Towards a chip architecture for acceleration of Deep Neural Networks using Analog Memory

Stefano Ambrogio
Pritish Narayanan
Hsinyu Tsai
Charles Mackin
An Chen
Robert M. Shelby
Geoffrey W. Burr

IBM Research – Almaden

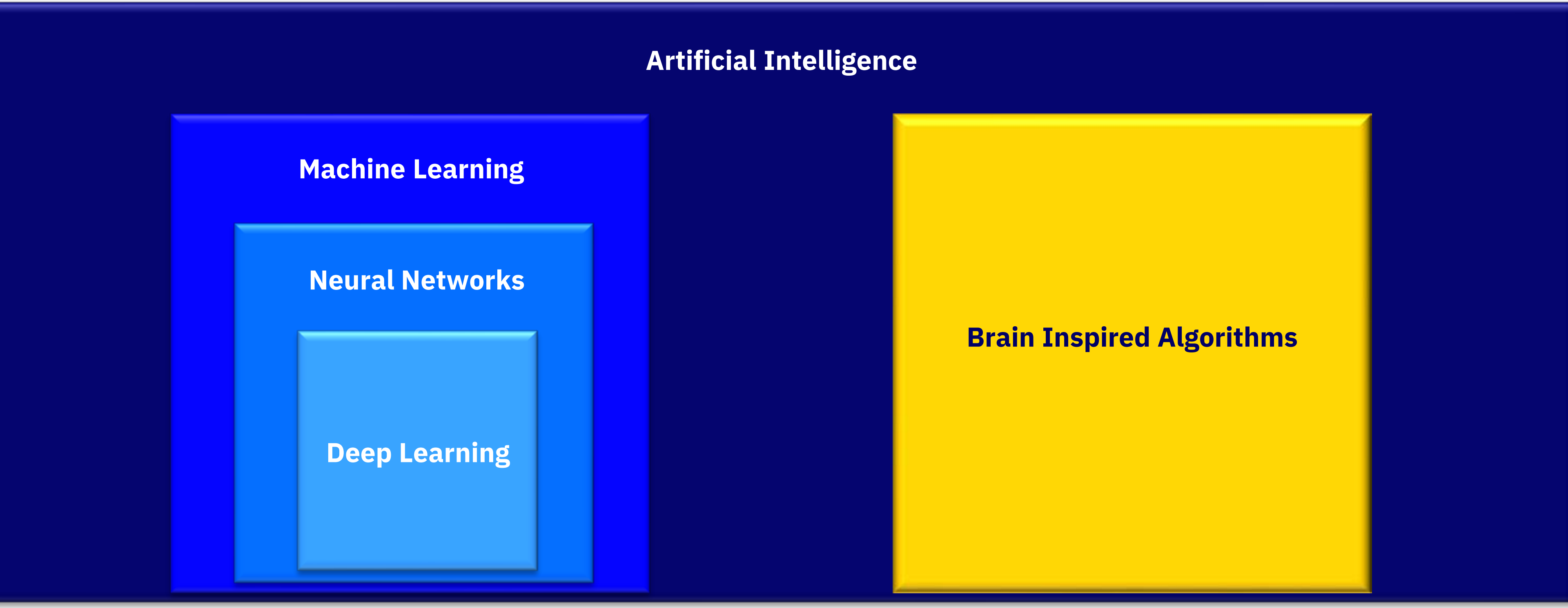
Outline

- Introduction
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Outline

- Introduction
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

What is AI?



Artificial Intelligence

Machine Learning

Neural Networks

Deep Learning

Brain Inspired Algorithms

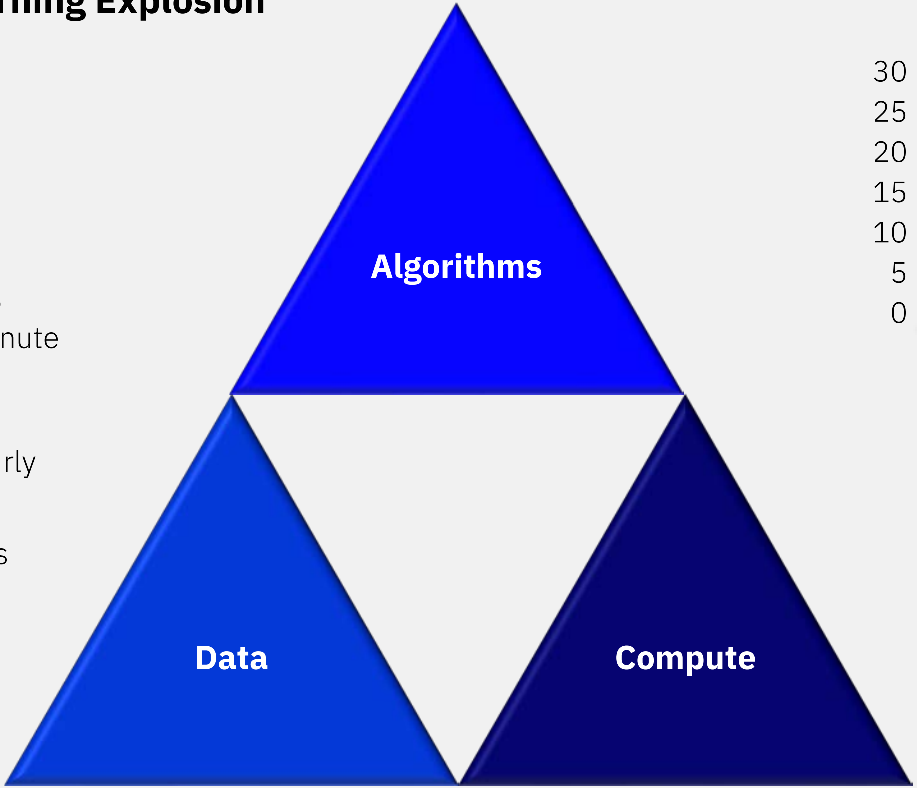
2012: AI foundations

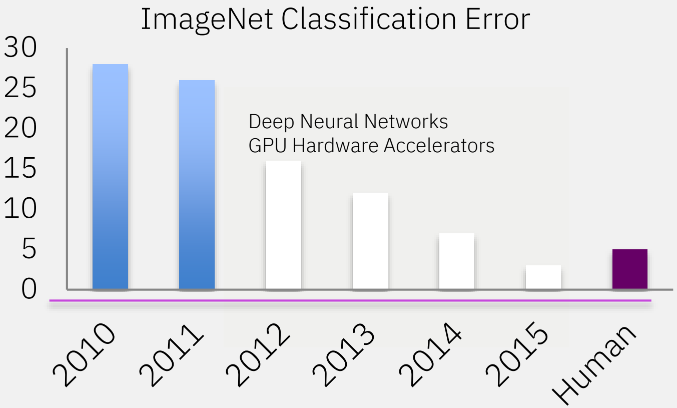
The Deep Learning Explosion

YouTube
400 hours of video uploaded every minute

Walmart
2.5 petabytes of customer data hourly

Facebook
350 million images uploaded daily




ImageNet Classification Error

Deep Neural Networks
GPU Hardware Accelerators

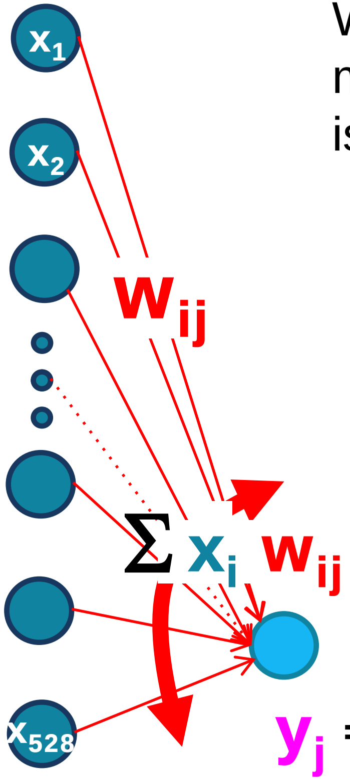
AI hardware, present & near-future: high-level view

	TODAY	VERY SOON	LATER ON…?
Forward Inference (in the cloud & at the edge)	**CPUs** **& GPUs**	TPU1 **Custom digital accelerators**	**Analog-memory-based accelerators** **?**
Training (mostly in the cloud)		TPU2	

Outline

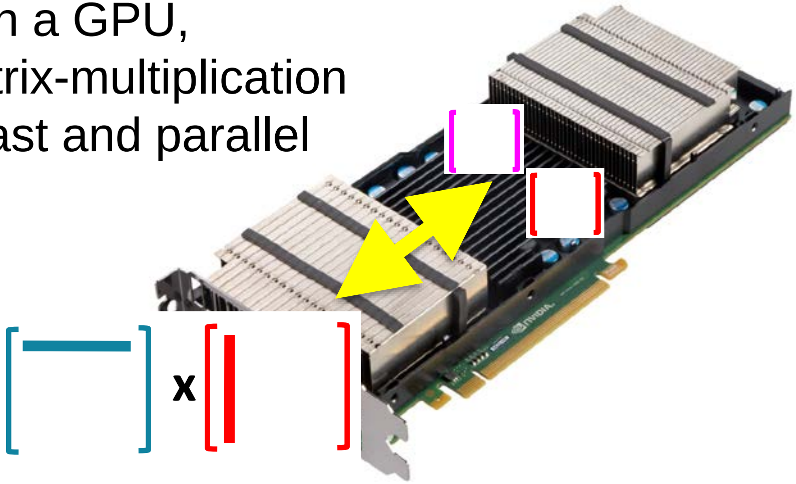
- Introduction
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Computation needed: “Multiply-accumulate”



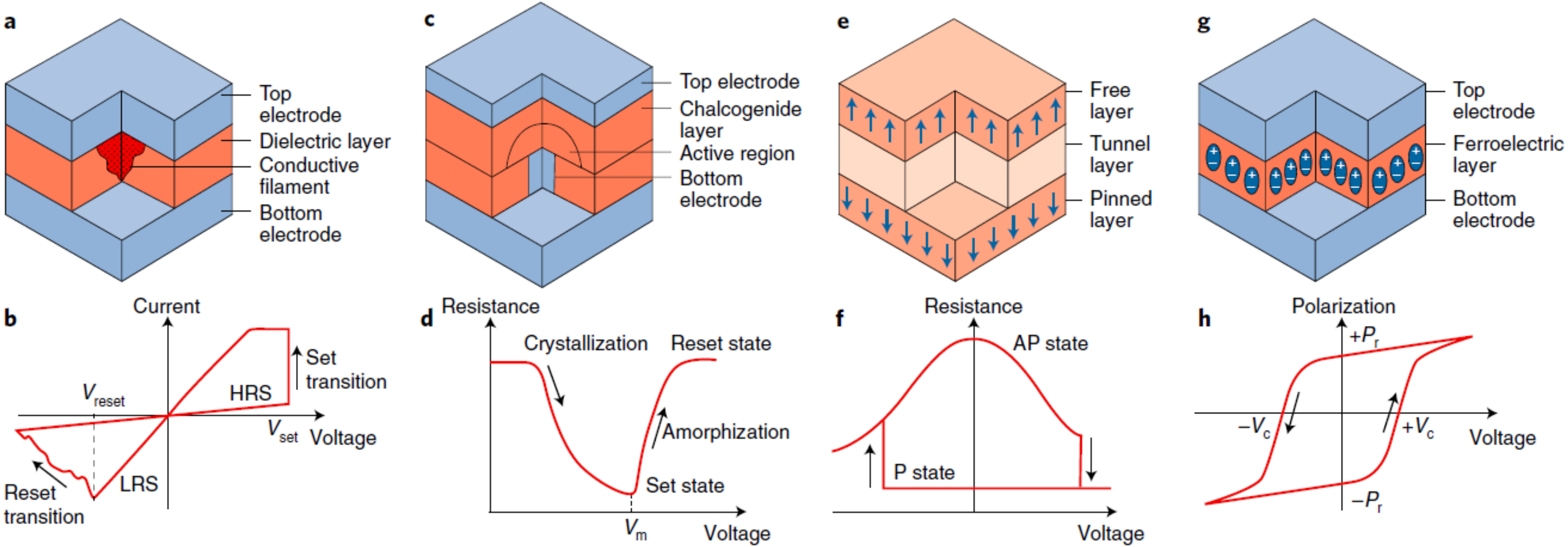
$y_j = f(\Sigma x_i w_{ij})$

With a GPU,
matrix-multiplication
is fast and parallel



but x and w values must arrive from DRAM
and new y values sent back to DRAM

Emerging devices for memory and computing



Resistive Memory (RRAM)

Phase-Change Memory (PCM)

Magnetic Memory (MRAM)

Ferro-Electric Memory (FeRAM)

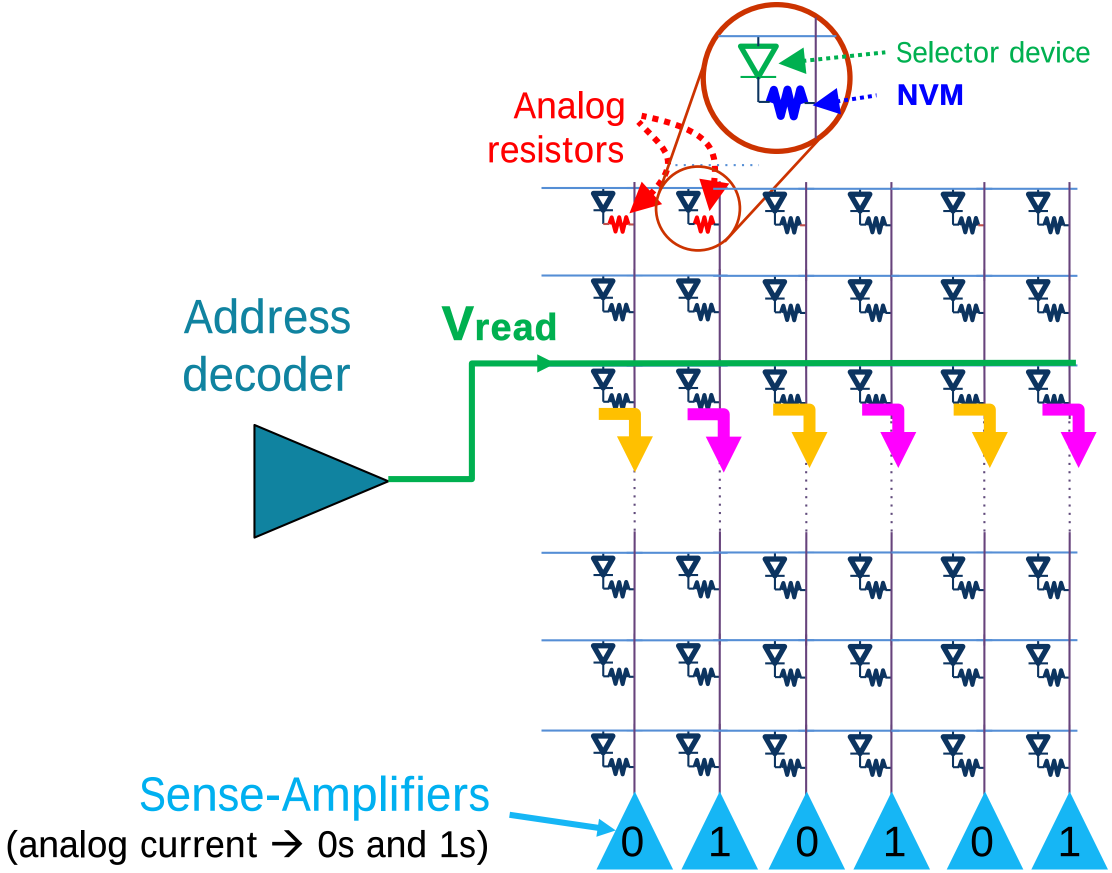
- Information encoded in the device conductance

D. Ielmini, H.-S. P. Wong, Nature Electronics (2018)

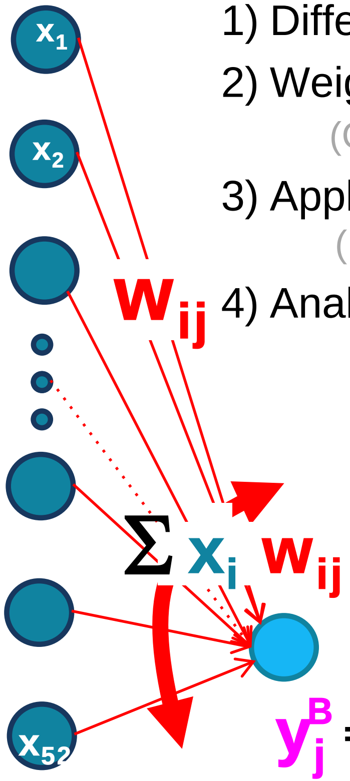
NVM (Non-Volatile Memory): usually for storing digital data (0s and 1s)

NVM technologies include:
MRAM (Magnetic RAM)
PCM (Phase-Change Memory)
RRAM (Resistance RAM)

Like conventional memory (SRAM/DRAM/Flash), an NVM is addressed one row at a time, to retrieve previously-stored digital data.



Multiply-accumulate with NVM: computed at the data, by physics



1) Different peripheral circuitry

2) Weights **w** → conductances **G^+, G^-**

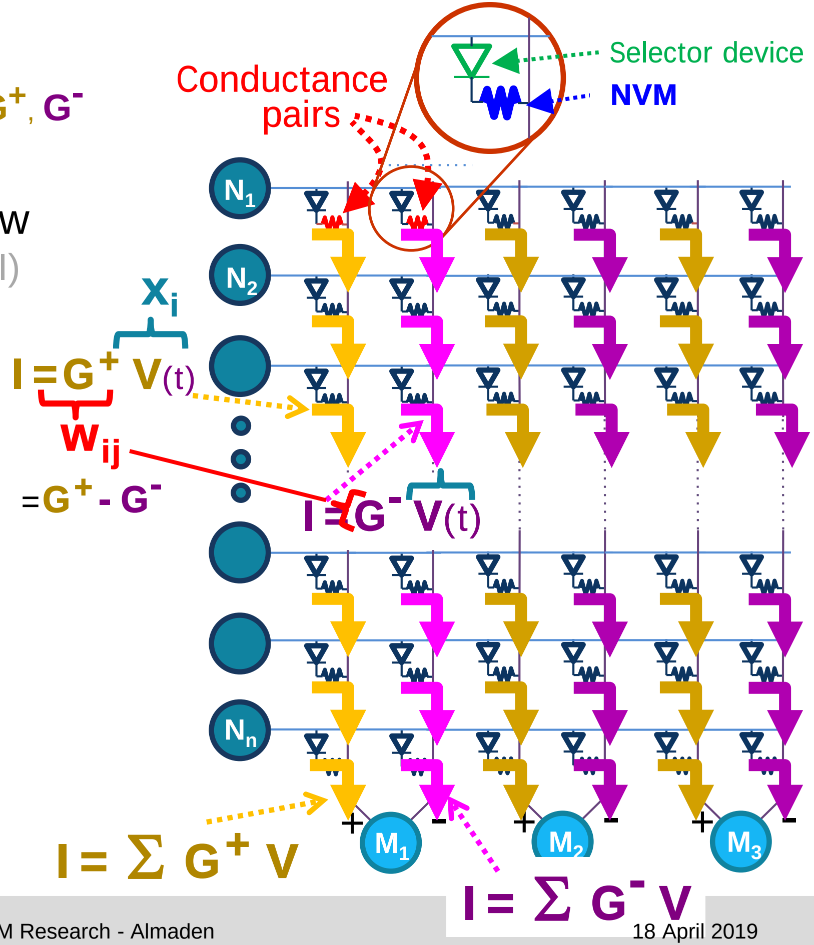
(Ohm's Law: V= IR → I = GV)

3) Apply "**x**" voltages to **every** row

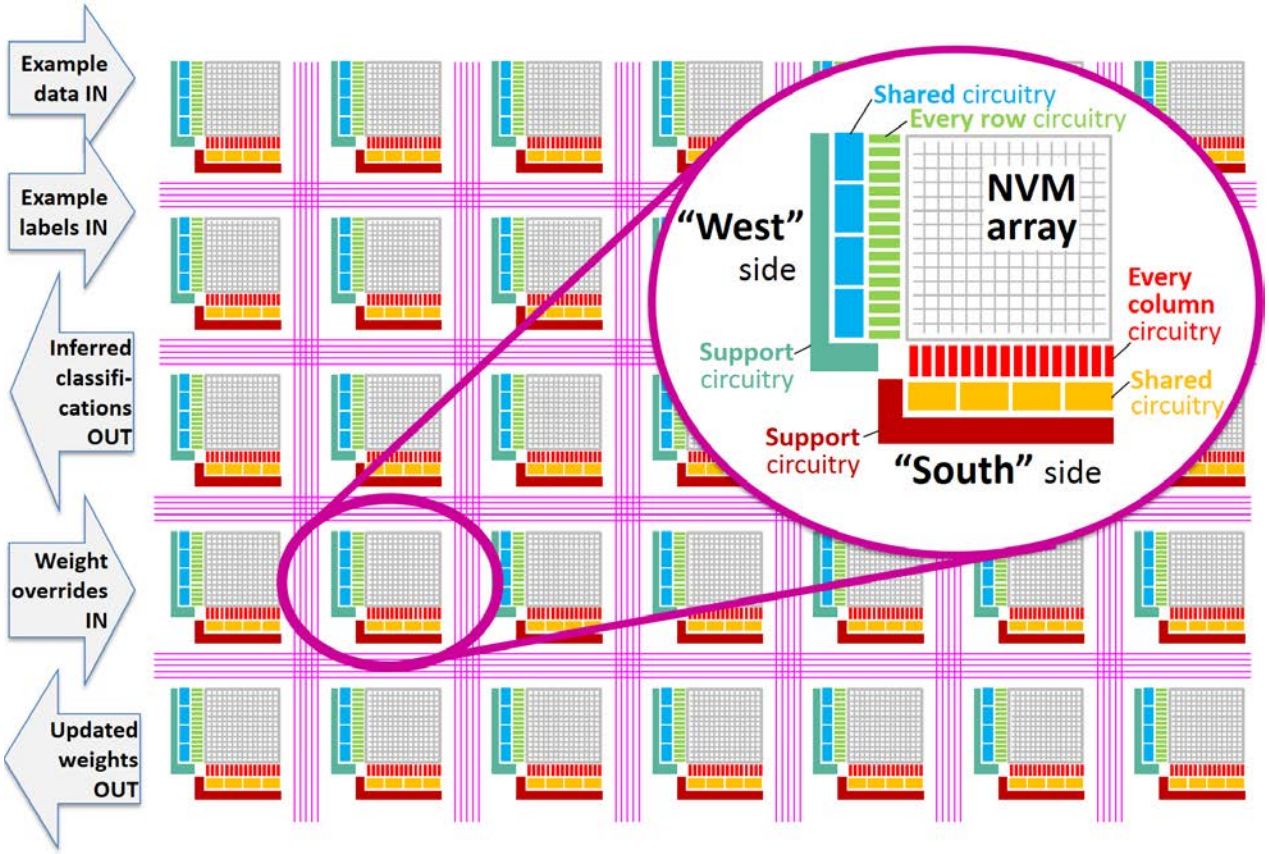
(Kirchhoff's Current Law → Σ I)

4) Analog measurement

$$y_j^B = f(\Sigma\ x_i\ w_{ij})$$

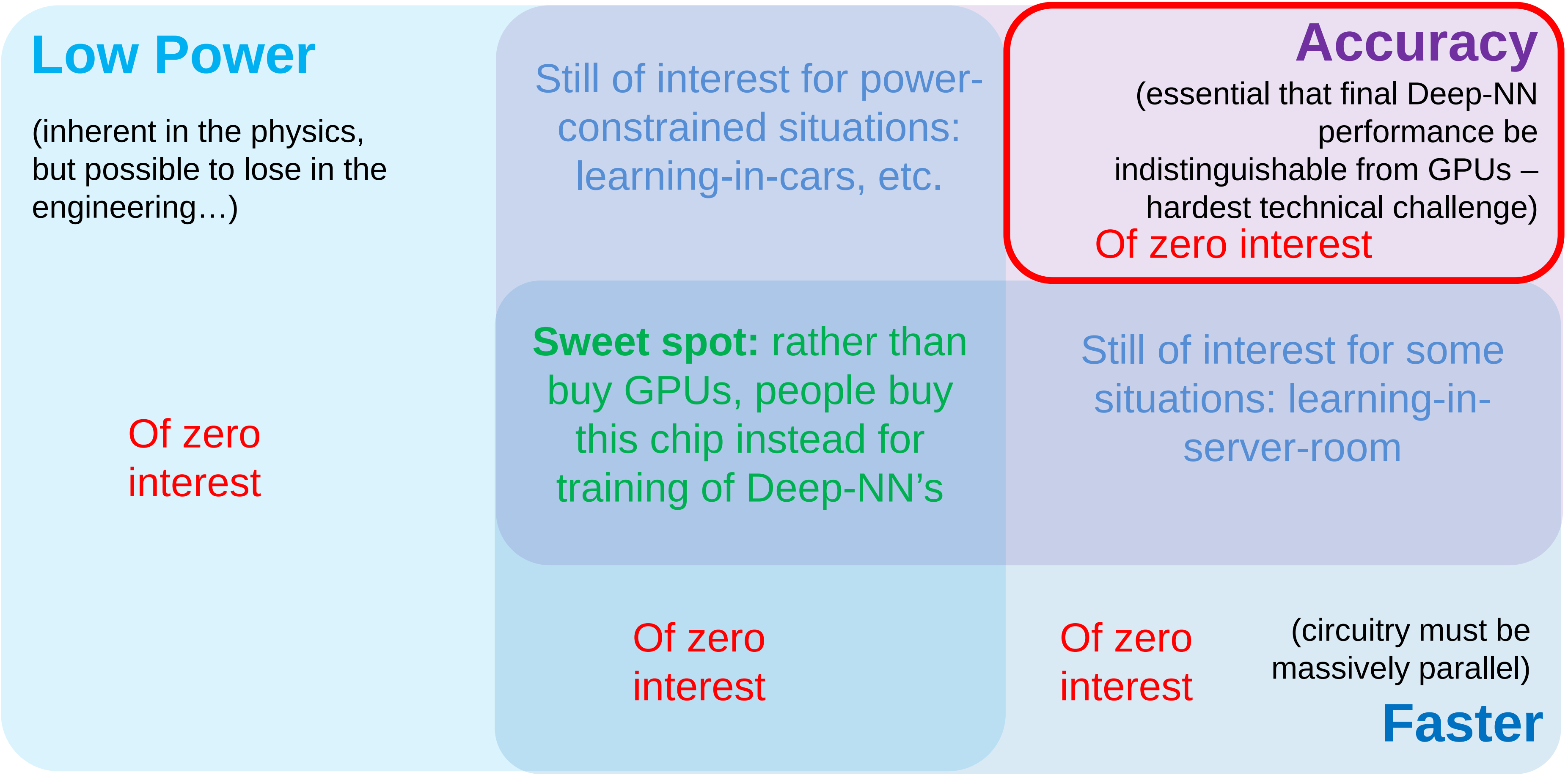


Vision: NVM-based Deep Learning Chip



- Support multiple deep learning algorithms
- Reconfigurable routing: Map different neural net topologies to the same chip
- Weight override mechanism for distributed learning

Maximizing the future business case (vs. a GPU)



Low Power

(inherent in the physics, but possible to lose in the engineering…)

Of zero interest

Still of interest for power-constrained situations: learning-in-cars, etc.

Accuracy

(essential that final Deep-NN performance be indistinguishable from GPUs – hardest technical challenge)

Of zero interest

Sweet spot: rather than buy GPUs, people buy this chip instead for training of Deep-NN's

Still of interest for some situations: learning-in-server-room

Of zero interest

Of zero interest

(circuitry must be massively parallel)

Faster

Outline

- Introduction
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Our journey towards high DNN accuracy

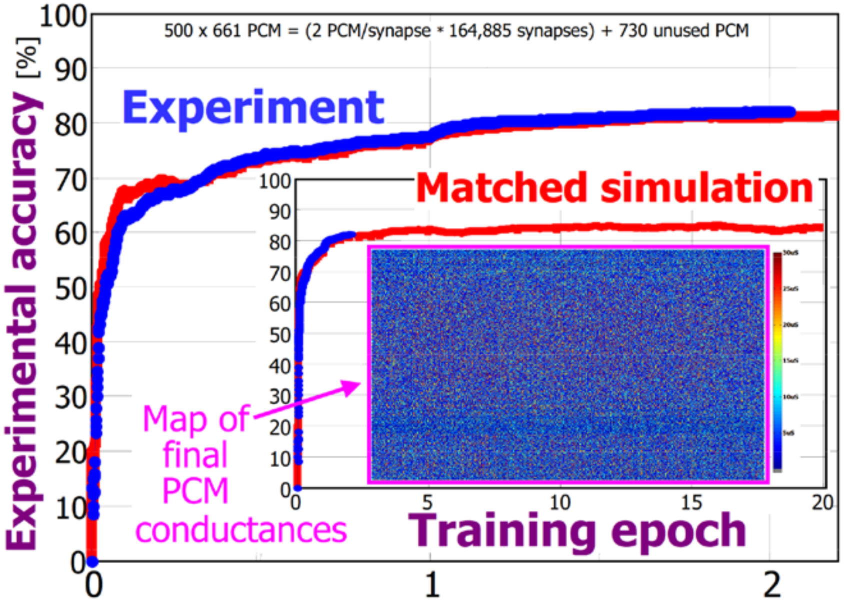
Where we were in **2014**

- Experiments on MNIST Dataset
- 82% accuracy w/ 5,000 examples,
- Too slow for 60,000 examples

"What a GPU would get" with this network…

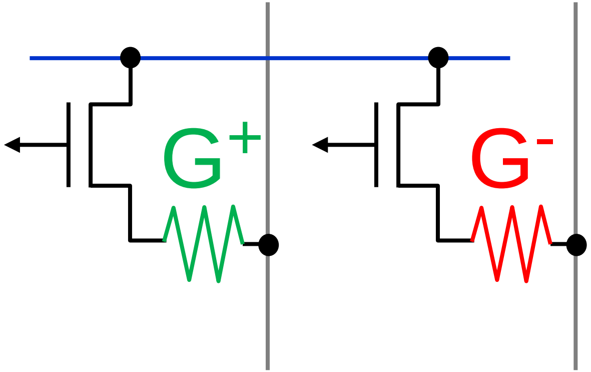
97-98% TEST accuracy w/ 60,000 examples

94% TEST accuracy w/ 5,000 examples



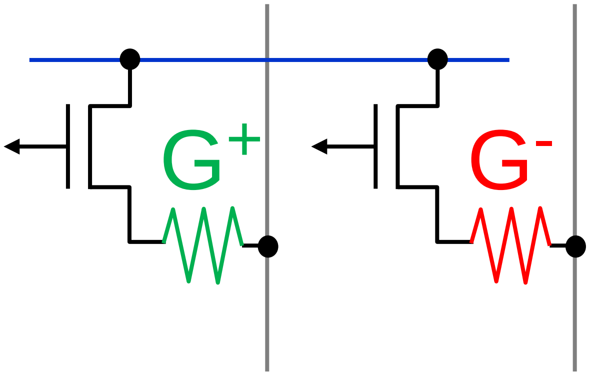
G. W. Burr, R. M. Shelby, et al., *IEDM Technical Digest*, 29.5, (2014).

} Non-idealities in Real PCM Devices

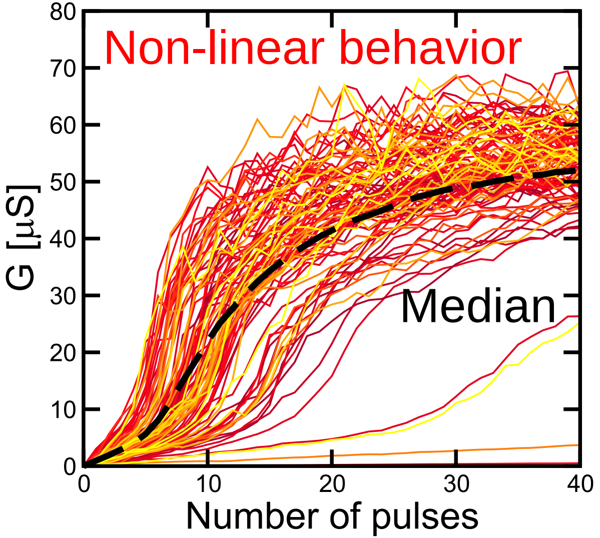


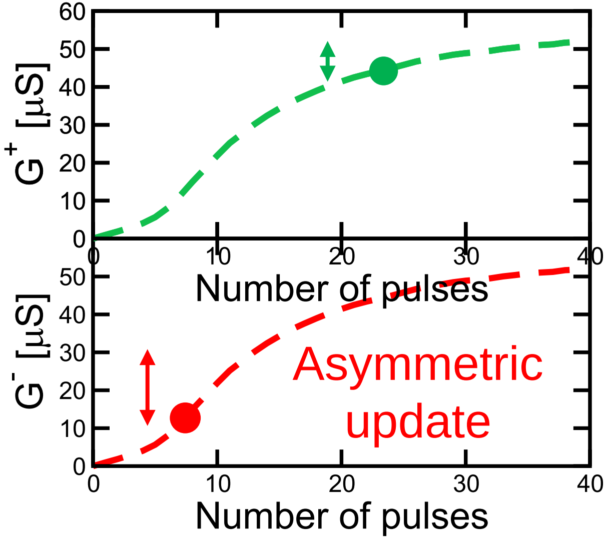
$$W = G^{+} - G^{-}$$

Study: 2-PCM: Asymmetric Conductance Response



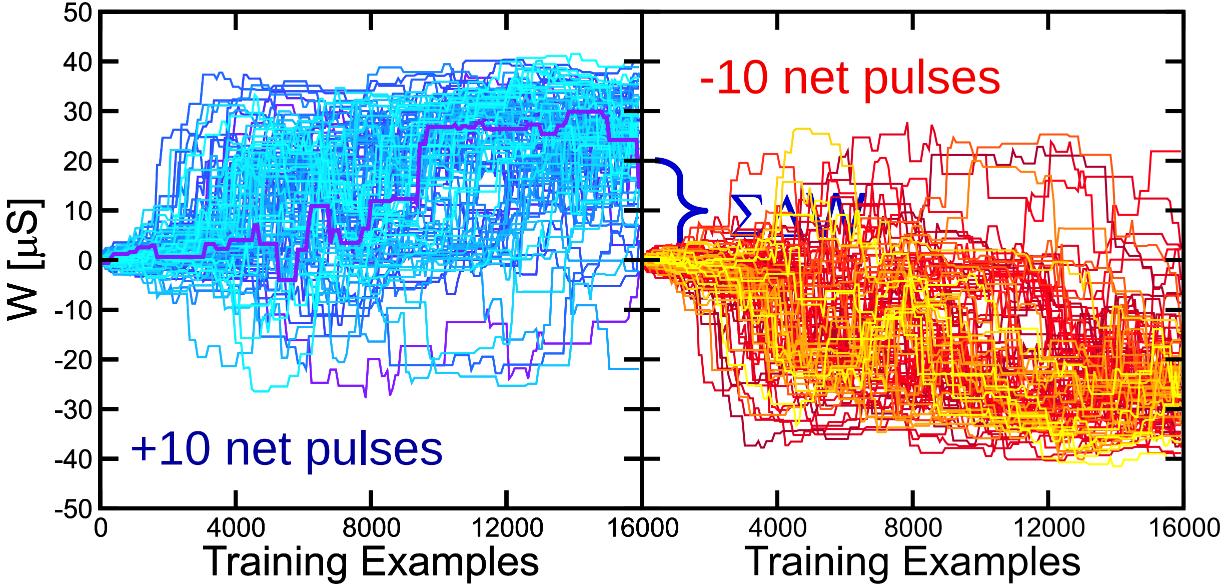
$W = G^+ - G^-$

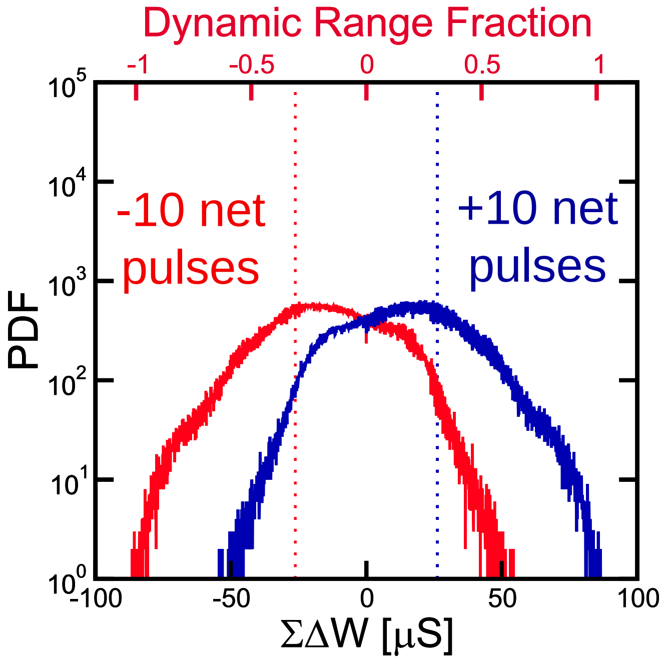




- 2-PCM unit cell is non-linear and asymmetric
- Symmetry is crucial to balance UP and DOWN steps and accurately implement open-loop weight update
- Strong impact on Neural Network training accuracy

2-PCM scheme: dependence on applied pulses





- ΣΔW distributions are overlapped, preventing a clear distinction of increase and decrease weight requests
- MNIST accuracy is lower than accuracy achieved with TensorFlow on a same size network

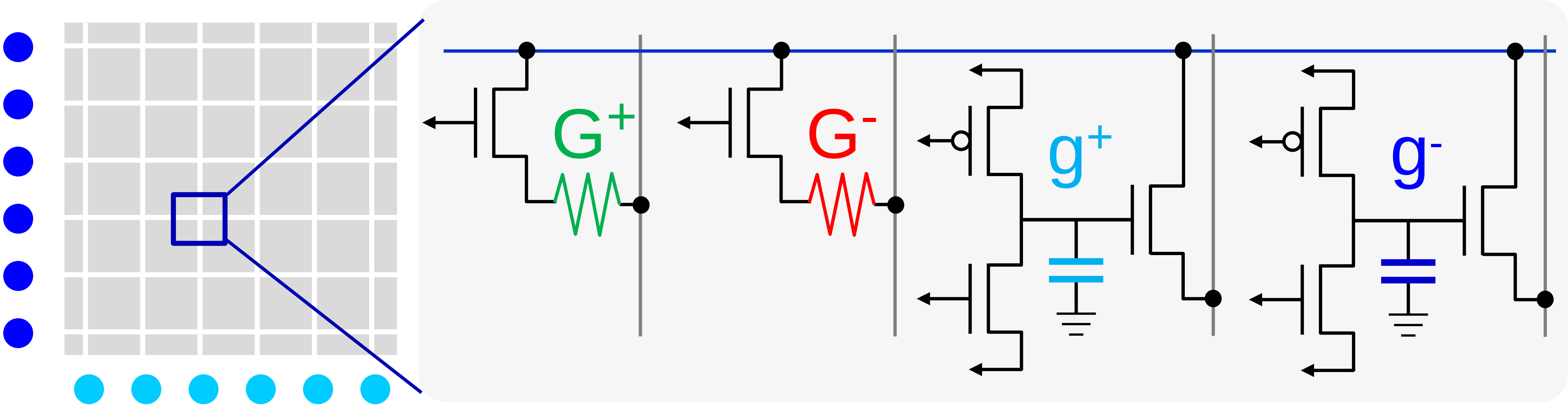
MNIST Accuracy
TensorFlow: 97.94%
2-PCM: 93.77%

Novel 2T2R + 3T1C unit cell

Most Significant Pair (MSP)

Least Significant Pair (LSP)

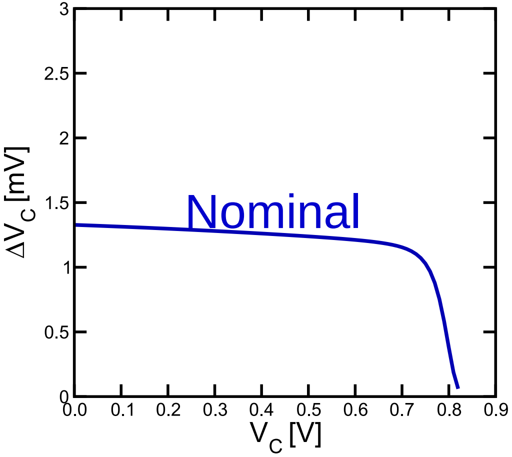
$$W = F \times (G^{+} - G^{-}) + g^{+} - g^{-}$$

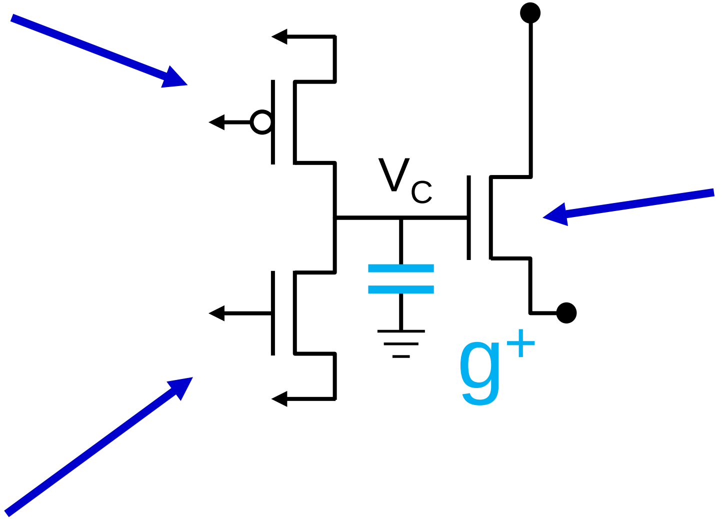


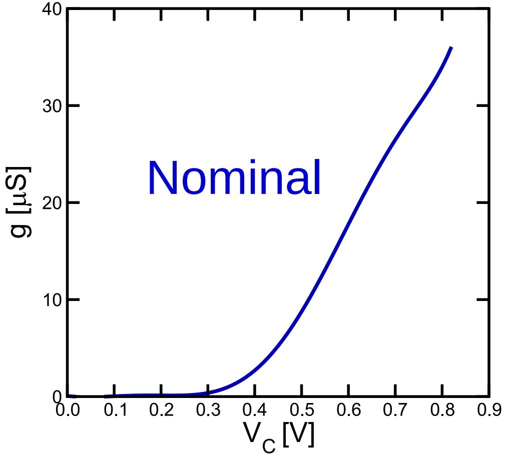
S. Ambrogio et al, *Nature*, 558, 60 (2018)

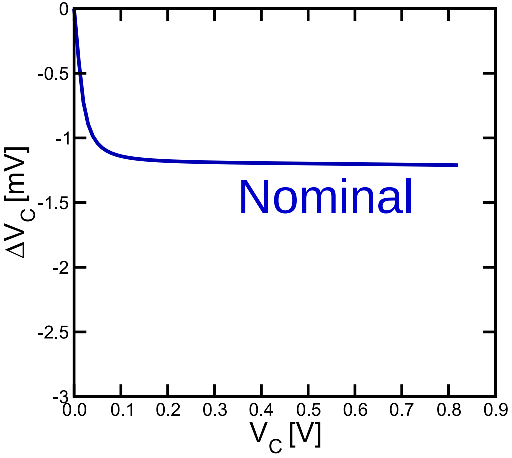
- Symmetry → Weight update performed on g+ only
  - g^{-} shared among many columns (e.g. 128 columns)
- Dynamic Range → Gain factor F (e.g. F = 3)
- Non-Volatility → Weight transferred to PCMs infrequently (every 1000s of images)

Novel unit cell: 2T2R + 3T1C, nominal behavior



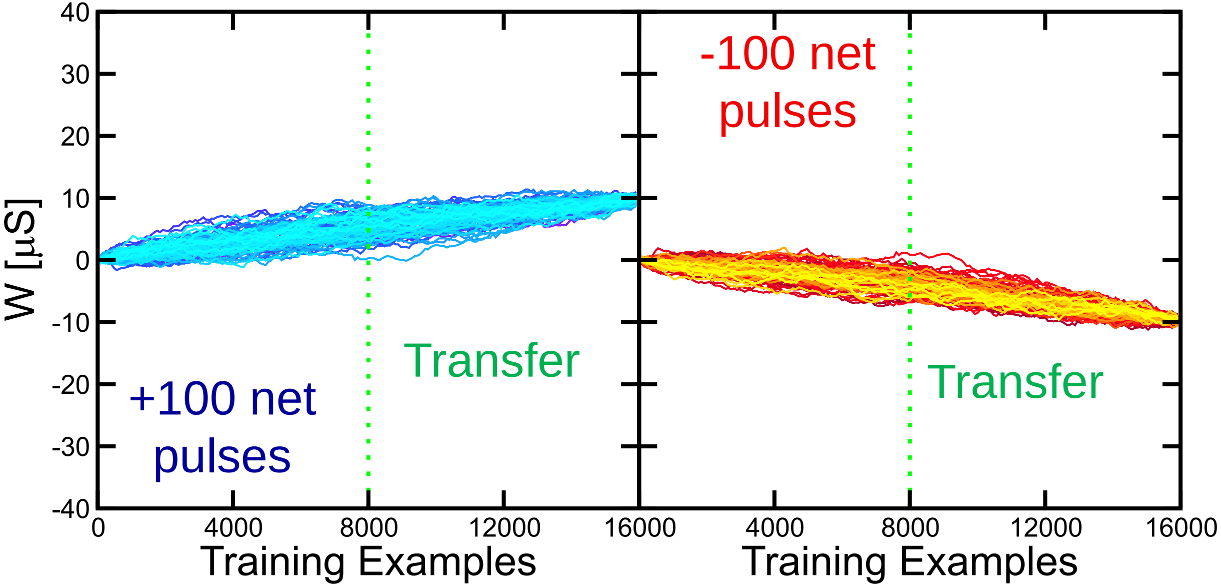


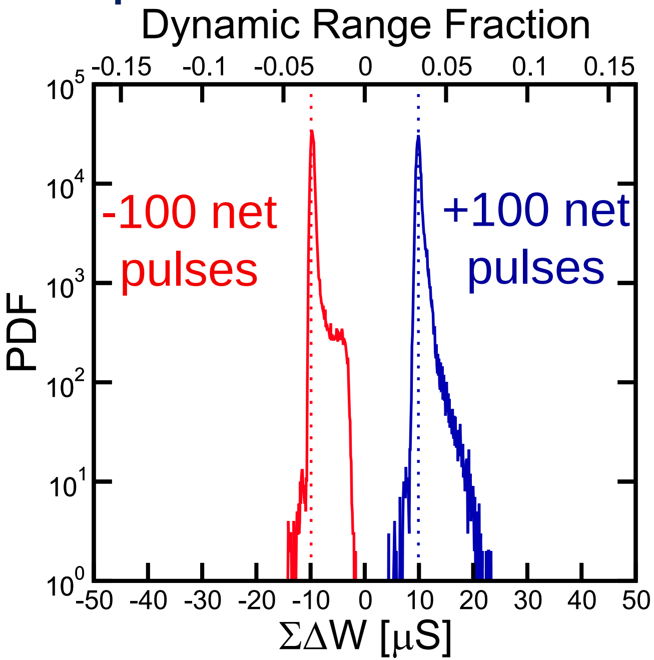




- PMOS charges the capacitor, increasing g+ and W
- NMOS discharges the capacitor, decreasing g+ and W
- Read MOS shows a linear dependence of g on V_C
- PMOS and NMOS provide the same current, balancing UP and DOWN weight updates

2T2R+3T1C scheme: dependence on applied pulses

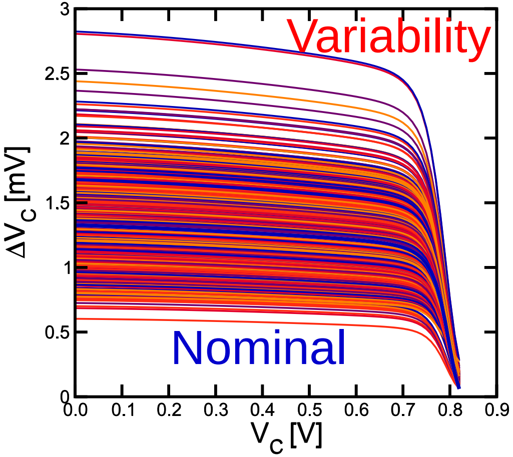


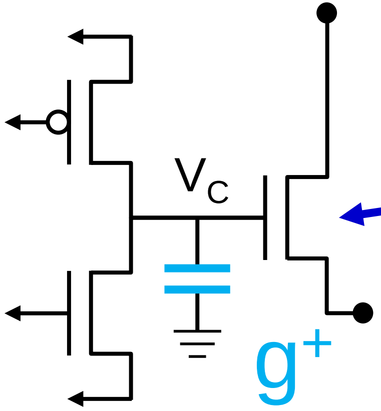


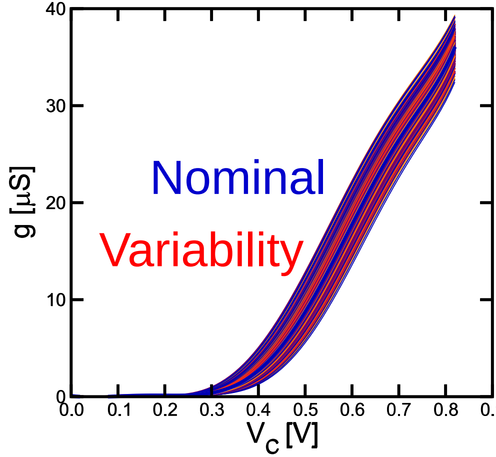
- Higher number of requested pulses due to very small g^+ update
- MNIST accuracy is equivalent to accuracy achieved with TensorFlow on a same size network

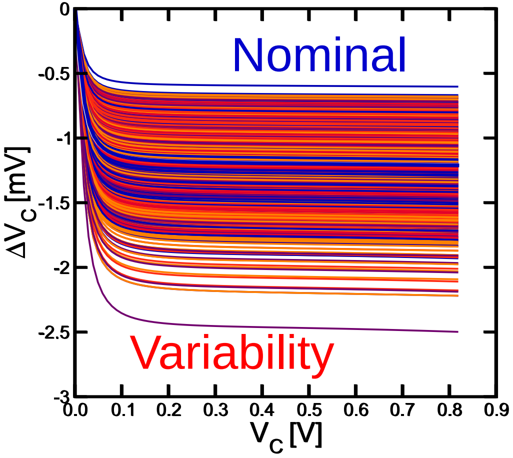
MNIST Accuracy
TensorFlow: 97.94%
2T2R+3T1C: 98.10%

Novel unit cell: 2T2R + 3T1C, CMOS variability



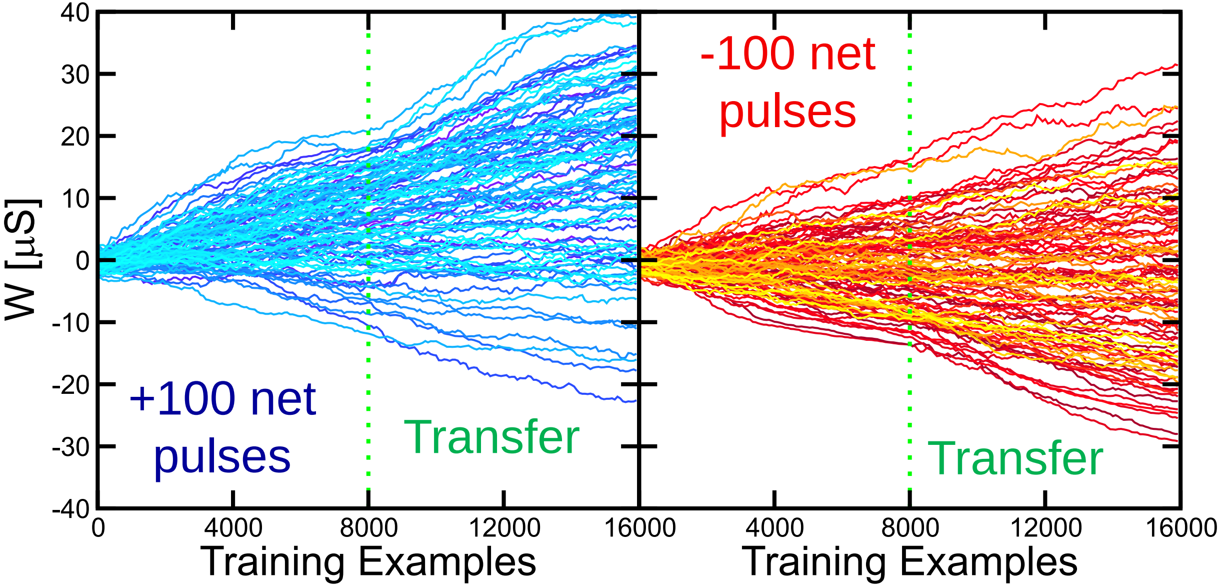


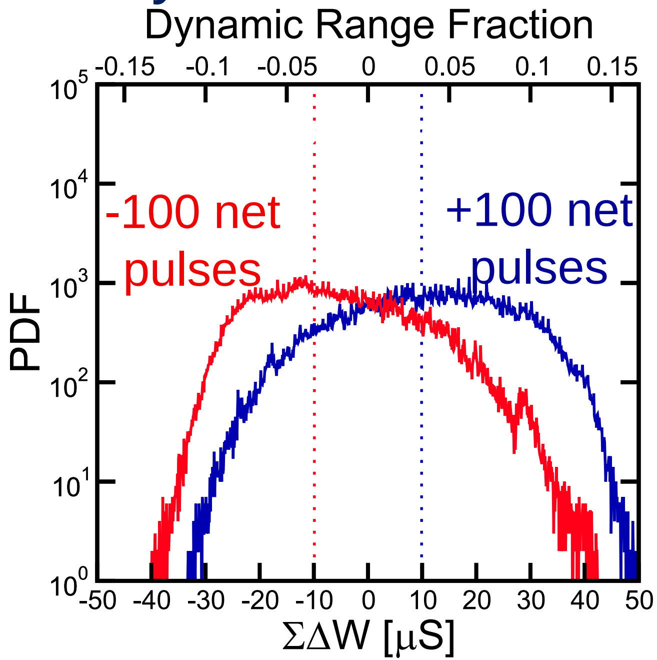




- PMOS charges the capacitor, increasing g+ and W
- NMOS discharges the capacitor, decreasing g+ and W
- Read MOS shows a linear dependence of g on V_C
- PMOS and NMOS never provide the same current, causing UP and DOWN weight updates asymmetry

2T2R+3T1C scheme: impact of CMOS variability

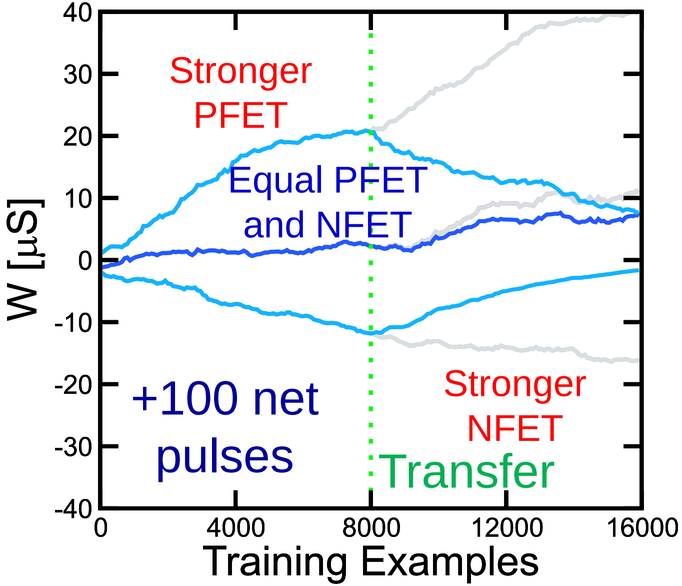


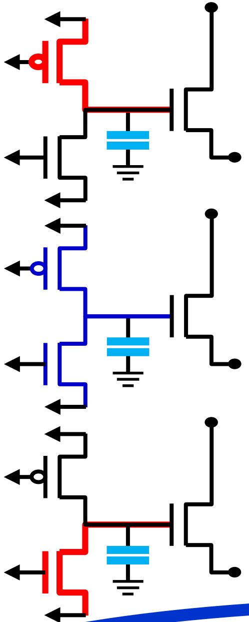


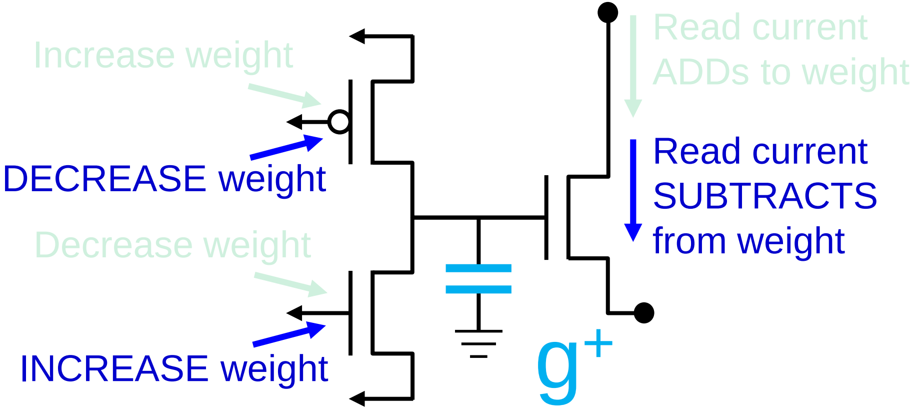
- Asymmetry in PMOS and NMOS strongly broadens ΣΔW distributions
- MNIST accuracy is highly degraded with respect to accuracy achieved with TensorFlow

MNIST Accuracy
TensorFlow: 97.94%
2T2R+3T1C: 98.10%
+Variability: 92.42%

2T2R+3T1C scheme: polarity inversion







Increase weight

DECREASE weight

Decrease weight

INCREASE weight

Read current ADDs to weight

Read current SUBTRACTS from weight

g^+

$W = F \times (G^+ - G^-) + g^+ - g^-$

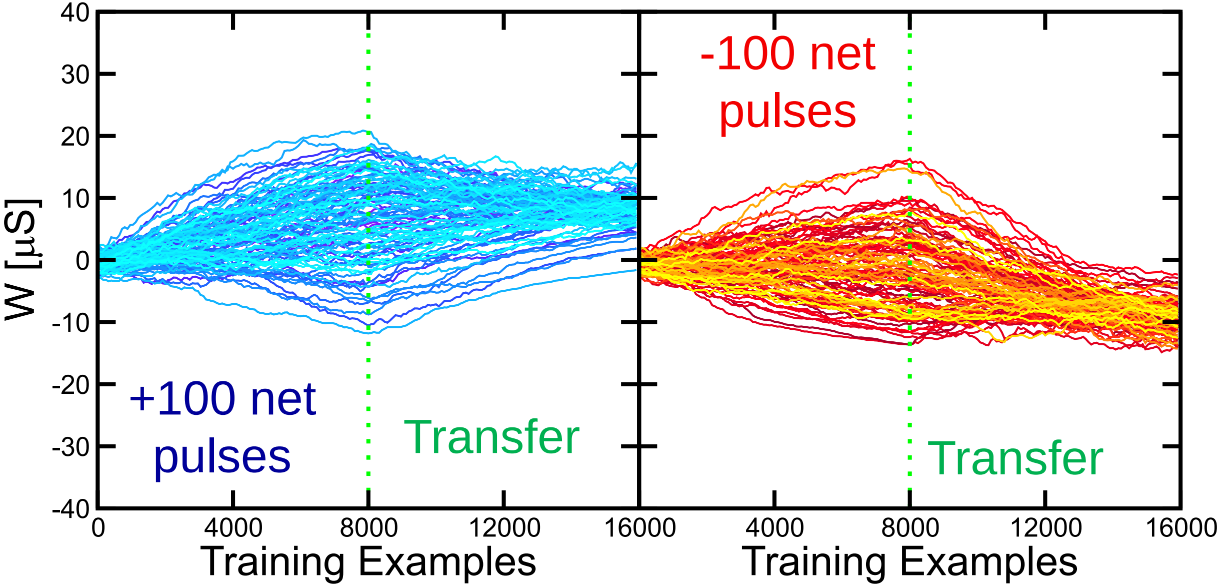
Transfer

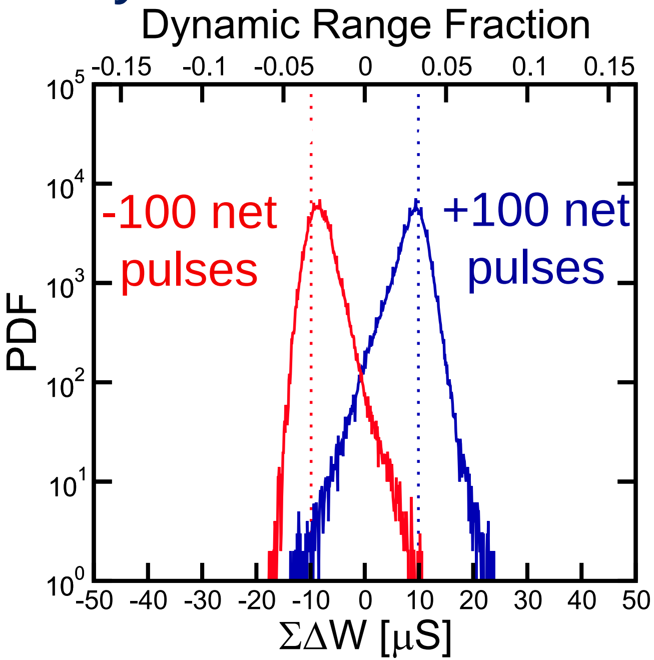
$W = F \times (G^+ - G^-) - (g^+ - g^-)$

Polarity inversion: Invert the sign of the lower significance conductance between transfers to higher significance pair

S. Ambrogio et al, *Nature*, 558, 60 (2018)

2T2R+3T1C scheme: CMOS variability, polarity inversion

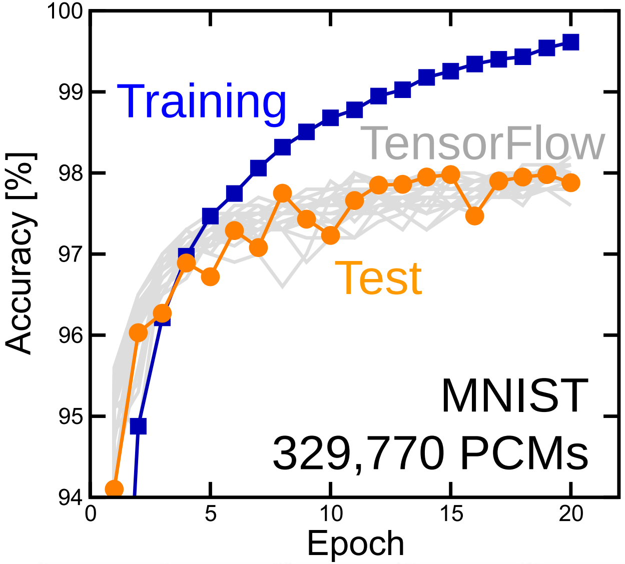


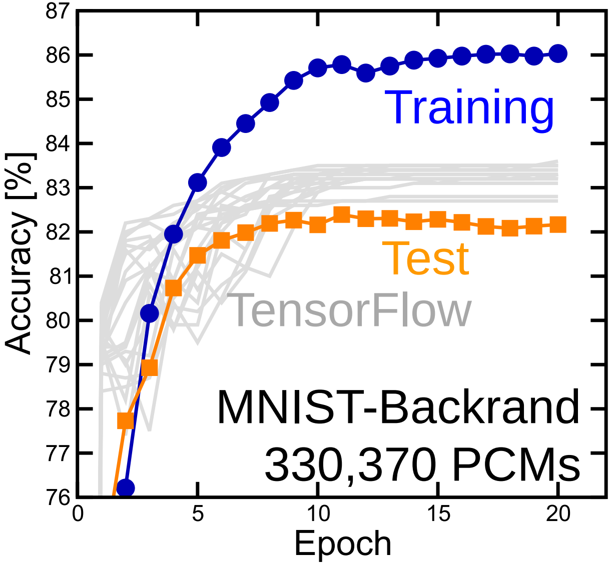


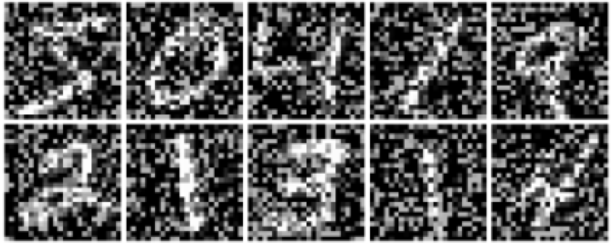
- Asymmetry in PMOS and NMOS is averaged by polarity inversion
- MNIST accuracy is equivalent to accuracy achieved with TensorFlow

MNIST Accuracy
Tensorflow: 97.94%
Polarity Inv: 97.95%

Accuracy on MNIST and MNIST backrand





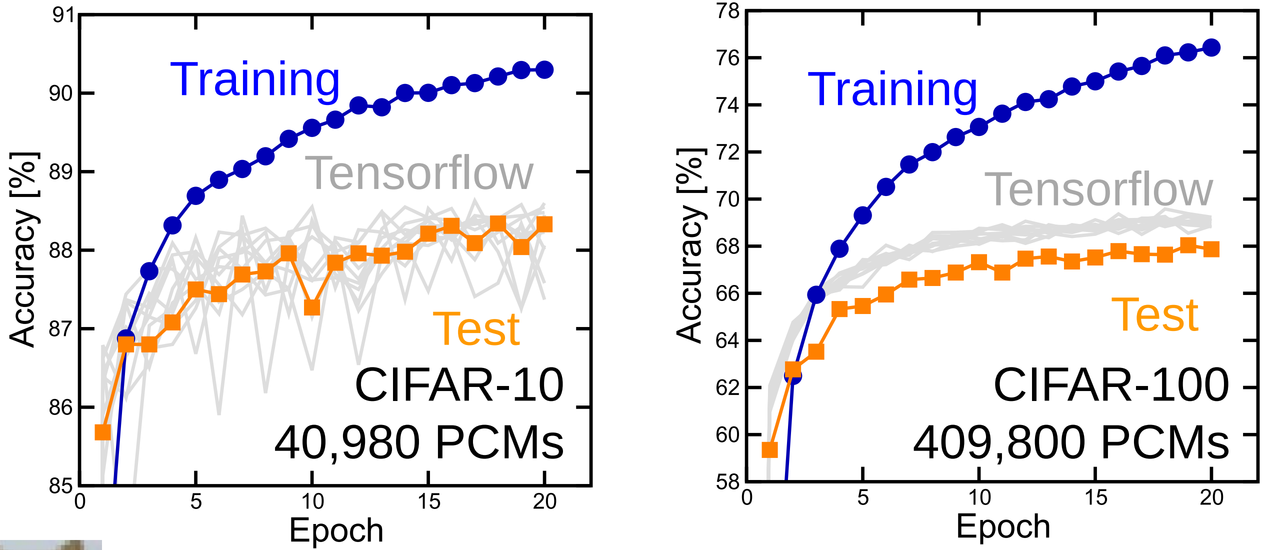


S. Ambrogio et al, *Nature*, 558, 60 (2018)

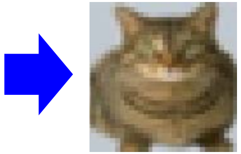
Mixed hardware-software experiment: every synaptic weight → 2 real PCM devices

Transfer learning from ImageNet to CIFAR-10/100

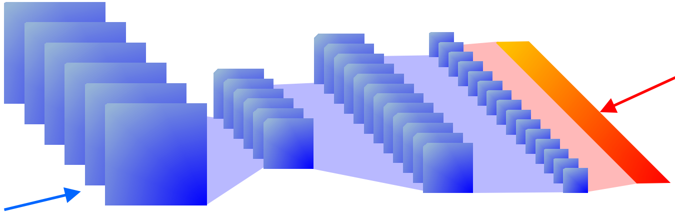
Mixed hardware-software experiment



ImageNET



CIFAR-10/100



Only train last fully-connected layer

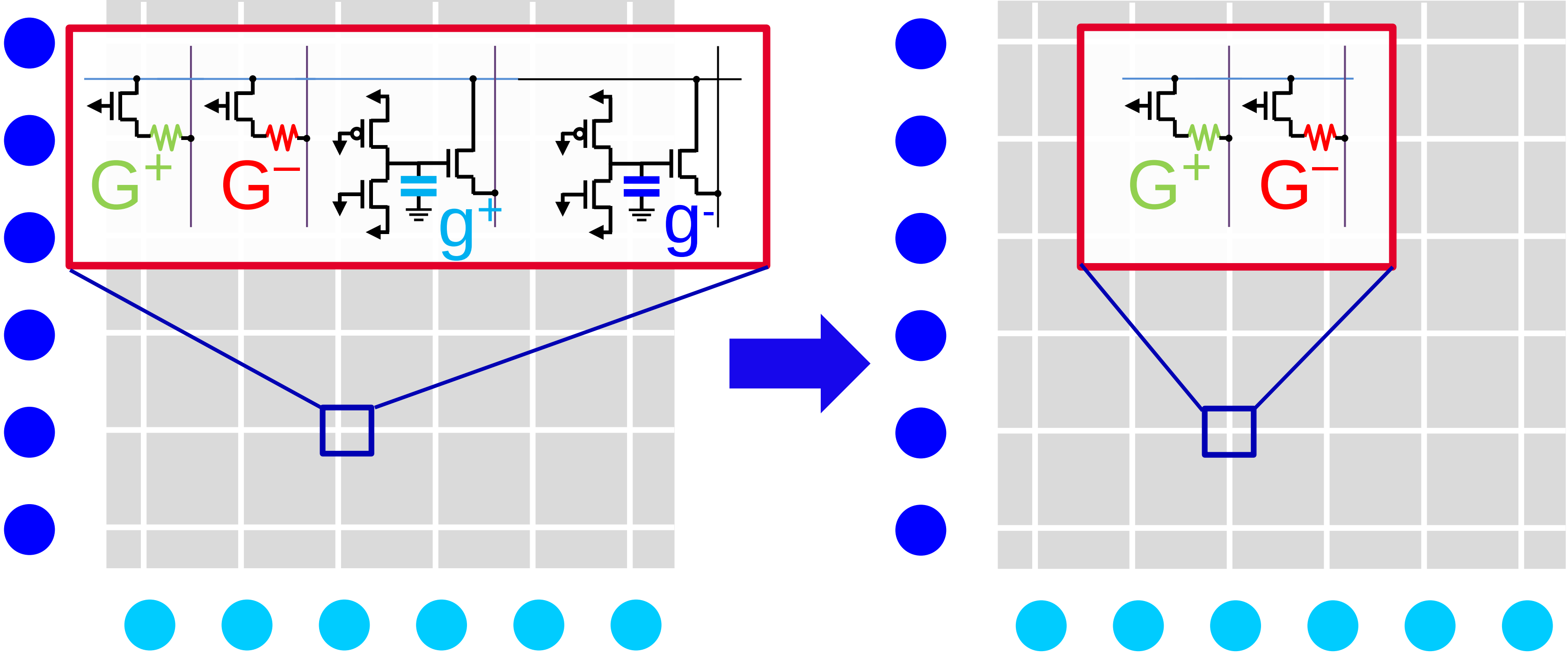
Fully Connected layer

Transfer Learning: Use pre-trained, scaled weights from ImageNET for convolution layers

Convolutional and Subsampling layers

Full 2-Analog Memory structure

$$W = F \times (G^+ - G^-) + g^+ - g^-$$



- Single pair of devices performing the entire training

Single device requirements

- Several specifications are requested to single resistive device in order to obtain software-equivalent accuracies

- A minimum of 1000 different conductance steps are required → extremely hard to obtain

- A maximum 5% of asymmetry between up and down conductance updates

  → need for very linear and symmetric devices

Our solution → Multiple conductances of varying significance, diversification of requirements

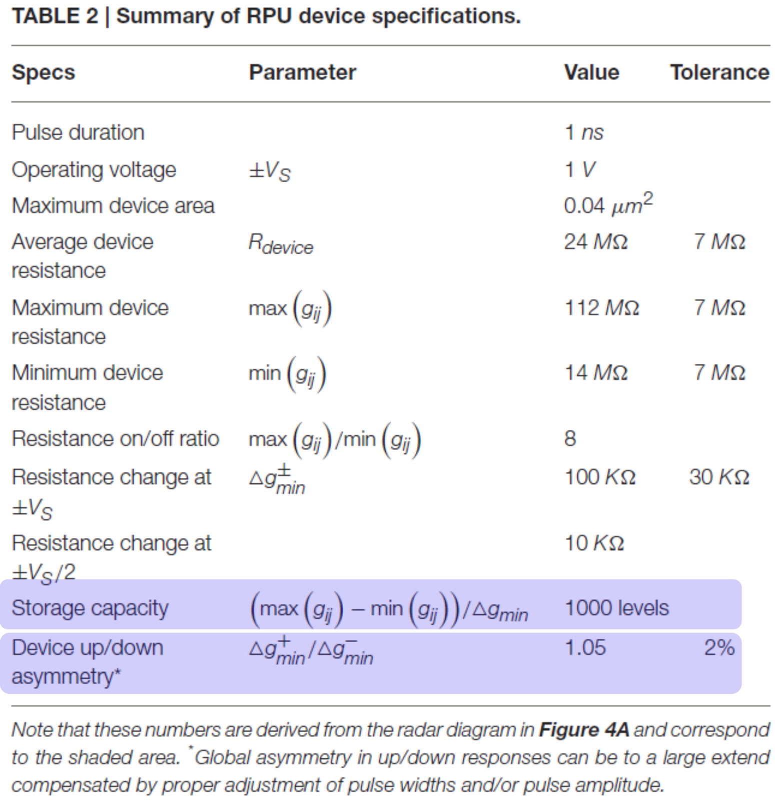


TABLE 2 | Summary of RPU device specifications.

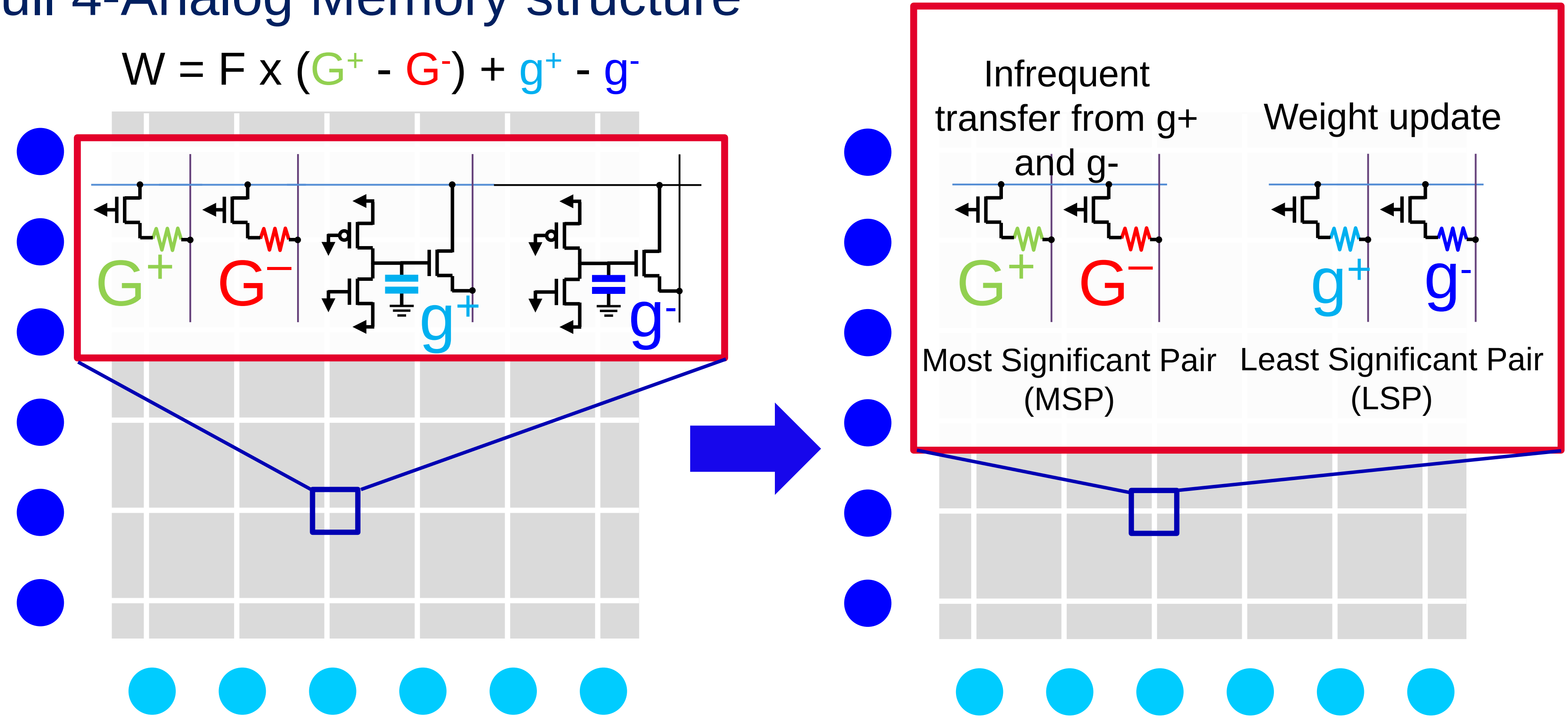
Specs	Parameter	Value	Tolerance
Pulse duration		1 ns	
Operating voltage	$\pm V_S$	1 V	
Maximum device area		0.04 μm^2	
Average device resistance	R_{device}	24 $M\Omega$	7 $M\Omega$
Maximum device resistance	$\max\left(g_{ij}\right)$	112 $M\Omega$	7 $M\Omega$
Minimum device resistance	$\min\left(g_{ij}\right)$	14 $M\Omega$	7 $M\Omega$
Resistance on/off ratio	$\max\left(g_{ij}\right)/\min\left(g_{ij}\right)$	8	
Resistance change at $\pm V_S$	$\Delta g_{min}^{\pm}$	100 $K\Omega$	30 $K\Omega$
Resistance change at $\pm V_S/2$		10 $K\Omega$	
Storage capacity	$\left(\max\left(g_{ij}\right) - \min\left(g_{ij}\right)\right)/\Delta g_{min}$	1000 levels	
Device up/down asymmetry*	$\Delta g_{min}^{+}/\Delta g_{min}^{-}$	1.05	2%

*Note that these numbers are derived from the radar diagram in **Figure 4A** and correspond to the shaded area. *Global asymmetry in up/down responses can be to a large extend compensated by proper adjustment of pulse widths and/or pulse amplitude.*

T. Gokmen, Y. Vlasov, *Frontiers in neuroscience* 10, 333 (2016)

Full 4-Analog Memory structure

$$W = F \times (G^+ - G^-) + g^+ - g^-$$



Infrequent transfer from g+ and g-

Weight update

Most Significant Pair (MSP)

Least Significant Pair (LSP)

- Most Significant Pair: Infrequent, **Closed Loop Programming** Operation
- Least Significant Pair: Frequent, **Open Loop Programming** Operation

Suggestions for new analog memory devices

- **Larger unit cell with two components**

1. More-significant pair of non-volatile conductances (e.g., PCM) stores “most” of the weight info

    - Non-linear conductance update → OK
    - DOES need to be able to tune these conductances rapidly in a CLOSED-LOOP manner

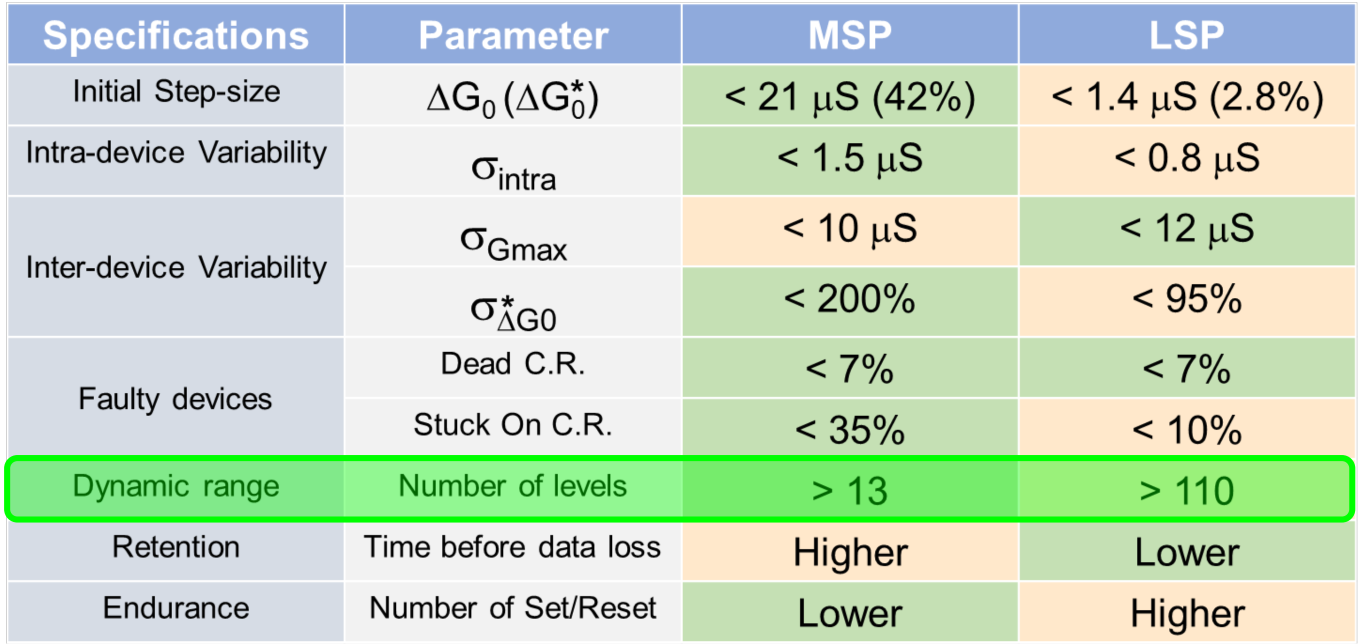
2. We perform all the OPEN-LOOP programming using a “less-significant” pair of conductances

    - Poor retention → OK
    - Significant device-to-device fixed variabilities → OK
    - DOES need to offer highly linear conductance update

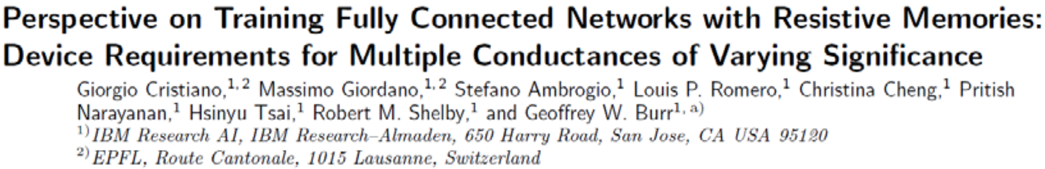
→ **Reduces the difficulty of device requirements**

S. Ambrogio et al, *Nature*, 558, 60 (2018)
G. Cristiano et al, J. Appl. Phys. 124 (15), 151901 (2018)

Comparison of device specifications for MSP and LSP



Specifications	Parameter	MSP	LSP
Initial Step-size	$\Delta G_0 \, (\Delta G_0^*)$	$< 21\ \mu S\ (42\%)$	$< 1.4\ \mu S\ (2.8\%)$
Intra-device Variability	σ_{intra}	$< 1.5\ \mu S$	$< 0.8\ \mu S$
Inter-device Variability	σ_{Gmax}	$< 10\ \mu S$	$< 12\ \mu S$
	$\sigma^*_{\Delta G0}$	< 200%	< 95%
Faulty devices	Dead C.R.	< 7%	< 7%
	Stuck On C.R.	< 35%	< 10%
Dynamic range	Number of levels	> 13	> 110
Retention	Time before data loss	Higher	Lower
Endurance	Number of Set/Reset	Lower	Higher



Perspective on Training Fully Connected Networks with Resistive Memories: Device Requirements for Multiple Conductances of Varying Significance

Giorgio Cristiano,[1,2] Massimo Giordano,[1,2] Stefano Ambrogio,[1] Louis P. Romero,[1] Christina Cheng,[1] Pritish Narayanan,[1] Hsinyu Tsai,[1] Robert M. Shelby,[1] and Geoffrey W. Burr[1, a)]

[1)] *IBM Research AI, IBM Research–Almaden, 650 Harry Road, San Jose, CA USA 95120*

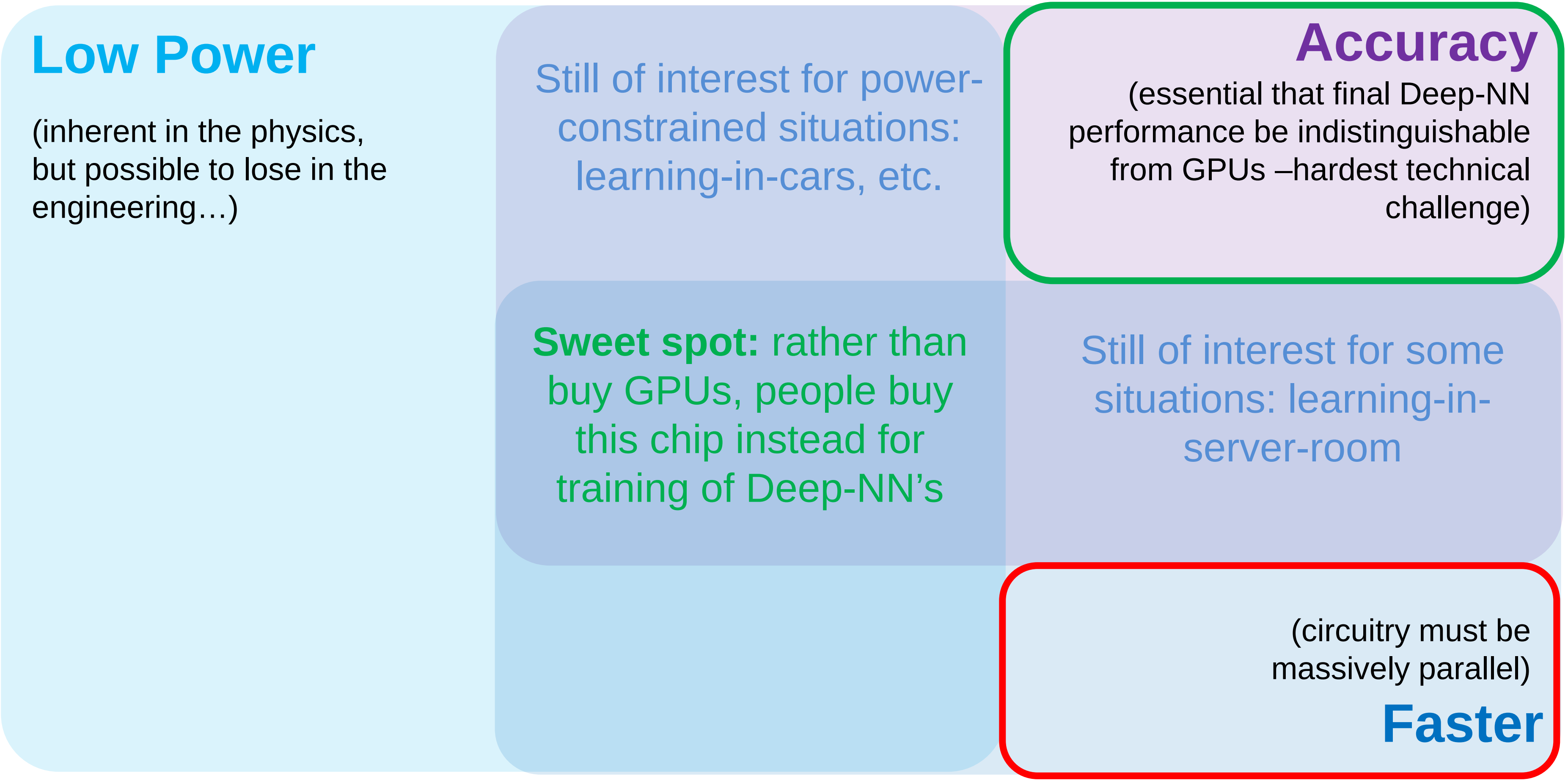
[2)] *EPFL, Route Cantonale, 1015 Lausanne, Switzerland*

G. Cristiano et al, J. Appl. Phys. 124 (15), 151901 (2018)

Outline

- Introduction
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Long-term: maximizing the future business case (vs. a GPU)



Low Power

(inherent in the physics, but possible to lose in the engineering…)

Still of interest for power-constrained situations: learning-in-cars, etc.

Accuracy

(essential that final Deep-NN performance be indistinguishable from GPUs –hardest technical challenge)

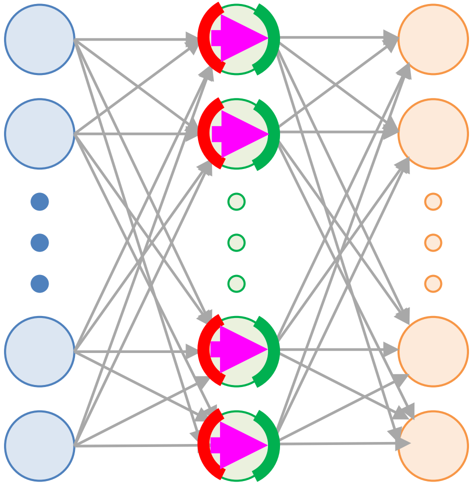
Sweet spot: rather than buy GPUs, people buy this chip instead for training of Deep-NN's

Still of interest for some situations: learning-in-server-room

(circuitry must be massively parallel)

Faster

Suggestions from circuit design work



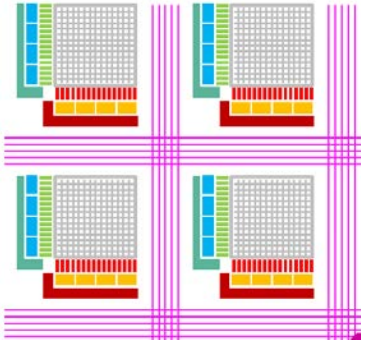
1) Parallelism is key

2) Avoiding ADC (Analog-to-Digital Conversion) saves time, power and area

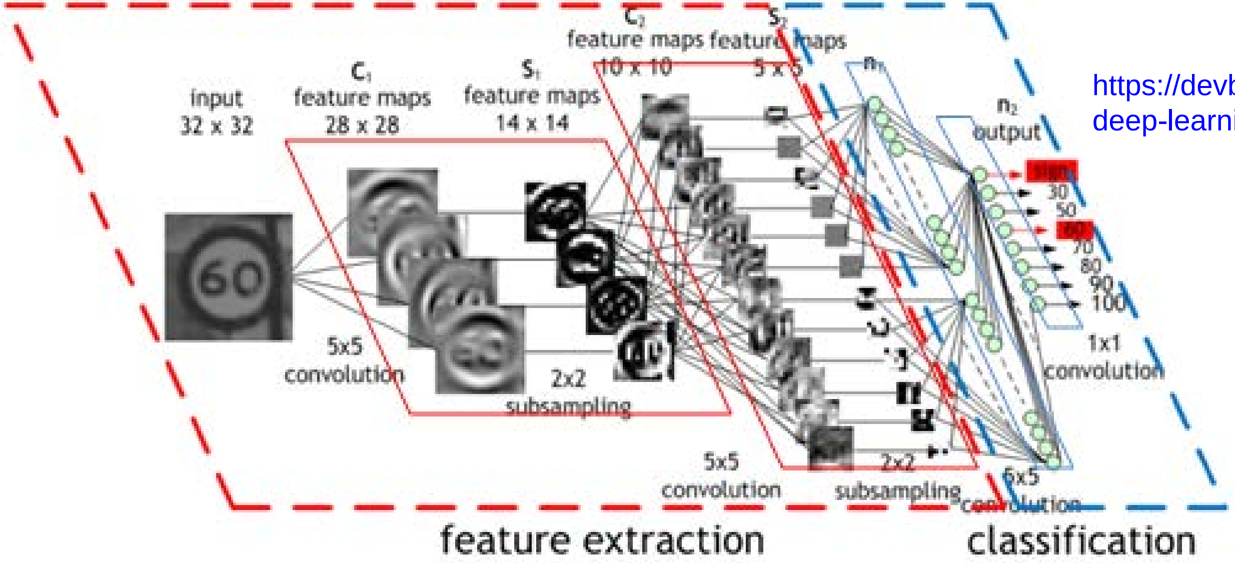
3) Do the necessary computations (squashing functions) but be as “approximate” as you can (get away with)

4) Need to get vectors of data from the bottom of one array to the edge of the next one

5) Digital accelerators are at their best w/ **convolutional** layers; Analog-memory accelerators are at their best w/ **fully-connected** layers.



Impact on Convolutional Neural Networks



https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/

- Only the last layers in a Convolutional Neural Network are Fully Connected due to memory constraints
- Hardware accelerators could easily implement FC layers, what could be the impact on CNN topology and performance?

Outline

- Introduction
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Conclusion

- **AI is introducing novel tools to develop solutions to everyday challenges**
  - Brain Inspired approach
  - Deep Learning approach

- **NVM-based crossbar arrays can accelerate the training of Deep Machine Learning compared to GPU-based training**
  - Multiply-accumulate performed at the data
  - Possible 500x speedup and orders-of-magnitude lower power

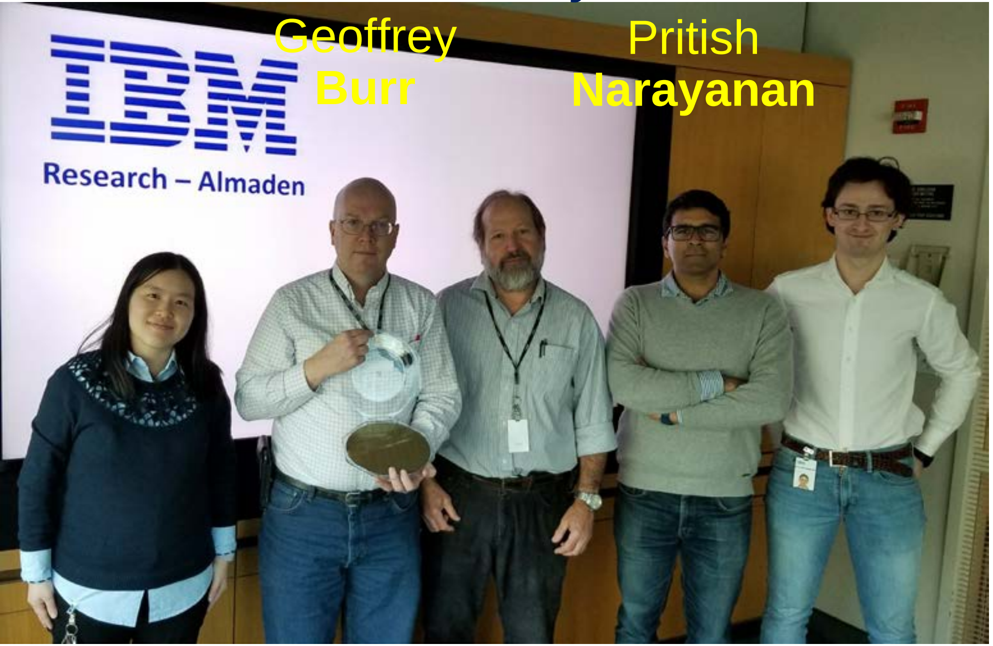
- **Experimental results on a 2T2R+3T1C unit cell demonstrate software-equivalent training accuracy**
  - MNIST, MNIST-backrand, CIFAR-10 and CIFAR-100 tested

- **Need area-efficient peripheral circuitry**
  - Tradeoffs balancing simplicity and area-efficiency against impact on ANN performance

stefano.ambrogio@ibm.com

Photos of us with “our first wafer of PCM-based circuit designs”

Sidney **Tsai** Bob **Shelby** Stefano **Ambrogio**



Geoffrey **Burr** Pritish **Narayanan**

Kohji **Hosokawa**

G. **Burr** P. **Narayanan**

Not shown: Scott C. **Lewis (YKT)**

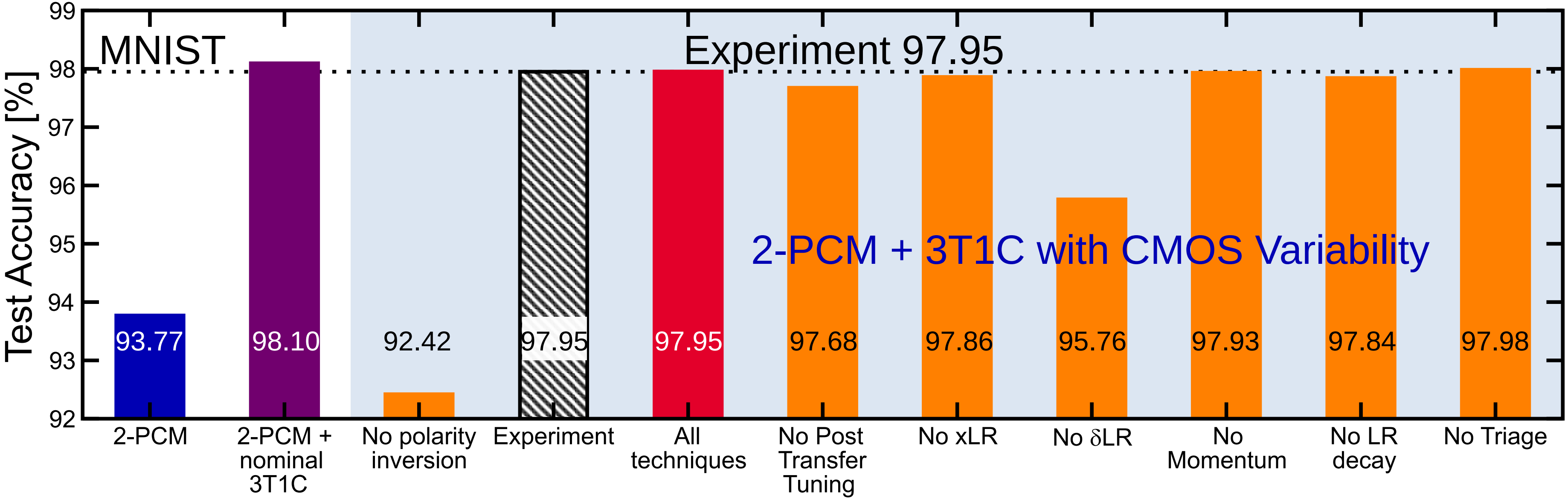
Thank you!

stefano.ambrogio@ibm.com

What do we mean by “mixed-hardware-software experiment”?

	Full software simulation	**Mixed-hardware-software experiment**	**Full hardware experiment**
NVM devices	Make a few NVM & measure, then capture in a statistical model → **not very accurate!**	**On-chip memory array** (the real yield, variability, non-ergodic statistics, etc.)	**On-chip memory array**
CMOS Periphery, Neurons, etc.	**Modeled in software (SPICE) → accurate!**	**Modeled in software (SPICE) → accurate!**	**Real CMOS implementation**

Impact of different techniques



- Polarity inversion shows the largest impact on accuracy
- Other techniques show varying importance depending on the training dataset (MNIST, MNIST backrand, CIFAR-10/100)