Hardware-Software Co-design of Slimmed Optical Neural Networks

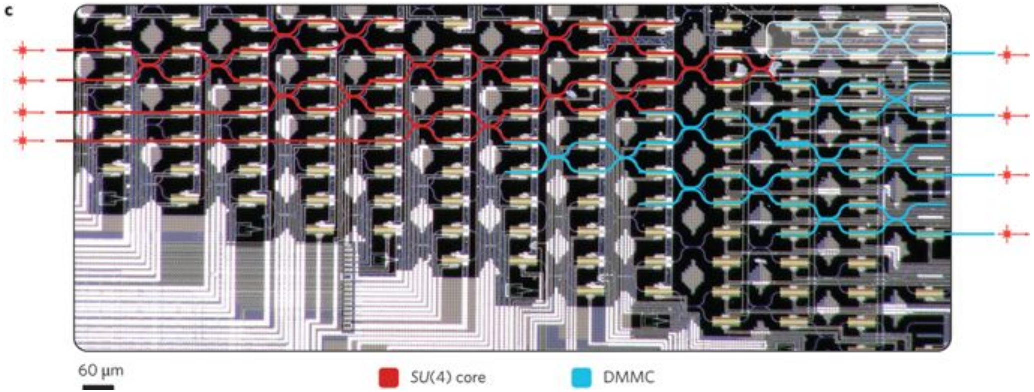
Zheng Zhao[1], Derong Liu[1], Meng Li[1], Zhoufeng Ying[1], Lu Zhang[2], Biying Xu[1], Bei Yu[2], Ray Chen[1], David Pan[1]

The University of Texas at Austin[1]

The Chinese University of Hong Kong[2]

Introduction

- Emergence of dedicated AI accelerators
  - Optical neural network processor: light in and light out
    - Speed-of-light floating point matrix-vector multiplication
    - >100GHz detection rate
    - Ultra-low energy consumption if configured
  - Great number of components, sensitivity to noise

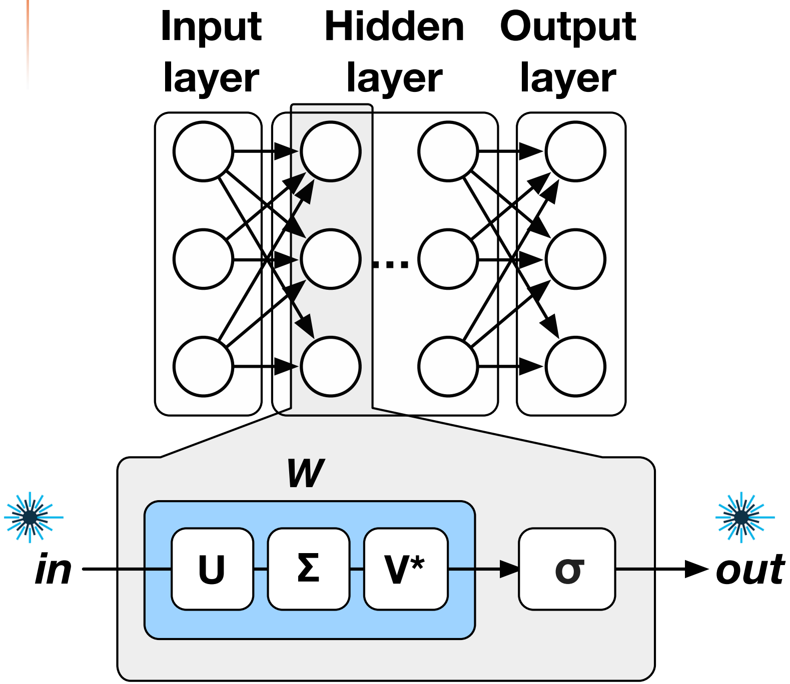


[Shen+, *Nature Photonics* 2017]

LIGHTELLIGENCE

LIGHTMATTER

Previous Optical Neural Network (ONN)



[Shen+, *Nature Photonics* 2017]

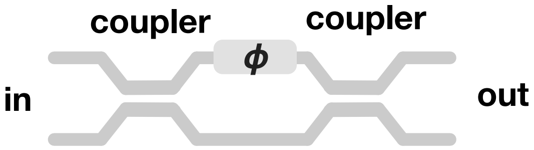
- SVD decompose **W** = **U Σ V***
- **U** and **V*** are unitary matrices

Most area expensive

- **Σ** is a diagonal matrix
  - Diagonal values are non-negative real
  - Implemented by optical attenuators
- **σ** is non-linear activation
  - Implemented by saturable absorber

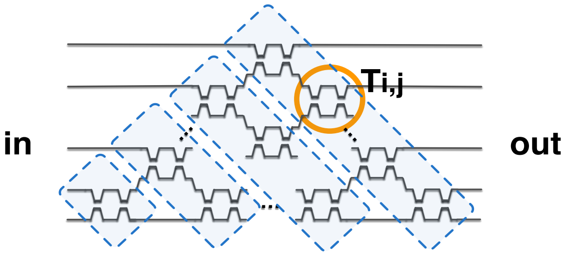
Implementing Unitary U and V*

- Mach-Zehnder interferometers (MZI) for **U** and **V***
  - A single MZI implements a 2-dim unitary



$$\boldsymbol{out} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \cdot \boldsymbol{in}$$

  - An array of n(n-1)/2 MZIs implements an n-dim unitary

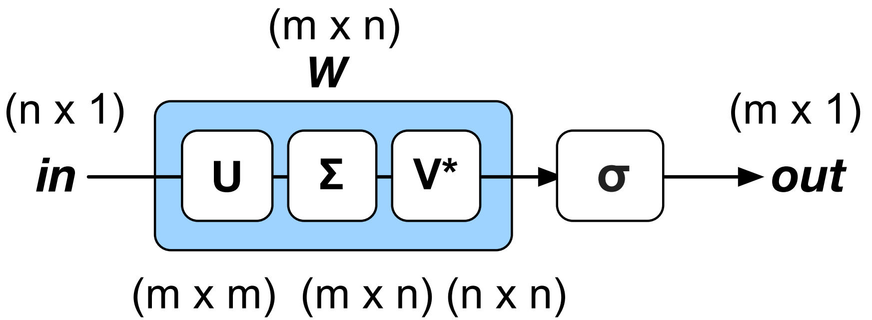


$$T_{i,j} = \begin{array}{cl} \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & \cos\phi & 0 & \sin\phi & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix} & \begin{array}{l} \\ i^{th}\text{ row} \\ \\ j^{th}\text{ row} \\ \\ \end{array} \\ \begin{array}{ccccc} \quad & i^{th}\text{ col.} & \quad & j^{th}\text{ col.} & \quad \end{array} & \end{array}$$

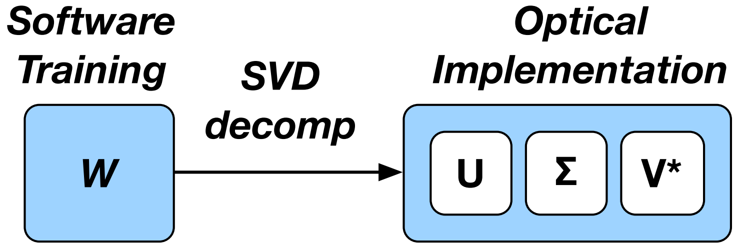
$$\boldsymbol{out} = \prod_{i=n}^{2} \prod_{j=1}^{i-1} T_{ij} \cdot \boldsymbol{in}$$

- Given an n-dim unitary, φ’s can be uniquely computed

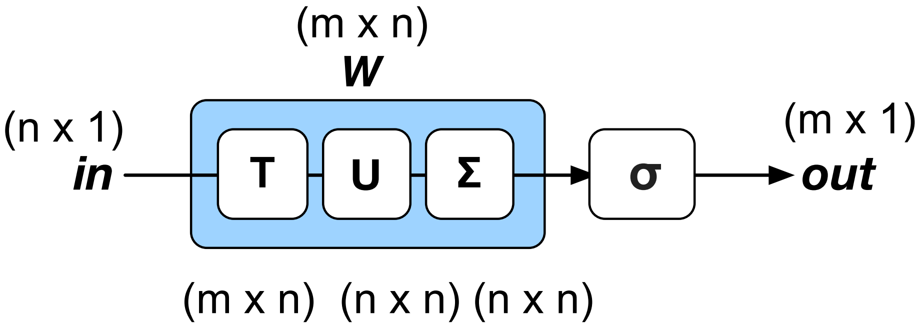
Previous ONN overview



- Layer size measured by # of MZIs = m(m-1)/2+n(n-1)/2
- Software training and hardware implementation
  - Train **W** directly in software → SVD-decomp to obtain **U**, **Σ**, **V***



Slimmed Architecture



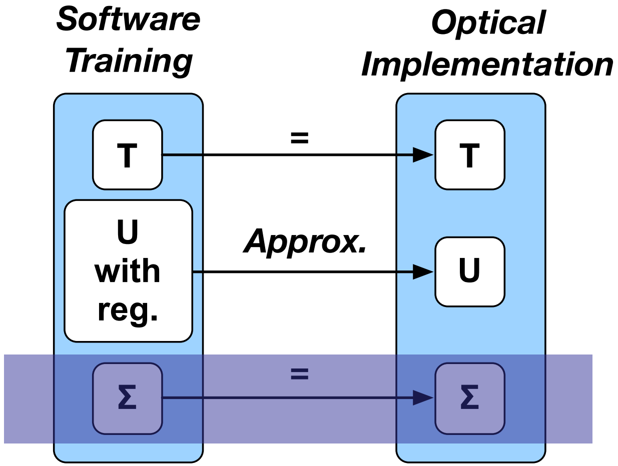
- **T**: sparse tree network
- **U**: unitary network
- **Σ**: diagonal network

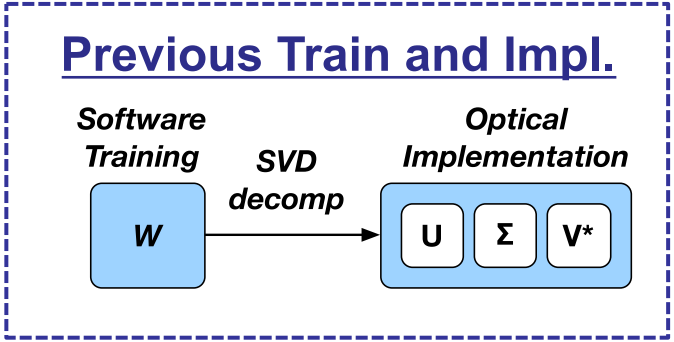
same constraints as the previous architecture

- Use less # of MZIs = n(n-1)/2
  - 1 unitary matrix to maintain the expressivity
  - An area-efficient tree network to match the dimension

Co-design Overview

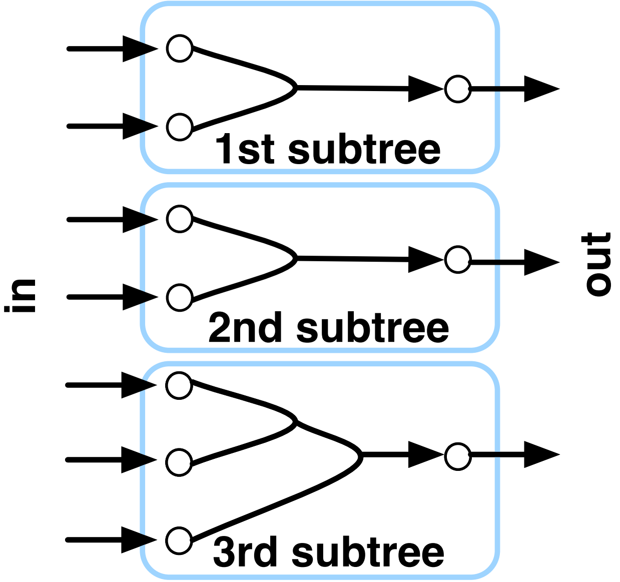
- An arbitrary weight **W** is **not TUΣ-decomposable**
- Co-design solution: training and implementation are coupled
  - **T** and **Σ**: Train the device parameters, constraints embedded
  - **U**: Add unitary regularization then approximate with true unitary

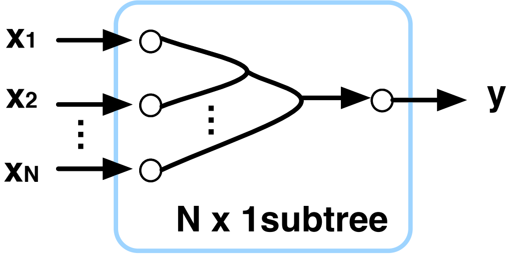




Sparse Tree Network

- Sparse Tree network (**T**) to match the different dimension
  - Suppose in-dim > out-dim
  - α: linear transfer coefficient



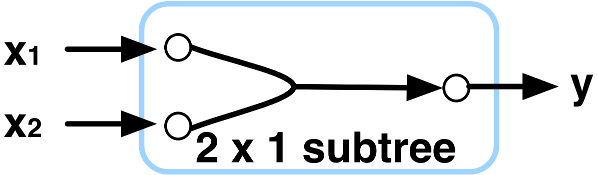


$$y = \sum_{i=1}^{N} \alpha_i \cdot x_i$$

$$out = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{2,1} & \alpha_{2,2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} \end{pmatrix} \cdot in.$$

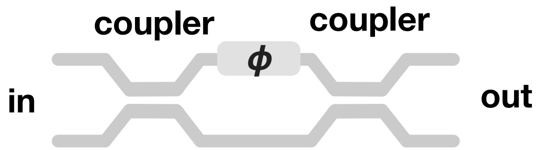
Sparse Tree Network Implementation

- Implemented with MZIs or directional couplers
- A 2 x 1 subtree



$$y = \alpha_1 \cdot x_1 + \alpha_2 \cdot x_2$$

can be Implemented with a single-out MZI or directional coupler



$$\boldsymbol{out} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \cdot \boldsymbol{in}$$

$$out_1 = \cos\phi \cdot in_1 + \sin\phi \cdot in_2$$

$\alpha_1 = \cos\phi$ and $\alpha_2 = \sin\phi$ $\qquad$ $\alpha_1^2 + \alpha_2^2 = 1$ (energy conservation)

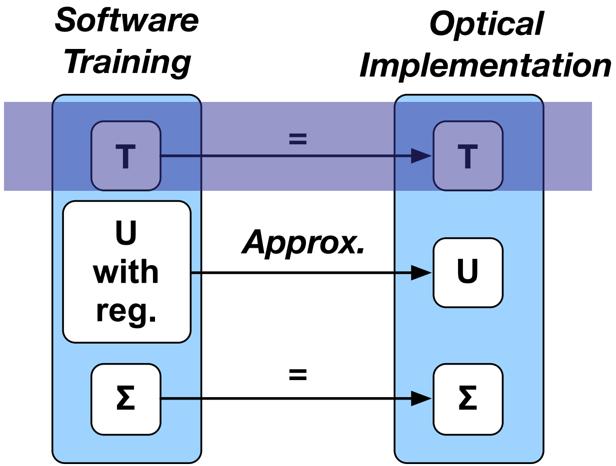
Sparse Tree Network Implementation

- Any N-input subtree with arbitrary α's satisfying energy conservation

$$\sum_{i=1}^{N} \alpha_i^2 = 1, -1 \leq \alpha_i \leq 1, i = 1, \cdots, N$$

can be implemented it by cascading (*N-1*) single-out MZIs.

- Energy conservation embedded in training



Unitary Network in Training

- For unitary network **U** satisfying **UU*** = **I**, add the regularization

$$\mathbf{reg} = \|\mathbf{UU^*} - \mathbf{I}\|_F$$

- Training loss function

Loss = Data Loss + Regularization Loss

leading to a **near-implementable** ONN with high accuracy

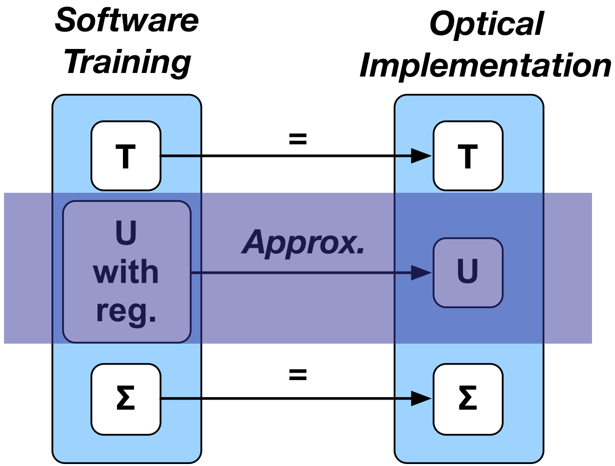
- Trained **Ut** ~ unitary but only true unitary is implementable by MZIs

Unitary Network in Implementation

- Approximate **Ut** by a true unitary **Ua**
- SVD-decompose **Ut** = **PSQ*** → **Ua** = **PQ***

- **Claim.** Minimize the regularization ⇔ find the best approximation

$$\textit{Min. } \mathbf{reg} \Leftrightarrow \textit{Min. } \| \mathbf{Ut} - \mathbf{Ua} \|_F$$



Simulation Results

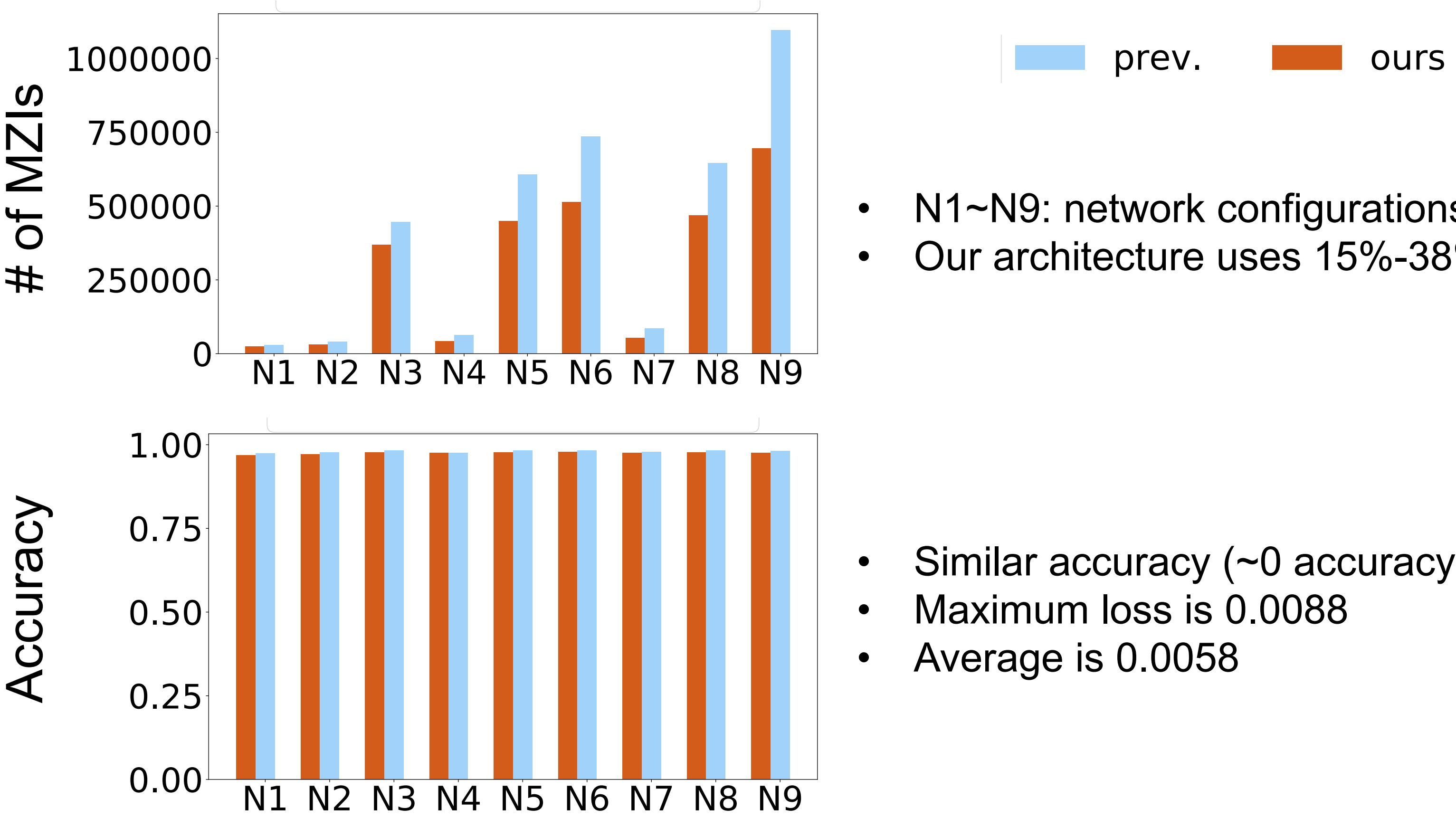
- Implemented in TensorFlow for various ONN setup

N1: (14×14)-100-10	**N4:** (14×14)-150-150-10	**N7:** (14×14)-150-150-150-10
N2: (14×14)-150-10	**N5:** (28×28)-400-400-10	**N8:** (28×28)-400-400-200-10
N3: (28×28)-400-10	**N6:** (28×28)-600-300-10	**N9:** (28×28)-600-600-300-10

- Tested it on Intel Core i9-7900X CPU and an NVIDIA TitanXp GPU

- Performed on the handwritten digit dataset MNIST

Simulation Results

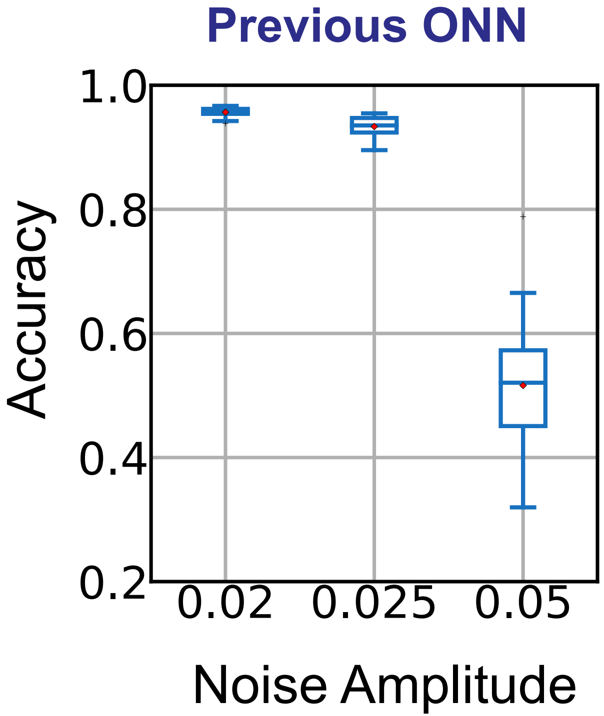


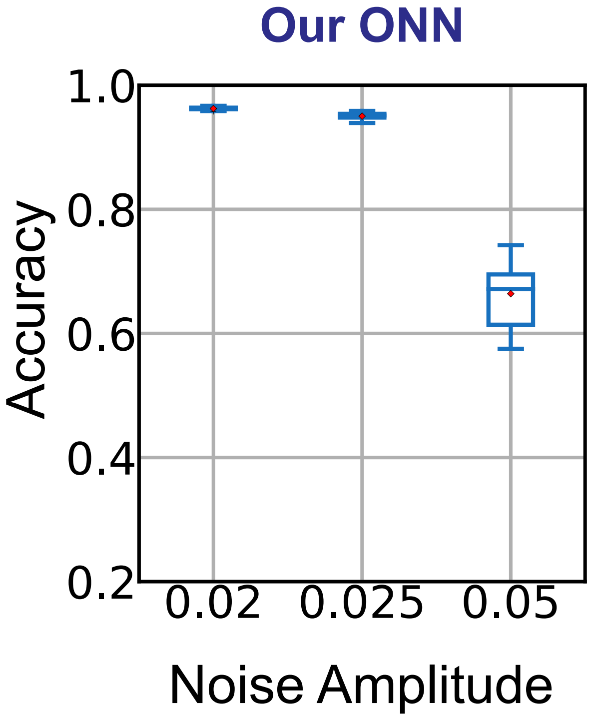
- N1~N9: network configurations
- Our architecture uses 15%-38% less MZIs

- Similar accuracy (~0 accuracy loss)
- Maximum loss is 0.0088
- Average is 0.0058

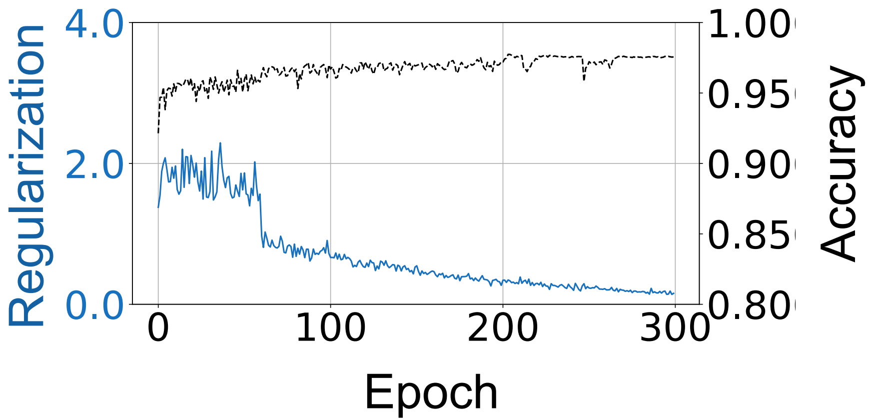
Noise Robustness

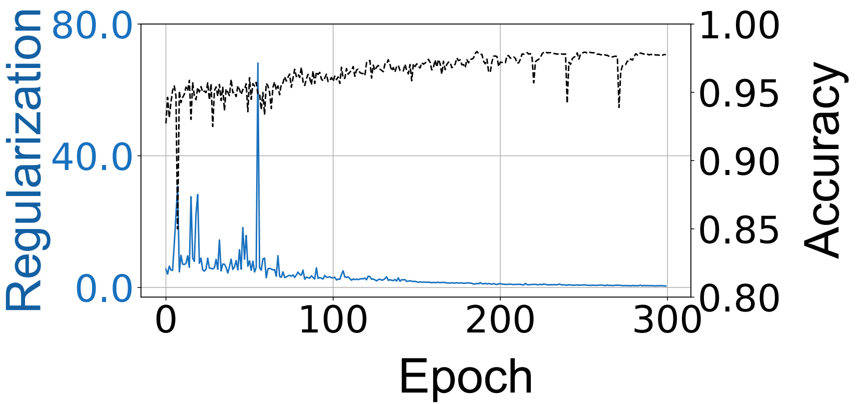
- Better resilience due to less cascaded components

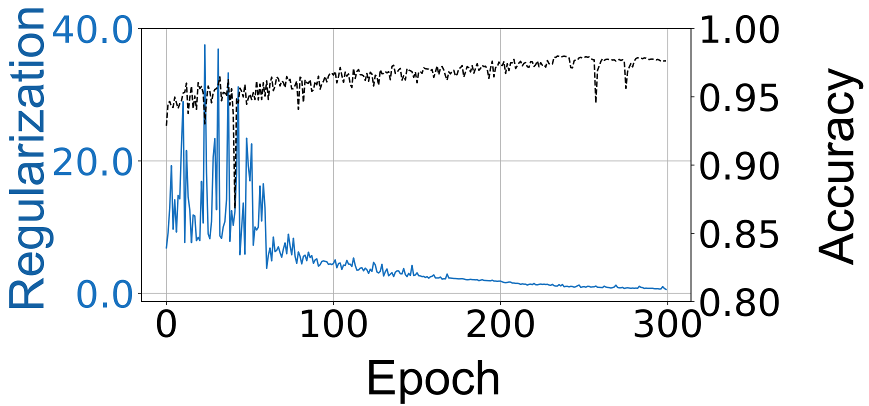




Training Curve







- Converged in 300 epochs
- Balance of the accuracy and the unitary approximation

Contributions of This Work

- An new architecture for ONN
  - Area-efficiency
  - ~0 accuracy loss
  - Better robustness to noise
- Hardware and software co-design methodology
  - Software-embedded hardware parameters
  - Hardware constraints guaranteed by software

Future Work

- Better MZI pruning methods
  - ~0 phase MZI → pruned + accuracy recover
  - MZI-sparse unitary matrix
- Design for robustness
  - Adjust noise distribution in training
- Online training
- ONN for other neural network architectures
  - CNN, RNN, etc.

Thanks
Q&A