ELight: Enabling Efficient Photonic In-Memory Neurocomputing with Life Enhancement

Hanqing Zhu, Jiaqi Gu, Chenghao Feng, Mingjie Liu, Zixuan Jiang, Ray T. Chen, David Z. Pan

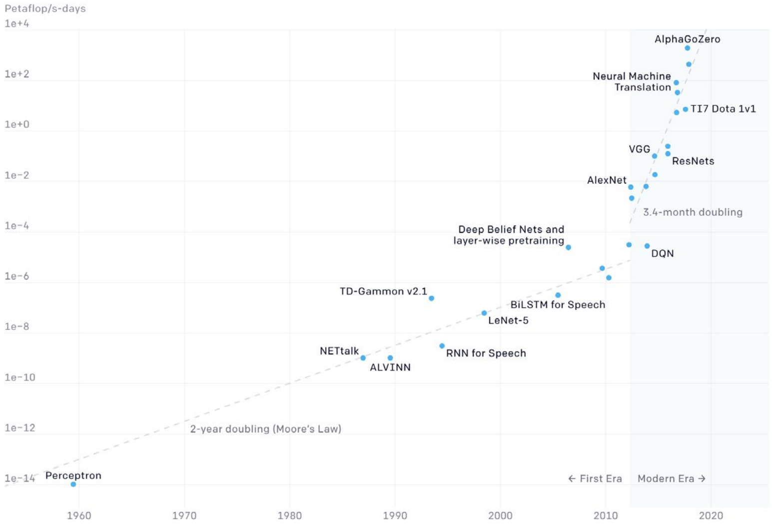
Dept. of Electrical and Computer Engineering
The University of Texas at Austin

This work is supported in part by MURI and ONR

AI Acceleration and Challenges

- ML models and dataset keep increasing
  - Low latency
  - Low power
  - High bandwidth

- AI compute requirement has **5**× the doubling rate of Moore's law

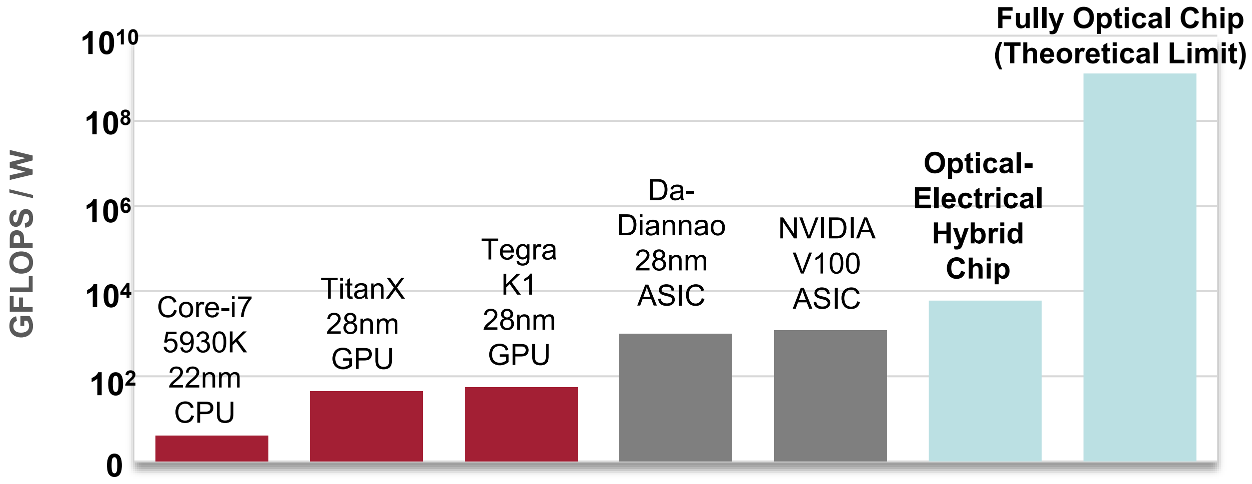


https://openai.com/blog/ai-and-compute/

Photonic AI

- Use light to continue Moore's law
- Promising technology for next-generation AI accelerator

Ultra-high speed & Ultra-low energy



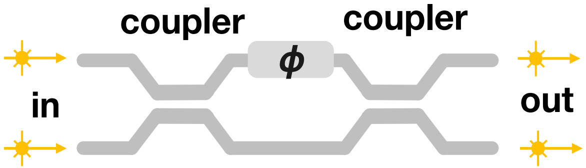
[Shen+, *Nature Photonics* 2017]

LIGHTMATTER LIGHTELLIGENCE OPTELLIGENCE LightOn CogniFiber XANADU LUMINOUS OPTIUS light powered computing

Optical Computing Basics

Computing Basics

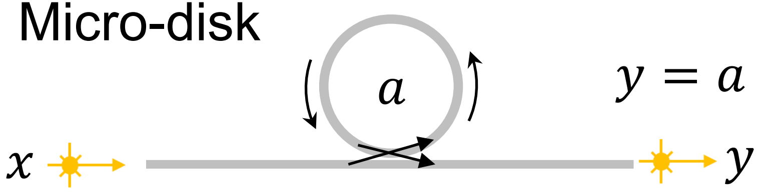
MZI



$$\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$$

2×2 Unitary Matmul (~100×20 um^2)

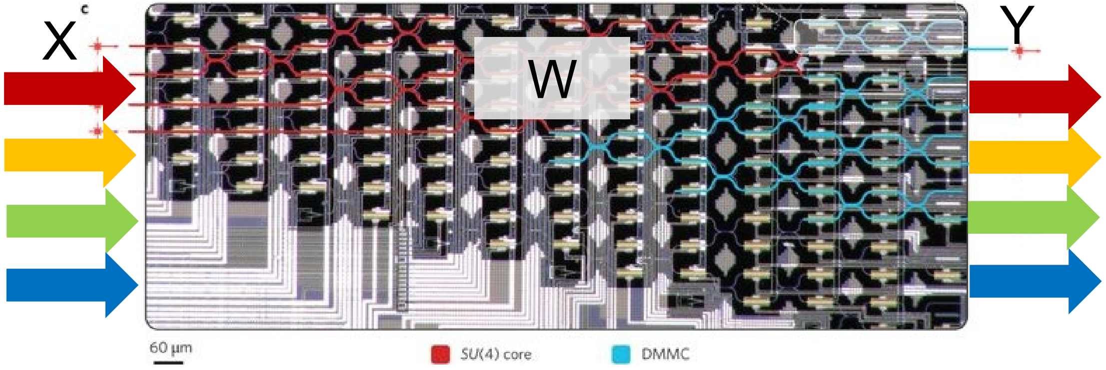
Micro-ring/
Micro-disk



$y = a \cdot x$

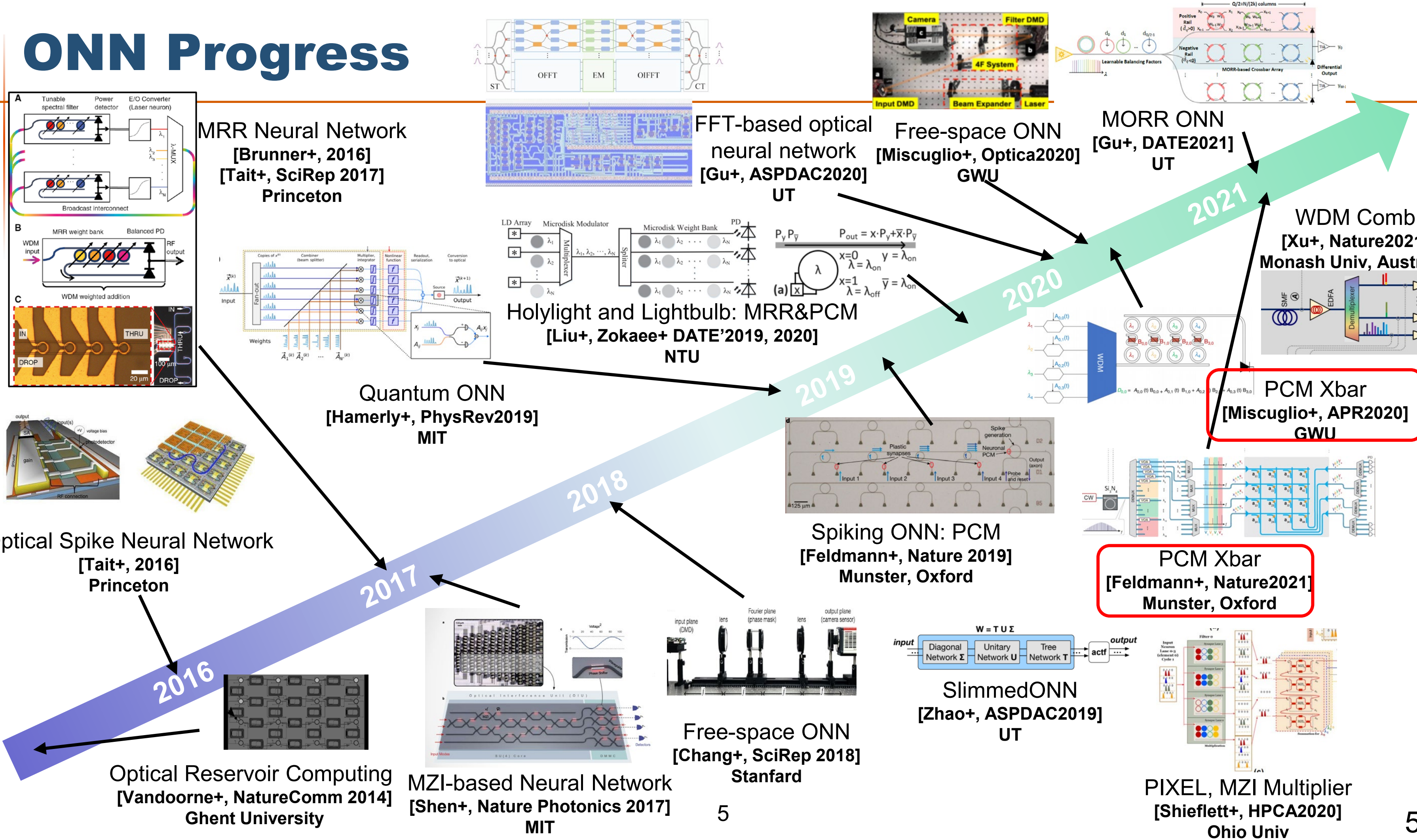
Scalar Mul. (~10×10 um^2)

Optical Neural Network



Photonic tensor unit for analog GEMM [MIT's Nat. Photonics'17]

ONN Progress

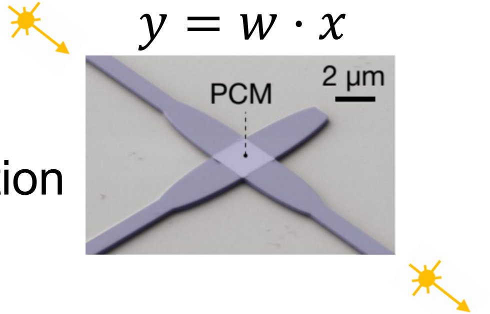


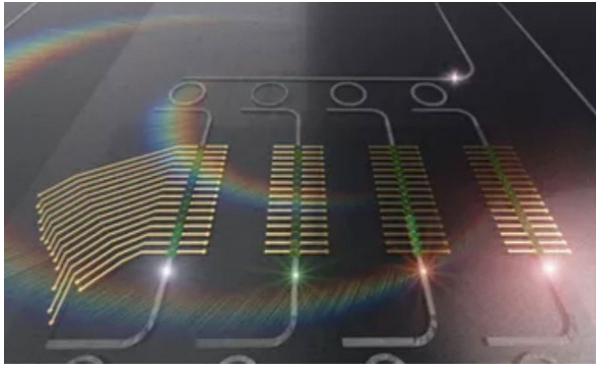
5

Computing with Photonic Phase Change Material

- Recall photonic devices: MZI, Microring, micro-disks, …
- A new device for non-volatile computing
  - Phase change material (PCM)
  - Modulate the light transmission to achieve multiplication
  - Store the transmission as a non-volatile memory

$$y = w \cdot x$$



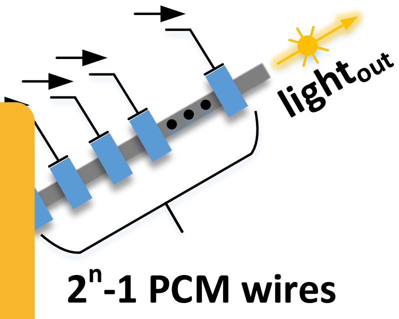


[Miscuglio+, APR'20]

[Feldmann+, Nature'21]

Barriers Towards Practical Deployment

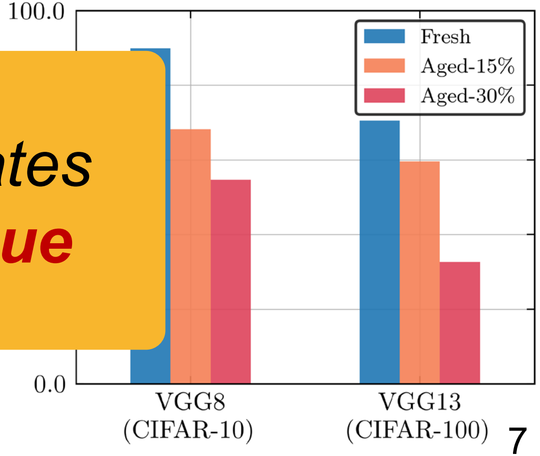
- Limited PCM rewrite!
  - Max: 1
  - Aging



bit PCM memory cell
[Miscuglio +, *APR*'20]

PCM has ***limited rewrite times***
Short endurance

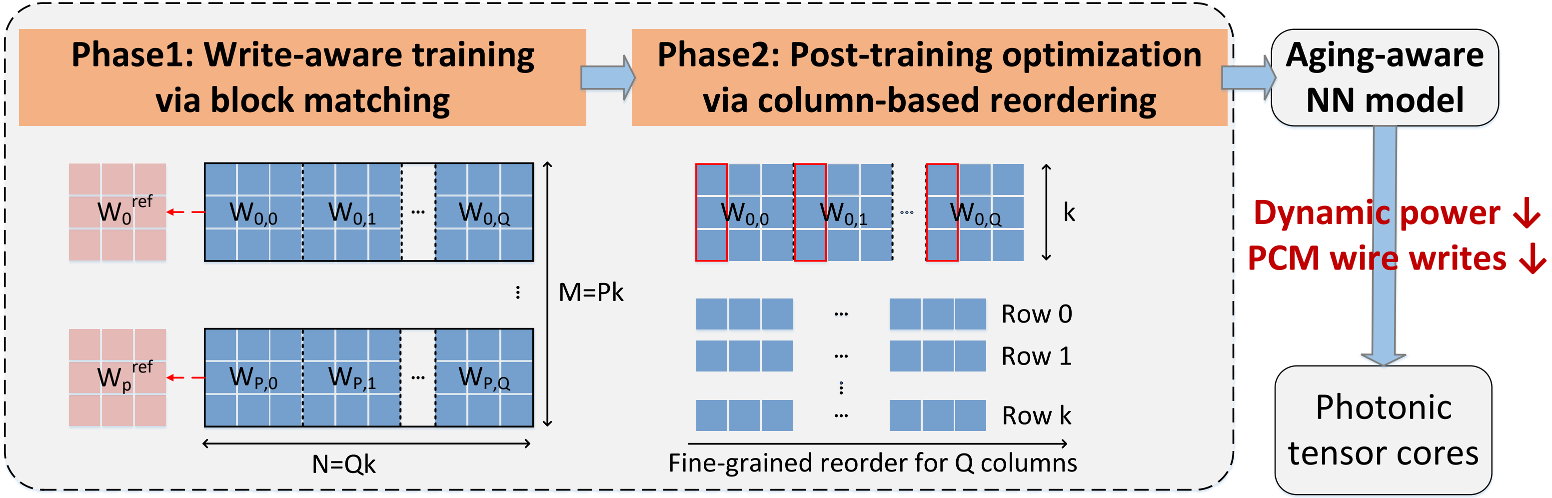
- Why we care inference endurance?
  - Limited
  - Limited
  - Need f



Frequent reuse *further escalates*
Dynamic power & Aging issue

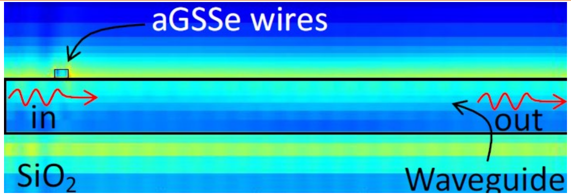
Our Proposed Aging-aware PCM-ONN: ELight

- A two-phase aging-aware optimization framework
  - **Minimize** PCM **write** operations in **inference**
- Achieved
  - $> 20\times$ fewer write operations
  - **Minimized** # max writes on a single PCM cell
  - $> 30\times$ less dynamic energy cost

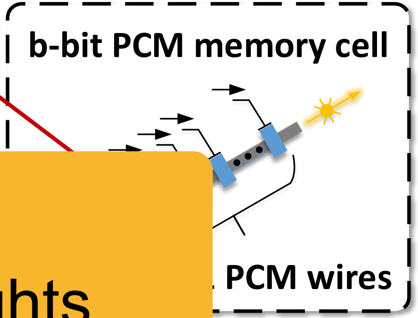


Distribution-aware Quantization

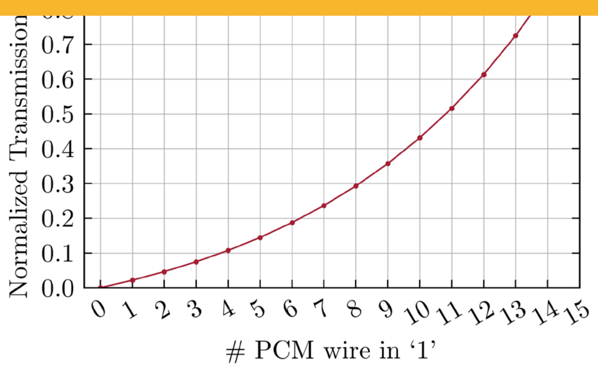
- 1 PCM wire: $y = cx$
- b-bit PCM cell: $y = c^b x$
- Transmission factor (t) follows power-of-c model

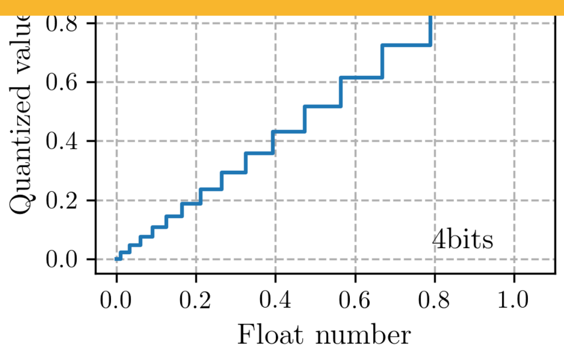


Amorphous state (0)
Crystalline state (1)



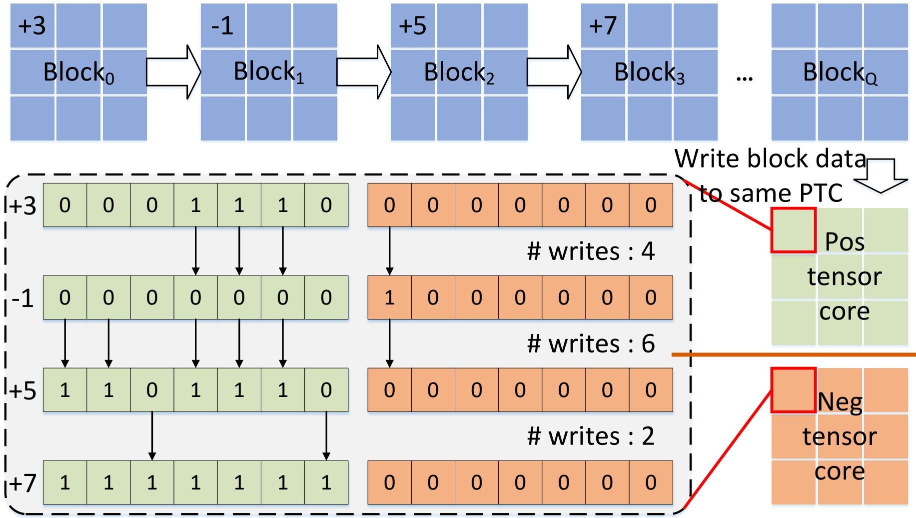
Be Aware of Distribution !
→ ***More*** quantization levels in ***small*** weights
→ ***boost expressivity***

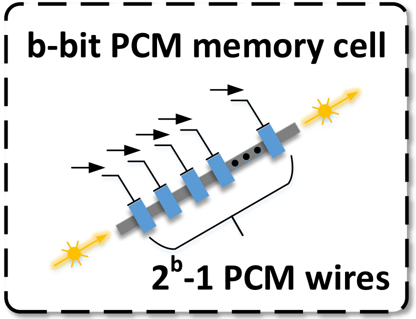




Augmented Redundant Write Elimination (ARWE)

- Block matrix multiplication
- Assume each PTC is assigned with one row of weight sub-blocks
- ARWE: Preserve current states at the most
  - Redundant write elimination scheme [Yang+, ISCAS]
  - Easy to compare values as weights are known and pre-stored





$$WT(w', w) = |l^+(w') - l^+(w)| + |l^-(w') - l^-(w)|$$

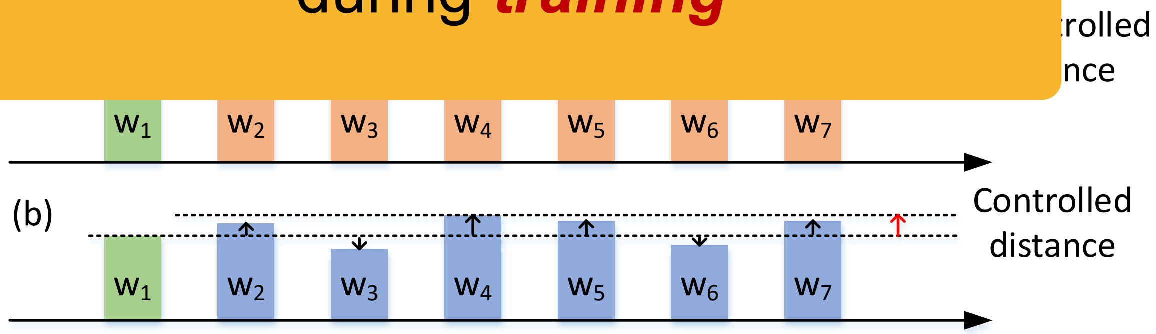
Write-aware Training: Weight Matching

- Significant rewrite operations still exists with ARWE!
  - Deployment of a 5-bit VGG8 model trained on CIFAR10

	Layer 2	Layer 3	Layer 4	Layer 5
# total writes	1.14×10^6	4.87×10^6	1.66×10^7	3.26×10^7
# max writes	294	534	914	1425

- Pull w

Orchestrate ***weight similarity*** during ***training***

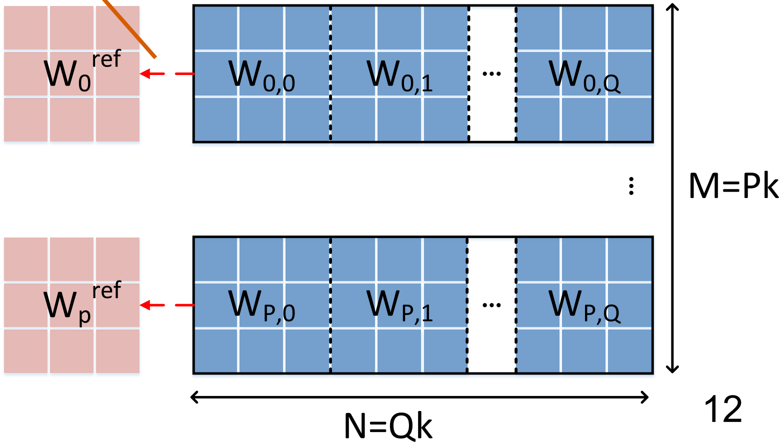


Write-aware Training: Block Matching

- Average a group weight blocks into one reference block
- Compute level difference between two blocks with L2 norm
  - Penalize large value deviation but Allow slight value deviation
  - Preserve model expressivity

$$LD(W^{ref}, W) = \sum_{i}^{k}\sum_{j}^{k} \|\tilde{l}^{+}(w_{ij}^{ref}) - \tilde{l}^{+}(w_{ij})\|^2 + \|\tilde{l}^{-}(w_{ij}^{ref}) - \tilde{l}^{-}(w_{ij})\|^2.$$

- Match weight blocks with the reference block



Write-aware Training: Optimization issue

- $\mathcal{L}_{BM}$ is not differentiable
  - Recall the function to get transmission levels

$$l^{+}(w)=\begin{cases}(2^b-1)-\mathtt{Clip}(\mathtt{R}(\log_t(s|w|+\delta)),0,2^b-1), & w\geq 0\\ 0, & w<0\end{cases}$$

$$l^{-}(w)=\begin{cases}0, & w\geq 0\\ \mathtt{Clip}(\mathtt{R}(\log_t(s|w|+\delta)),0,2^b-1)-(2^b-1), & w<0\end{cases}.$$

STE

- Not all gradients need to be propagated back
  - Only gradients from physically stored value are valid

Mask out invalid gradients

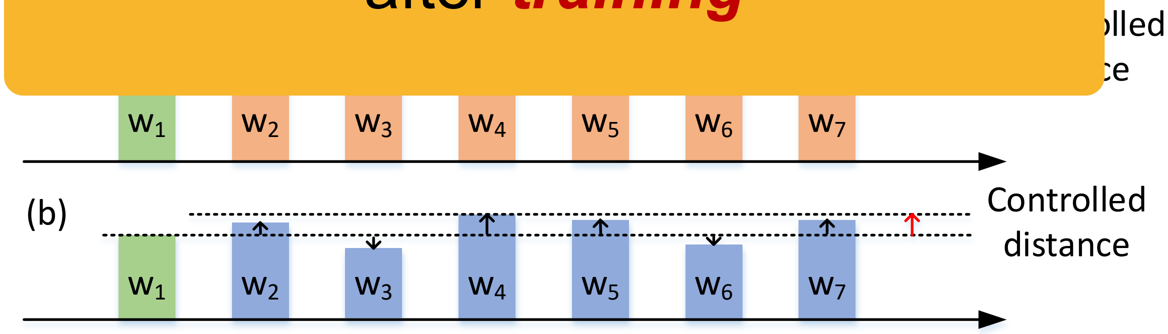
$$\mathcal{L}_{BM}=\sum_{l}^{L}\sum_{t}^{G}\sum_{i}^{n_g}\frac{1}{\beta B}LD(B_i^t,B_{avr}^t)$$

$$LD(B,A)=\sum_{i}^{k}\sum_{j}^{k}\|\tilde{l}^{+}(b_{ij})-\tilde{l}^{+}(a_{ij})\|^2+\|\tilde{l}^{-}(b_{ij})-\tilde{l}^{-}(a_{ij})\|^2$$

Post-training Optimization: Need of Reordering

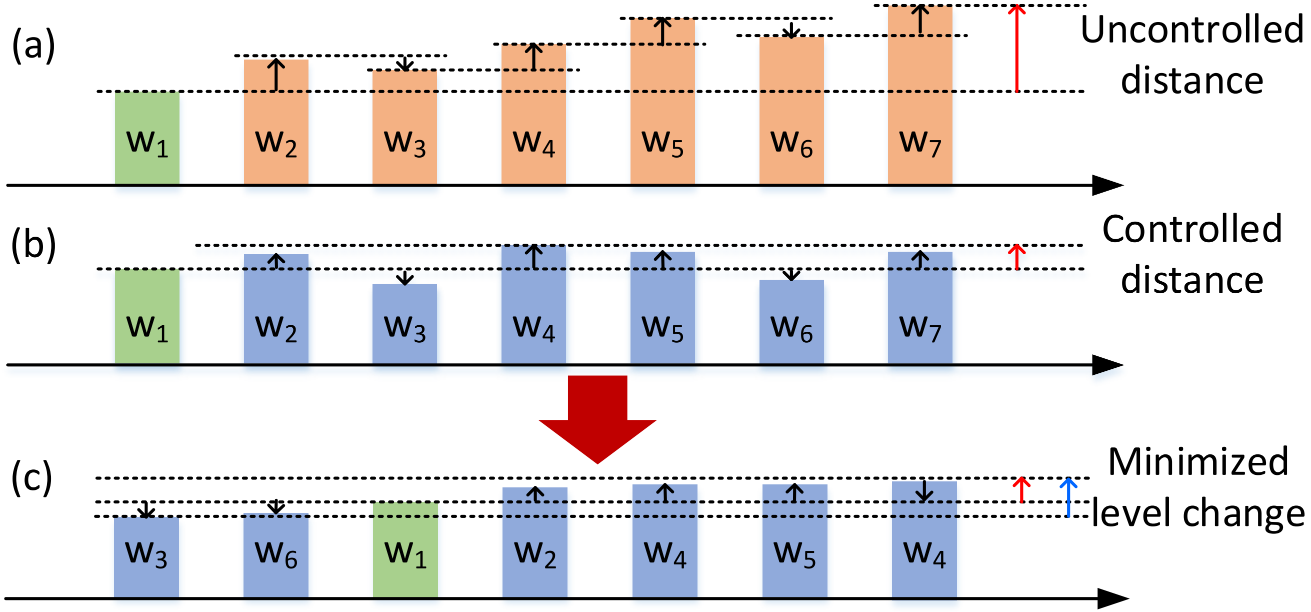
- We only pull weights close to a reference
- No order information is introduced during our optimization!
  - Cannot guarantee best-of-reduction
- Find an o

Consider ***order information*** after ***training***



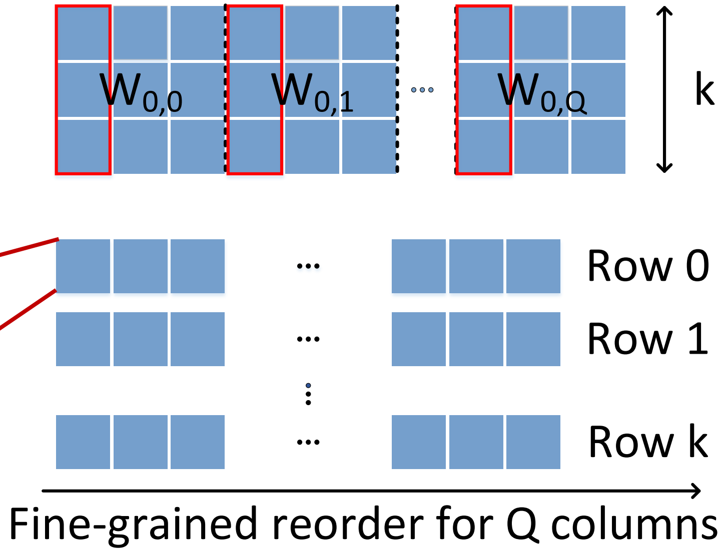
Post-training Optimization: One-shot Reordering

- Simply reorder the weight sequences
- Efficient reordering with negligible overhead

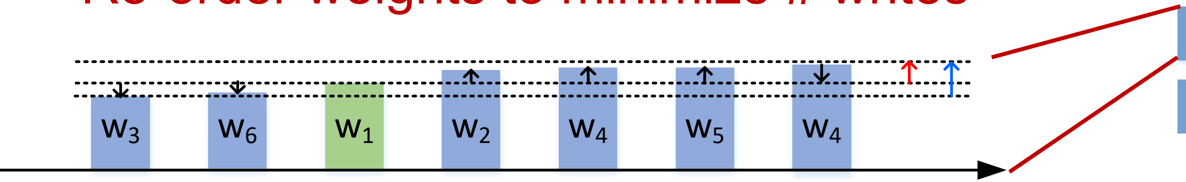


Post-training Optimization: One-shot Reordering

- Take one more step before real deployment
- One-shot recording concurrently for different columns
- No affect on the computation results
  - General matrix multiplication

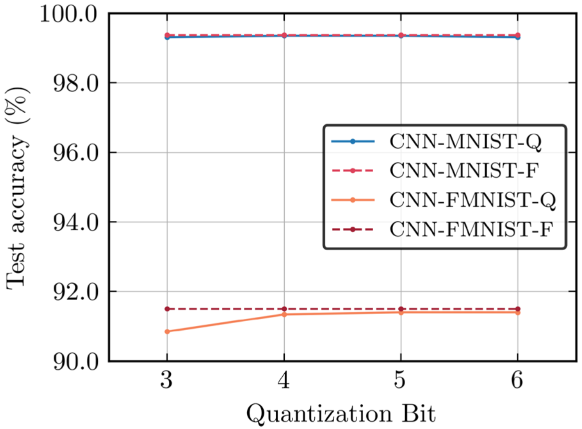


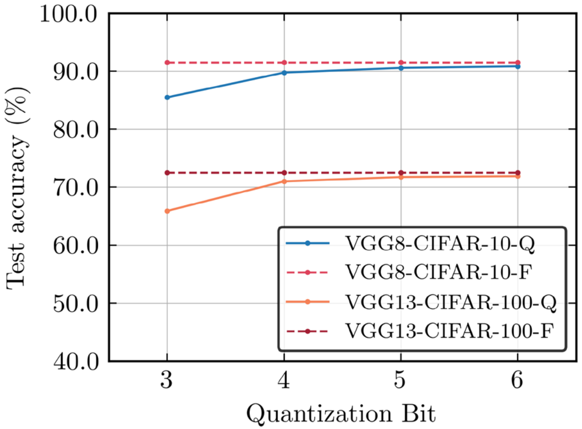
Re-order weights to minimize # writes



Experimental Results: Quantization

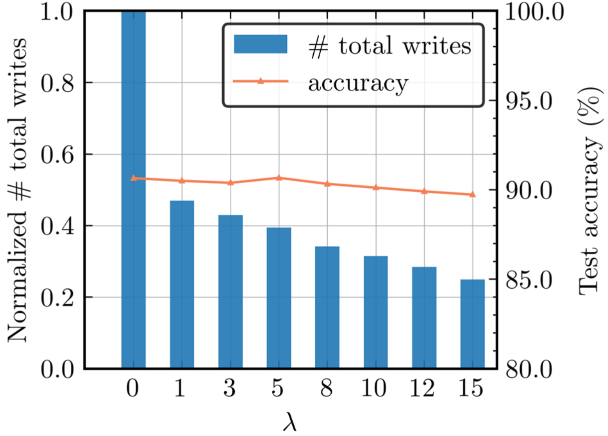
- Experiments settings
  - Photonic PCM memory: $3 - 6$-bit
  - Photonic tensor core: 16×16 and 64×64
  - Models: Simple CNN, VGG8 and VGG13
  - Dataset: MNIST, FashionMNIST, CIFAR10 and CIFAR100
- Distribution-aware quantization
  - Small accuracy loss with > 4-bit

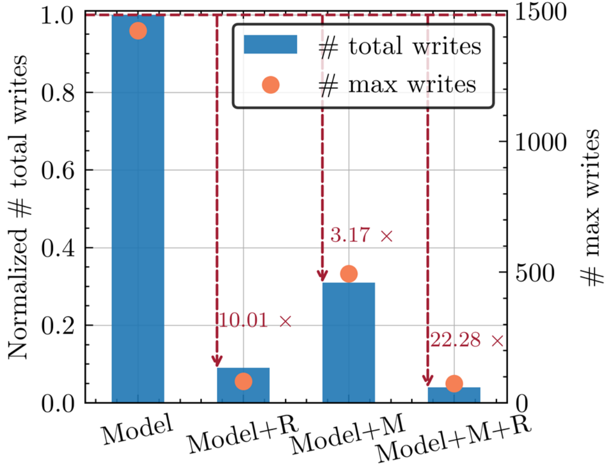




Experimental Results: # total Write

- 5-bit VGG8 on CIFAR10
- Write-aware training (M)
  - Trade-off between accuracy and write elimination
  - $< 1\%$ accuracy drop with sweet parameters
- Post-training reordering (R)
  - Further cut down redundant writes
  - Achieve $22.3\times$ reduction on #writes with M





Experimental Results: Endurance & Energy

- ♦ $> 20\times$ fewer write operations
- ♦ Minimized # max writes on a single PCM cell
- ♦ $> 30\times$ less dynamic energy cost

ONN Lifetime ↑
Dynamic energy ↓

Network	Dataset	Bitwidth	λ	Acc(%)/AC	*# total writes* ↓ (×)		Energy cost ↓ (×)		*# max writes*	
					-	+R	-	+R	-	+R
VGG8	CIFAR-10	3	0	86.71	1	6.52	1	9.27	128	15
			8	86.02/-0.69	22.12	**46.11**	6.63	**69.29**	14	**7**
		4	0	89.75	1	7.84	1	11.31	401	36
			10	89.94/+0.19	3.83	**24.45**	3.92	**35.48**	95	**19**
		5	0	90.56	1	10.01	1	14.35	1425	82
			10	90.12/-0.44	3.17	**22.28**	3.20	**31.17**	494	**74**
		6	0	90.83	1	12.31	1	16.89	4464	180
			5	89.88/-0.95	6.82	**26.35**	7.15	**32.48**	1560	**146**
VGG13	CIFAR-100	4	0	70.99	1	9.66	1	13.84	542	39
			10	70.44/-0.55	3.54	**29.25**	3.57	**42.02**	173	**33**
		5	0	71.73	1	12.06	1	17.29	1771	84
			3	71.95/+0.22	2.19	**21.93**	2.21	**31.41**	921	**55**
		6	0	71.88	1	14.37	1	17.62	4926	182
			3	70.97/-0.91	3.11	**22.65**	3.19	**29.85**	3577	**156**

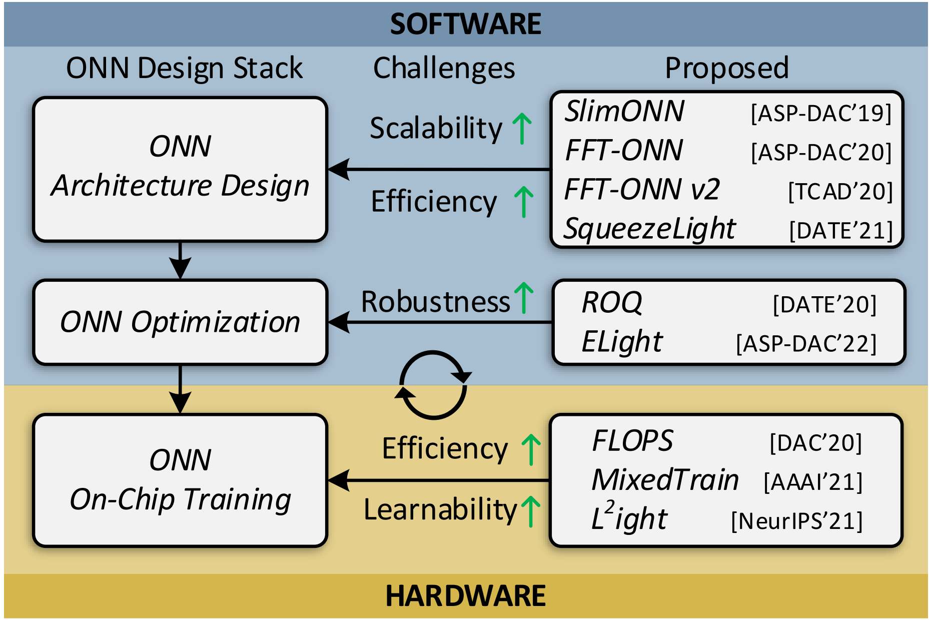
Conclusion and Future work

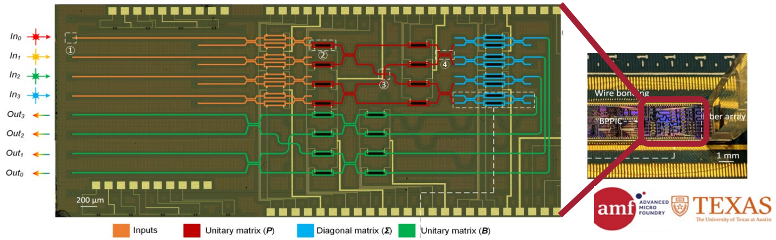
- The ***first*** aging-aware optimization framework for Photonic in-memory computing
- Expressivity: Distribution-aware quantization
- Lifetime enhancement: $> 20 \times$ fewer write operations
- Energy efficiency: $> 30 \times$ less dynamic energy cost
- Push forward the real deployment of Photonic in-memory computing

- Future direction
  - Preserve the accuracy of NN model with aged PCM cells
  - Counter other non-ideal factors such as device-to-device variations
  - Investigate the effect of temporal drift

Our Recent Work & Open-source Framework

- Build ultra-fast (light-speed), ultra-energy efficient, and highly robust optical neural accelerators with photonic integrated circuits





Circuit-Architecture-Algorithm Co-Design!

PyTorch-ONN Library

Thanks!

Q & A?