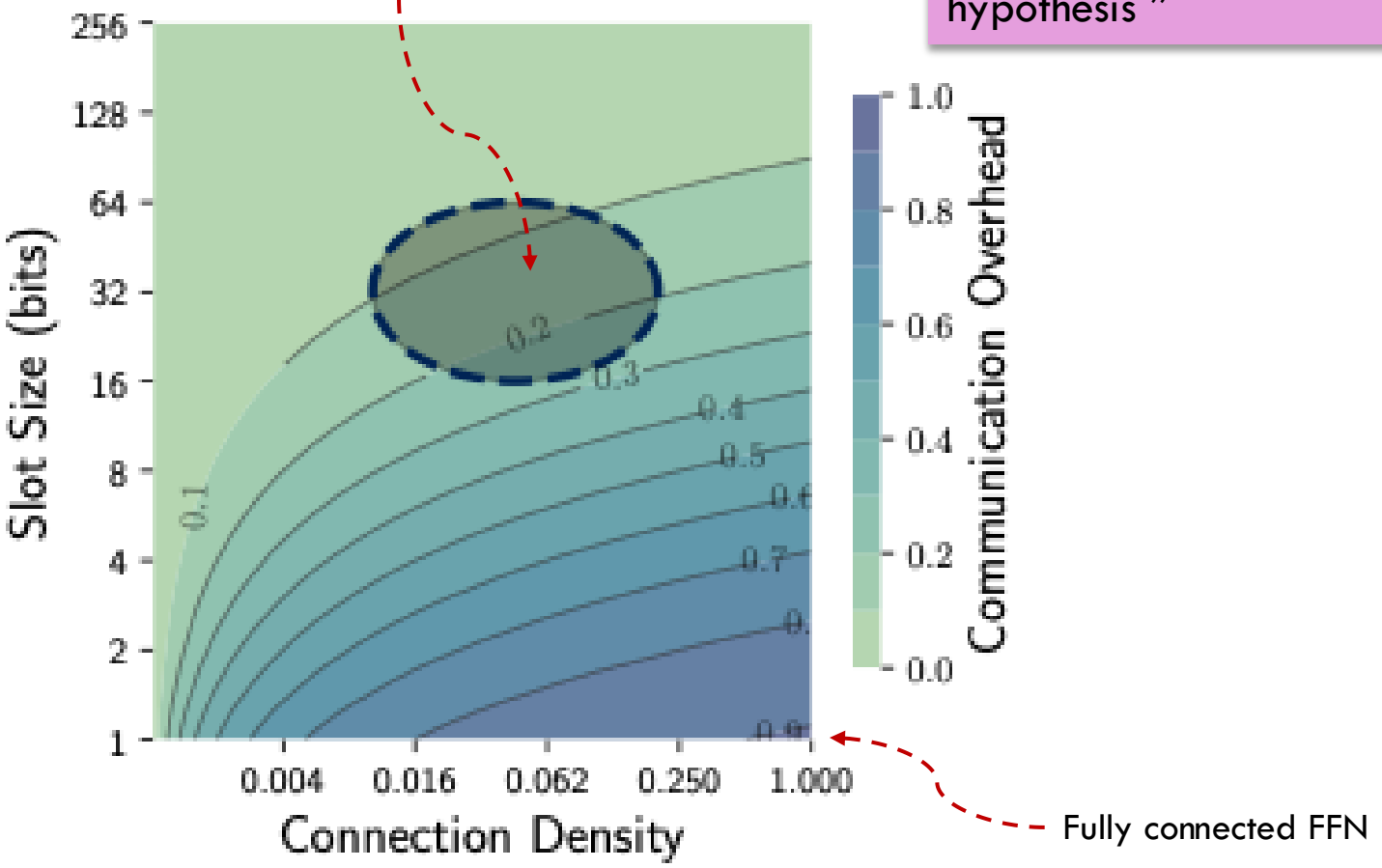
1. Packet-switched communication
2. Feedforward block-sparse layer
3. Address bits (A) $\propto \log_2$ (#chunks x connection density)
4. Data bits (D) $\propto$ chunk size

Communication Overhead (O)

$$O(A, D) = \frac{A}{A + D}$$

Overcoming Communication Bottlenecks

Best performance parameters



Fully connected FFN

Echoing the "lottery-ticket hypothesis*"

Assumptions

1. Packet-switched communication
2. Feedforward block-sparse layer
3. Address bits (A) $\propto \log_2$ (#chunks x connection density)
4. Data bits (D) $\propto$ chunk size

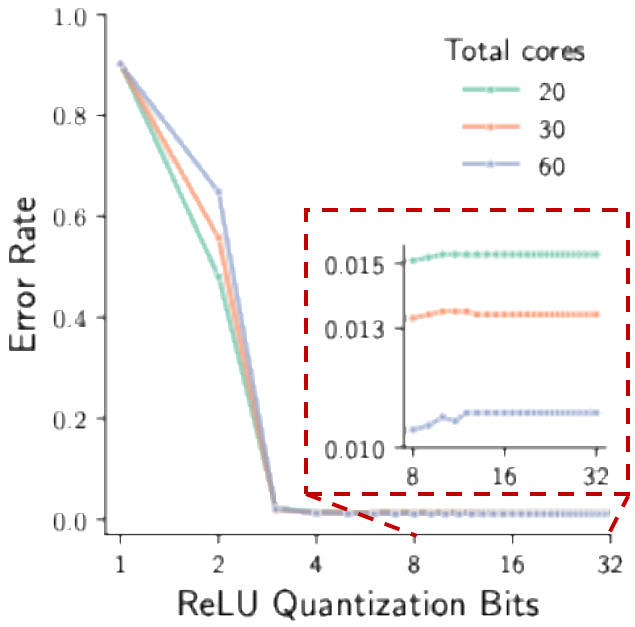
Communication Overhead (O)

$$O(A, D) = \frac{A}{A + D}$$

*Frankle, J. & Carbin, M. *arXiv* (2018).

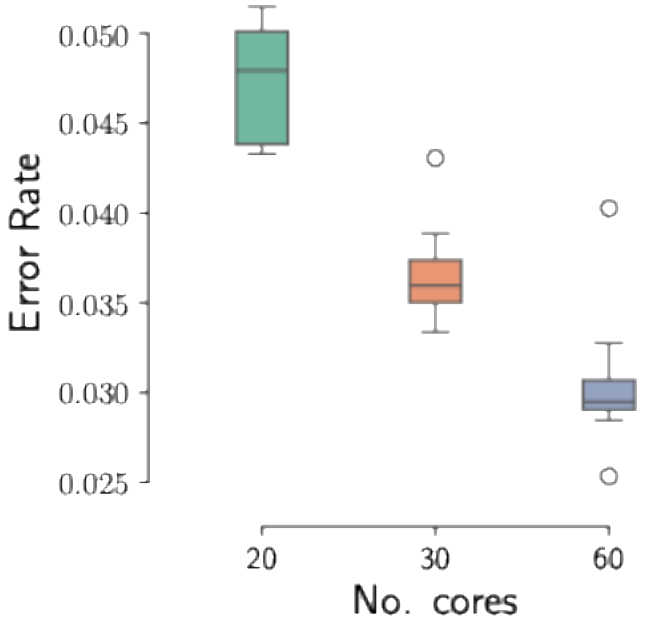
Quantized ScRRAMBLe Networks

Post Training Quantization



- Full precision training
- Quantize ReLU-activation post training
- Error rate returns to near full-precision with only about 4-bits[†]

Binary activation



- 1-bit (binary) activation during training
- Straight-through estimator
- Error-rate falls to near full precision baseline

[†]Wan, W. *et al. Nature* **608,** 504–512 (2022).

Designing CIM accelerators with a new generation of neural networks

Insights for hardware design

1. Structured sparsity → efficient design.

2. Routing can be part of the compute and not just data transfer.

3. Weight sharing is effective in MVM, much like convolutions.

How can we make neural networks ***block-sparse*** for CIM?

Insights into NN design

1. Block-sparsity as a regularizer.

2. Vector representations can eliminate expensive off-chip digital controllers + enable generative modeling

3. Block-sparse networks as a stand-in for FFNs.

Acknowledgements

Funding Sources

Office of Naval Research
ONR
Science & Technology

NSF

MICROELECTRONICS COMMONS
BRIDGING THE GAP FROM LAB TO FAB

Designing CIM accelerators with a new generation of neural networks

Insights for hardware design

1. Structured sparsity → efficient design.

2. Routing can be part of the compute and not just data transfer.

3. Weight sharing is effective in MVM, much like convolutions.

How can we make neural networks ***block-sparse*** for CIM?

Insights into NN design

1. Block-sparsity as a regularizer.

2. Vector representations can eliminate expensive off-chip digital controllers + enable generative modeling

3. Block-sparse networks as a stand-in for FFNs.