WHERE **INNOVATION** BEGINS

DESIGN AUTOMATION CONFERENCE

JULY 9–13, 2023

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Tutorial: A Journey to Optical Computing: From Physics Fundamentals to Hardware-Software Co-Design, Automation, and Application

Organizers: Jiaqi Gu[1,2], Ulf Schlichtmann[3], Zhengqi Gao[4], Cunxi Yu[5]

Presenters: Jiaqi Gu[1,2], Chenghao Feng[1,6], Ulf Schlichtmann[3], Zhengqi Gao[4], Cunxi Yu[5]

[1]The University of Texas at Austin, [2]Arizona State University, [3]Technical University of Munich, [4]Massachusetts Institute of Technology, [5]University of Utah, [6]Alpine Optoelectronics

Outline of Tutorial

- Tutorial I: Fundamentals of Optical Computing and Integrated Photonics for High-Performance Digital Logic and Efficient Machine Learning
  - Jiaqi Gu, Chenghao Feng (UT Austin, Arizona State University)
- Tutorial II: LightRidge: An End-to-end Agile Design Framework for Diffractive Optical Neural Networks
  - Yingjie Li, Cunxi Yu (University of Utah)
- Tutorial III: Topology and Physical Layout Optimization of Photonic Networks-on-Chip and PIC Variation Analysis
  - Ulf Schlichtmann (Technical University of Munich)
- Tutorial IV: Integrated Programmable Photonic Circuits
  - Zhengqi Gao (MIT)

Tutorial I:
Fundamentals of Optical Computing and Integrated Photonics for High-Performance Digital Logic and Efficient Machine Learning

Presenters: Jiaqi Gu[1,2], Chenghao Feng[1,3]

Contributors: Hanqing Zhu[1], Zhoufeng Ying[1], Zheng Zhao[1], Ray T. Chen[1], David Z. Pan[1]

[1]The University of Texas at Austin, [2]Arizona State University, [3]Alpine Optoelectronics

jqgu@utexas.edu; fengchenghao1996@utexas.edu

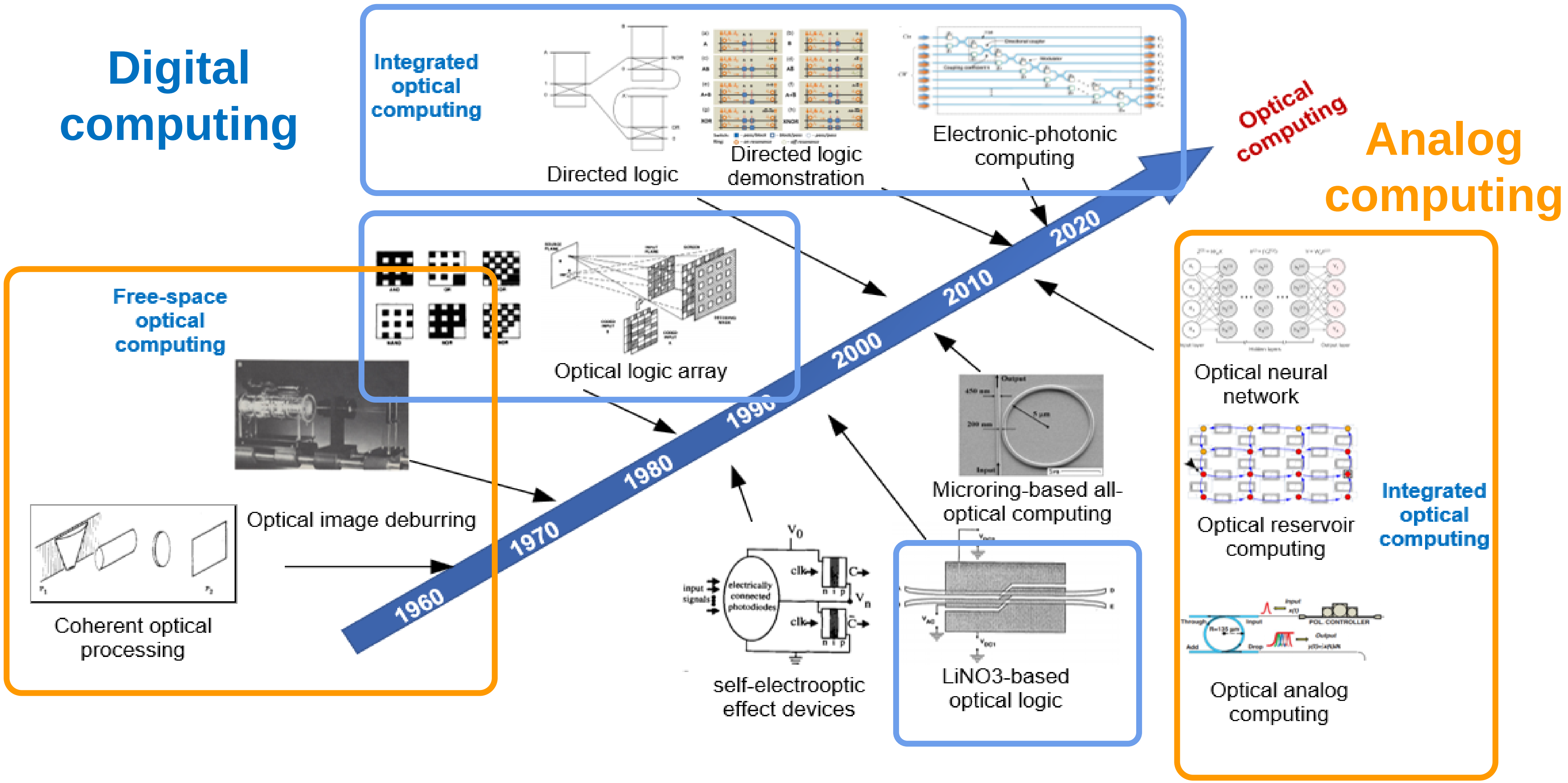
Outline of Tutorial I

- Introduction to Optical Computing
- Design and Demonstration of Electronic-Photonic Digital Computing
- Analog Photonic Computing for Optical Neural Networks
  - Coherent Photonic Tensor Core
  - Incoherent Photonic Tensor Core

Outline of Tutorial I

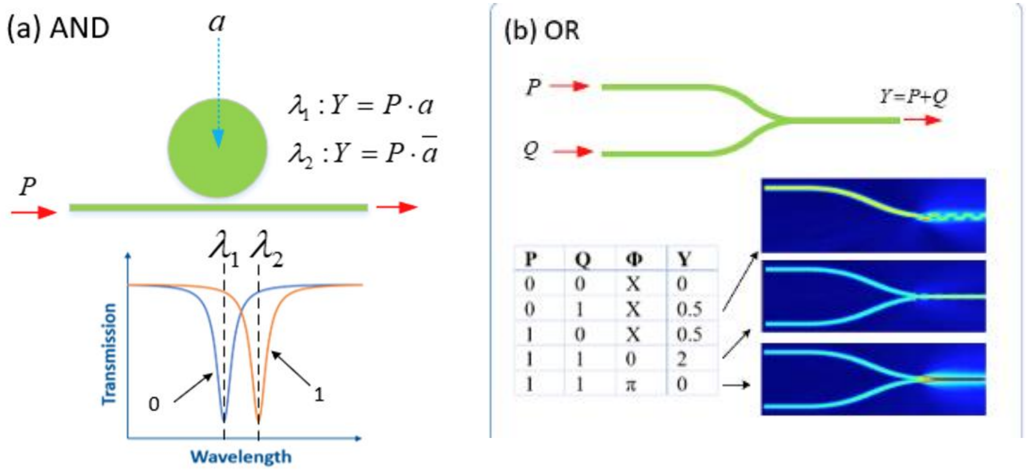
- Introduction to Optical Computing
- Design and Demonstration of Electronic-Photonic Digital Computing
- Analog Photonic Computing for Optical Neural Networks
  - Coherent Photonic Tensor Core
  - Incoherent Photonic Tensor Core

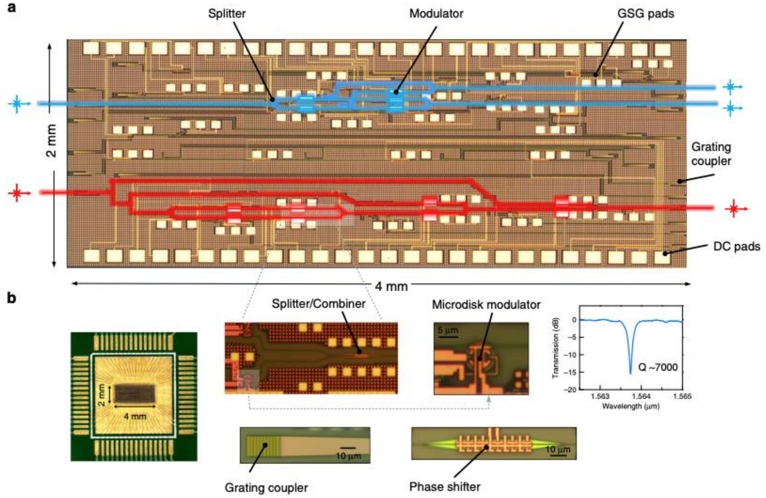
The Timeline of Optical Computing



Digital vs. Analog Photonic Computing

Digital computing (Logic, ALU, Control)

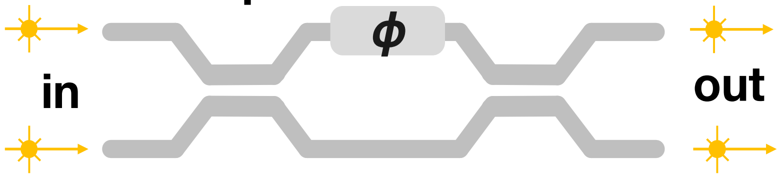




[Ying et al, Nature Comm. 2020]

Analog computing (ML, Optimization, Linear Algebra)

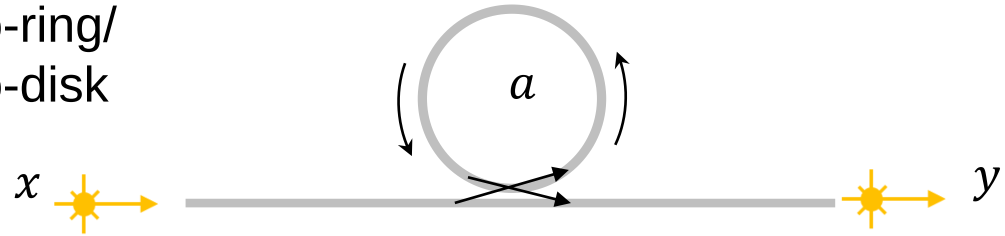
MZI



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

2×2 Unitary Matmul (~100×20 μm^2)

Micro-ring/
Micro-disk



$y = a \cdot x$

Scalar Mul. (~10×10 um^2)

WDM+PD

$y = \sum x_i$

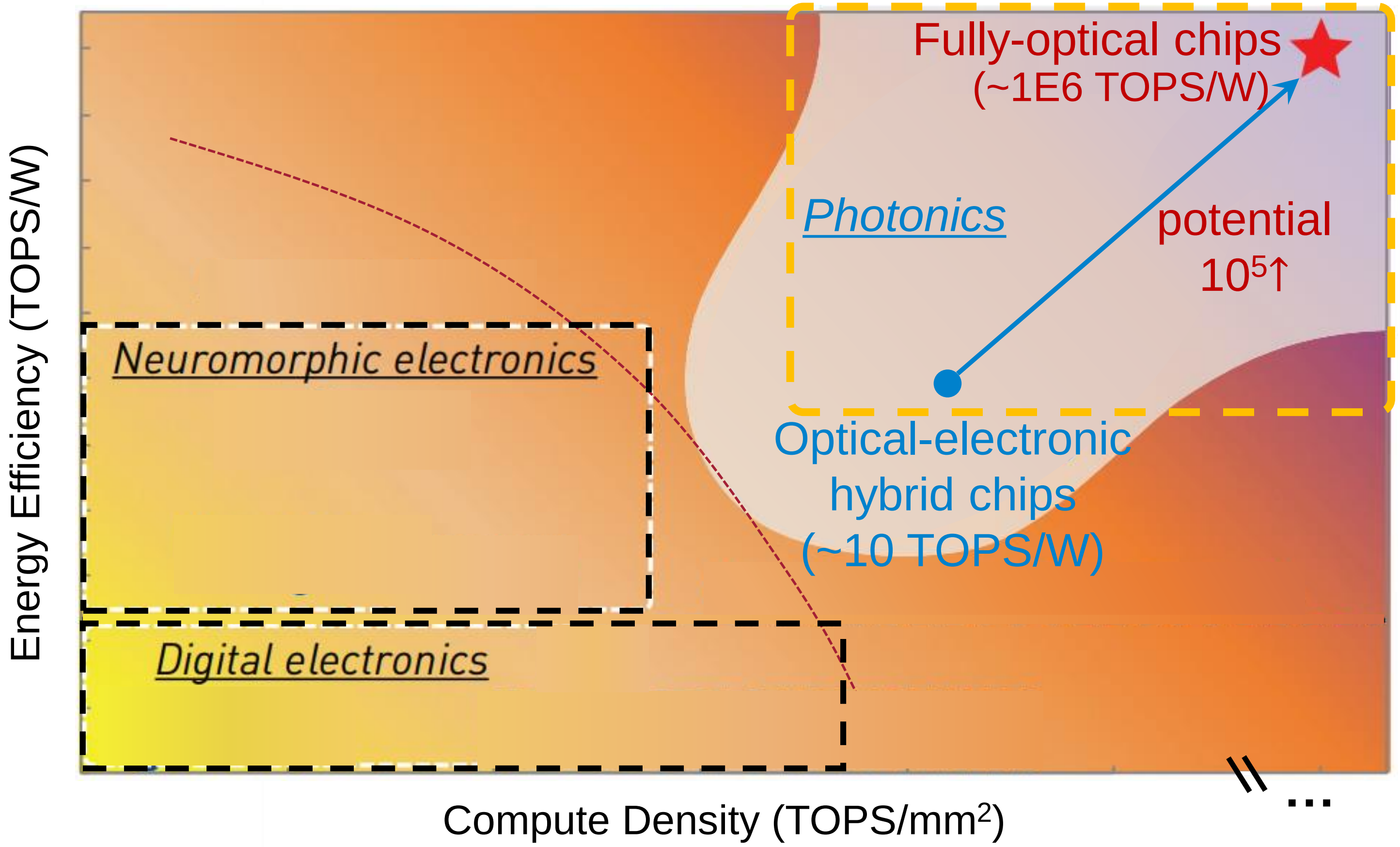
[Lightelligence]

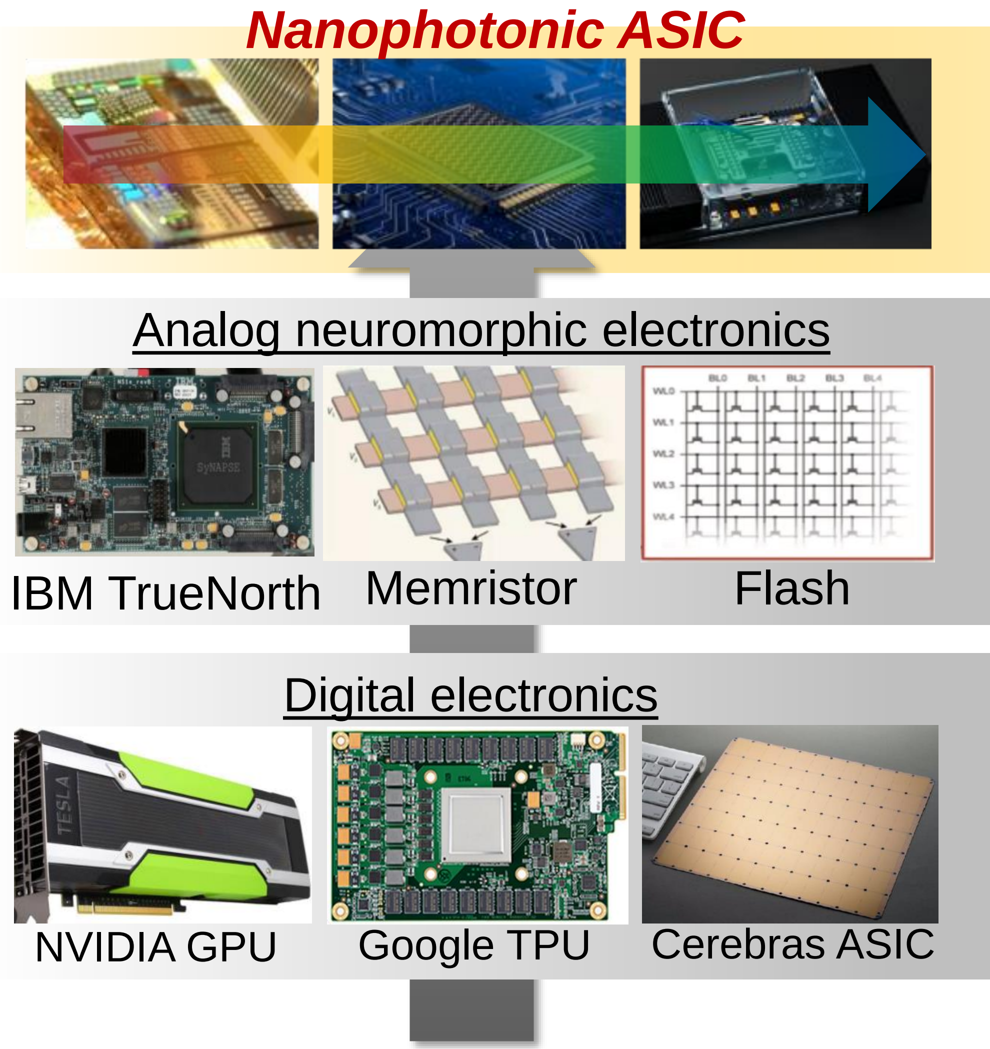
WDM-based Summation

Photonic Computing Chips

- Evolve from <u>electronics</u> to <u>integrated photonics</u>

What is unique about photonics?





Source: Mitchell A. Nahmias, Bhavin J. Shastri, Alexander N. Tait, Thomas Ferreira de Lima and Paul R. Prucnal, "Neuromorphic photonics," Optics & Photonics News, Jan 2018.

Electrical Computing vs Photonic Computing

High speed

Delay 100 ns ~ 1 μs
A few hundred clock cycles

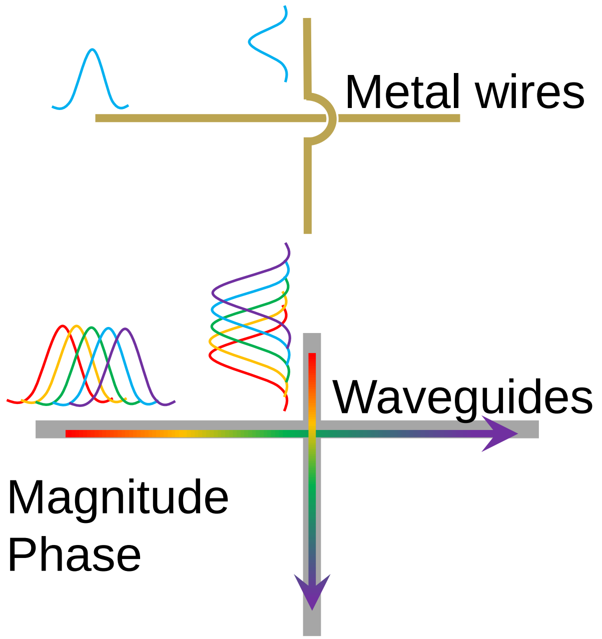
Electronic Comp Unit

Delay ≪ 1 ns

Photonic Comp Unit

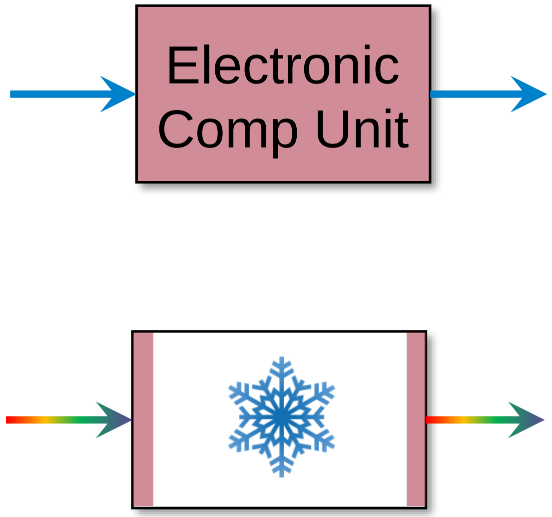
Computing as light propagate

Massive parallelism



Light propagate in parallel

High energy efficiency



Passive circuits consumes near zero static power

Application Potentials of Photonic Computing

- ***Ultra-fast, efficient*** digital control / ALU
- ***Energy-efficient, real-time*** machine intelligence

Fast edge/mobile processing

High-throughput datacenter processing

Smart commun. network, distributed computing

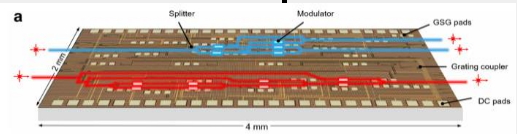
Scientific comp., optimization, bio / material..

Outline of Tutorial I

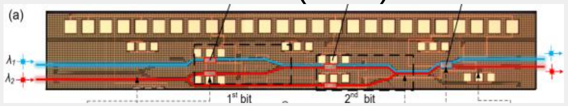
- Introduction to Optical Computing
- Design and Demonstration of Electronic-Photonic Digital Computing
- Analog Photonic Computing for Optical Neural Networks
  - Coherent Photonic Tensor Core
  - Incoherent Photonic Tensor Core

Progress in Optical Digital Computing

Arithmetic operations

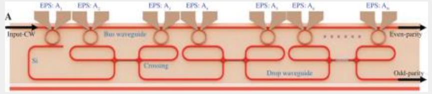


Full adder (2020)

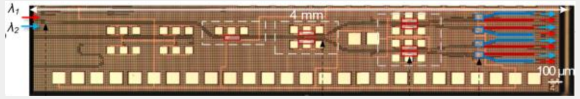


Comparator (2021)

Control circuits

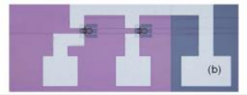


Parity checker (2018)



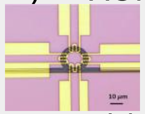
Decoder and multiplexer (2020)

Bitwise Logic operations



NOR gate (2011)

XOR gate (2012)



Multi-operand (2019)

Optical building block library

System Specification

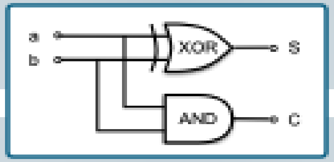
High-level Synthesis

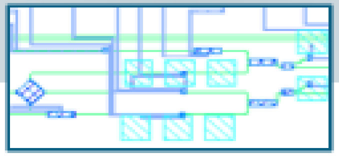
Logic Synthesis & Mapping

Physical Design

Fabrication

```
module adder;
  input a, b;
  input cin;
  ....
endmodule
```





Relatively mature EDA tools

BDD-based Synthesis (2016~)

Power Efficient Synthesis (2018~)

WDM Synthesis (2019)

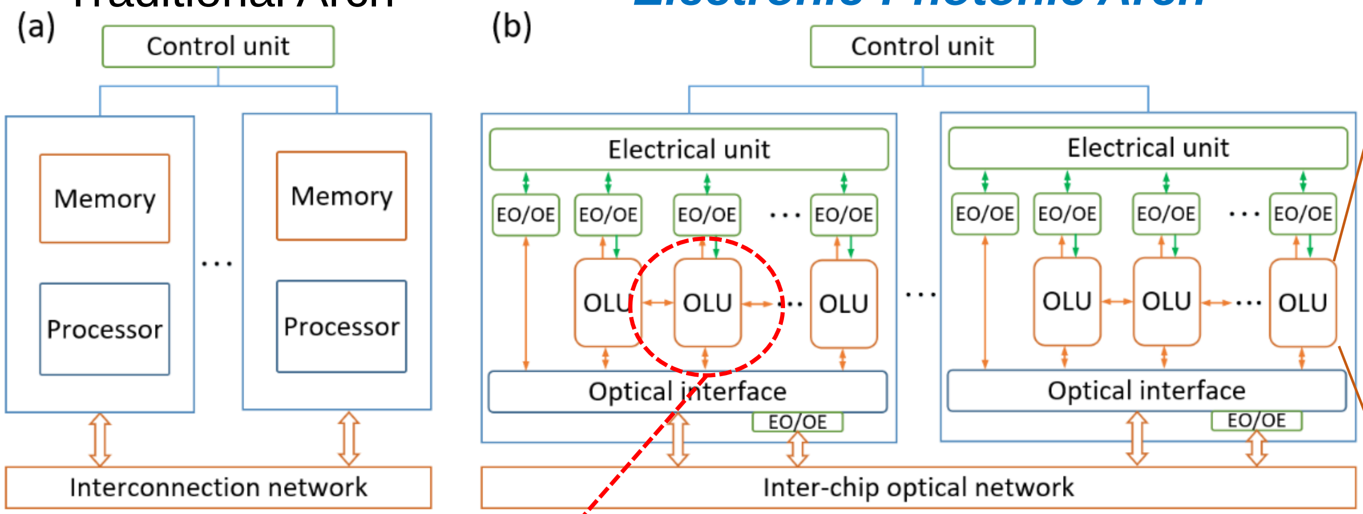
Preliminary EPDA tools

Optical Interconnect Synthesis (2016~)

WDM-based Electronic-Photonic Computing Circuits

Traditional Arch

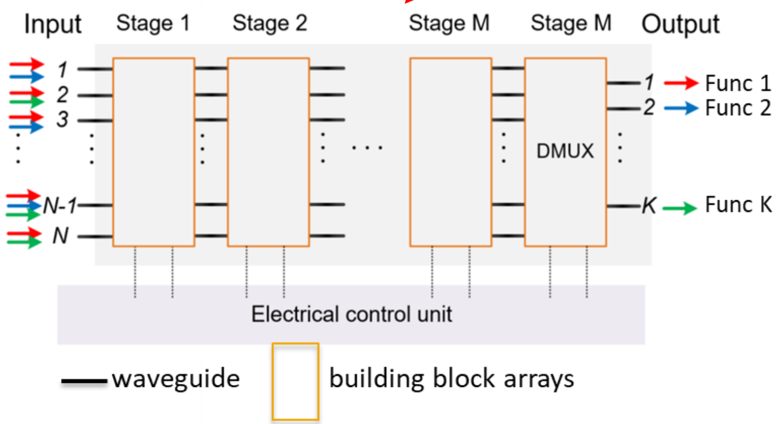
Electronic-Photonic Arch



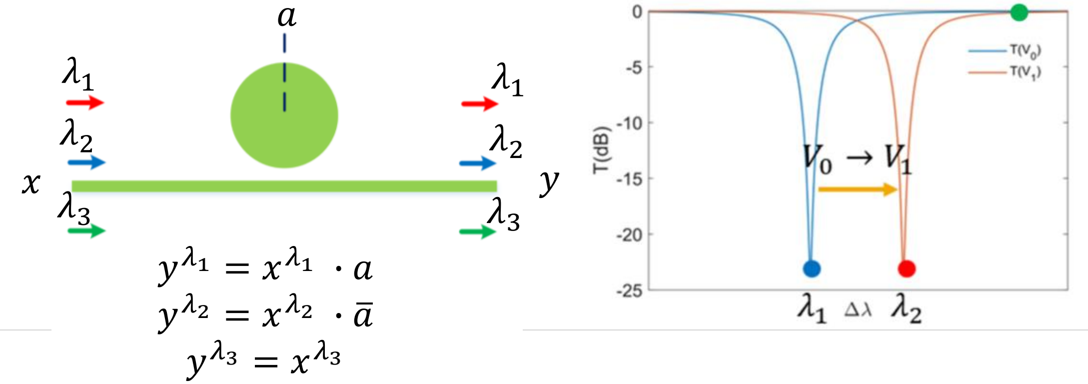
OLU Features

- Inheritance: light in and light out
- Continuity: no OE/EO conversion
- Independence: no product between two optical basis
- Parallelism: multiple input λs, multiple logic functions

Multiple wavelengths reuse the same unit !



(c) WDM-based Optical logic unit [2]

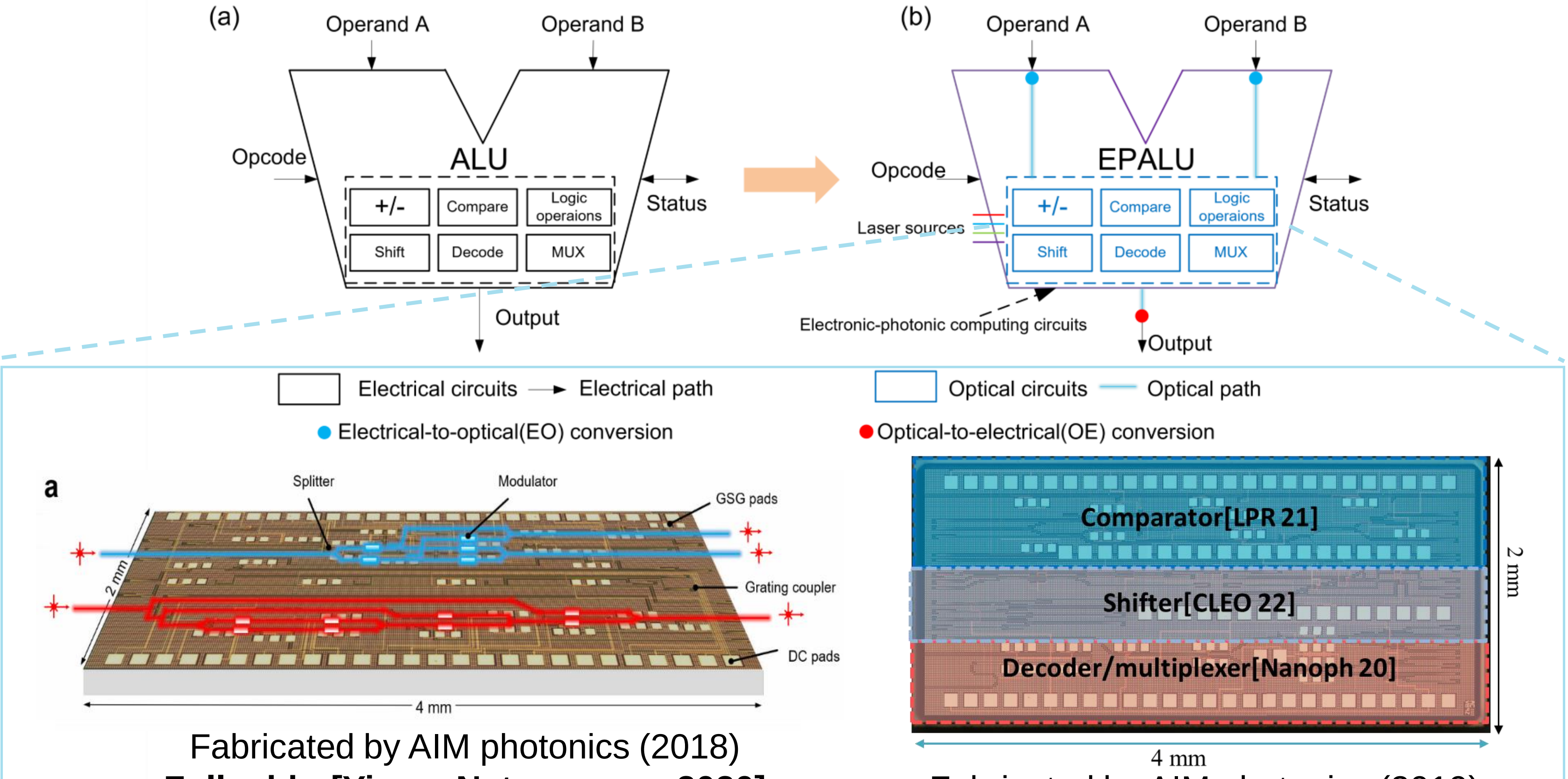


(d) Hardware efficiency improvement of electrooptic (EO) logic gates using wavelength division multiplexing(WDM)

[1] Ying, Z. *et al.*, **JSTQE** 2018
[2] Feng, C. *et al.*, **Photonics West** 2020

Photonic-Electronic Arithmetic Logic Unit (ALU)

- From electrical ALU to <u>high-speed</u> and <u>energy-efficient</u> photonic-electronic ALU
- We demonstrate 20Gb/s photonic-electronic digital computing chips
- For general-purpose digital computing

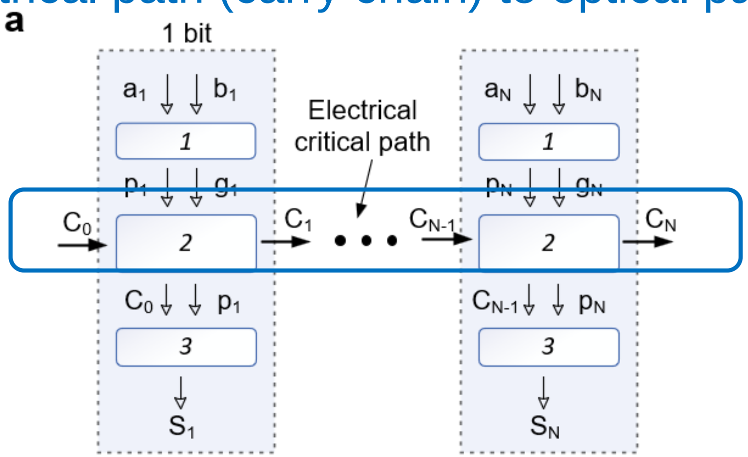


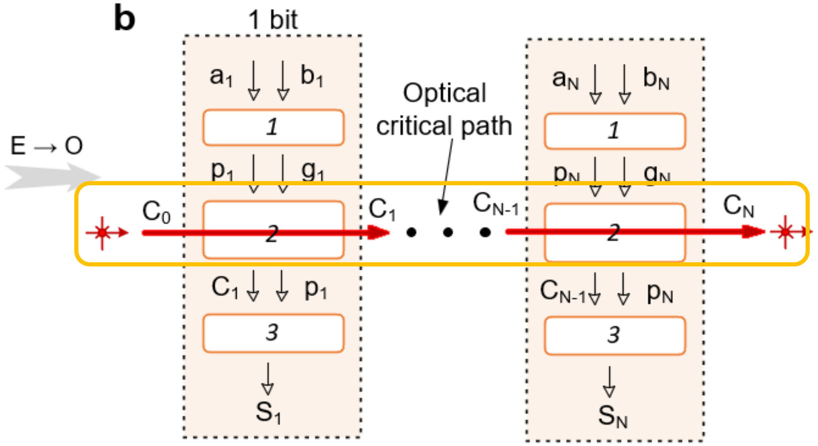
Fabricated by AIM photonics (2018)
Full adder[Ying+, Nat commun 2020]

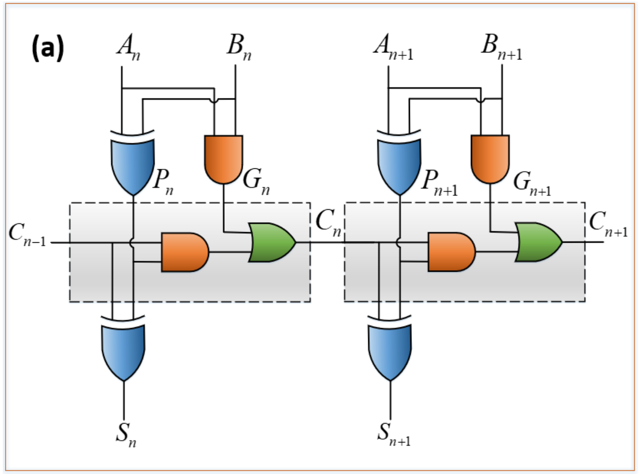
Fabricated by AIM photonics (2019)

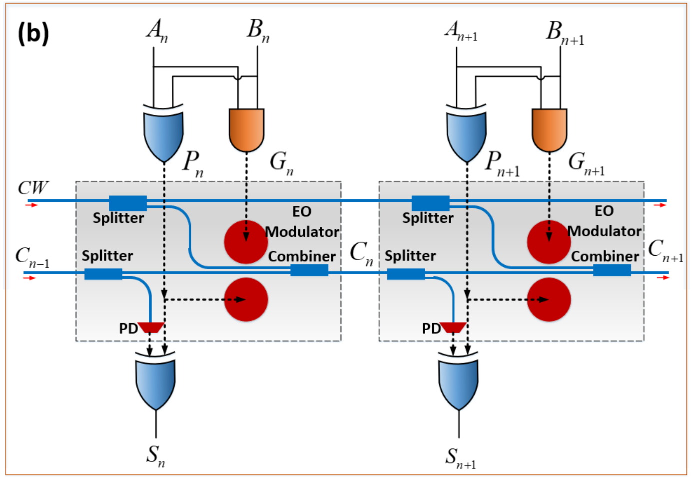
WDM-based Electronic Photonic Carry-Select Adder

Replace electrical path (carry chain) to optical path



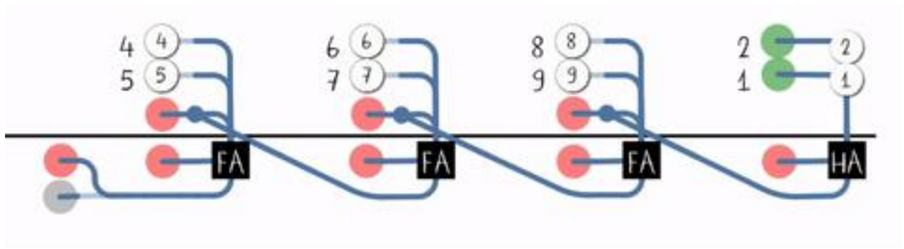




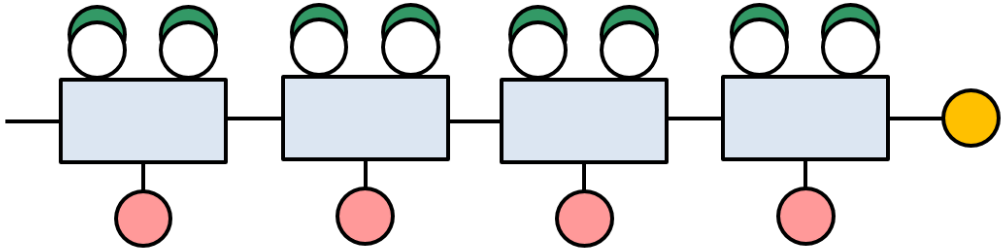


Advantages of Using Optics to Implement Additions

Electrical full adder



Optical full adder

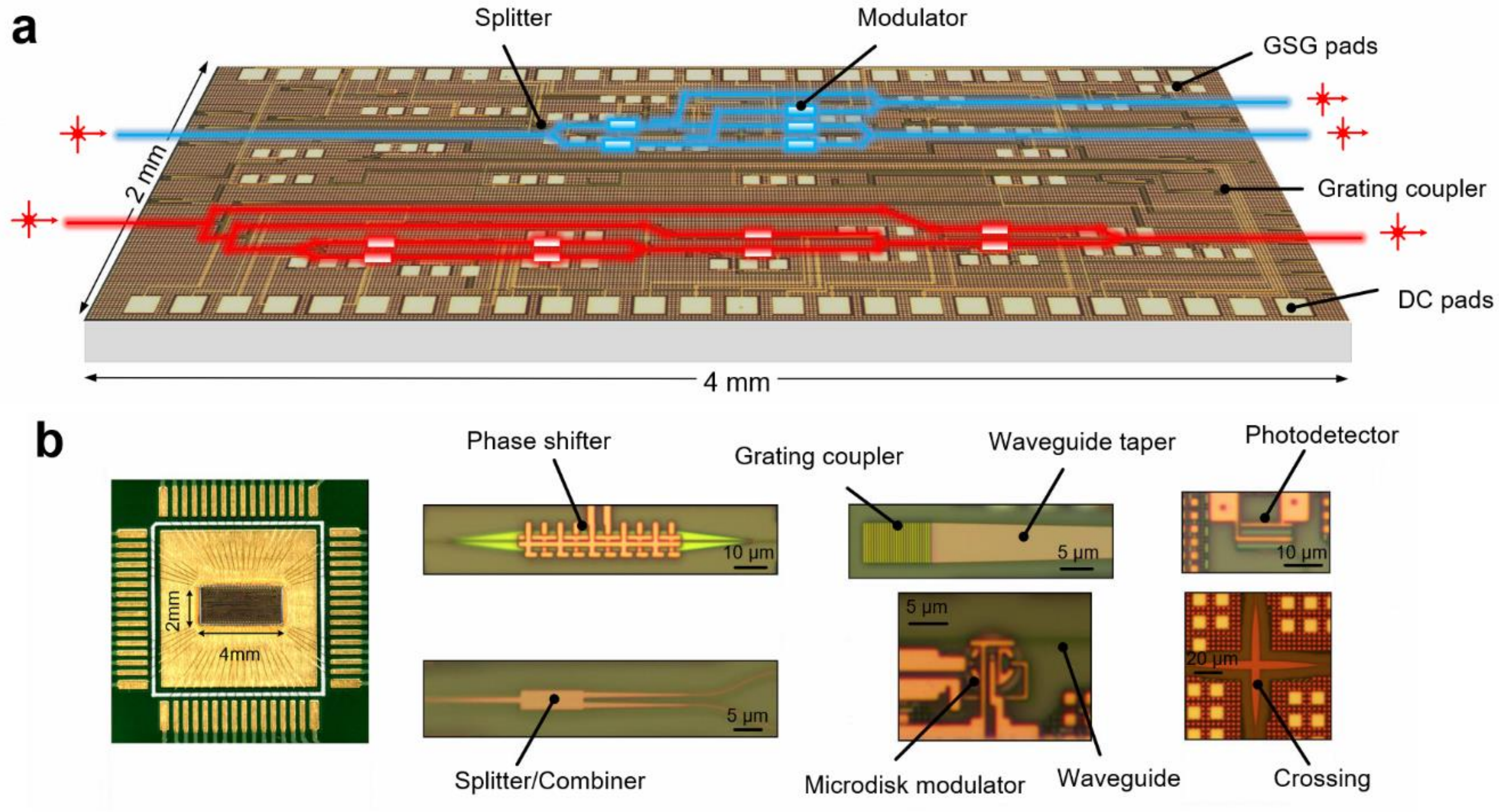


Latency of electrical full adder → **Reduced to** → **Latency of optical full adder**

$$T = T_{p,g} + T_{epb} \times n \qquad T_{opb} \ll T_{epb} \qquad T = T_{p,g} + T_{sw} + T_{opb} \times n$$

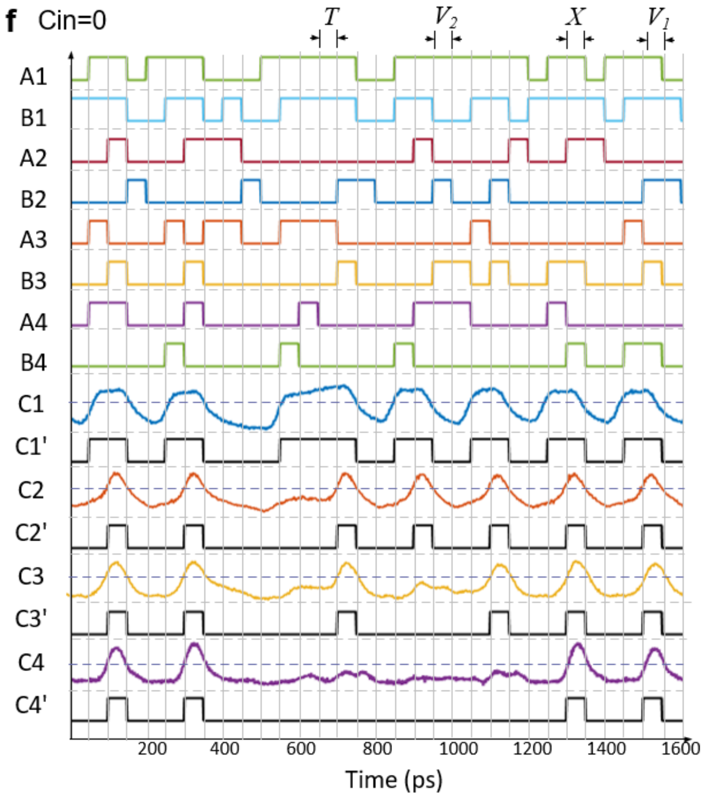
Delay for generating P and G	Switch time of modulators	Optical propagation delay per bit	Electrical delay per bit
$T_{p,g}$	T_{sw}	T_{opb}	T_{epb}

Chip Layout of the Electronic Photonic Full Adder



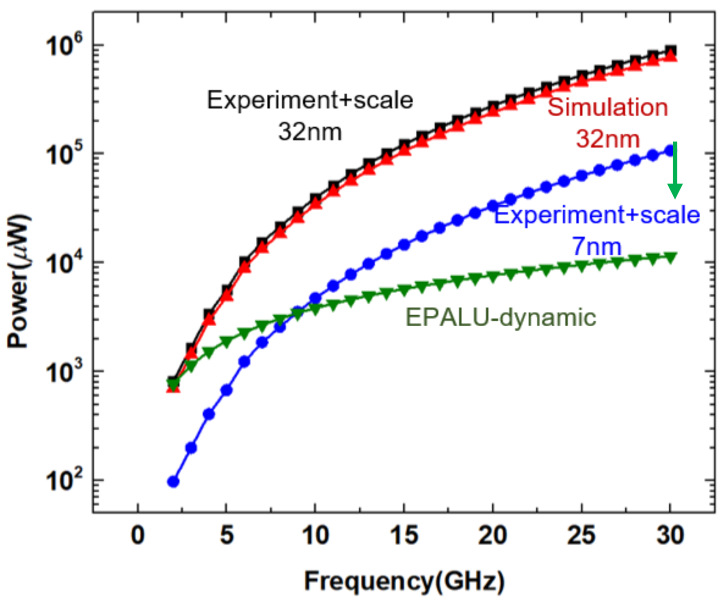
Comparison with the State-of-the-art Transistors

- Compared to 32 nm / 7nm (scaling) from Intel
- 4× faster (20 GHz vs 5 GHz)
- >10× more energy-efficient

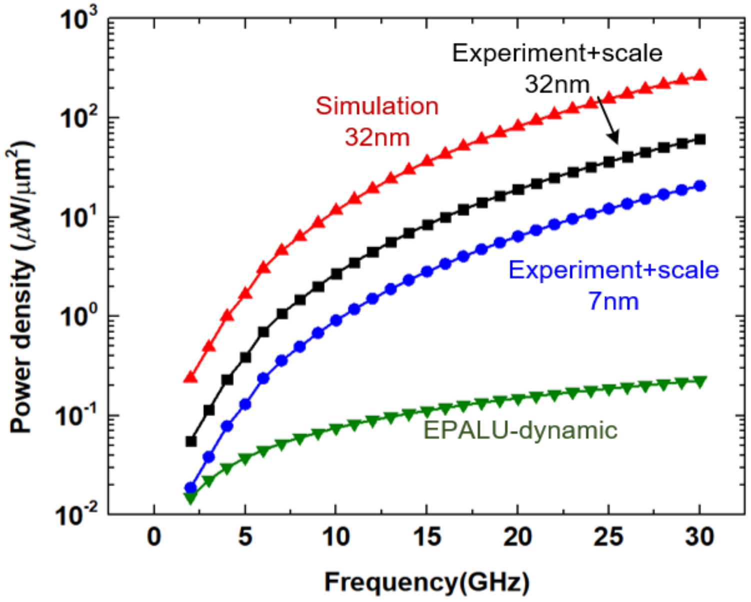


20 Gb/s

10× Low power than 7nm @20 GHz

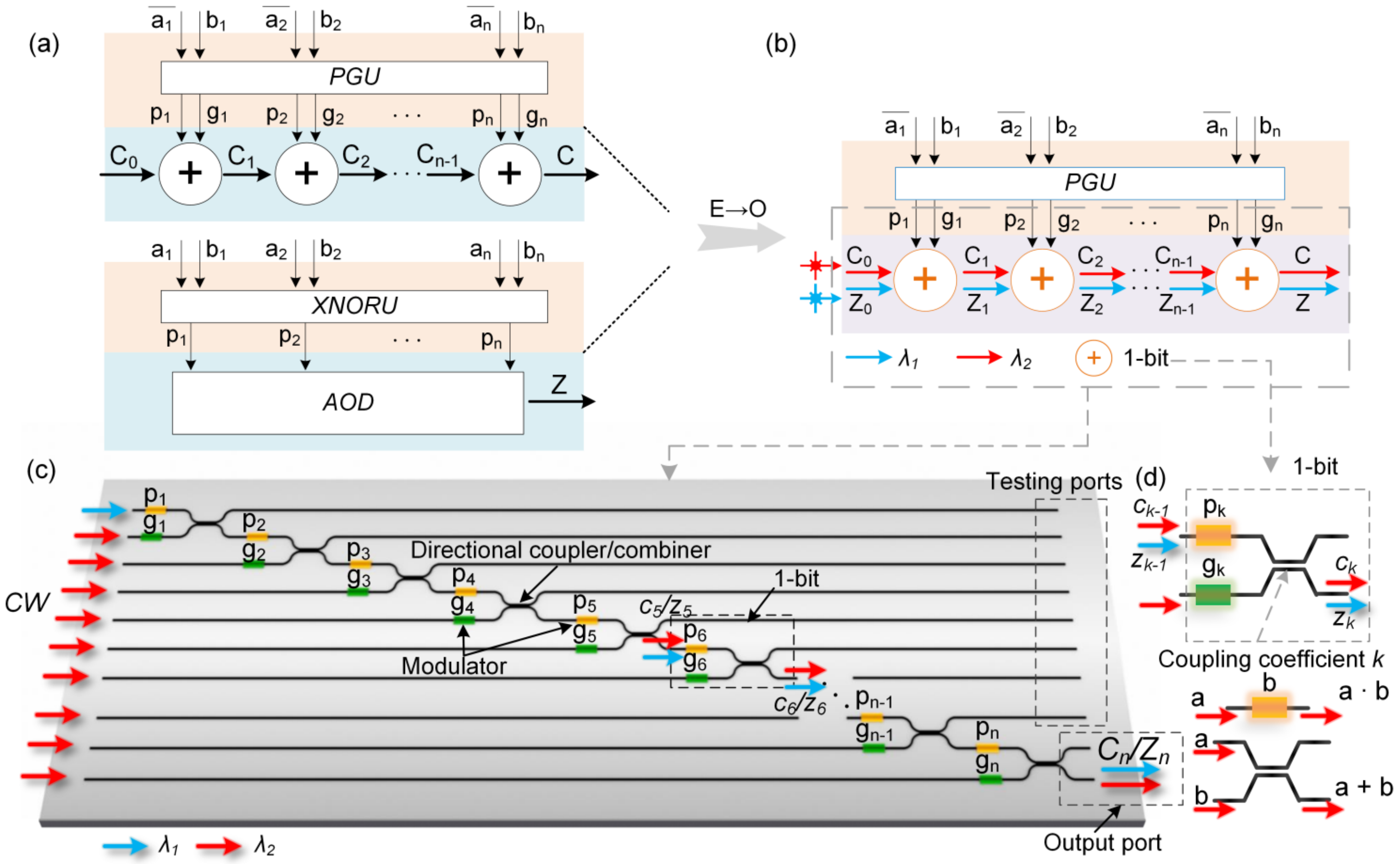


>70× Low power density than 7 nm @ 20 GHz



WDM-based Photonic-Electronic Unsigned Comparator

WDM-based comparator: Different λs, different functions



$C: A < B?$

$Z: A = B?$

Truth table

C	Z	Result
0	1	A=B
0	0	A>B
1	0	A<B

$$p_k = a_k \otimes \overline{b_k}$$

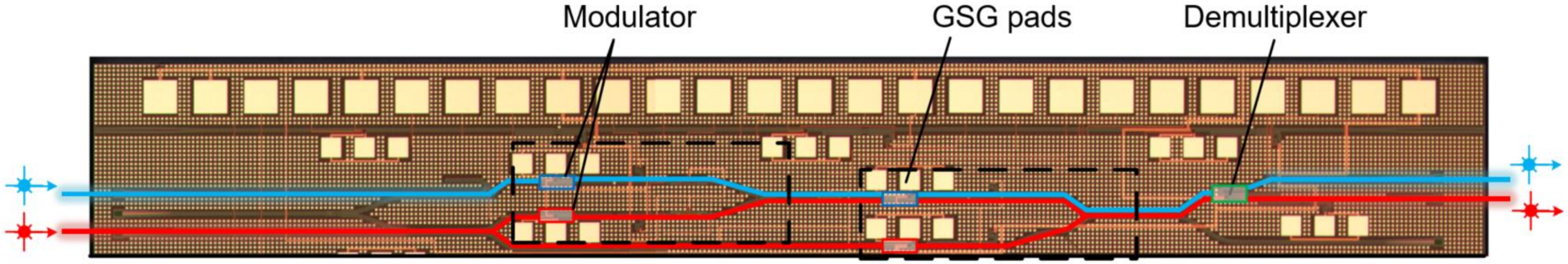
$$g_k = a_k \cdot \overline{b_k}$$

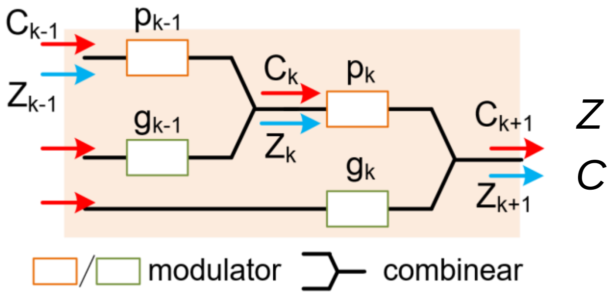
$$c_k = p_k \cdot c_{k-1} + g_k$$

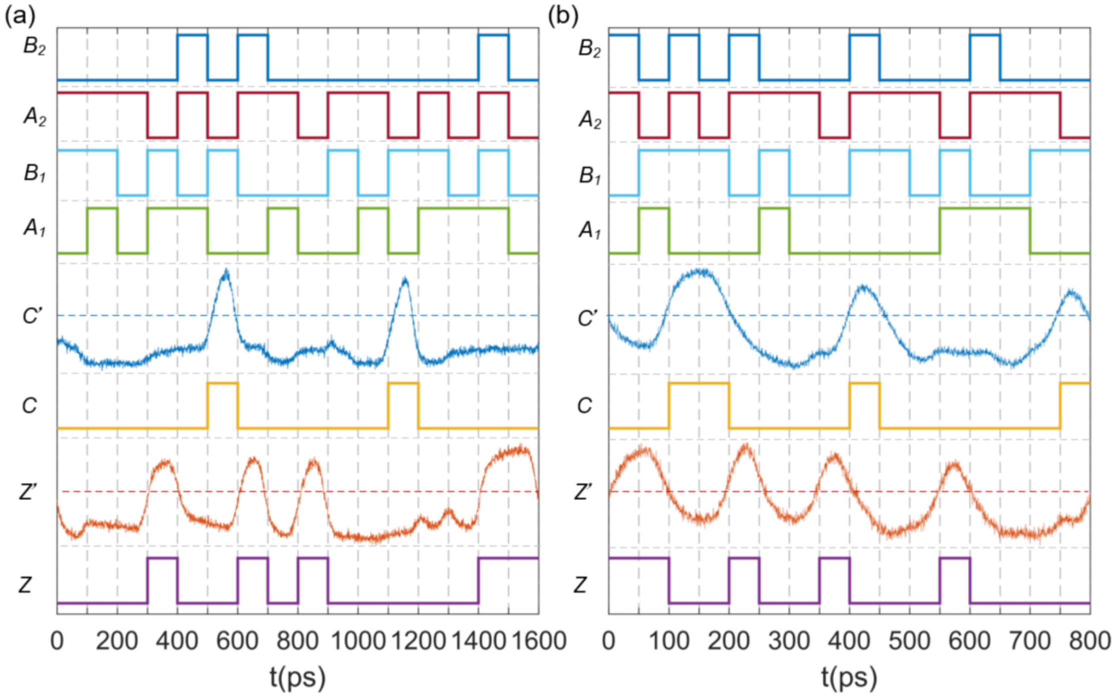
$$z_k = p_k \cdot z_{k-1}$$

[Feng C. *et al.*, **Laser & Photonics Reviews,** 2021]

Experimental Results (2-bit unsigned comparator)







@10 Gb/s

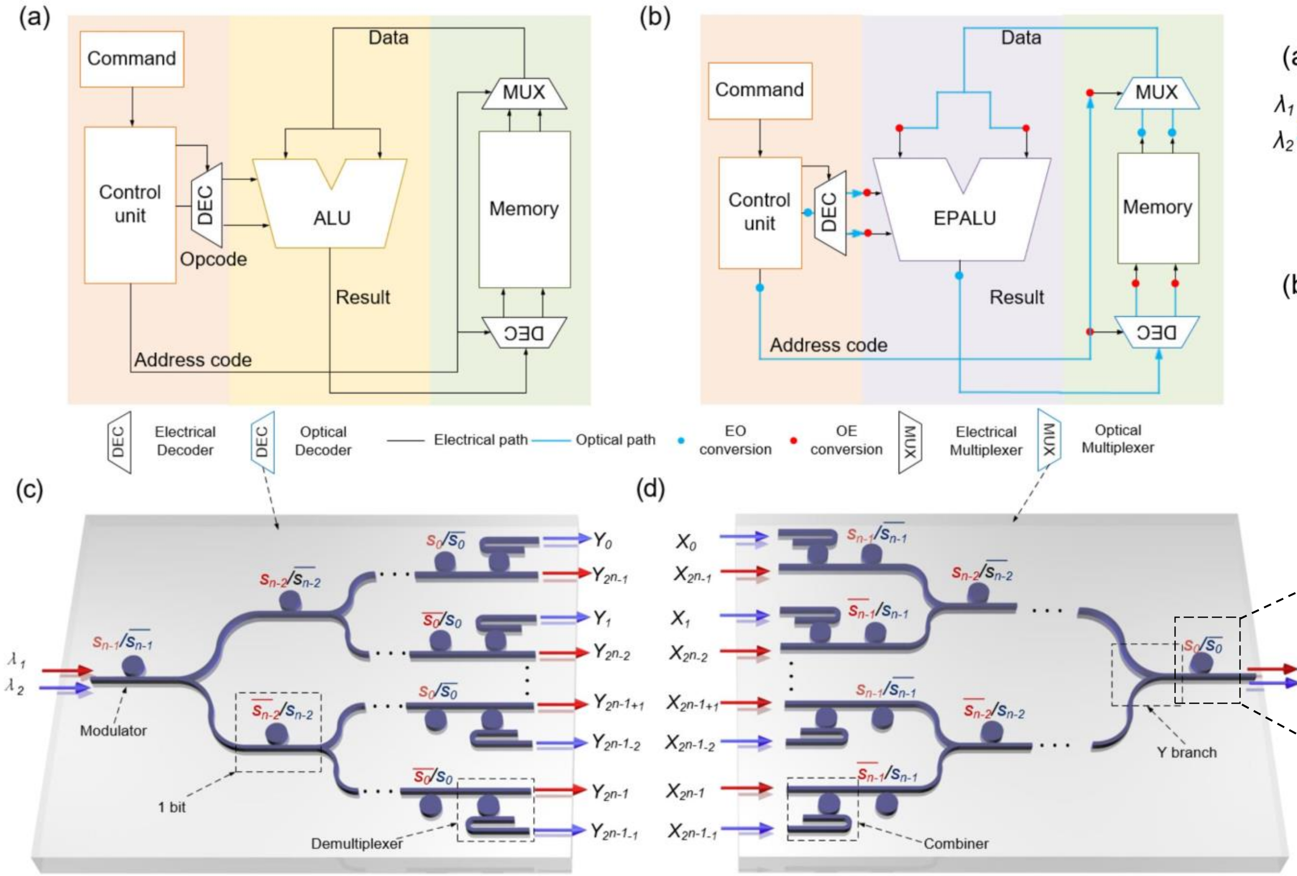
@20 Gb/s

C	Z	Result
0	1	A=B
0	0	A>B
1	0	A<B

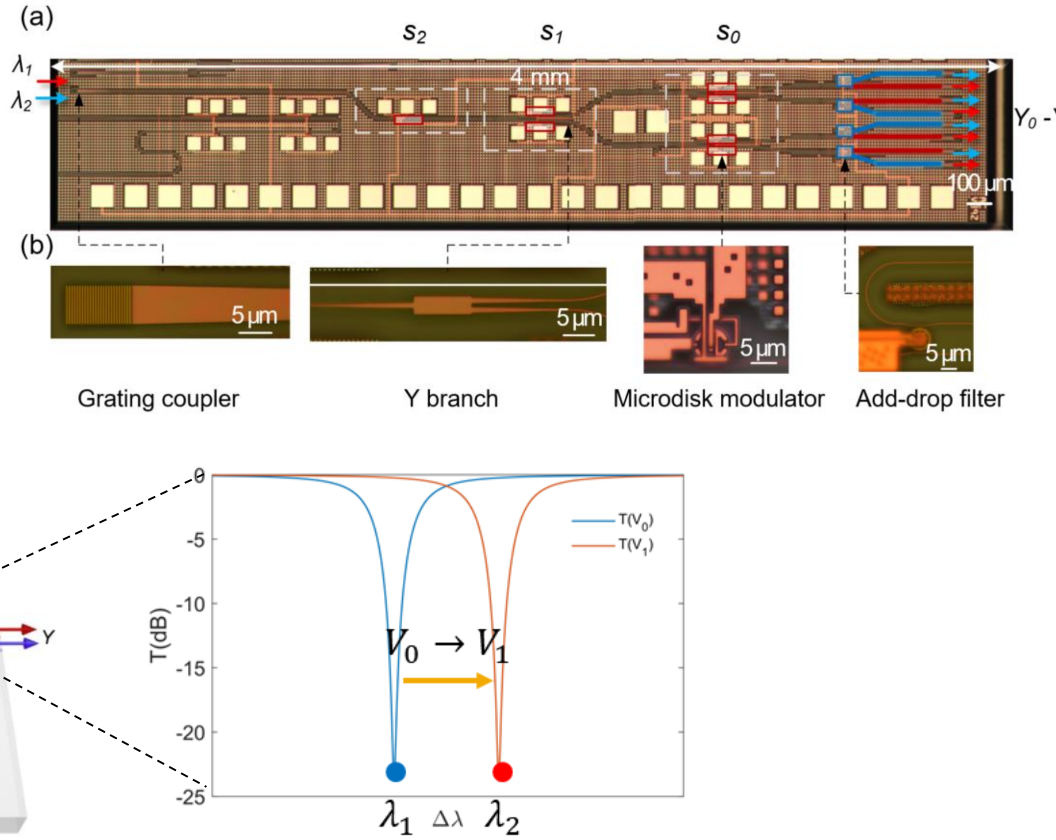
Truth table

WDM-based Electronic-Photonic Switching Network

- Applications: decoder, multiplexer, demultiplexer



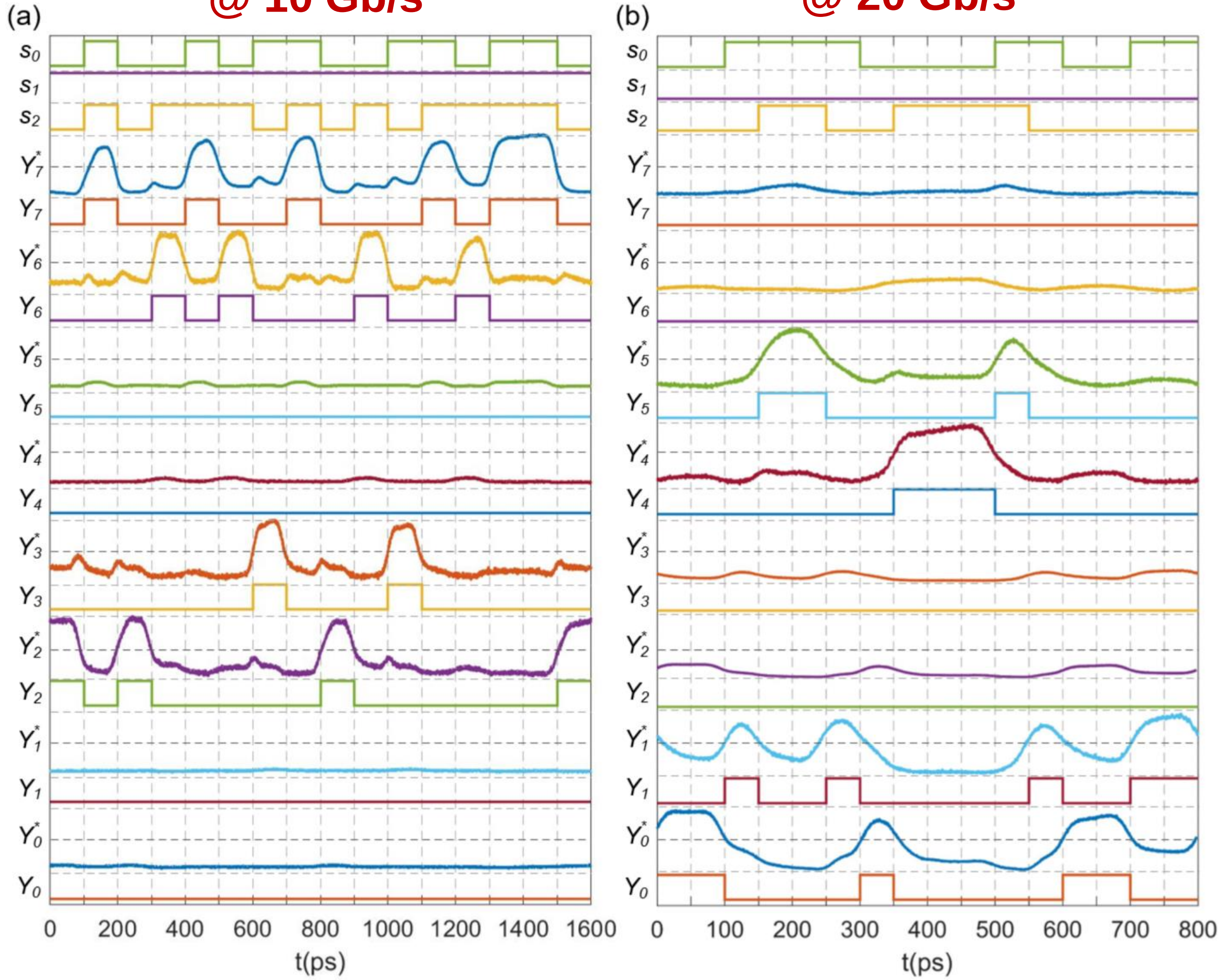
Schematic of the WDM-based switching network

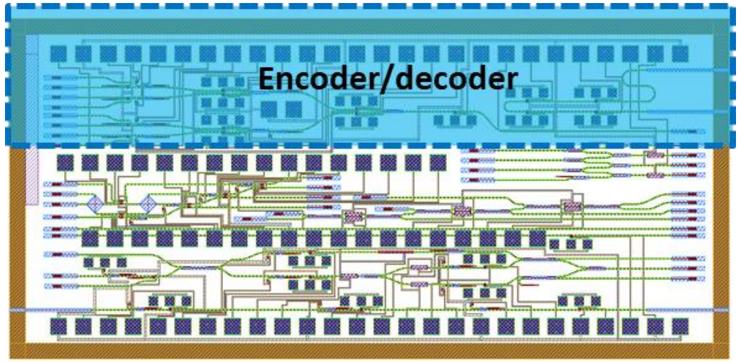


[Feng C. *et al.*, **Nanophotonics**, 2020]

Experimental Results of the 3-8 Optical Decoder

@ 10 Gb/s **@ 20 Gb/s**





Output	Input
Y_7	$X_3X_2X_1$
Y_6	$X_3X_2\overline{X_1}$
Y_5	$X_3\overline{X_2}X_1$
Y_4	$X_3\overline{X_2X_1}$
Y_3	$\overline{X_3}X_2X_1$
Y_2	$\overline{X_3}X_2\overline{X_1}$
Y_1	$\overline{X_3X_2}X_1$
Y_0	$\overline{X_1X_2X_3}$

Truth table

Outline of Tutorial I

- Introduction to Optical Computing
- Design and Demonstration of Electronic-Photonic Digital Computing
- Analog Photonic Computing for Optical Neural Networks
  - Coherent Photonic Tensor Core
  - Incoherent Photonic Tensor Core

Hardware Limits in Machine Learning

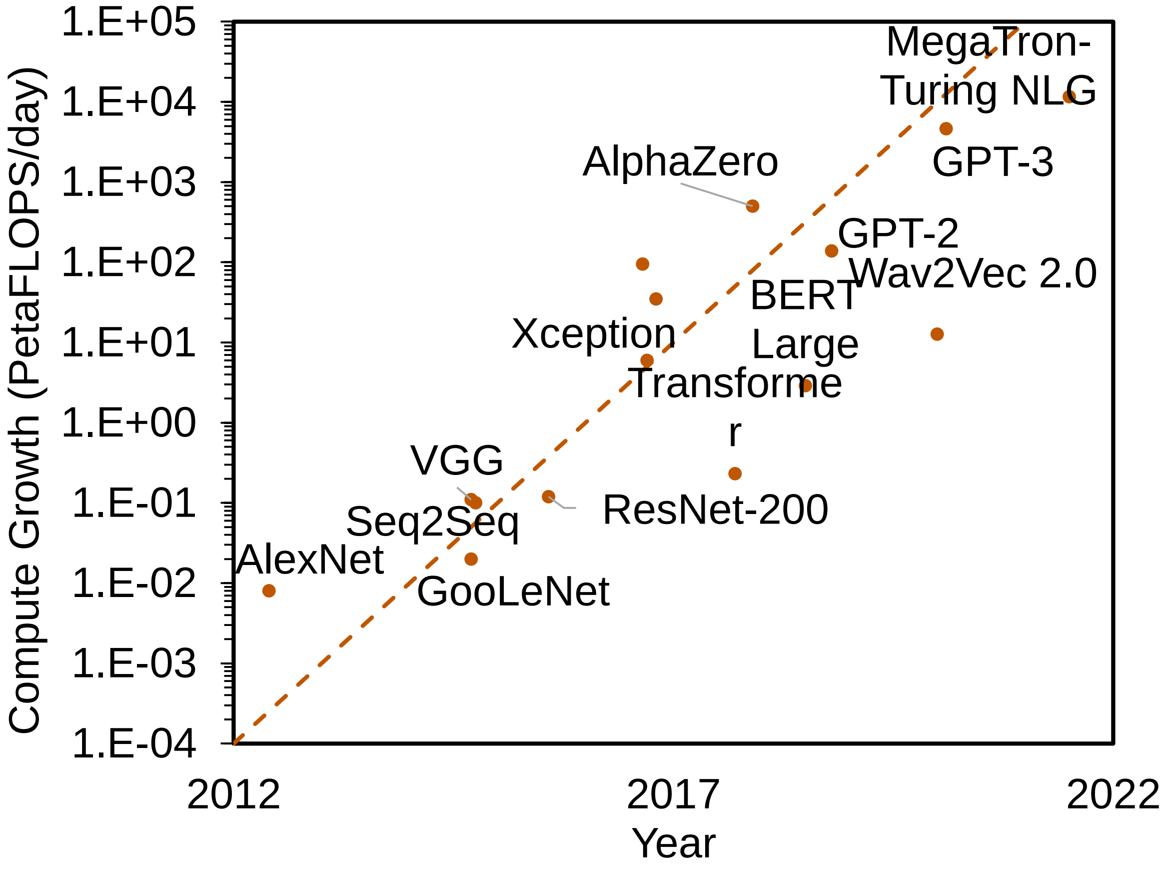
ML Compute

×2 every **3.5** months

~5× the doubling rate

Moore's Law

×2 every **18** months

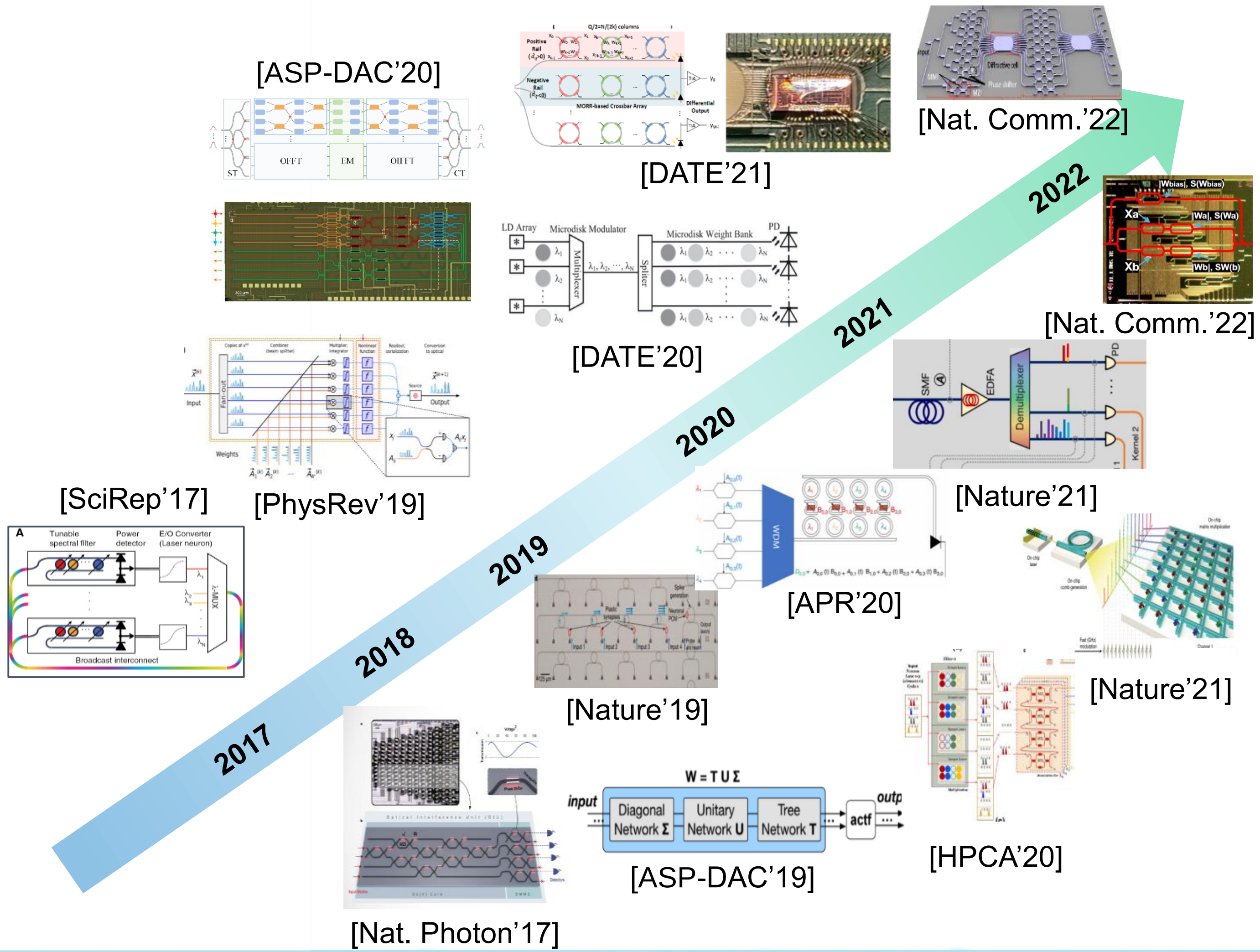


Need **specialized / domain-specific computing hardware** for speed and efficiency breakthrough

Source: https://openai.com/blog/ai-and-compute/
Source: https://spectrum.ieee.org/nvidias-next-gpu-shows-that-transformers-are-transforming-ai

Photonic AI is Booming

Photonic Neural Network Trends in Academia

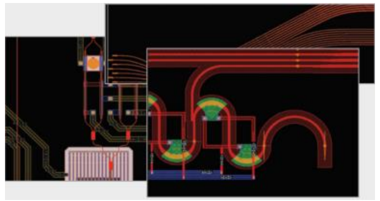


Foundry / EPDA Support in Industry

Photonic Computing Chip Designs

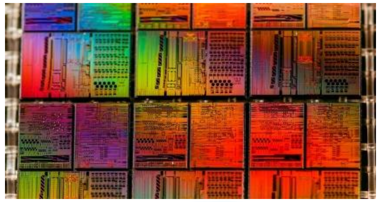
LIGHTMATTER
Lighton
XANADU
LIGHTELLIGENCE
OPTELLIGENCE
LUMINOUS
OPTIUS light powered computing
OpenLight
SALIENCE LABS

Electronic-Photonic Design Automation Tools



Ansys LUMERICAL
SYNOPSYS PHOTONIC SOLUTIONS
cadence PHOTONICS

PDK / Tape-out / Packaging Support



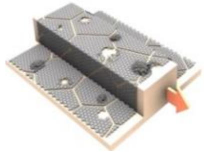
amf ADVANCED MICRO FOUNDRY
AIM photonics
SiEPIC
GlobalFoundries
Tower Semiconductor

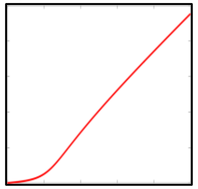
Photonic AI Computing Basics

- Principle: light modulation, interference, photodetection
- Good at ultra-fast, parallel linear operations in the *analog* domain

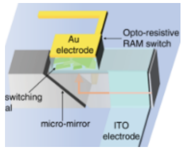
Nonlinear

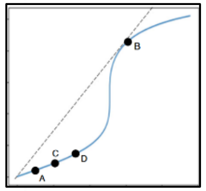
Absorber



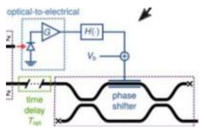


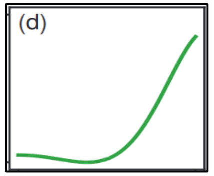
Optical RRAM





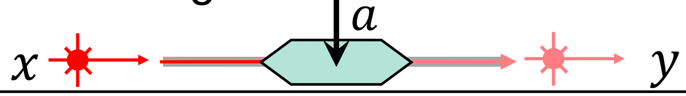
E/O Convert

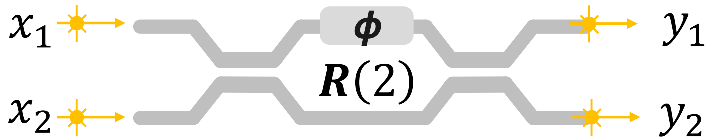


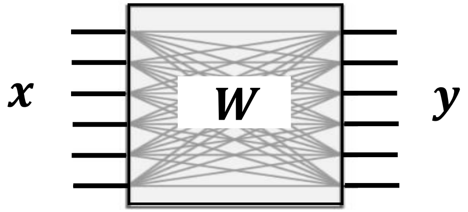


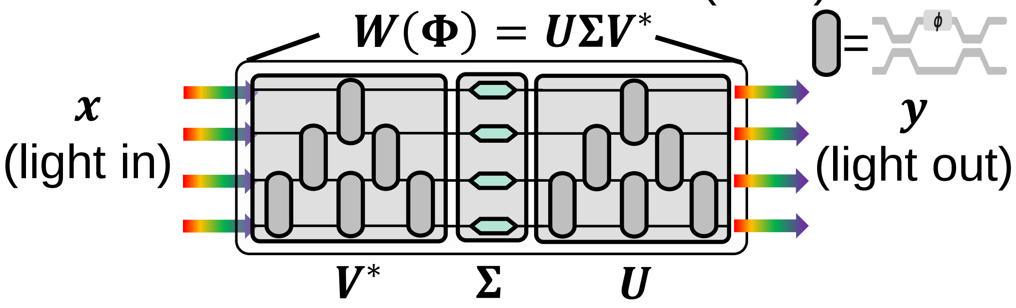
…

Computing Primitives	**Photonic Implementation**
Scalar Multiply $y = a \cdot x$	Light Modulation
2×2 Unitary Matrix Multiply $\boldsymbol{y} = \boldsymbol{R}(2) \times \boldsymbol{x}$, $\boldsymbol{R}(2) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$	Mach-Zehnder Interferometer (MZI)
Matrix-Vector Multiply (MVM) $\boldsymbol{y} = \boldsymbol{W} \times \boldsymbol{x}$	Photonic Tensor Core (PTC) $\boldsymbol{W}(\boldsymbol{\Phi}) = \boldsymbol{U\Sigma V}^*$







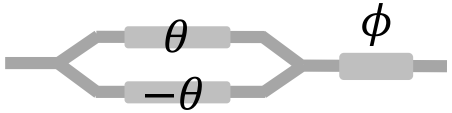


One-shot computing at speed-of-light!

Photonic Tensor Core (PTC) Categories

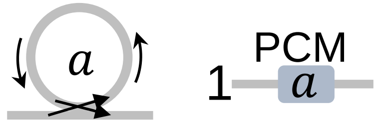
- **Encoding**

Coherent $|x|e^{j\phi(x)}$: magnitude + phase

MZM 1 

$\cos\theta \cdot e^{j\phi}$

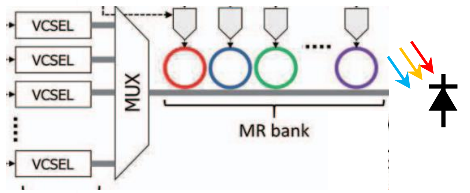
Incoherent $|x|$: magnitude-only

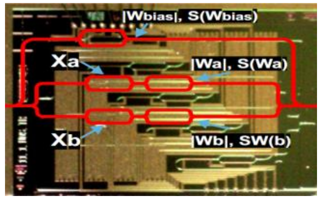
MRR 

$a \geq 0$

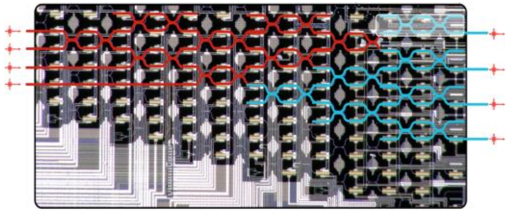
- **MVM principle**

Direct MVM: $y_i = \sum_j w_{ij} x_j$



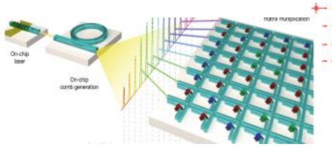


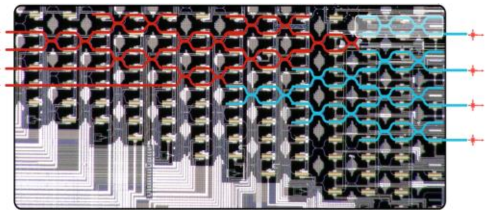
Indirect MVM: $\boldsymbol{y} = \boldsymbol{W}(\Phi)\boldsymbol{x}$



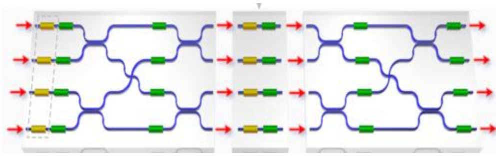
- **Matrix Expressivity**

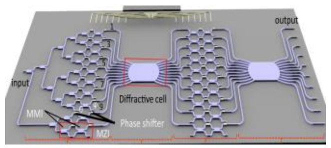
Universal linear: $W \in \mathbb{R}^{N\times N}$ or $\mathbb{C}^{N\times N}$





Subspace linear: $W \subseteq \mathbb{R}^{N\times N}$ or $\mathbb{C}^{N\times N}$





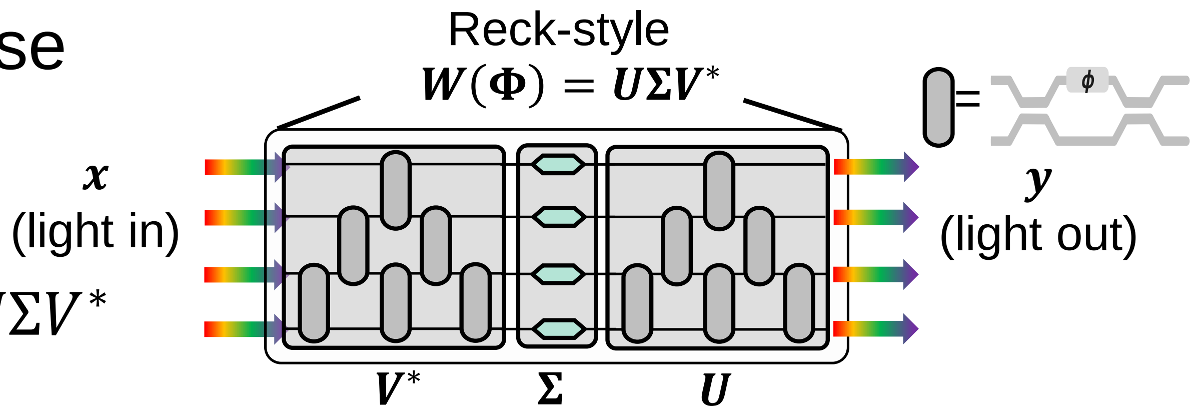
Coherent ONN Architectures

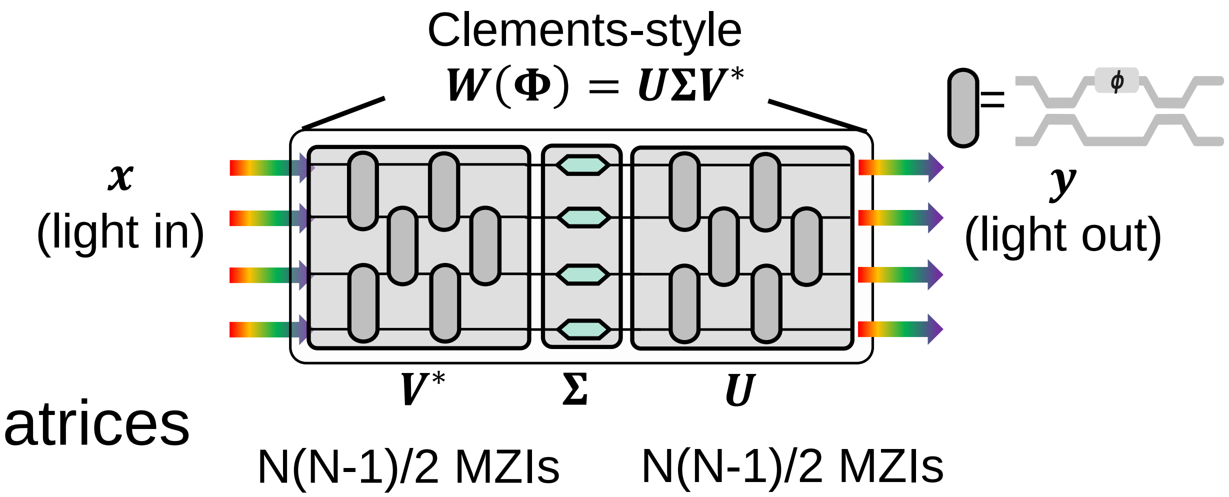
- Encoding: $|x|e^{j\phi(x)}$ magnitude + phase
- Computing: interference (indirect)
- MZI array [Shen+, Nat. Photon'17]
  - Singular value decomposition $W = U\Sigma V^*$
  - Phase decomposition

$$\boldsymbol{U}(N) = \boldsymbol{D} \prod_{i=N}^{2} \prod_{j=1}^{i-1} \boldsymbol{R}_{ij}(\phi_{ij})$$

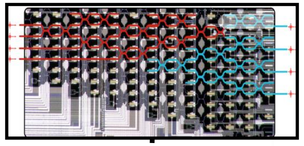
$$\boldsymbol{R}(2) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$$

  - Universal linear units for arbitrary matrices





Universal vs. Specialized Photonic Tensor Cores



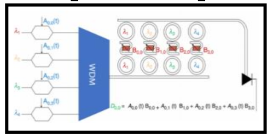
[Nat. Photon.'17]

Universal Linear

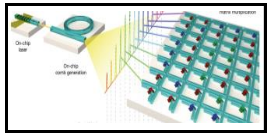
Trade off expressivity vs. efficiency

Specialized Hardware for Subspace Linear

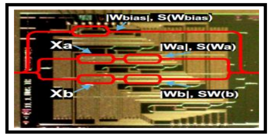
[APL'19]



[Nature'21]

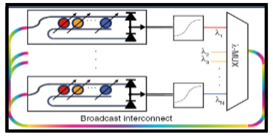


[Nature'22]

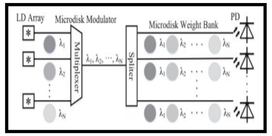


…

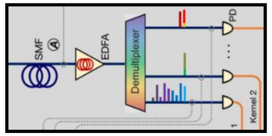
[SciRep'17]



[DATE'19]

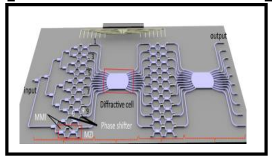


[Nature'21]

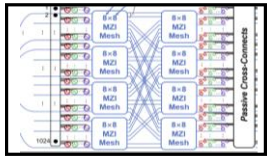


Cross-disciplinary Research

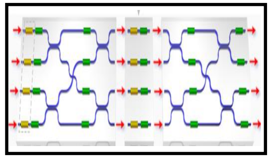
[Nat. Comm.'22]



[APL Photon.'21]



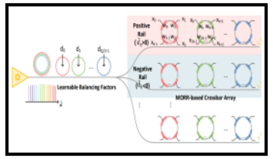
[ACS Photon.'22]



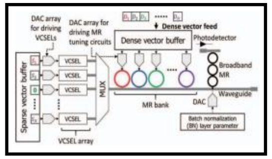
…

Photonics

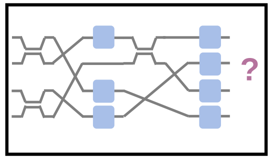
[DATE'21, TCAD'22]



[ASP-DAC'22]



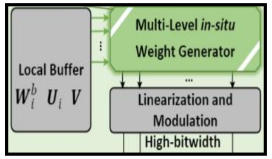
[DAC'22]



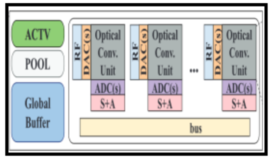
…

Design Automation

[ICCV'21]



[ICCAD'21]



[HPCA'23]

…

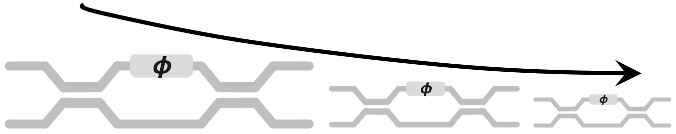
ML / Architecture

Specialized Coherent ONN Architectures

- Leverage the matrix redundancy → reduce hardware cost → subspace linear

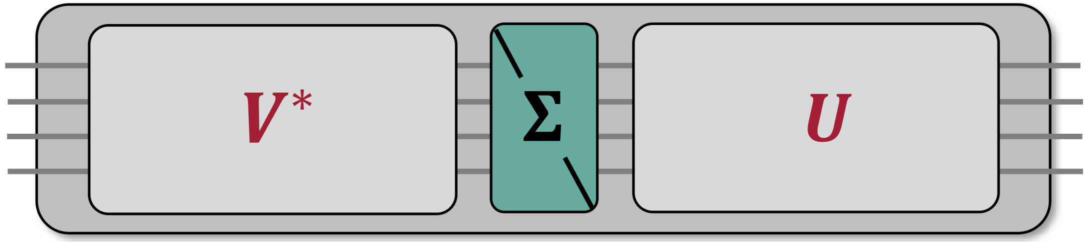
Universal Linear Op

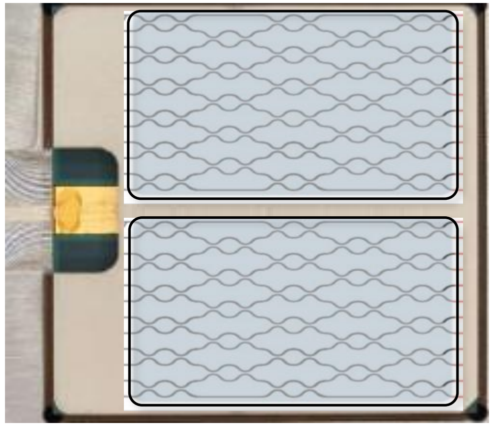
Bulky



Device scaling? Physical limits

Large MZI array $\mathcal{O}(n^2)$ MZIs

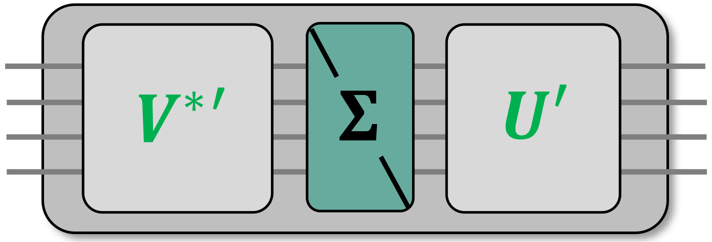


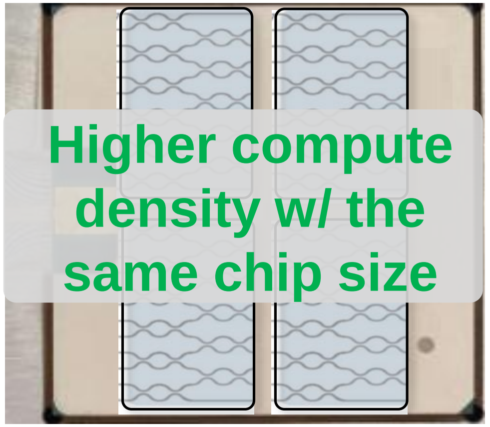


Instead of having general matrices…

Compress

Subspace matrices





Specialized Linear Op

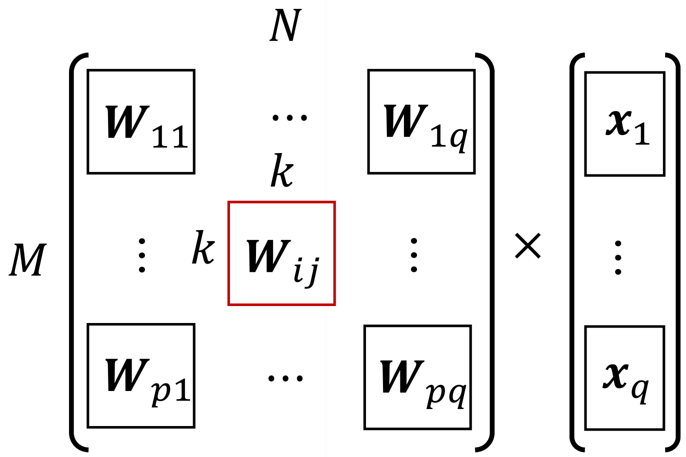
Compact

Compact photonic structure

J. Gu, Z. Zhao, C. Feng, M. Liu, R.T. Chen, D.Z. Pan, "Towards Area-Efficient Optical Neural Networks: An FFT-based Architecture," **ACM/IEEE ASP-DAC**, 2020. Best Paper Award

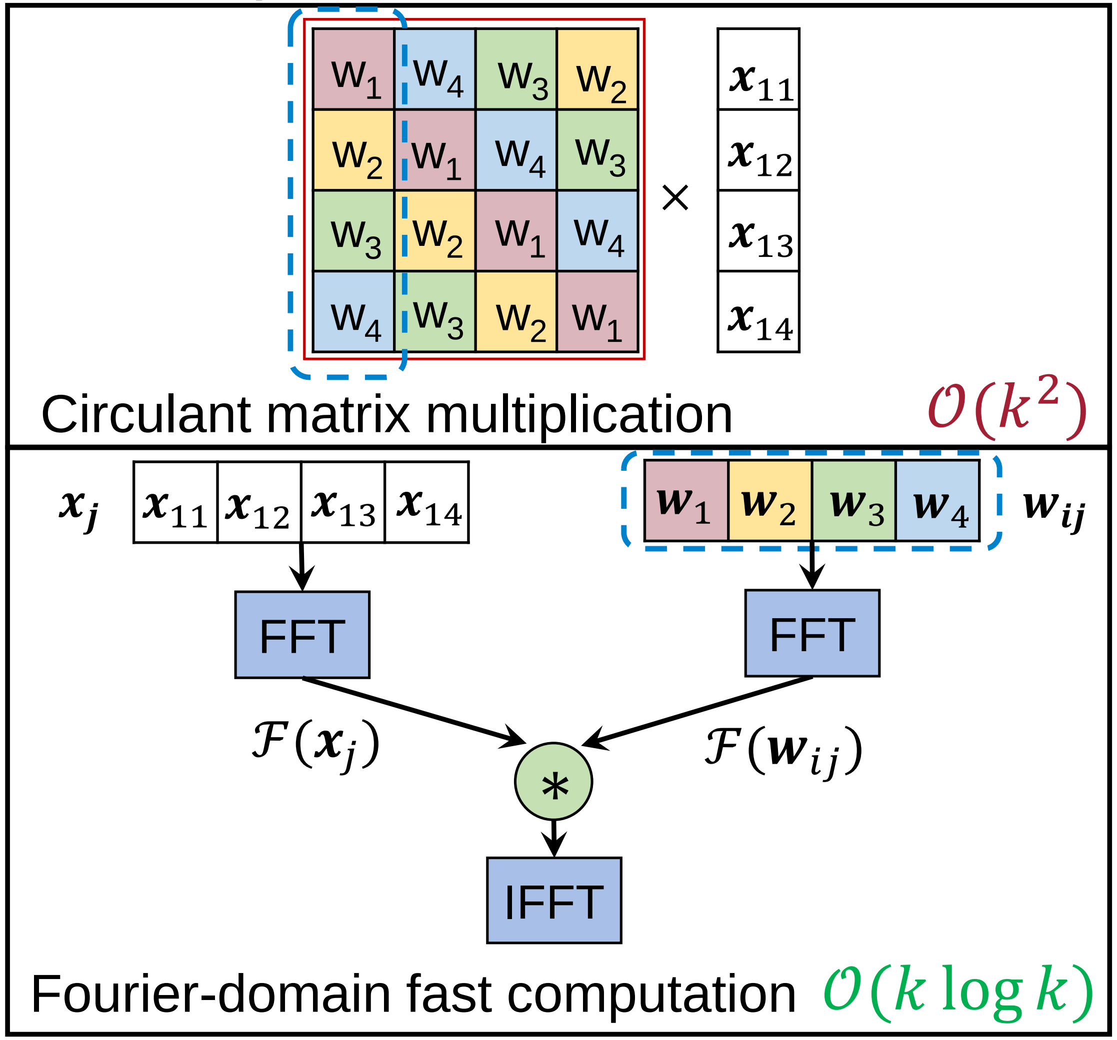
Butterfly-style Photonic Tensor Core

- Efficient circulant matrix multiplication in Fourier domain



Block-structured matrix

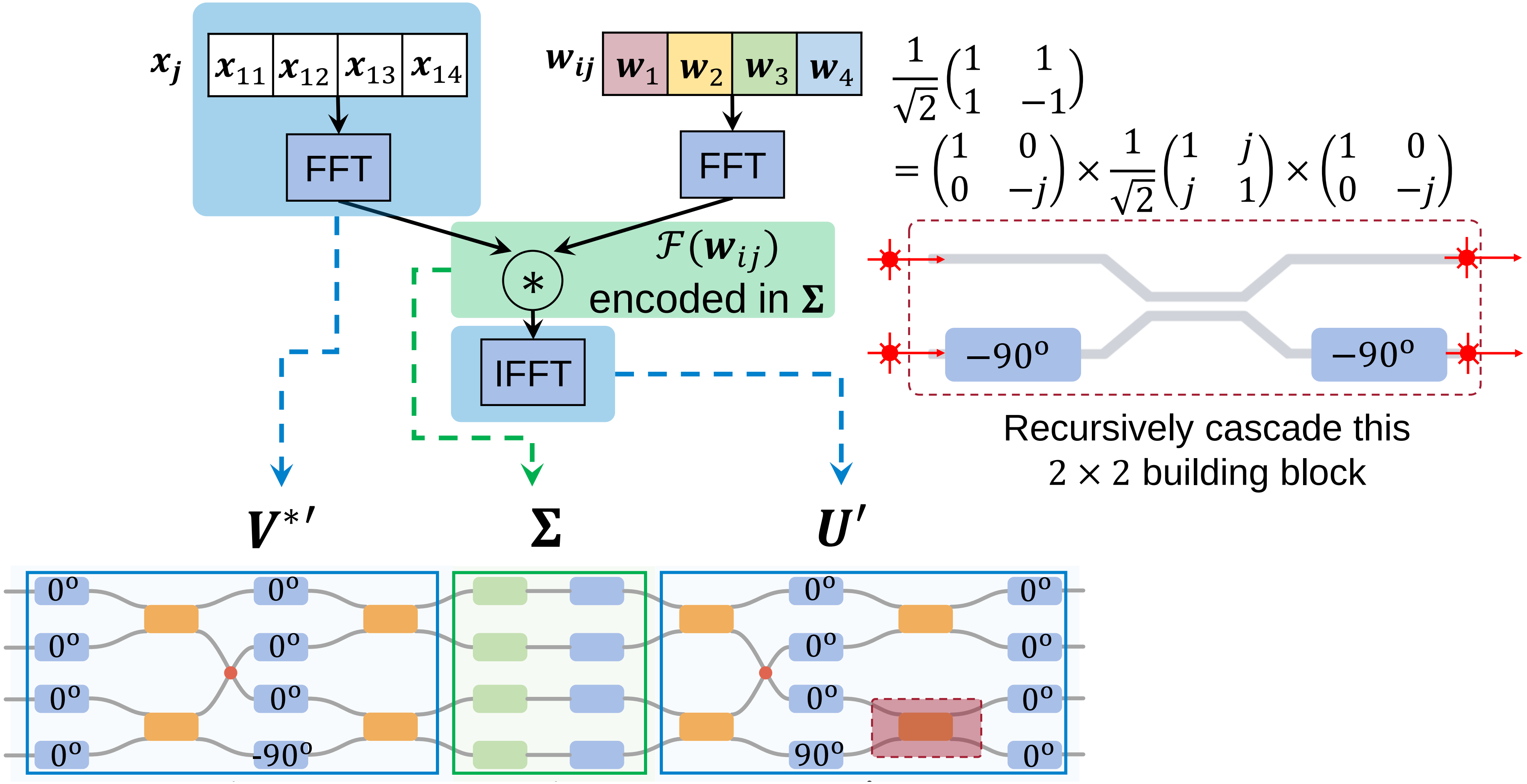
$$\boldsymbol{y_i} = \sum_j \boldsymbol{W_{ij}} \boldsymbol{x_j}$$



$$\boldsymbol{W}_{ij} \times \boldsymbol{x_j}$$

$$\mathcal{F}^{-1}\left(\mathcal{F}(\boldsymbol{w_{ij}}) \odot \mathcal{F}(\boldsymbol{x_j})\right)$$

Butterfly Photonic Mesh for Circulant MVM

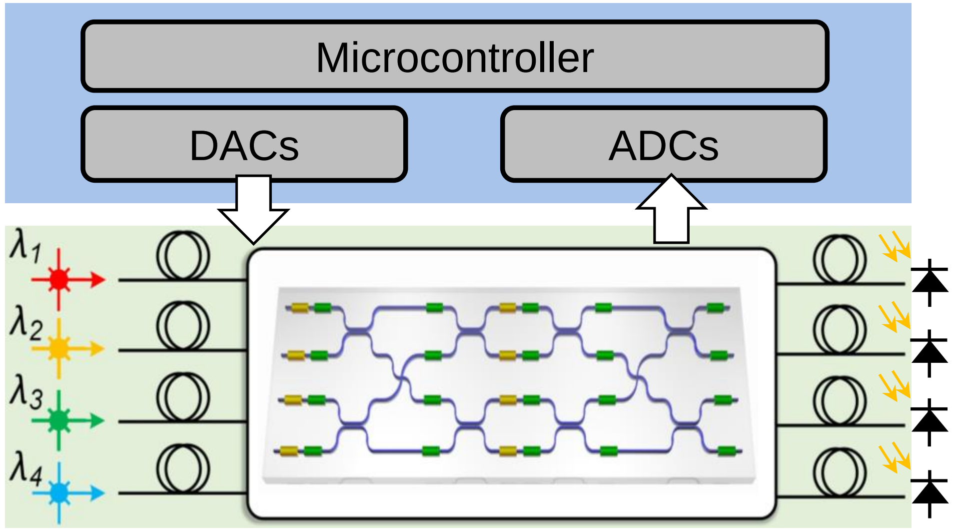


Photonic Neural Chip Tapeout & Demonstration

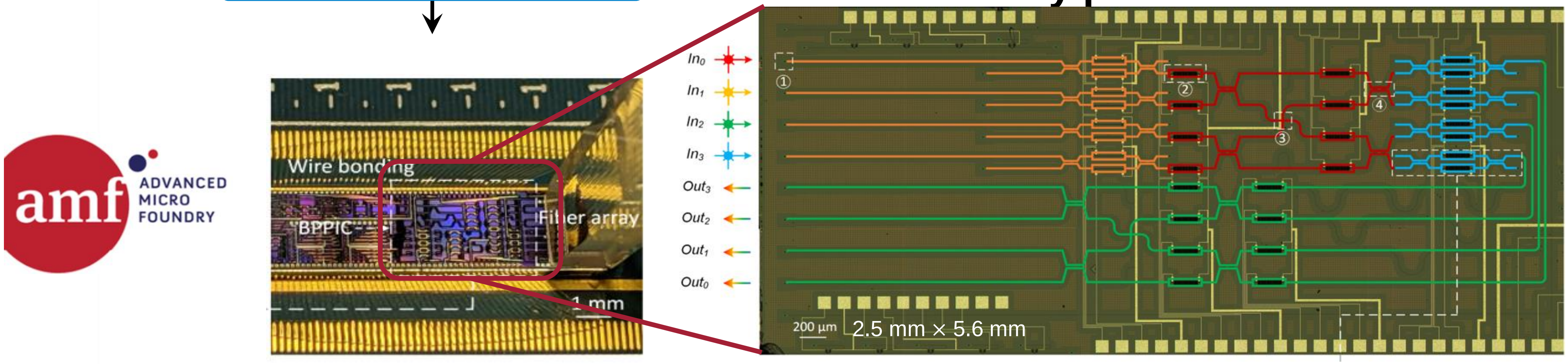
Dev. Flow

Design → Layout → Simulation → Tape-out by foundry → Testing → Model Training → Evaluation on ML Tasks

Electronic-photonic computing platform



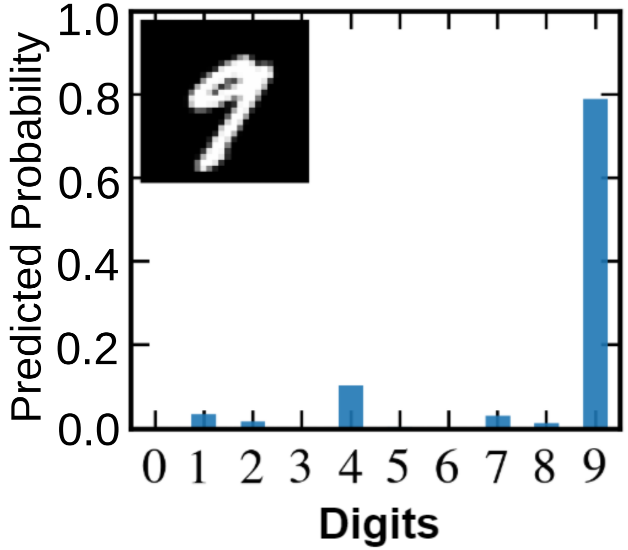
4×4 butterfly photonic tensor core



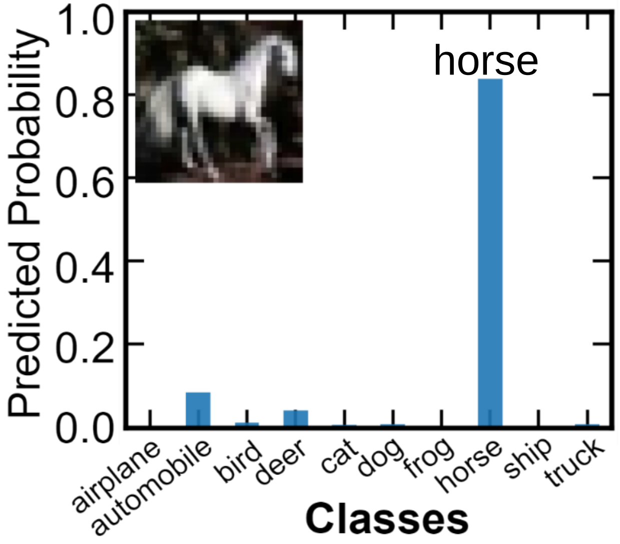
C. Feng*, J. Gu* (co-first), H. Zhu, Z. Ying, Z. Zhao, D.Z. Pan, R.T. Chen, "A compact butterfly-style silicon photonic-electronic neural chip for hardware-efficient deep learning", **ACS Photonics**, Nov. 30, 2022.

Evaluate on ML Tasks & Efficiency

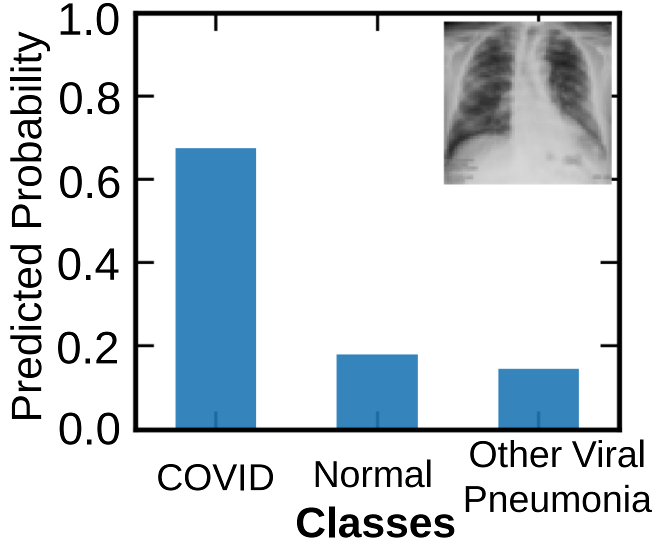
>94% accuracy
2-layer CNN (1.6k #params)
MNIST
3-bit weight resolution
Fixed butterfly transform



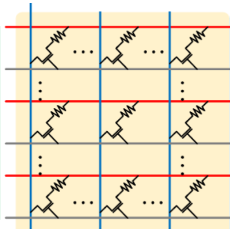
>85% accuracy
ResNet-20 (0.27M #param)
CIFAR-10
3-bit weight resolution
Fixed butterfly transform



96.5% accuracy
VGG8 (4M #params)
COVID Chest X-ray
3-bit weight resolution
Fixed butterfly transform



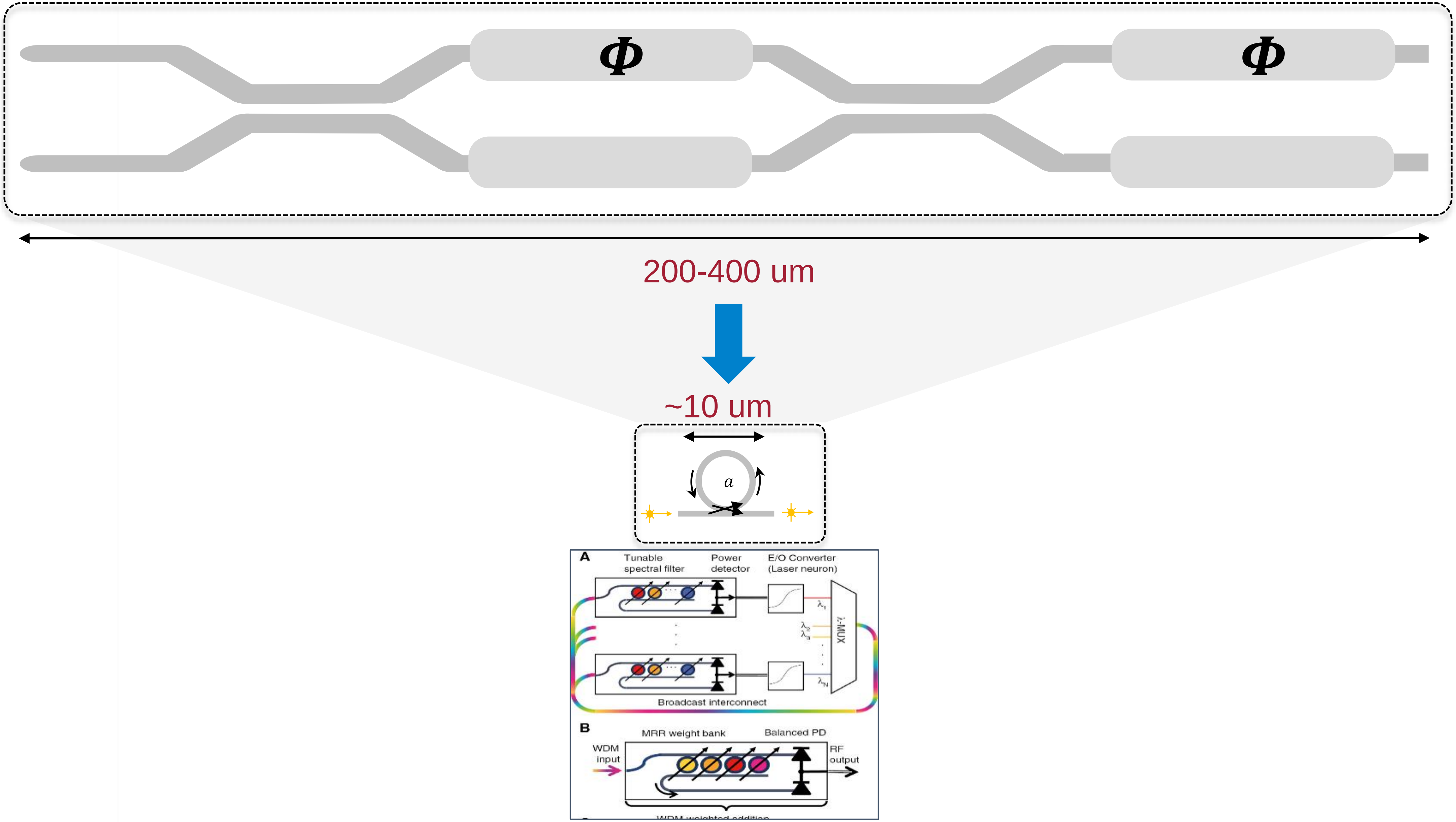
225 TOPS/mm²
2-4× smaller area & **5-13×** less optical delay than MZI-ONN [Nat. Photon.'17]



*Reference accuracy **85.6%**
ResNet-20 CIFAR-10
ReRAM Crossbar **4-bit** weight (GEMM)
[Wan *et al.*, **Nature**, Aug. 2022]

Outline of Tutorial I

- Introduction to Optical Computing
- Design and Demonstration of Electronic-Photonic Digital Computing
- Analog Photonic Computing for Optical Neural Networks
    - Coherent Photonic Tensor Core
    - Incoherent Photonic Tensor Core

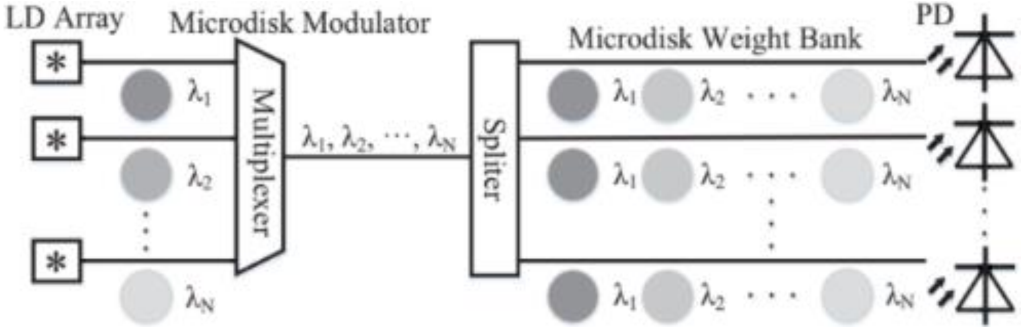

Φ
Φ
200-400 um
~10 um
a
A
Tunable spectral filter
Power detector
E/O Converter (Laser neuron)
λ-MUX
Broadcast interconnect
B
MRR weight bank
Balanced PD
WDM input
RF output

Incoherent ONN Architectures

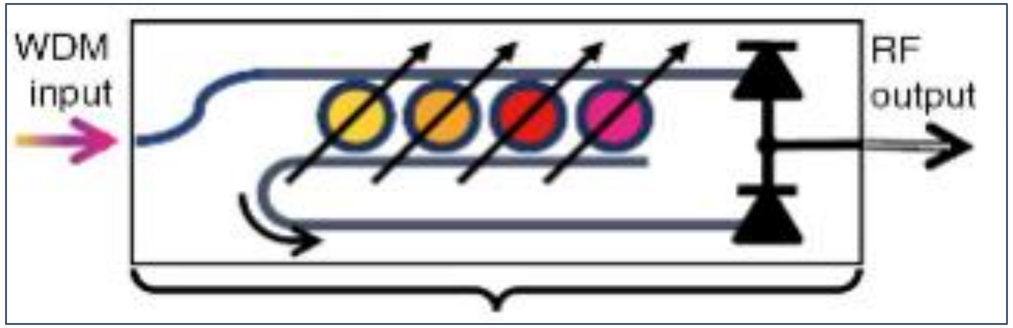
- Encoding: $|x|$
- Computing: Multi-wavelength modulation + photodetection
- Microring resonator (MRR) weight bank
- $y = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n$
- All-pass MRR weight bank
  - $w_i = a_i \in [0, 1]$
- Add-drop MRR weight bank
  - $w_i = a_i - (1 - a_i) = 2a_i - 1 \in [-1, 1]$

- Compact in size
- Can we do it better?
  - Bottleneck by 1 Op/device

All-pass MRR weight bank



Add-drop MRR weight bank



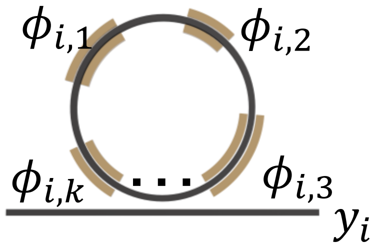
Customized Incoherent ONN Architectures

- ***Multi-operand*** optical neuron (MOON):
  - Single-device to implement vector-vector multiplications (beyond 1 OP/device)
- Built-in non-linear transfer function $T(\cdot)$:

$$y_i = T\left(\Sigma_j\left(\phi_j(V_j)\right)\right)$$

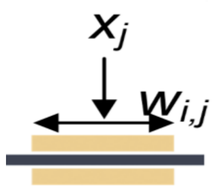
$$V_j = w_j \cdot x_j \text{ or } \phi_j = w_j \cdot x_j \ \ldots$$

$k \times$ higher compute density at the same cost

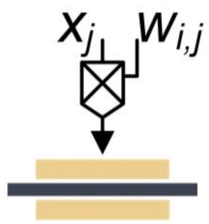


(a) Multi-operand microring

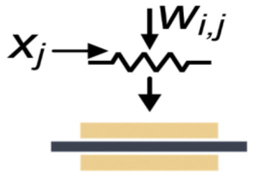
- Weight (w) and input (x) encoding:



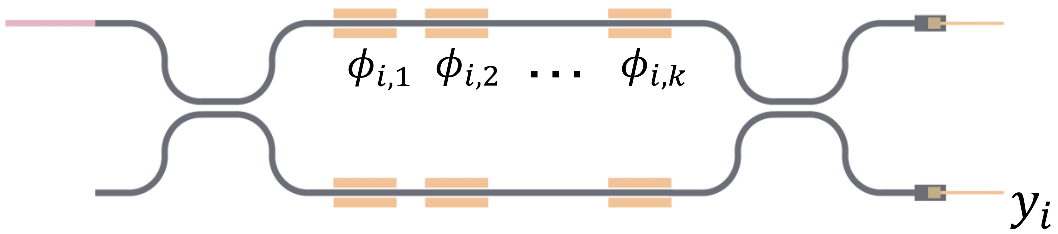
Fixed weight



Map product



Programmable weight



(b) Multi-operand MZI

(MOON) Multi-Operand Ring Resonators

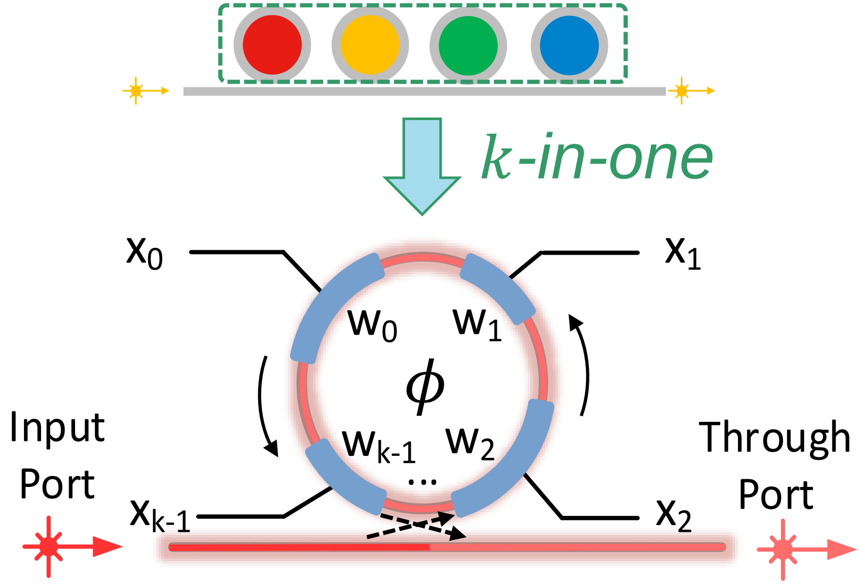
- **MORR**: k-segment controllers on one micro-ring
- Single-device length-k vector dot-product

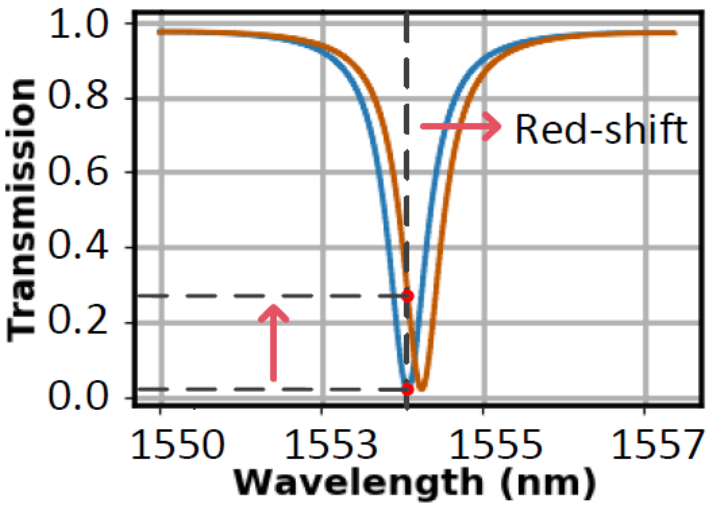
  Round-trip phase: $\phi \propto \sum_{i=0}^{k-1} w_i x_i^2$

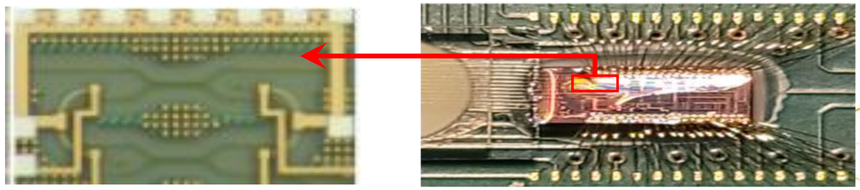
- Built-in nonlinearity
  - Half-Tanh-like nonlinear activation $f(\cdot) \in (0,1)$
  - Tunable smoothness (r, a)

$$f(\phi) = \left|\frac{r - a\, e^{-j\phi}}{1 - ra\, e^{-j\phi}}\right|^2$$

$$OUT = f(\phi) \cdot in \propto f\left(\sum_{i=0}^{k-1} w_i x_i^2\right) \cdot IN$$







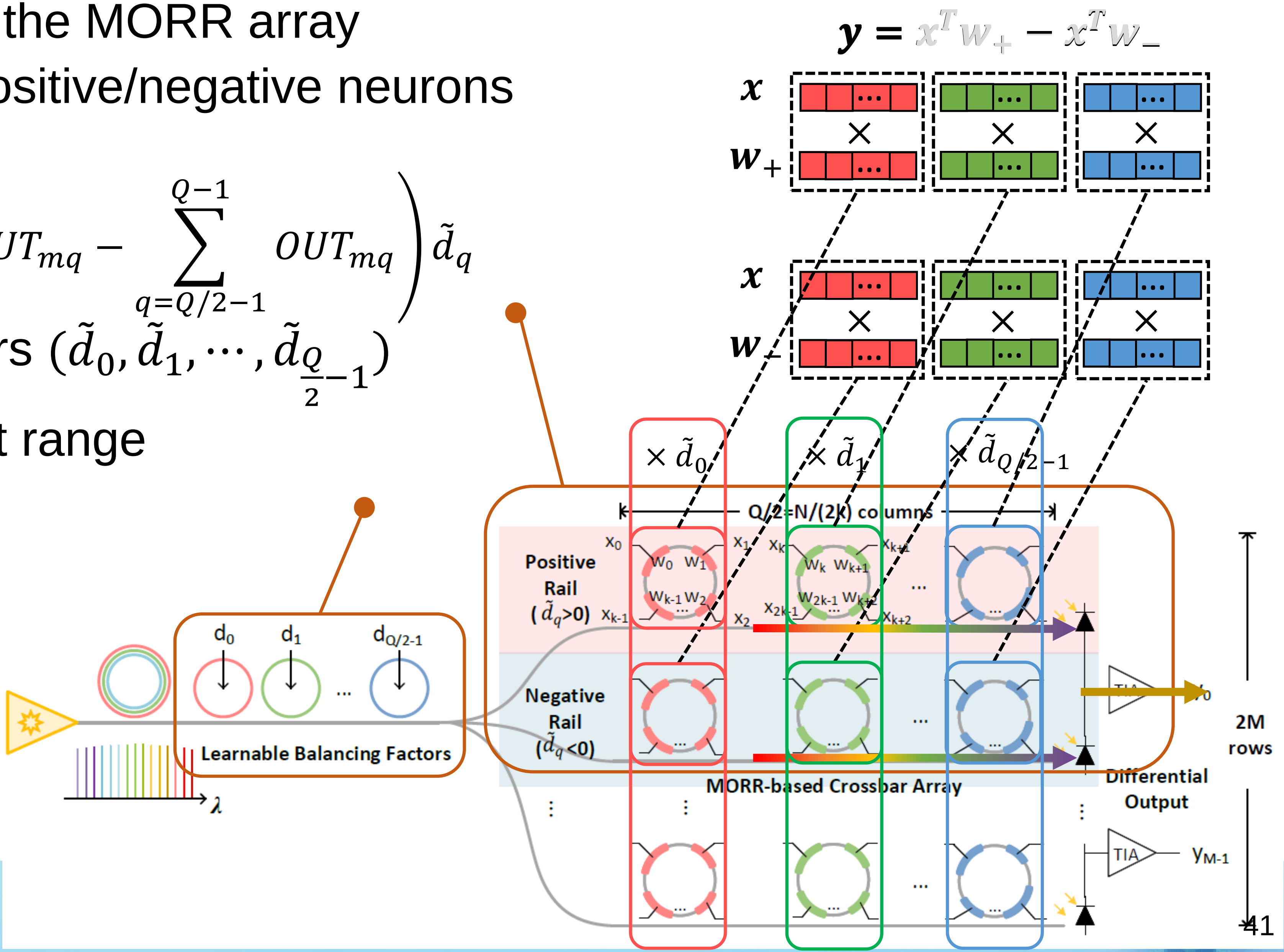
AIM photonics

MORR-based ONN: SqueezeLight [Gu+, DATE'21, TCAD'22]

- ***MatMul + Nonlinearity*** in the MORR array
- Differential rails support positive/negative neurons

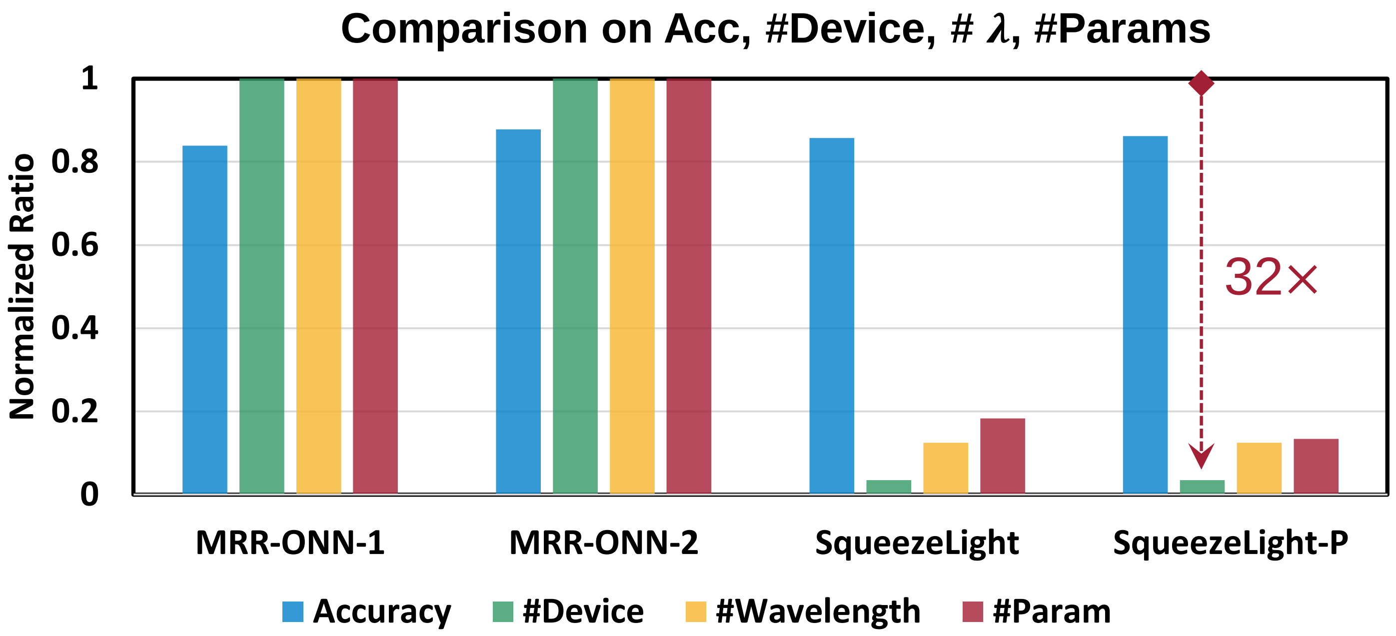
$$y_m = \left(\sum_{q=0}^{Q/2-1} OUT_{mq} - \sum_{q=Q/2-1}^{Q-1} OUT_{mq} \right) \tilde{d}_q$$

- Learnable balancing factors $(\tilde{d}_0, \tilde{d}_1, \cdots, \tilde{d}_{\frac{Q}{2}-1})$
  - Adaptive MORR output range
  - Enhanced expressivity



Cross-layer Scalability Evaluation

- Compare with SoTA MRR-ONNs on MNIST, FMNIST, CIFAR-10
- **23×-32×** less device usage
- **8×** fewer wavelength usage
- ***MORR array*** vs MRR array
  - w/ same area budget
  - **5.3×** higher TOPS/mm^2
  - **9.8×** higher TOPS/W
  - **63.5%** system energy reduction

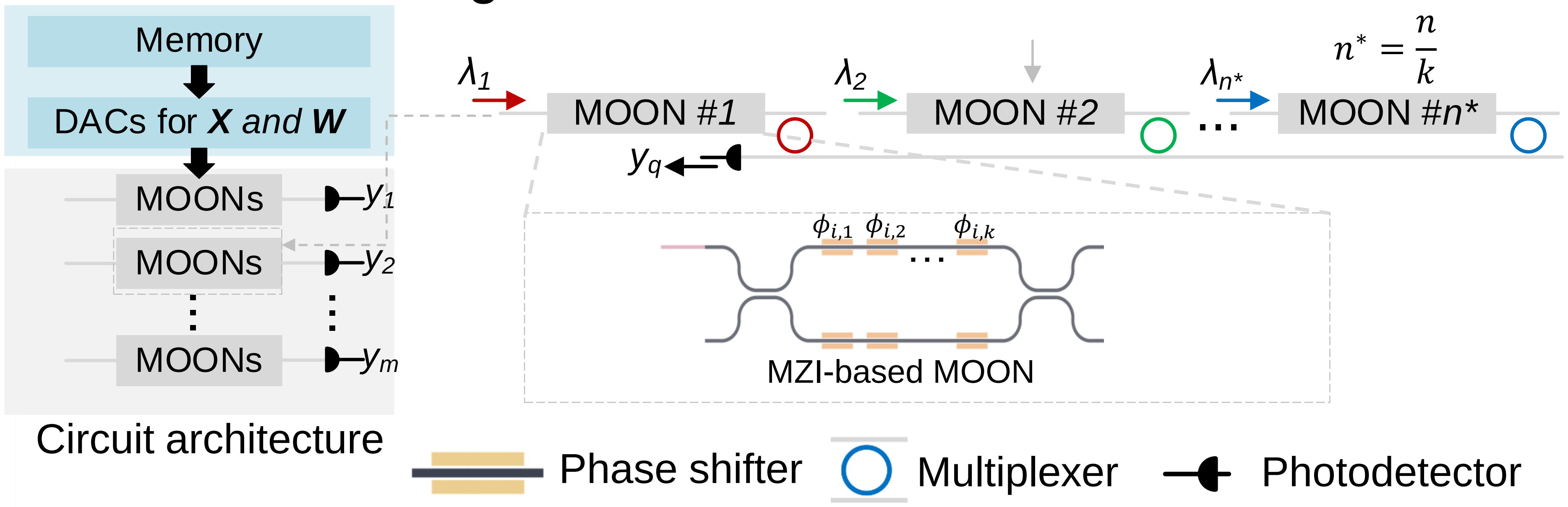


- Good expressivity & training scalability
- Robust to crosstalk/noises with special robustness optimization

MRR-ONN-1 [Liu+, DATE'2019]
MRR-ONN-2 [Tait+, SciRep 2017]

(MOON) Multi-Operand MZI

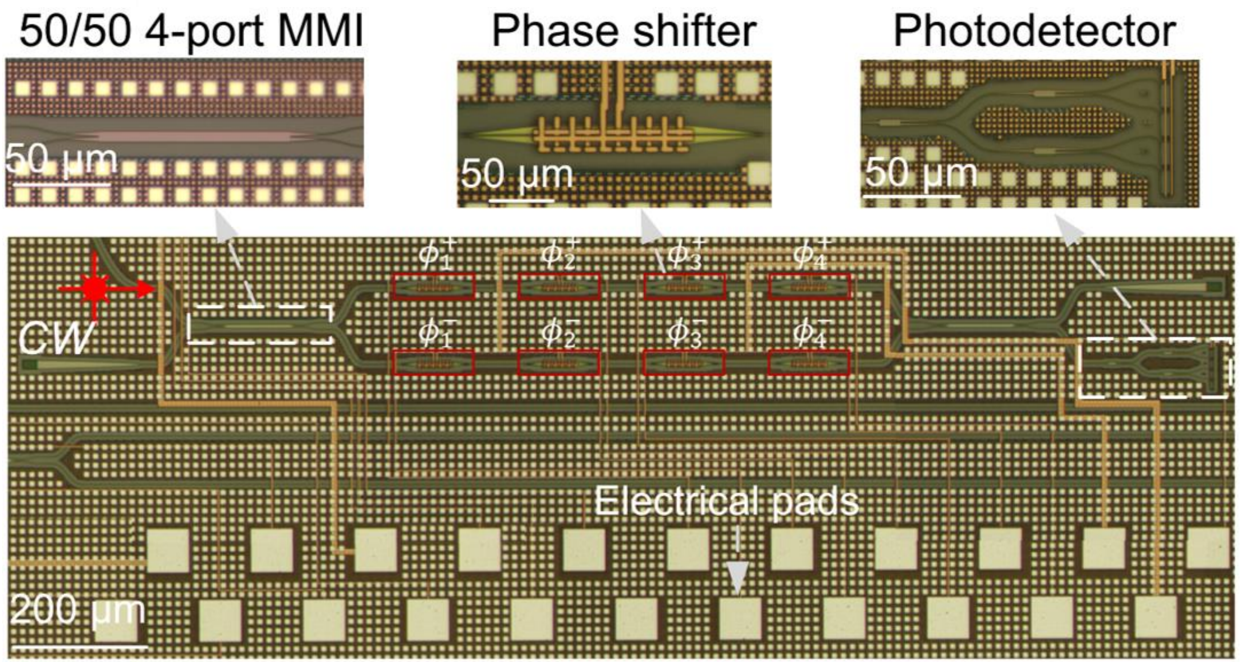
- Partition MZI controllers into k segments
- Dot-product + nonlinearity: $y = \cos(\sum_i w_i x_i)$
- Scale up to larger vectors with WDM
- Fewer cascaded device → lower insertion loss and delay
- Same power/area as a single MZI

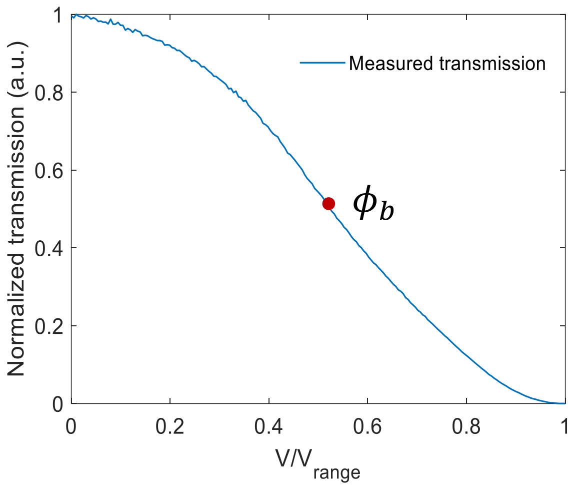


# of operands k depends on controlling/fabrication precision and chip layout: 4/8/16/…

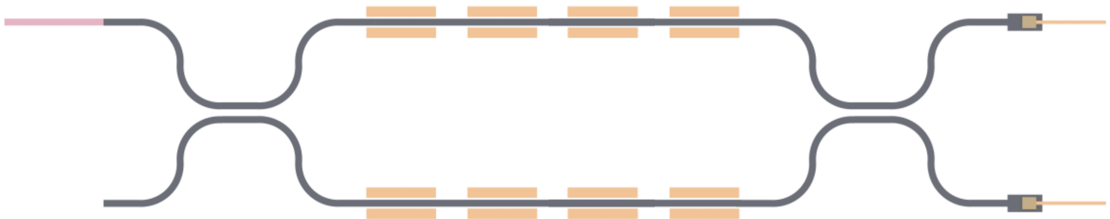
[Feng C. *et al.*, under submission]

MOMZI Chip Layout (4-op MOMZI)





Biased in the middle
Full-range output



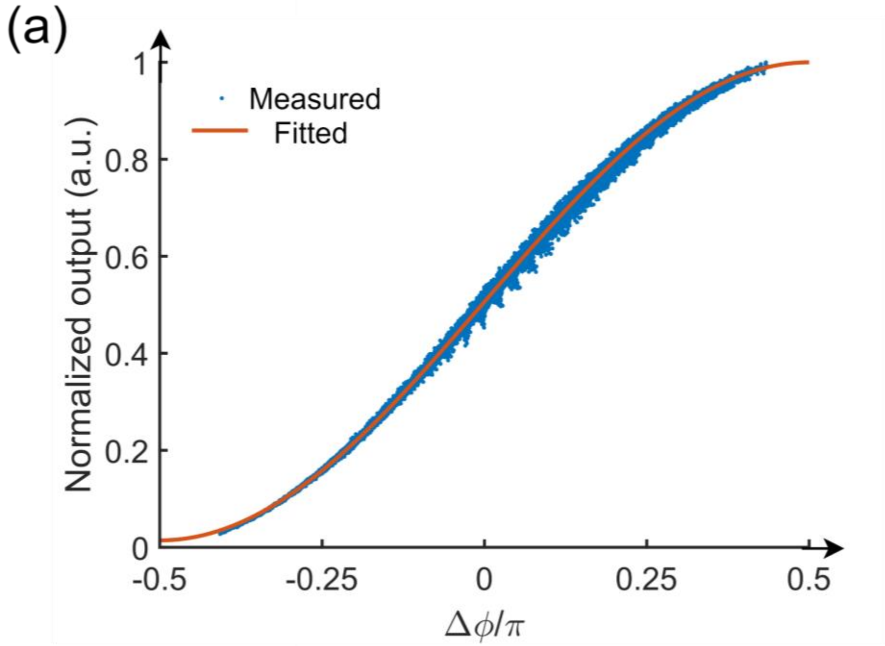
$$y = \frac{\mathrm{I_{in}}}{2} \Sigma_i \cos(\Sigma w_i x_i + \phi_b)$$

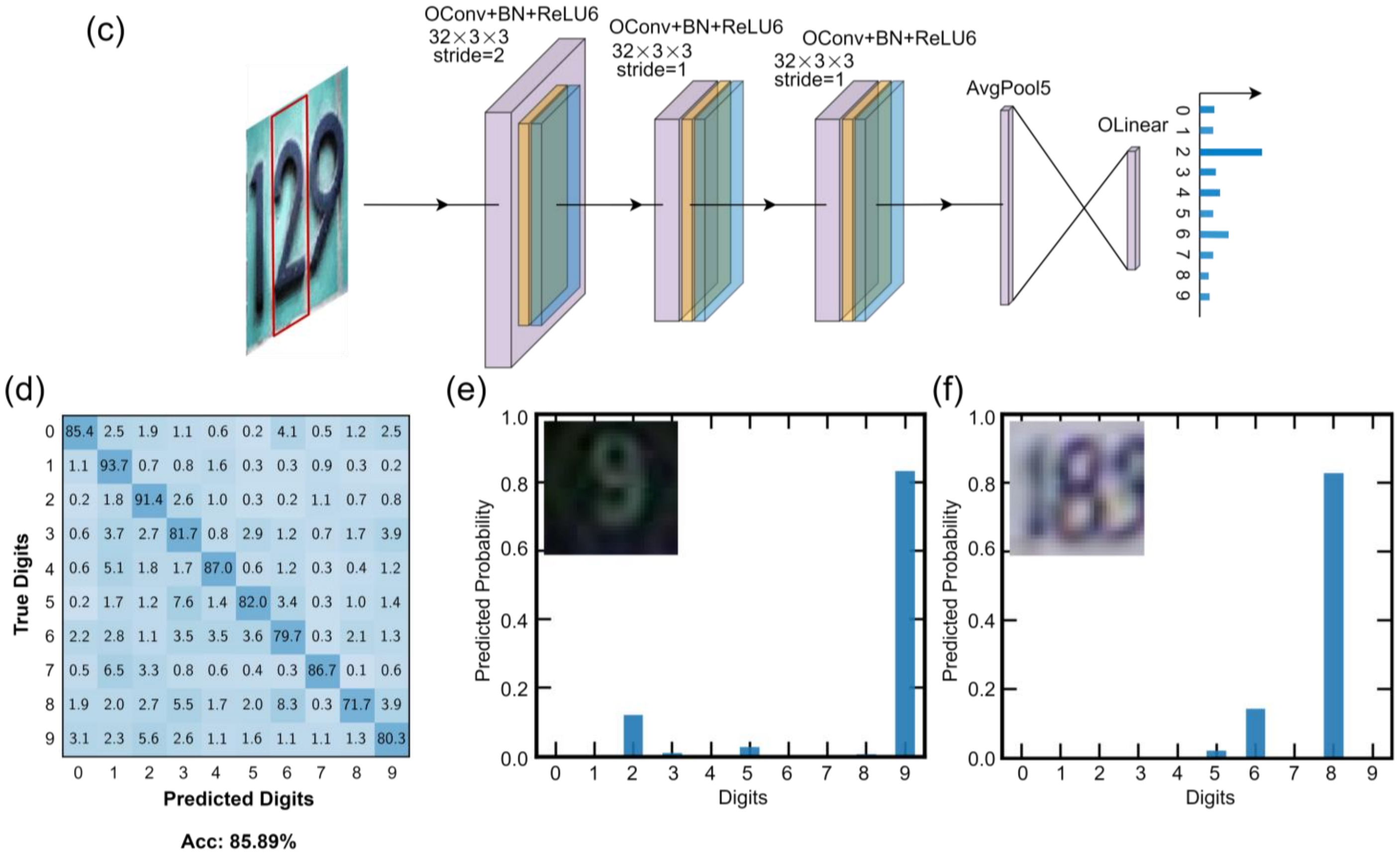
$$\phi_b \cong \frac{\pi}{2}$$

[Feng C. *et al.*, under submission]

Evaluation Results

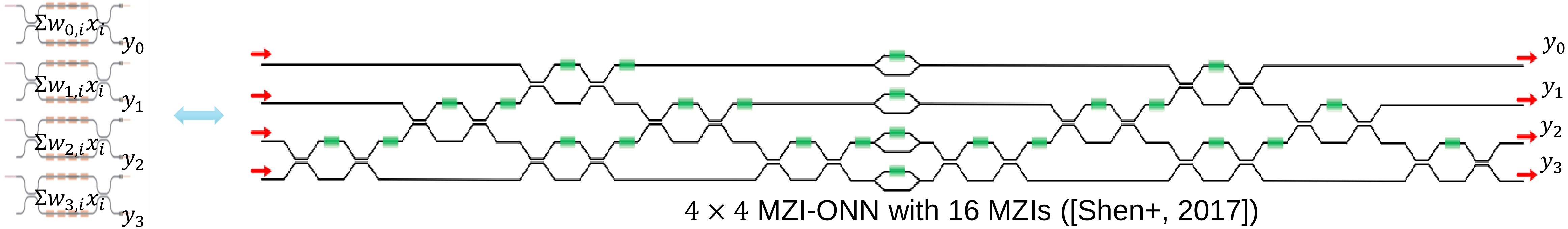
- Robust output with small noises (~0.5%)
- ~86% measured acc on 4-layer CNN SVHN
- 4-bit voltage control precision





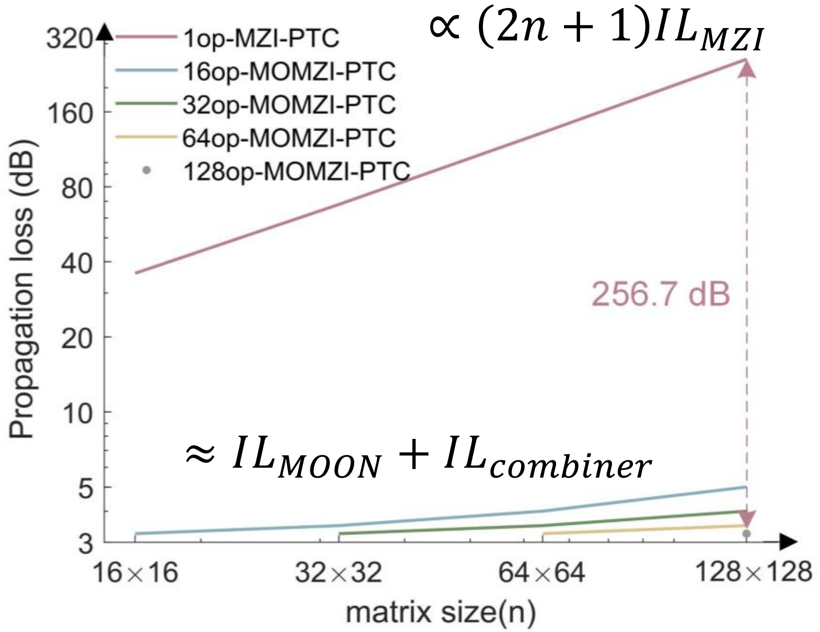
Performance Analysis

- >100 dB smaller insertion loss and 7.2× smaller footprint

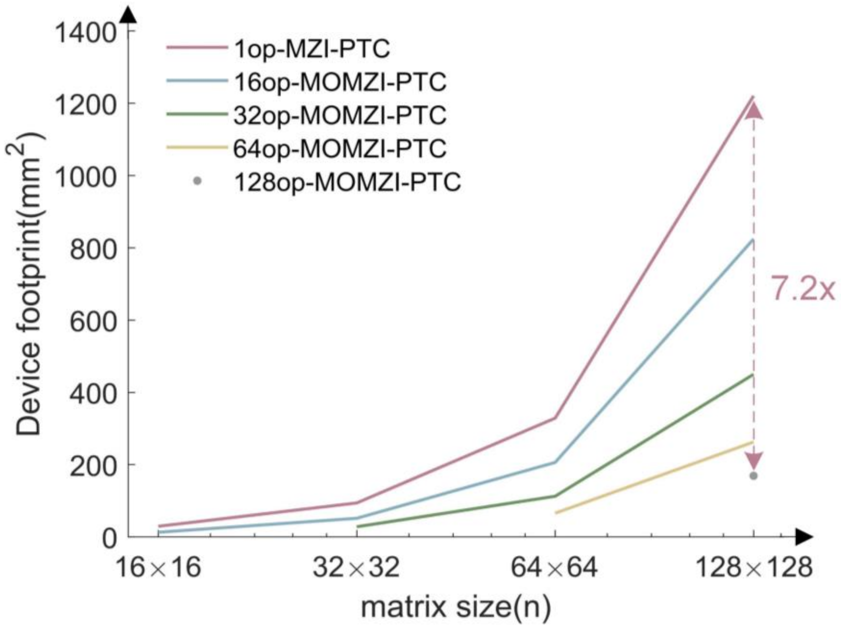


4×4 MOMZI-ONN

4×4 MZI-ONN with 16 MZIs ([Shen+, 2017])



(a) Propagation loss comparison



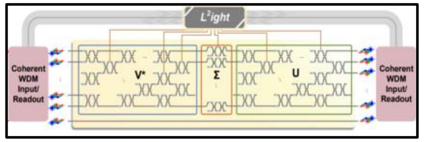
(b) Footprint comparison

Results based on AIM photonics PDK, For MZI-ONN, we use thermo-optical MZI switch for weight programming

Open-Source Photonic AI: TorchONN

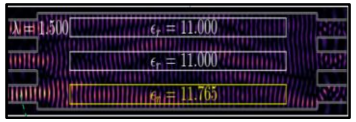
Torch ONN

On-chip Train

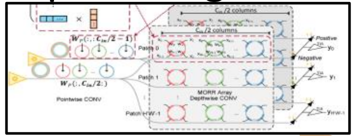


MORR Tape-out

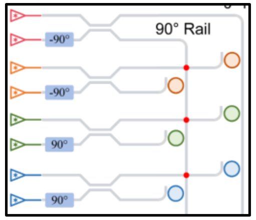
NeurOLight



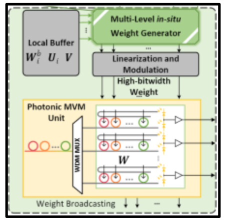
SqueezeLight-v2



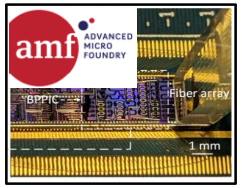
O2NN



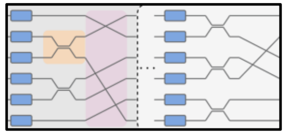
Mem-Eff ONN



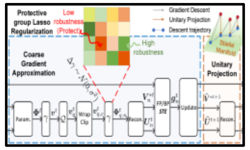
FFTON Tape-out



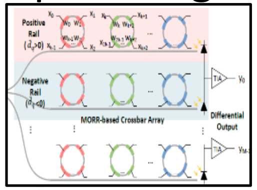
Auto-ONN



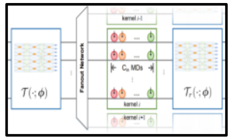
ROQ

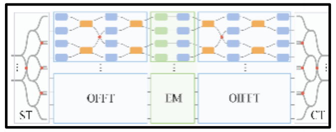


SqueezeLight



FFT-ONN-v2





FFT-ONN

2019 2020 2021 2022 2023 Future

Gen-1
Standard Device
Customized Circuit
Noise-aware Train

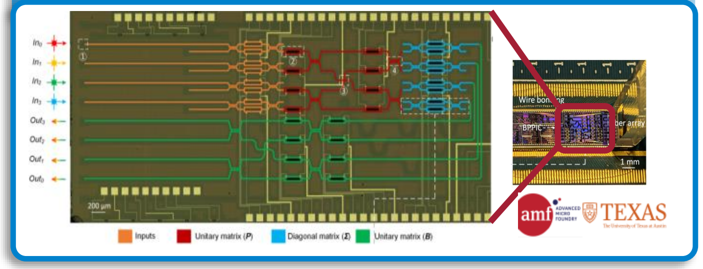
Gen-2
Support More Applications
(Dynamic MatMul; On-chip Train...)

Gen-3
Customized Devices
Efficient Architecture

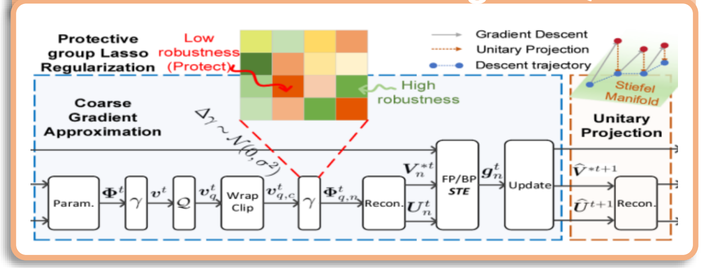
Gen-4
AI-based Automated ONN Design

Optics for AI
AI for Optics

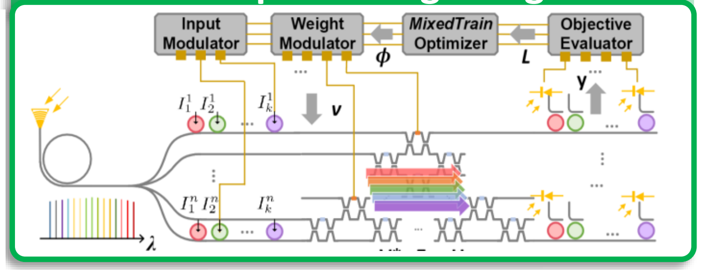
ONN Architecture: FFT-ONN



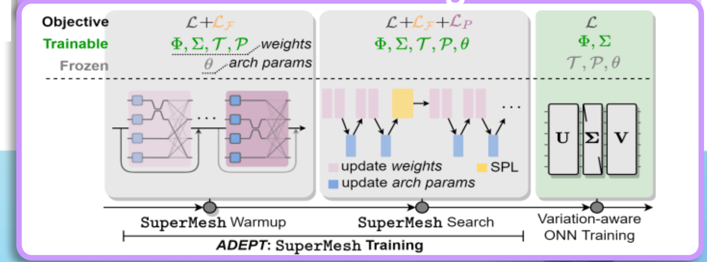
Robust ONN Training: ROQ



On-Chip Learning: L²ight



Automated PIC Design: ADEPT



Photonic AI Library TorchONN

- Construction: customized optical Conv/Linear layers
  - Modeling of various devices
    - PCM, MZI, MRR, MORR, …
  - Support various tensor core designs
    - MRR weight bank / MORR / MZI / FFT array…
    - CUDA backend for customized operators…
- Mapping: convert from electrical to optical
  - Decomposition or optimization-based map
- Co-design infrastructure
  - Device quantization & QAT
  - Noise injection & NAT
  - Circuit pruning & PAT
  - On-chip training support
    - Zeroth-order optimization
  - Circuit topology search (*SuperMesh*)

Torch ONN

A PyTorch Library for Photonic Integrated Circuit Simulation and Photonic AI Computing

```python
import torchonn as onn
from torchonn.models import ONNBaseModel

class ONNModel(ONNBaseModel):
    def __init__(self, device=torch.device("cuda:0")):
        super().__init__(device=device)
        self.conv = onn.layers.MZIBlockConv2d(
            in_channels=1,
            out_channels=8,
            kernel_size=3,
            stride=1,
            padding=1,
            dilation=1,
            bias=True,
            miniblock=4,
            mode="usv",
            decompose_alg="clements",
            photodetect=True,
            device=device,
        )
```

..
__init__.py
base_layer.py
fftonn_conv2d.py
fftonn_linear.py
morr_conv2d.py
morr_linear.py
mzi_conv2d.py
mzi_linear.py
pcm_conv2d.py
pcm_linear.py
super_conv2d.py
super_linear.py
super_mesh.py

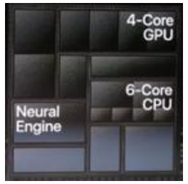
Tutorial II:
LightRidge: An End-to-end Agile Design Framework for Diffractive Optical Neural Networks

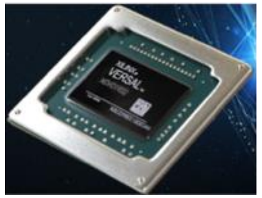
Yingjie Li, Ruiyang Chen, Minhan Lou, Berardi Sensale-Rodriguez, Weilu Gao, **Cunxi Yu**

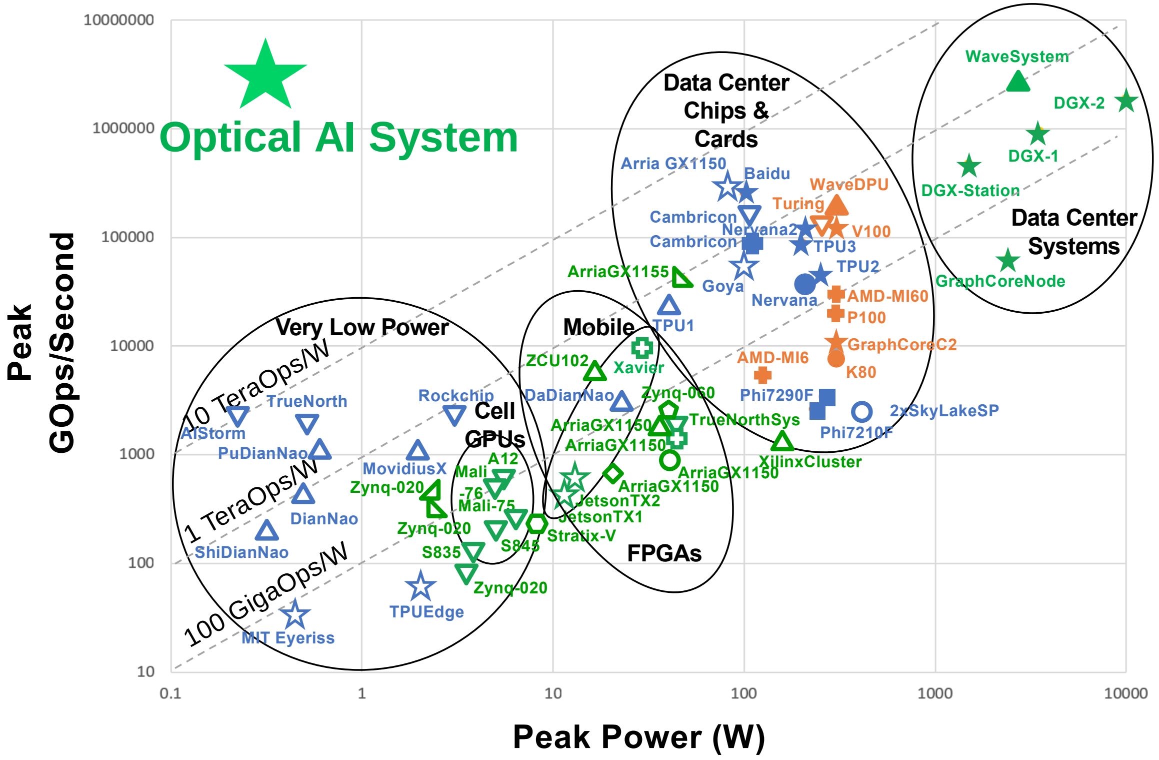
University of Utah

New Trends of Computing

- AI’s impacts in hardware system design
  - The raise of domain-specific computing
  - Beaten the trend of *Moore’s Law* (R.I.P Dr. Moore)
    - Doubling every **3.5 months** (18 months)



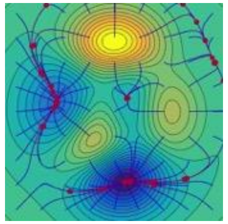
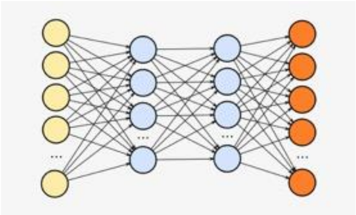




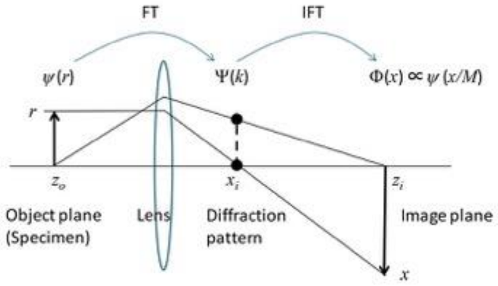
Reuther, A., Michaleas, P., Jones, M., Gadepally, V., Samsi, S., & Kepner, J. Survey and benchmarking of machine learning accelerators. HPEC 2019

Challenges in Optical AI System Design

- *A Computer Engineering* journey to Optical AI System
  - Challenge 1: Cross-disciplinary domain knowledge barriers

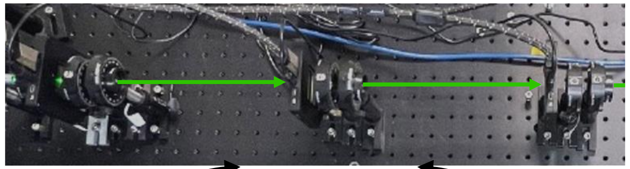
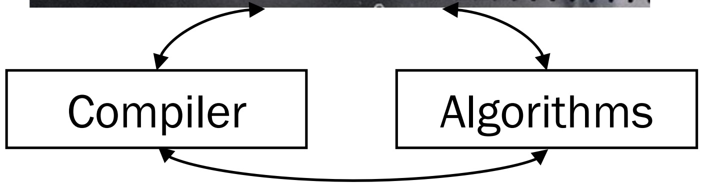
ML Algorithms & Neural Nets



Optics

```
#define NX 256
#define BATCH 10
#define RANK 1
...
{
    cufftHandle plan;
    cufftComplex *data;
    ...
    cudaMalloc((void**)&data, sizeof(cufftComplex)*NX*BATCH);
    cufftPlanMany(&plan, RANK, NX, &iembed, istride, idist,
        &oembed, ostride, odist, CUFFT_C2C, BATCH);
```

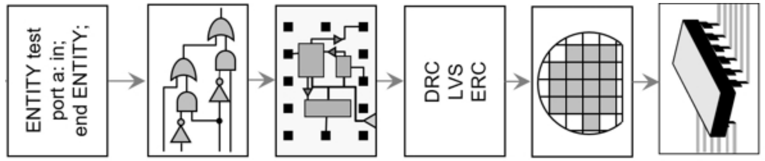
High Performance Programming

HW-SW Co-design

Challenges in Optical AI System Design

- *A Computer Engineering* journey to Optical AI System
  - **Challenge 1**: Cross-disciplinary domain knowledge barriers
  - **Challenge 2**: Lacks high-performance infrastructures for programming, modeling, training, exploration, fabrication, etc.




```
void test (ap_uint <4> in, ap_uint<4> out[3]) {
volatile ap_uint<4> temp[5];
temp[4]=in;
for(i=4;i>0;i--){
  #pragma HLS unroll
    temp[i-1] = temp[i];
}
out[2]=temp[0];
out[1]=temp[1];
out[0]=temp[2];
}
```

HLS-C/RTL & Compile

```
class MLP(nn.Module):
  '''
    Multilayer Perceptron.
  '''
  def __init__(self):
    super().__init__()
    self.layers = nn.Sequential(
      nn.Flatten(),
      nn.Linear(32 * 32 * 3, 64),
      nn.ReLU(),
```

PyTorch/TF

?

Optical Neural Networks

Data Preparation → NNs Modeling → Optimizations on DFGs → Deployment On HW

Challenges in Optical AI System Design

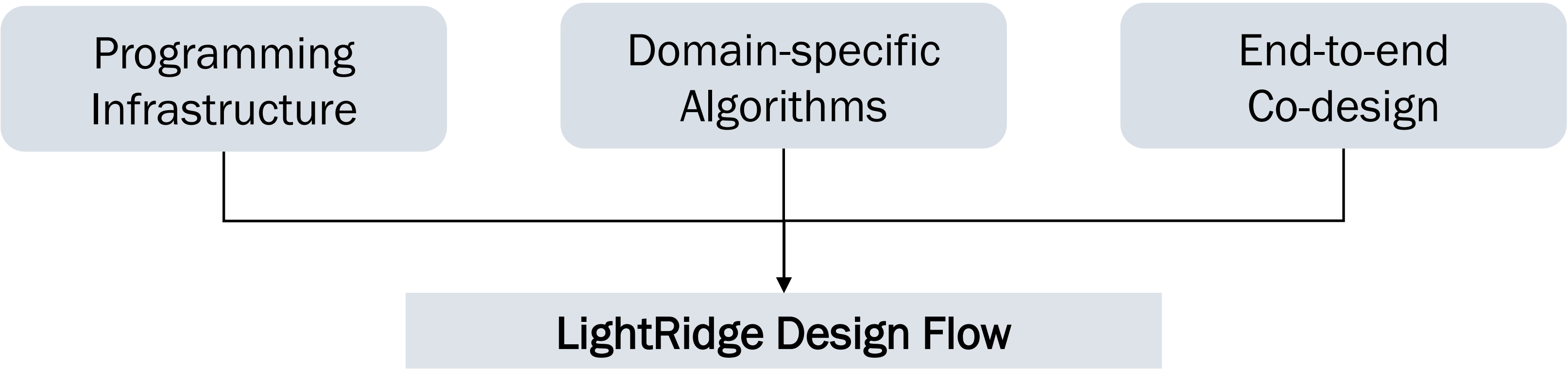
- *A Computer Engineering* journey to Optical AI System
  - **Challenge 1**: Cross-disciplinary domain knowledge barriers
  - **Challenge 2**: Lacks high-performance infrastructures for programming, modeling, training, exploration, fabrication, etc.
  - **Challenge 3**: Limited studies of physics-to-system co-design to enable seamless design-to-deployment

Numerical Emulation

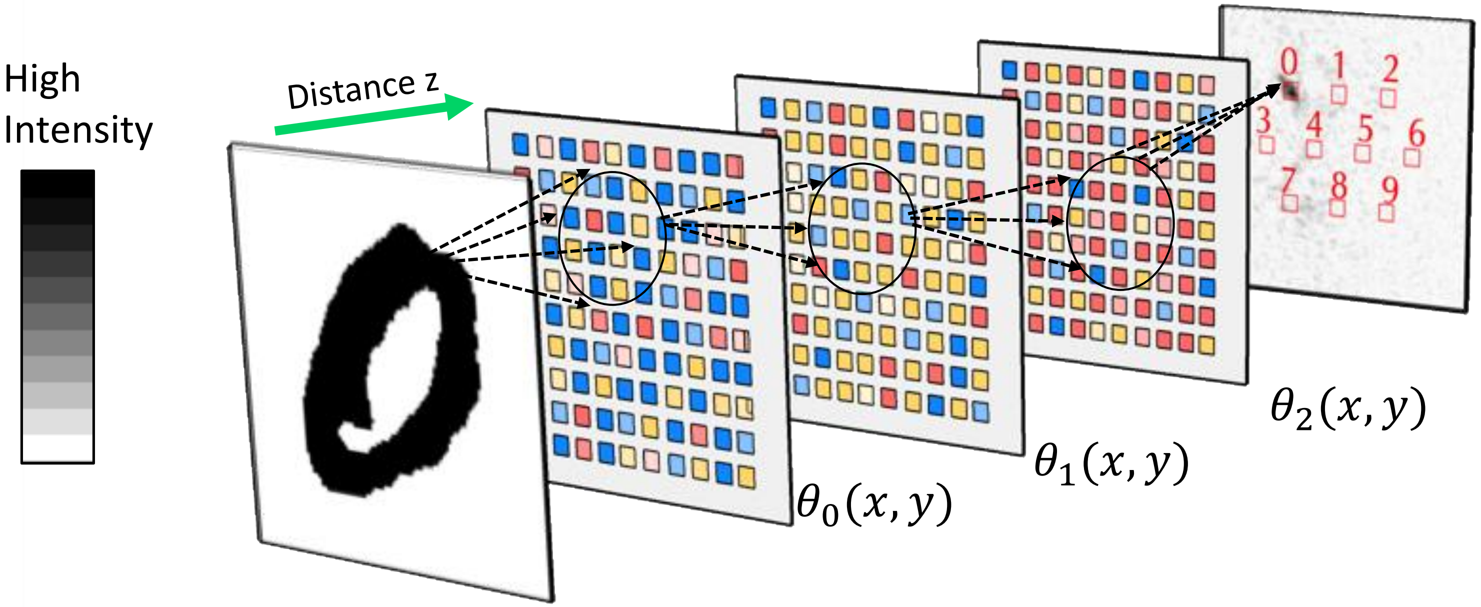
Physical Measurement

Challenges in Optical AI System Design

- *A Computer Engineering* journey to **Diffractive Optical Neural Networks**
  - **Challenge 1:** Cross-disciplinary domain knowledge barriers
  - **Challenge 2:** Lacks high-performance infrastructures for programming, modeling, training, exploration, fabrication, etc.
  - **Challenge 3**: Limited studies of physics-to-system co-design to enable seamless design-to-deployment



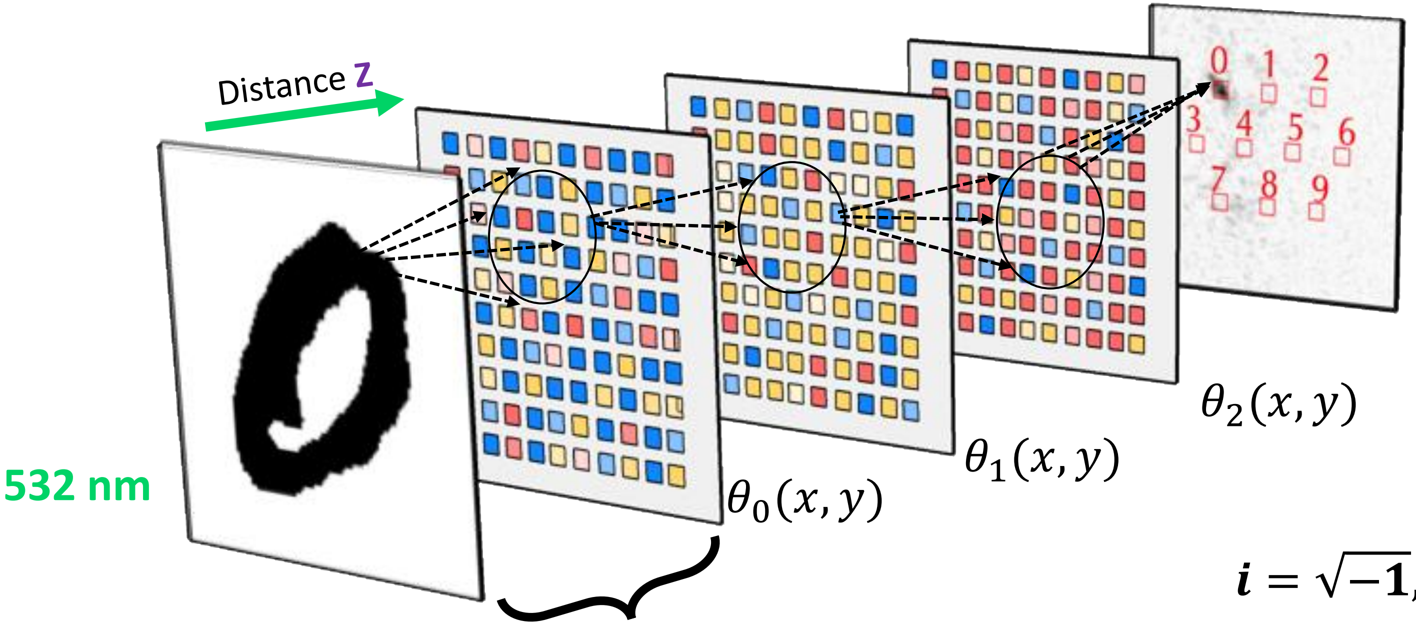
Background: Diffractive Neural Networks



	Conventional NNs	**Diffractive NNs**
Input	Images (Real)	**Light** (Complex)
Operator	Conv, Dense, Pool, …	**Light Diffraction** and **Phase Mod**
Propagation	Digital Computing	**Light Propagation** (Complex)
Output	Digital Output (Real)	**Light Intensity** (CPLEX-2-Real)

Lin, Xing, et al. "All-optical machine learning using diffractive deep neural networks." *Science* 361.6406 (2018): 1004-1008.

Background: Diffractive Neural Networks



$$i = \sqrt{-1}, k = \frac{2\pi}{\lambda}$$

① *Free-space Diffraction*

↓ Distance z

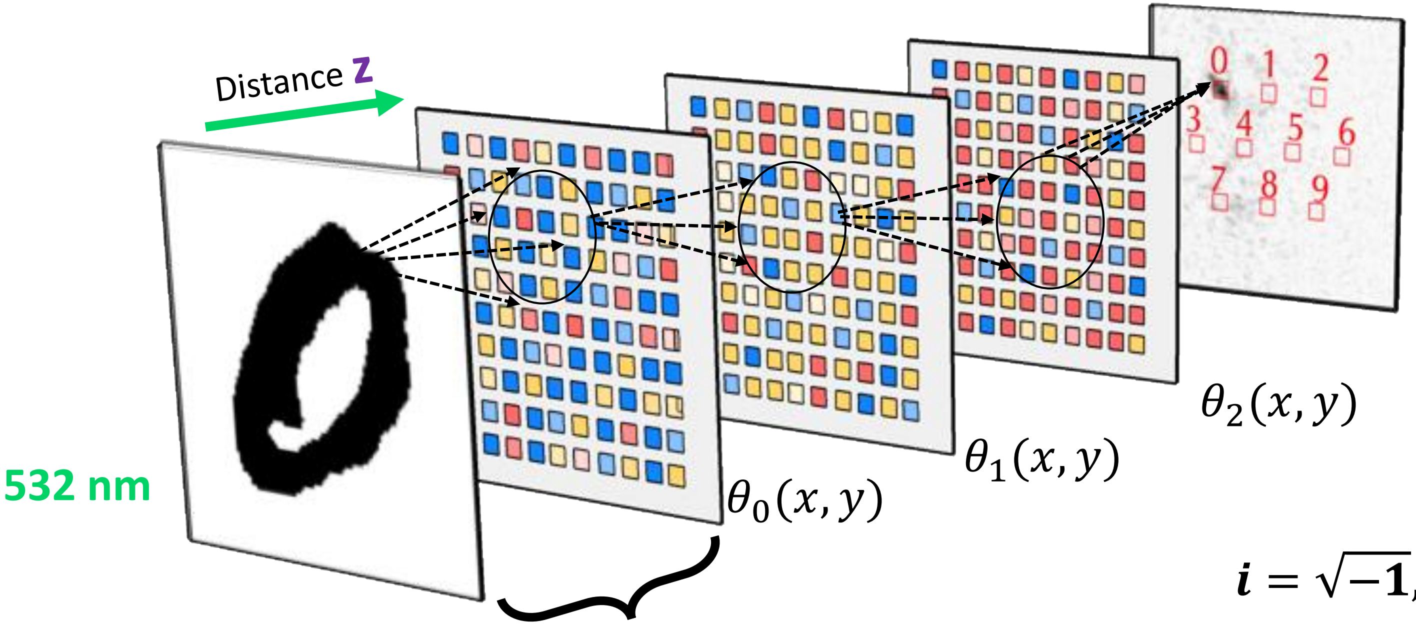
e.g., Fresnel approximation:

$$h(x,y,z) = \frac{\exp(ikz)}{i\lambda z}\exp\{\frac{ik}{2z}(x^2+y^2)\}$$

$$\text{DiffMod}\,(X_c(x,y),\theta_0) = \underbrace{\mathrm{iFFT_{2D}}\left(\mathrm{FFT_{2D}}(\mathrm{X_c}(x,y)) \times \mathrm{FFT_{2D}}(\mathrm{h}(x,y,z))\right)}_{\text{① Light Diffraction}}$$

Lin, Xing, et al. "All-optical machine learning using diffractive deep neural networks." *Science* 361.6406 (2018): 1004-1008.

Background: Diffractive Neural Networks



$$i = \sqrt{-1}, k = \frac{2\pi}{\lambda}$$

① *Free-space Diffraction*

↓ Distance z

② *Phase modulation*

↓ θ Trainable parameters

e.g., Fresnel approximation:

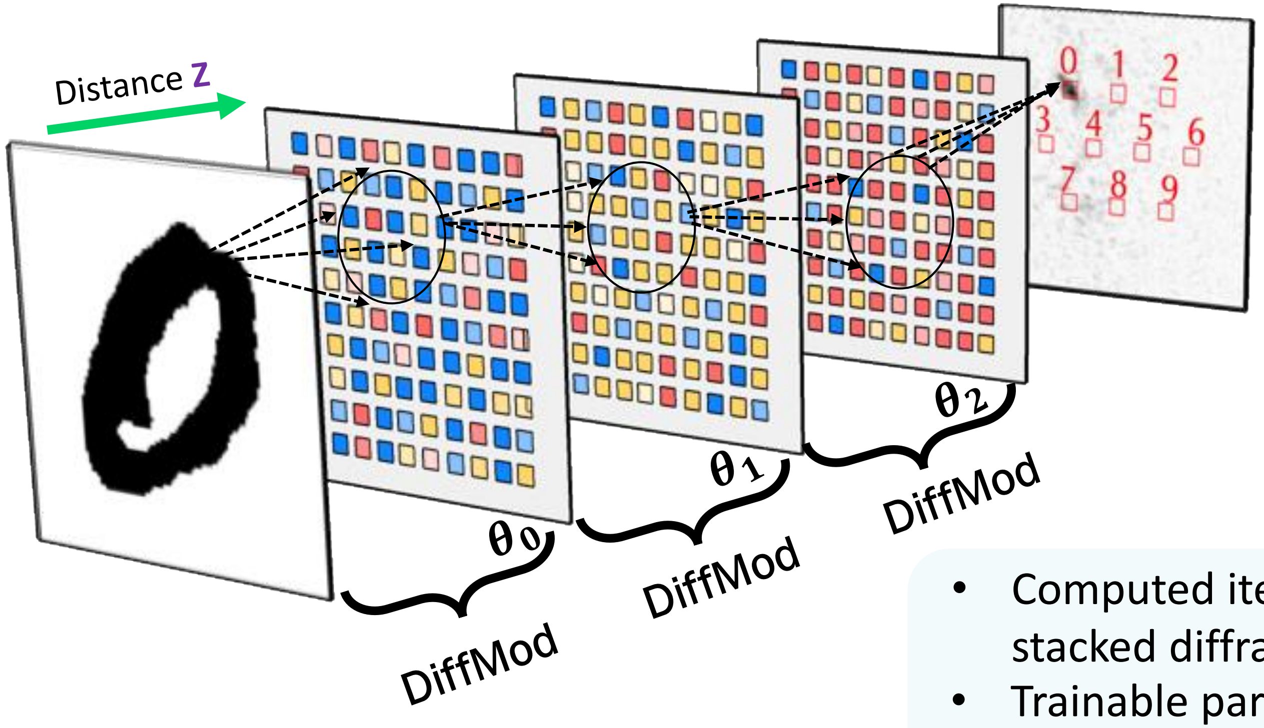
$$h(x, y, z) = \frac{\exp(ikz)}{i\lambda z} \exp\{\frac{ik}{2z}(x^2 + y^2)\}$$

DiffMod ($X_c(x, y)$, θ_0) = Light Diffraction $\times$ $\underbrace{(\cos\theta_0(x, y) + i \cdot \sin\theta_0(x, y))}_{\text{② Phase modulation}}$

↑ Complex MatMul

Lin, Xing, et al. "All-optical machine learning using diffractive deep neural networks." *Science* 361.6406 (2018): 1004-1008.

Background: Diffractive Neural Networks



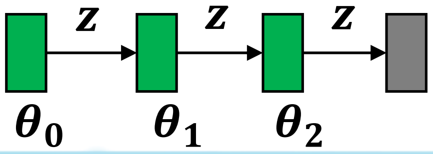
$$\mathbf{X_C} = \boldsymbol{A} + \boldsymbol{i} \cdot \boldsymbol{B}$$

$$\mathbf{I}(\mathbf{XC}) = \sqrt{\boldsymbol{A}^2 + \boldsymbol{B}^2}$$

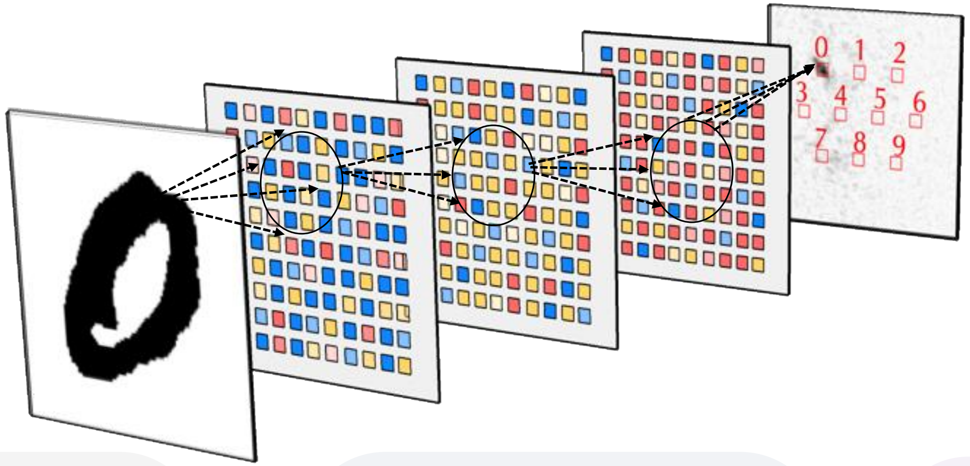
- Computed iteratively for all stacked diffractive layers
- Trainable parameters
  $\boldsymbol{\theta}=\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$

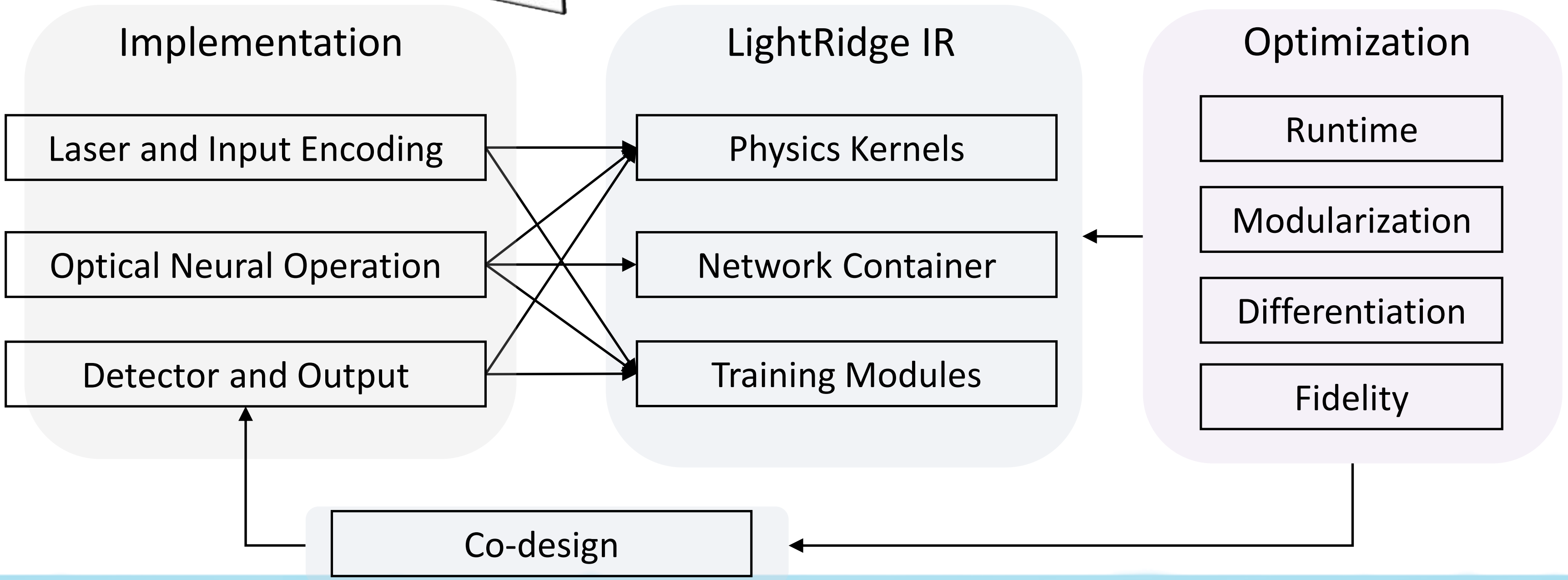
For example, 3-layer forward function:

$$\mathbf{I}(\mathbf{X_c}, \boldsymbol{\theta}) = \text{DiffMod}\,(\,\text{DiffMod}\,(\,\text{DiffMod}\,(\boldsymbol{X_c}(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{\theta}_0), \boldsymbol{\theta}_1\,), \boldsymbol{\theta}_2\,)$$



Overview of LightRidge Framework





Yingjie Li, Ruiyang Chen, Minhan Lou, Berardi Sensale-Rodriguez, Weilu Gao and **Cunxi Yu**. "*LightRidge: An Open-source Compiler Framework for Diffractive Optical ML Architectures.*" Workshop on Open-Source Computer Architecture Research (OSCAR) held in conjunction with ISCA (ISCA'49)

Overview of LightRidge Framework

Design Flow

Initialization

Model Definition

Training

Deployment

Compiler Code Blocks

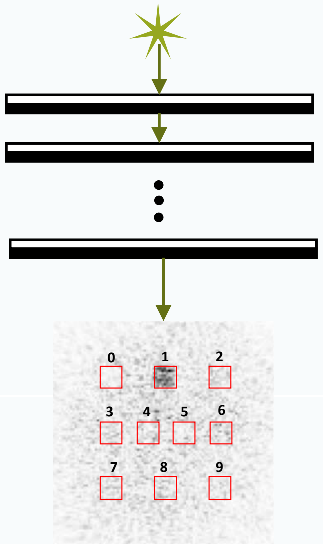
```
import lightridge as lr
lr.utils.data2cplex(DataLoader)
lr.init(DataLoader, wl, z)
```

```
# laser-specific definition
lr.laser(GaussBeam, wl, power)
# forward definition of DONNs
lr.model.sequential(
  lr.layers.DiffractLayer('Fresnel',…),
  lr.layers.DiffractLayer('Fresnel',…),
  ...
  lr.detector(dprofile, regions), )
```

```
lr.train(model, optimizer, loss,…)
lr.to(model, ['cuda:0', 'cuda:1'])
# spatial and integrated DSE
lr.dse(lprofile, wl, size, DataLoader)
```

```
# e.g, SLM Sys
model.to_device(amp_func, phase_func, …)
# e.g., 3D Print THz Sys
model.to_3d_render(index_dict, …)
```

Physical System



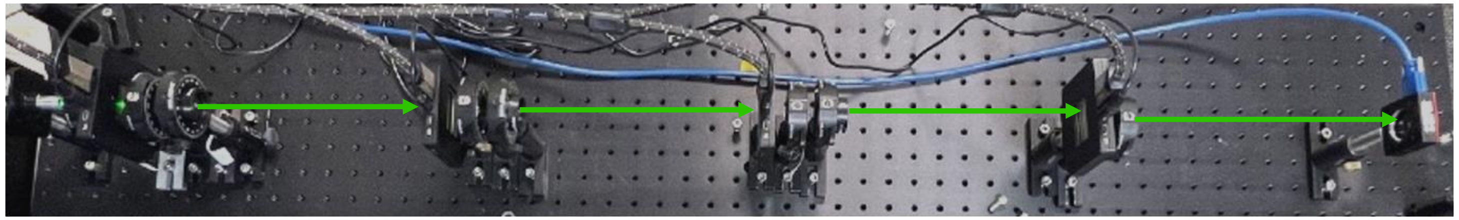
LightRidge API Example: DiffractiveLayer()

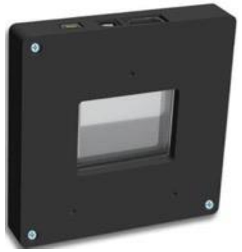
```
class DiffractiveLayer(torch.nn.Module):
    """ Implementation of diffractive layer via co-design.
    Args:
        - phase_func/intensity_fun: Callable device's phase/intensity response
        - wavelength: Float representing the wavelength of the laser source
        - pixel_size: Float representing the size of each pixel in the diffractive layer
        - resolution: Integer representing number of pixels
        - distance: Float representing the propagation distance between layers
        - amplitude_factor: Float the scaling factor in complex regularization
        - mesh_size: Integer specifying the mesh size used for diffraction approximation
        - name: String representing the name of the diffractive layer
        - approx: Callable for approximation method (default: lr.kernel.Fresnel)
                  - (Options: lr.kernel.Frauhofer, lr.kernel.Sommerfeld, lr.kernel.verify)
        - phase_mod: Boolean indicating phase modulation on or off (default: True)

    Shape:
        - Input: :math:`(*)`. Input can be of any shape
        - Output: :math:`(*)`. Output is of the same shape as input
     """
    def __init__(self, phase_func, intensity_func, wavelength,
        pixel_size, resolution, distance, amplitude_factor,
            name, approx=lr.kernel.Fresnel, phase_mod=True):
            super(DiffractiveLayer, self).__init__()
```

LightRidge API Example: *Forward Function*

- Example hardware-specific DONNs modeling
  - w.r.t 532 nm setups (z =11 inches) and LC 2012 SLMs (HOLOEYE)

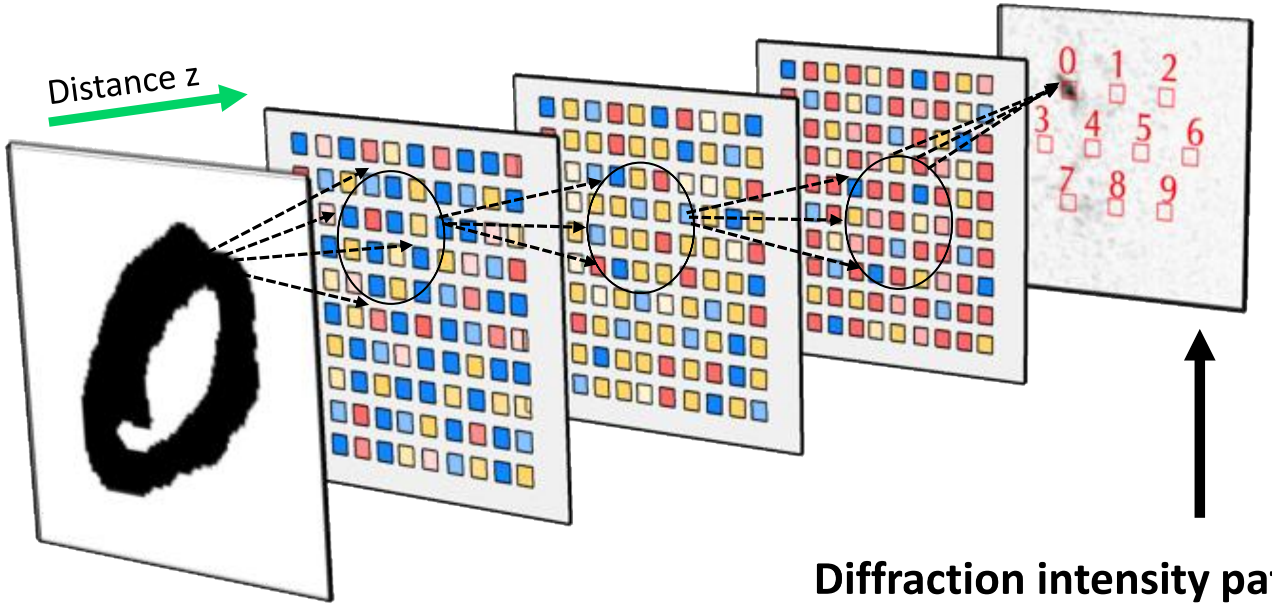




```
self.layers[1] = lr.layer.DiffractiveLayer(SLM1_phase, SLM1_amp,
        wavelength=5.32e-7,  pixel_size=3.6e-5,  resolution=100,  distance=0.2794,
        amplitude_factor=5, name='Diffractive_Layer1')
…
# a virtual layer for diffraction (w/o phase mod) before detector
self.layers[4] = lr.layer.DiffractiveLayer(self, …,       name='Last_Diffraction'
, … phase_mod = False)
```

```
# forward example of chain topology of DONNs
def forward(self, x):
        for index, layer in enumerate(self.layers):
                x = layer(x)
        output = self.detector(x)
```

Training – Example of Classification



Diffraction intensity pattern captured

Pre-defined detector region

Ground truth label t

Loss function $\mathcal{L}$

Normalized Light intensity

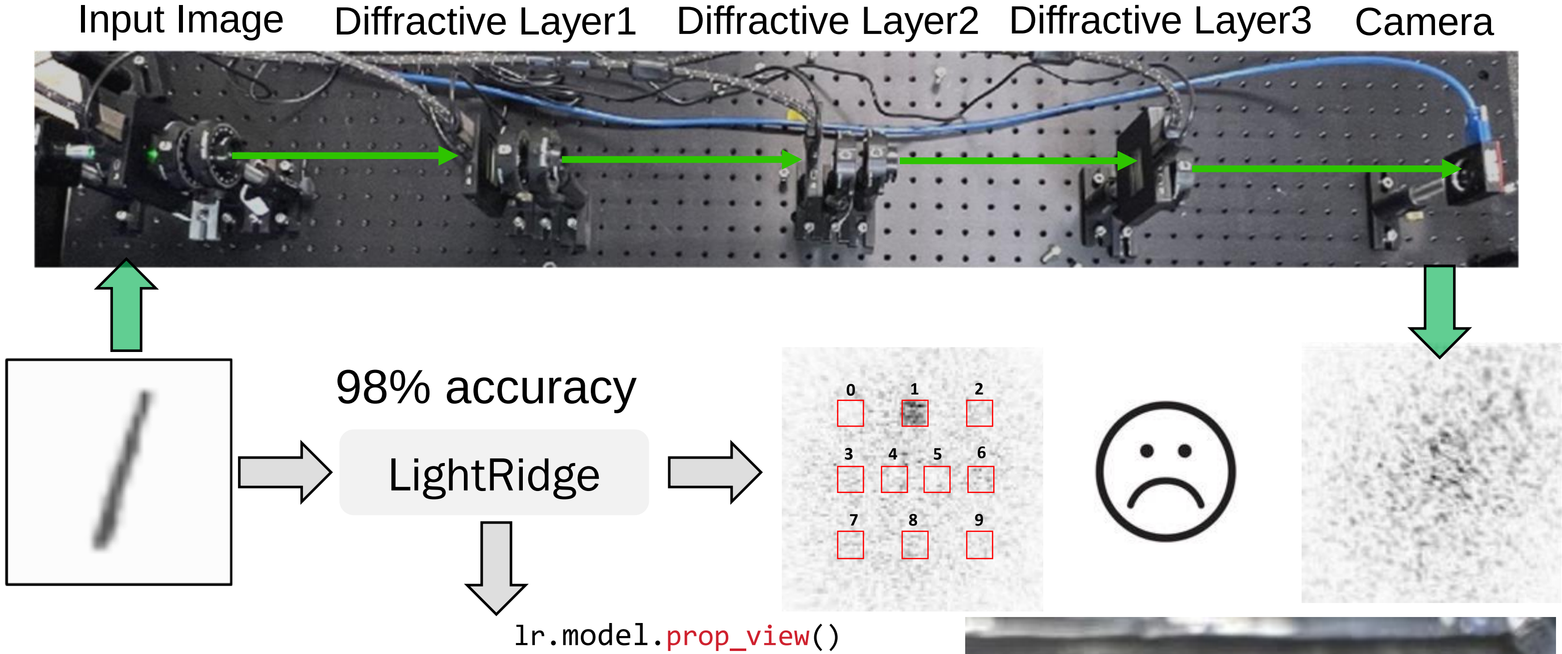
[0.7, 0.05, …, 0.03, 0.1]

`argmax = 0`

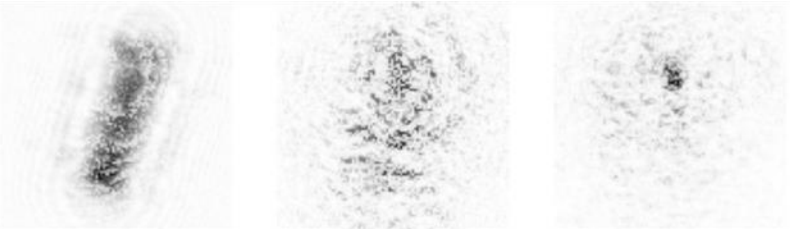
[1 , 0 , …, 0 , 0]

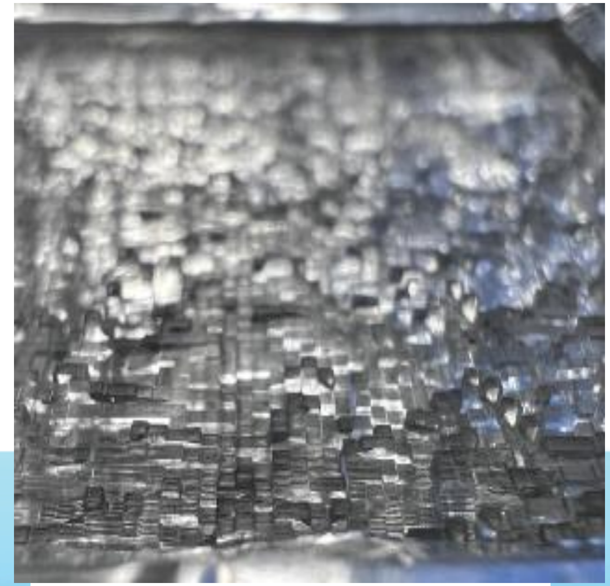
- Training via `Backprop` works!
  - Fully **tensorized** and **differentiable physics** kernels (Autograd)
  - Customizable loss w.r.t applications
    - e.g., classification, segmentation, etc.

Miscorrelation in Experimental Measurements



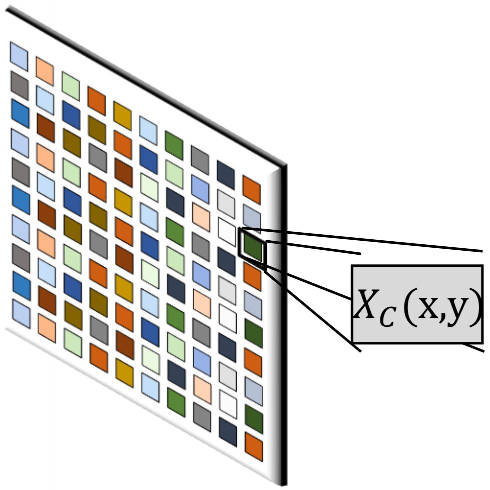
- **Issu**
  - Cc
  - Ur
  - Varies from device-to-device, wavelengths, environment, etc.





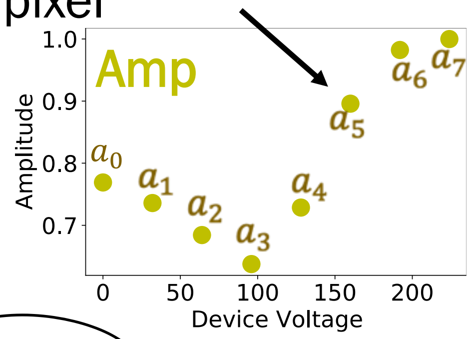
Co-design Formulation

- The formal view of co-design challenges
  - Most challenging scenarios
    - Coupled or not coupled response (phase only)
    - Individual response for different layer or every pixel

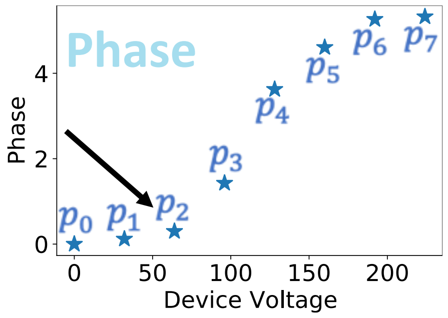


$$= A(x,y)\cos\theta(x,y) + i\,A(x,y)\sin\theta(x,y)$$

$$= 1\cdot\cos\theta(x,y) + i\,1\cdot\sin\theta(x,y)$$



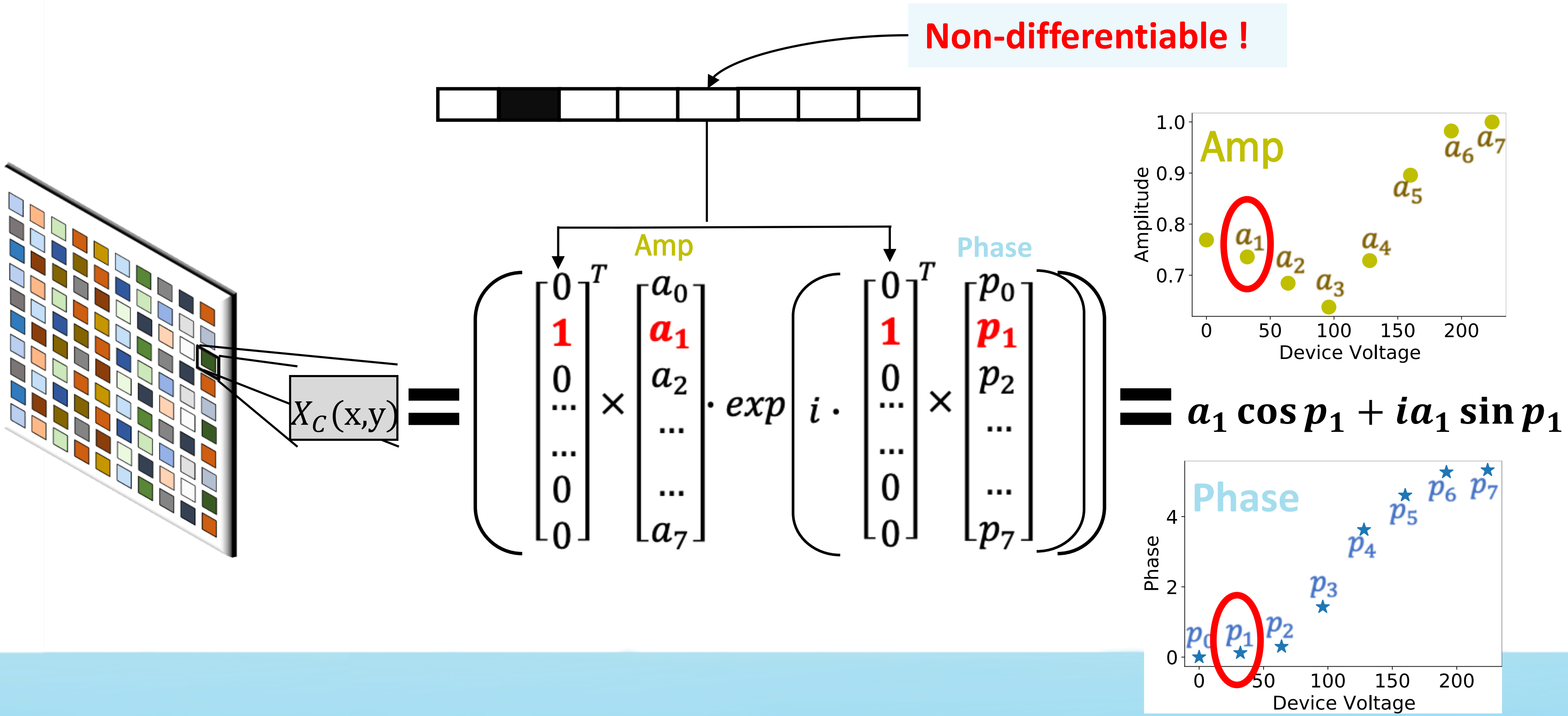
Phase modulation



A = 0.9, θ = 0.3π (0.94)

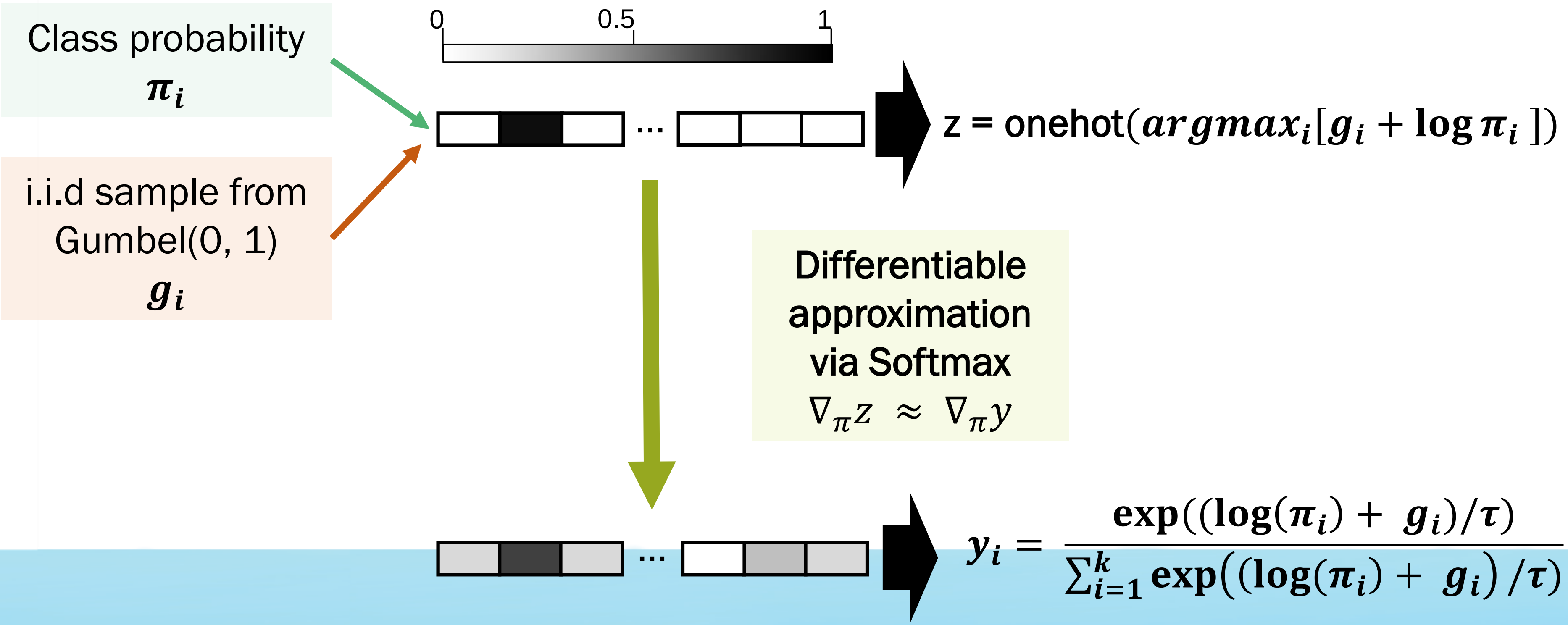
Co-design Formulation

- Discrete "trainable" parameters
  - Discrete parameters directly selects voltage index
  - Loss function remains unchanged



Gumbel-Softmax

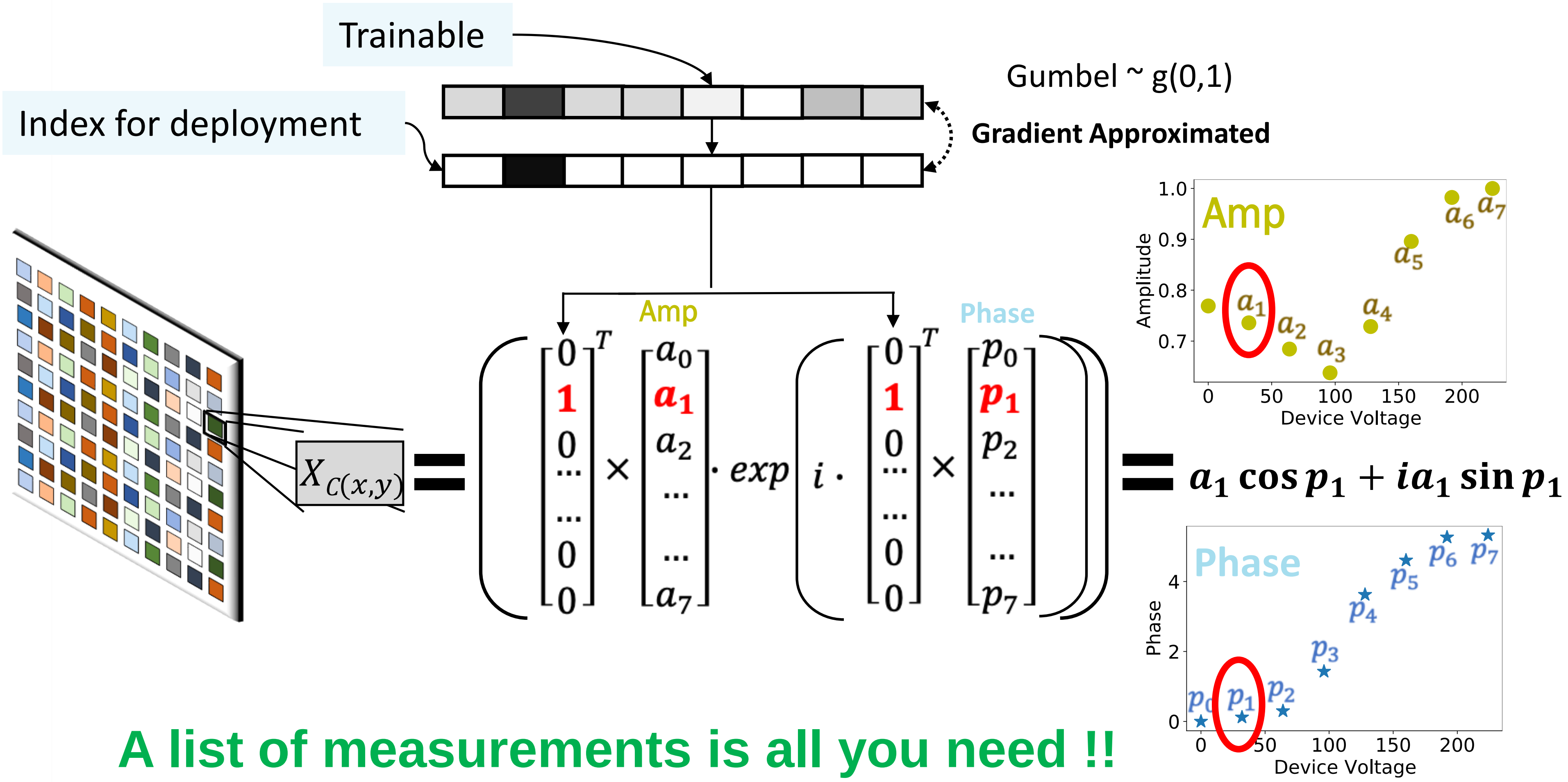
- *Gumbel-Softmax* (GS) for physics-aware discrete training
  - A differentiable approximation to sampling **discrete** data
  - **Straight-through GS** (STGS) for differentiable discrete sampling
    - Discretize GS sample back with argmax in the **forward pass**
    - GS sample in the **backward pass** to approximate the gradients



Jang, Eric, Shixiang Gu, and Ben Poole. "Categorical Reparameterization with Gumbel-Softmax." *ICLR (2017).*

Co-design Formulation

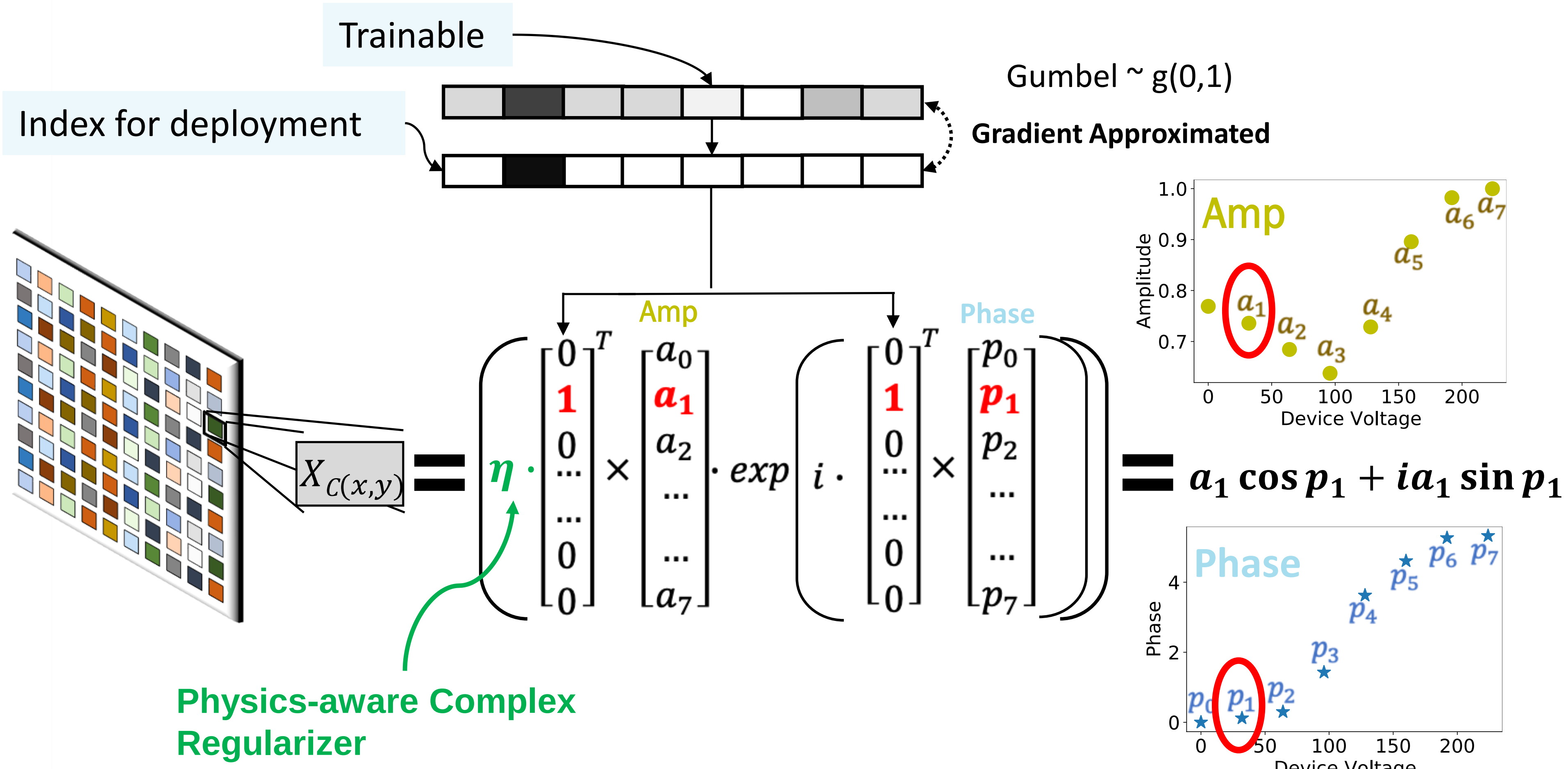
- Refine the formulation via Gumbel-Softmax



A list of measurements is all you need !!

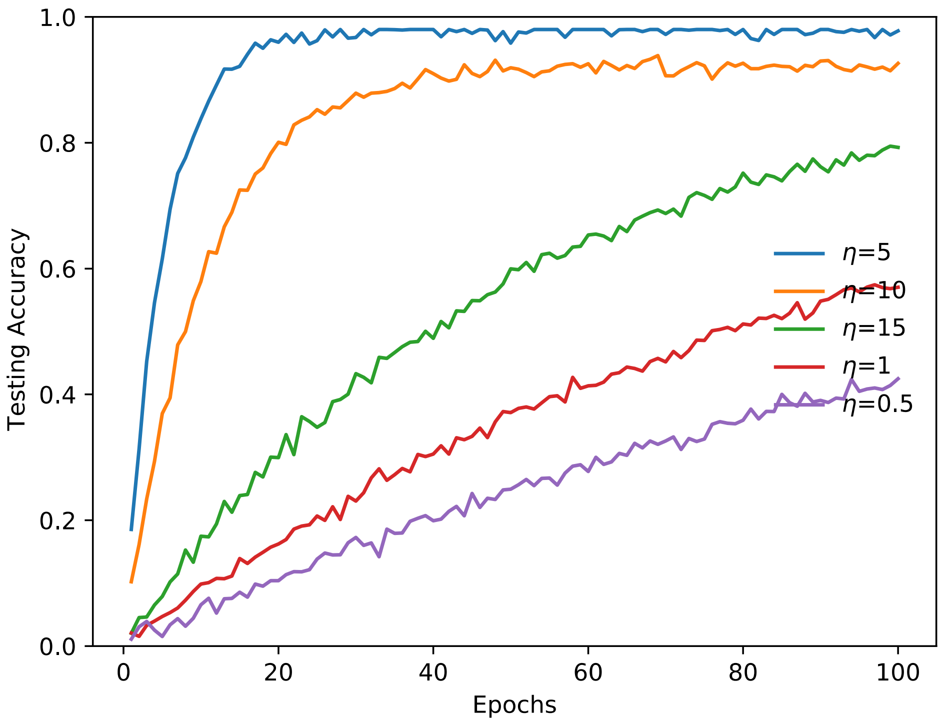
Domain-specific Complex Regularization

- Discrete training via Gumbel-Softmax



Domain-specific Complex Regularization

- Regularization is a new hyperparameter
  - Varies for different wavelength, depth, and distance
  - Tuning needs to combine Gumbel-Softmax temperature schedule



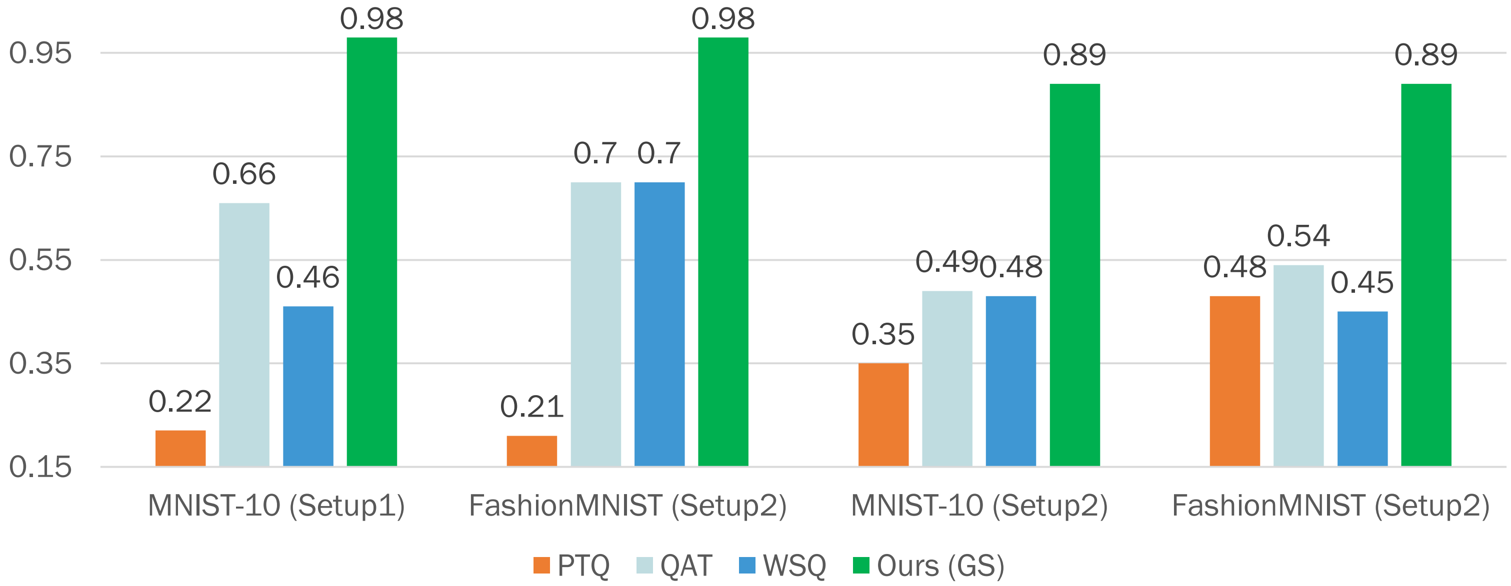
Li, Yingjie, Ruiyang Chen, Weilu Gao, and **Cunxi Yu**. "Physics-aware Differentiable Discrete Codesign for Diffractive Optical Neural Networks." In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD'22). 2022*

Comparisons with quantization methods

- Comparisons trained with a fitted continuous curve from a multi-polynomial regression model

	Post-training quantization (PTQ)	Quantization-aware training (QAT)	Weights sharing quantization (WSQ)
Pre-trained model (float32)	X	X	✓
Quantization method	Round after training	Hardware-aware training loss with **minibatch clipping**	Weights sharing with **KMeans** clustering

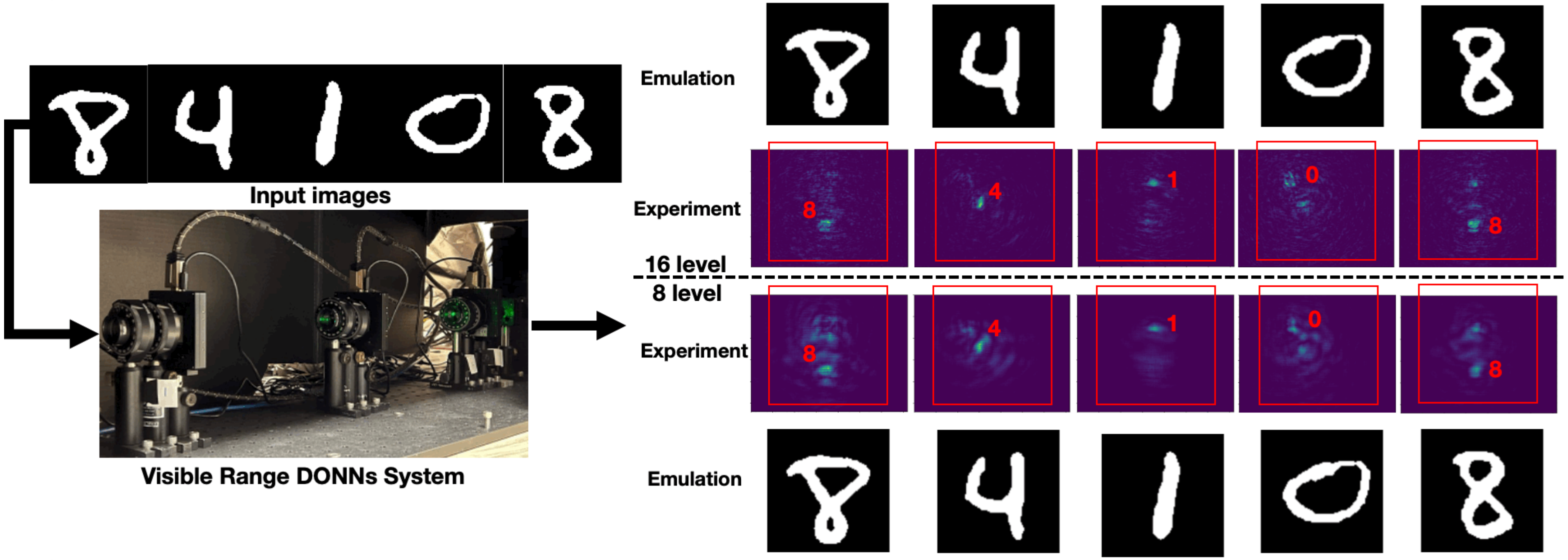
Comparisons with quantization methods




```
# file.csv includes amplitude/phase response in two rows
SLM1_amp, SLM1_phase, … = lr.utils.load_device([slm1.csv, slm2.csv,])
# plug-in in the layer definition
lr.layer.DiffractiveLayer(SLM1_phase, SLM1_amp, wavelength, …)
lr.layer.DiffractiveLayer(SLM2_phase, SLM2_amp, wavelength, …)
```

Li, Yingjie, Ruiyang Chen, Weilu Gao, and **Cunxi Yu**. "Physics-aware Differentiable Discrete Codesign for Diffractive Optical Neural Networks." In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD'22). 2022*

Experiments – Visible Range

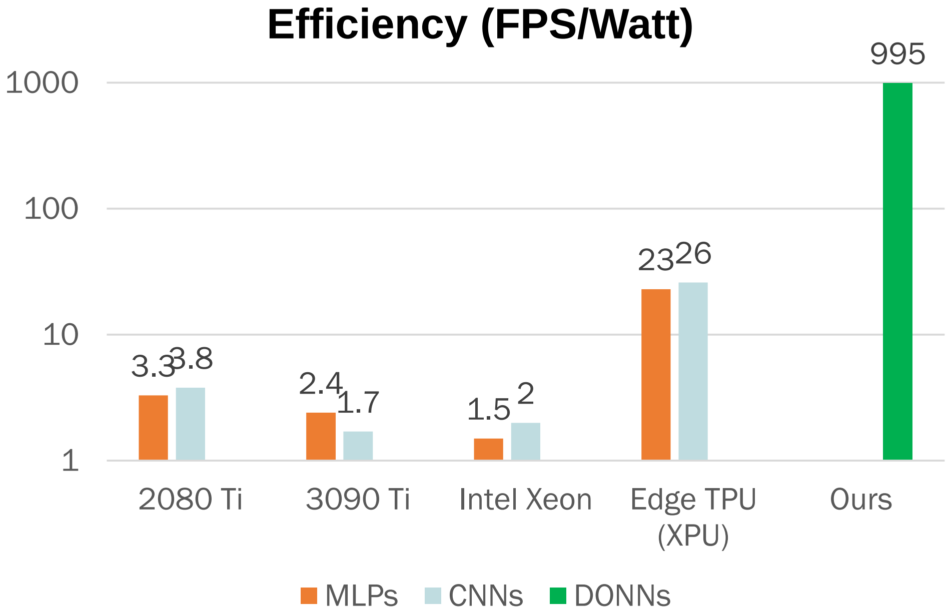


- Training and hardware setups
  - 10 min training on RTX 3090 Ti and straight out-of-box deployment
  - 98% accuracy in experimental evaluation on MNIST-10
    - Match LightRidge emulation results

Li, Yingjie, Ruiyang Chen, Weilu Gao, and **Cunxi Yu**. "Physics-aware Differentiable Discrete Codesign for Diffractive Optical Neural Networks." In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD'22). 2022*

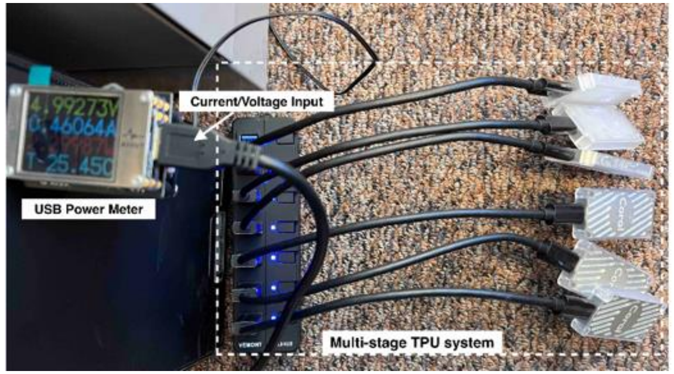
Experimental Energy Efficiency

- FPS/Watt at inference
  - Batch = 1
  - 3 orders vs GPPs
  - 50X vs XPU
  - CNNs/MLPs acc = 0.99
  - DONNs = 0.98



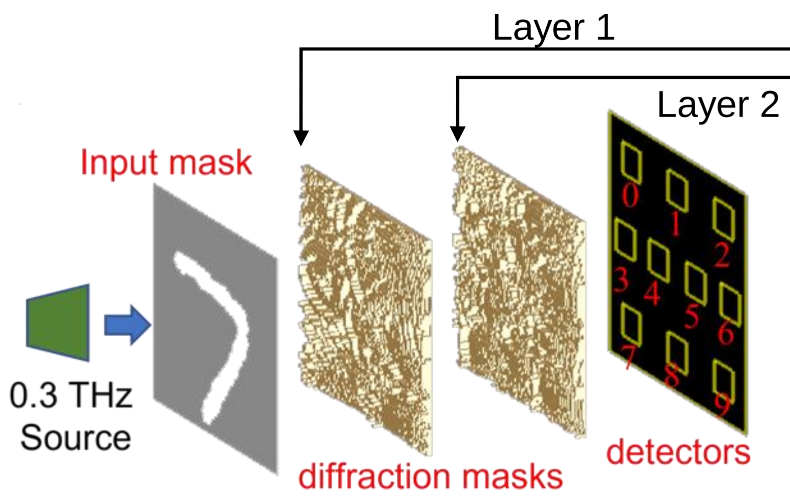
- Can be further optimized with monolithic fabrication and advance setups

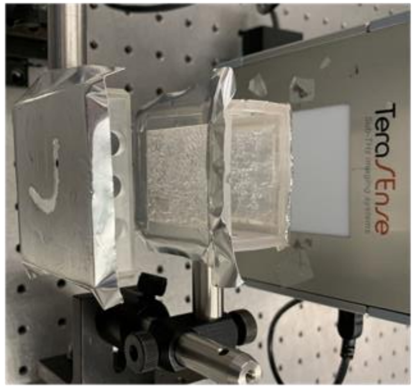
APC Metered Rack Supply PDU (CPU/GPU/DONNs measurement)



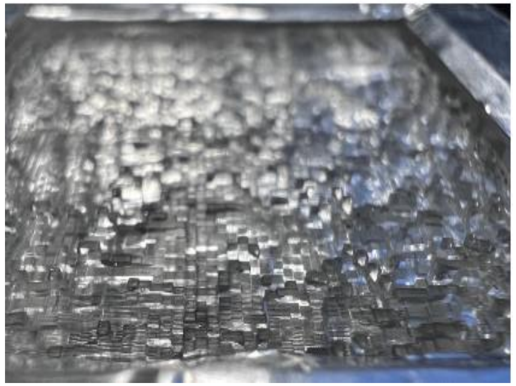
Google Edge TPU Setup

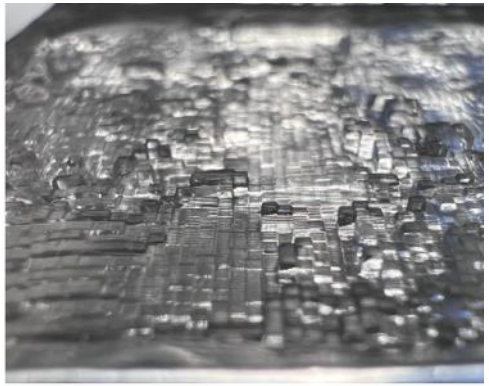
Experiments – THz Range





- THz hardware setups
  - Laser source 0.3 THz
  - 3D printed diffractive layers
    - Pixel dimension 0.5 mm
  - 93% accuracy in MNIST-10
  - *Physical sparsity*



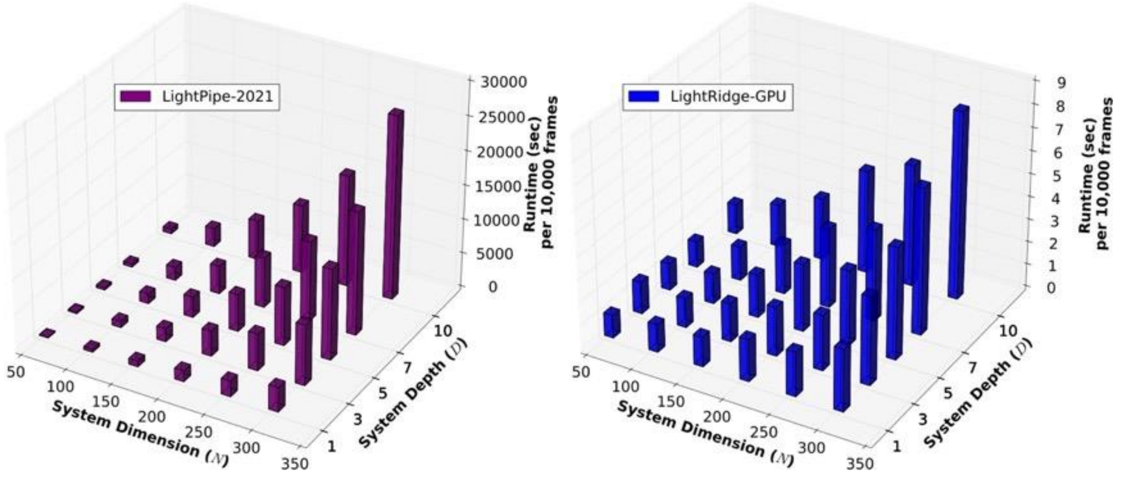


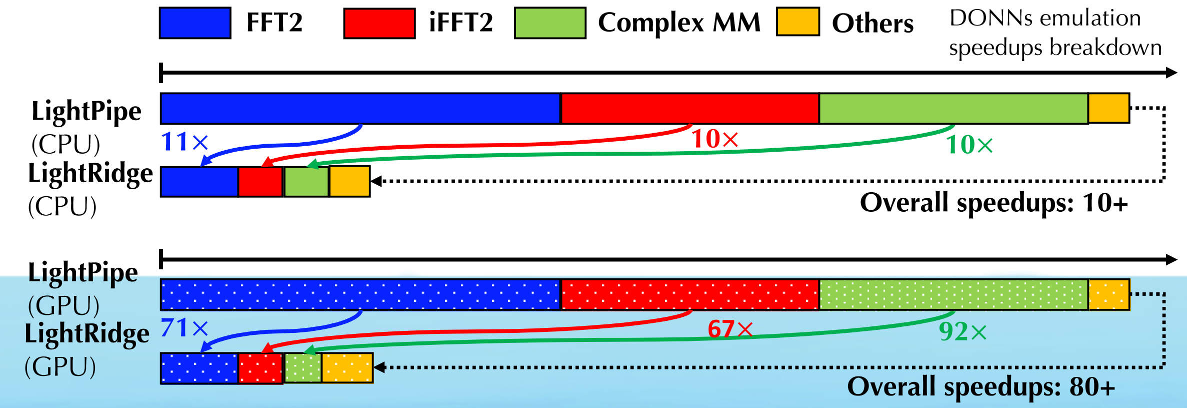
Lou, Minhan, Yingjie Li, **Cunxi Yu**, Berardi Sensale-Rodriguez, and Weilu Gao. "Effects of interlayer reflection and interpixel interaction in diffractive optical neural networks." *Optics Letters 48, no. 2 (2023): 219-222.*

Yingjie Li*, Shanglin Zhou*, Minhan Lou, Weilu Gao, Caiwen Ding, **Cunxi Yu**. "Physics-aware Roughness Optimization for Diffractive Optical Neural Networks". *Design Automation Conference (DAC'23)*

LightRidge Runtime Speedups

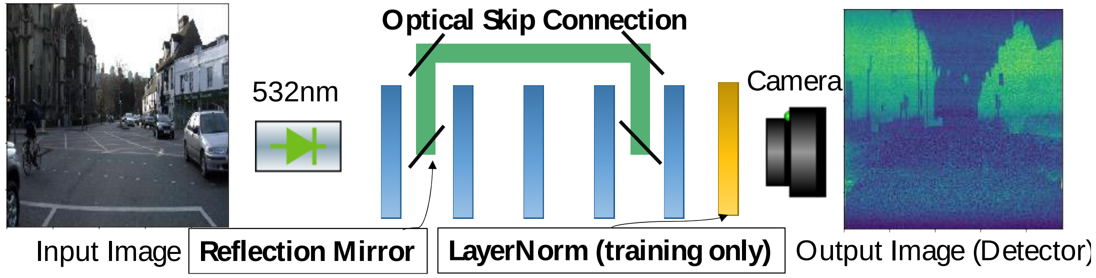
- LightRidge offers **orders of magnitude** speedups
  - Baseline: LightPipes(2021) and SOTAs [Science'18, Nature Photonics'21]
    - SOTAs reported 3-4 days training time for 5-layer DONNs
- Speedups breakdown
  - `DiffractMod` are the most critical
  - Deployment of `cuFFTC2C` and cache planning on ***h***





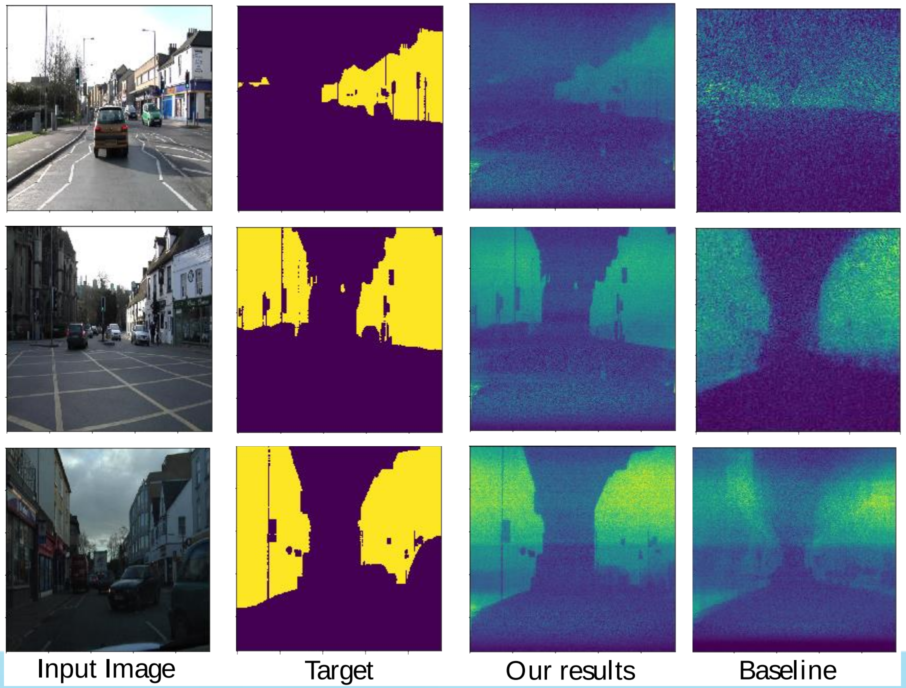
Advanced Architecture – All-Optical Segmentation

- All-optical segmentation task
  - *CityScape* dataset
  - “Optical skips”
    - Better gradient flow
  - *BCELoss*




```
for i in range(6)
  self.layers[i] = lr.layer.DiffractiveLayer()
self.layers[5].phase_mode = False
```

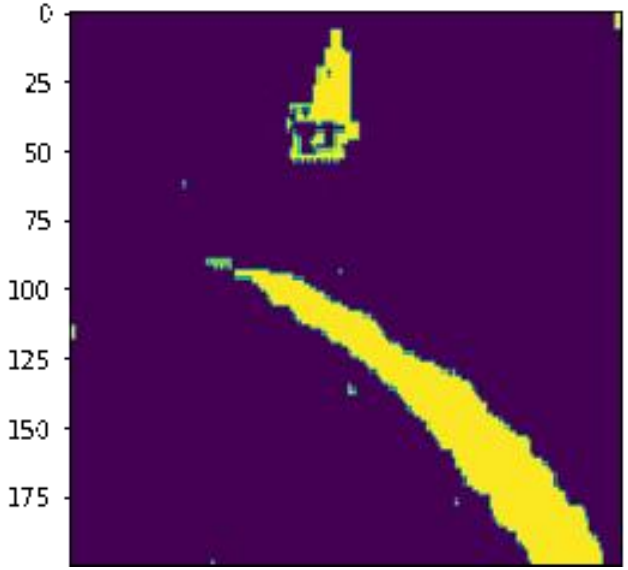
```
# optical ‘skip connection’
def forward(self, x):
  x0 = self.layers[0](x)
  x1 = self.layers[1](x0)
  x2 = self.layers[2](x1)
  x3 = self.layers[3](x2)
  # the skip
  x4 = self.layers[4](x3) + x0
  x  = self.layers[5](x4)
ln = torch.nn.LayerNorm()
output = self.detector(ln(x))
```



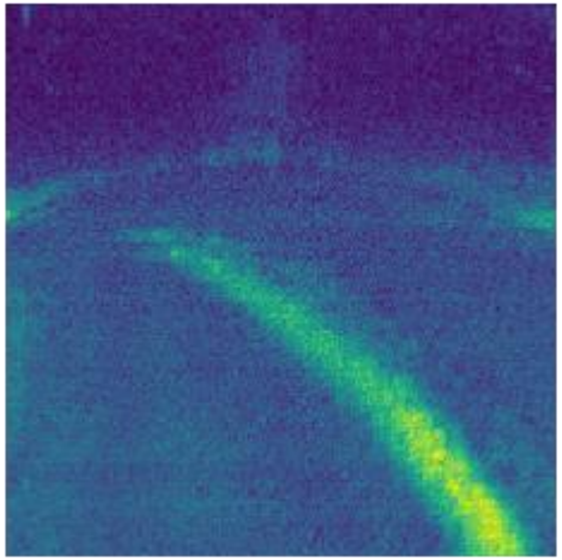
Advanced Architecture – All-Optical Segmentation

- Preliminary of all-optical autonomous driving
  - In-door lane following
  - Same architecture as segmentation task

Input



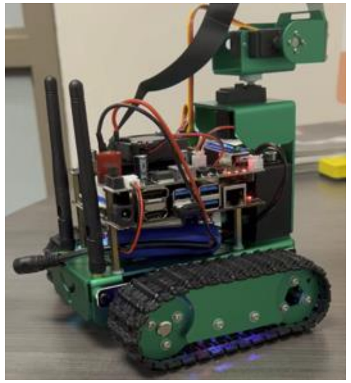
Label

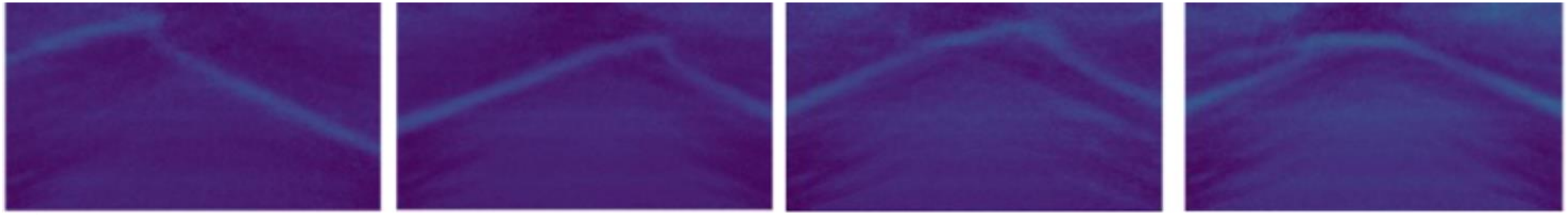


DONNs

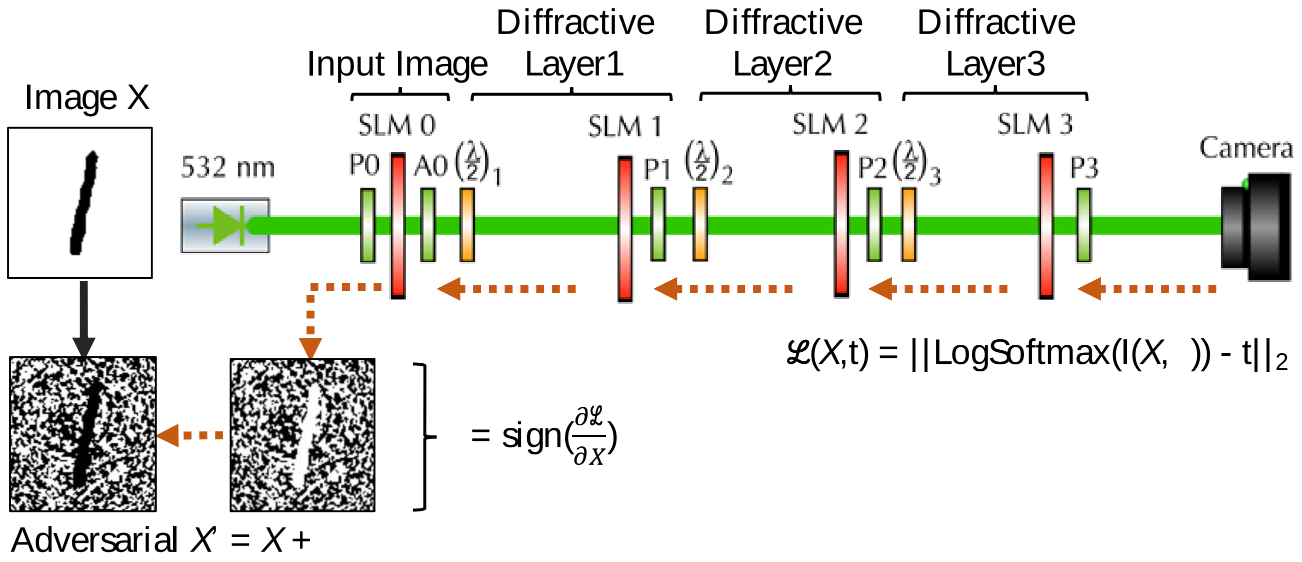
Advanced Architecture – All-Optical Segmentation

- Preliminary of all-optical autonomous driving
  - Out-door autonomous driving
    - University campus road (summer)
  - Same architecture as segmentation task





Adversaries of Light

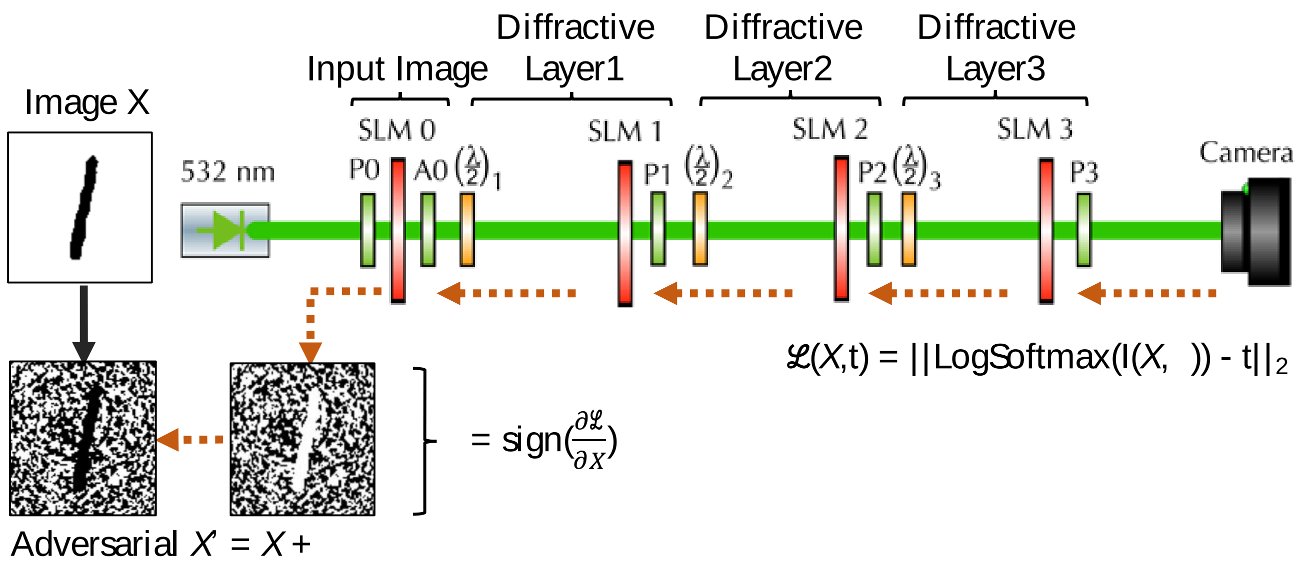


- The space of the adversaries in DONNs

Attack Types	HW System	Numerical
Real	Amplitude attack	$(A+\boldsymbol{p})e^{i\theta} = (A+\boldsymbol{p})\cos\theta + i\cdot(A+\boldsymbol{p})\sin\theta$
Complex	Phase attack	$Ae^{i(\theta+\boldsymbol{p})} = A\cos(\theta+\boldsymbol{p}) + i\cdot A\sin(\theta+\boldsymbol{p})$

Adversarial Perturbation $\boldsymbol{p}$

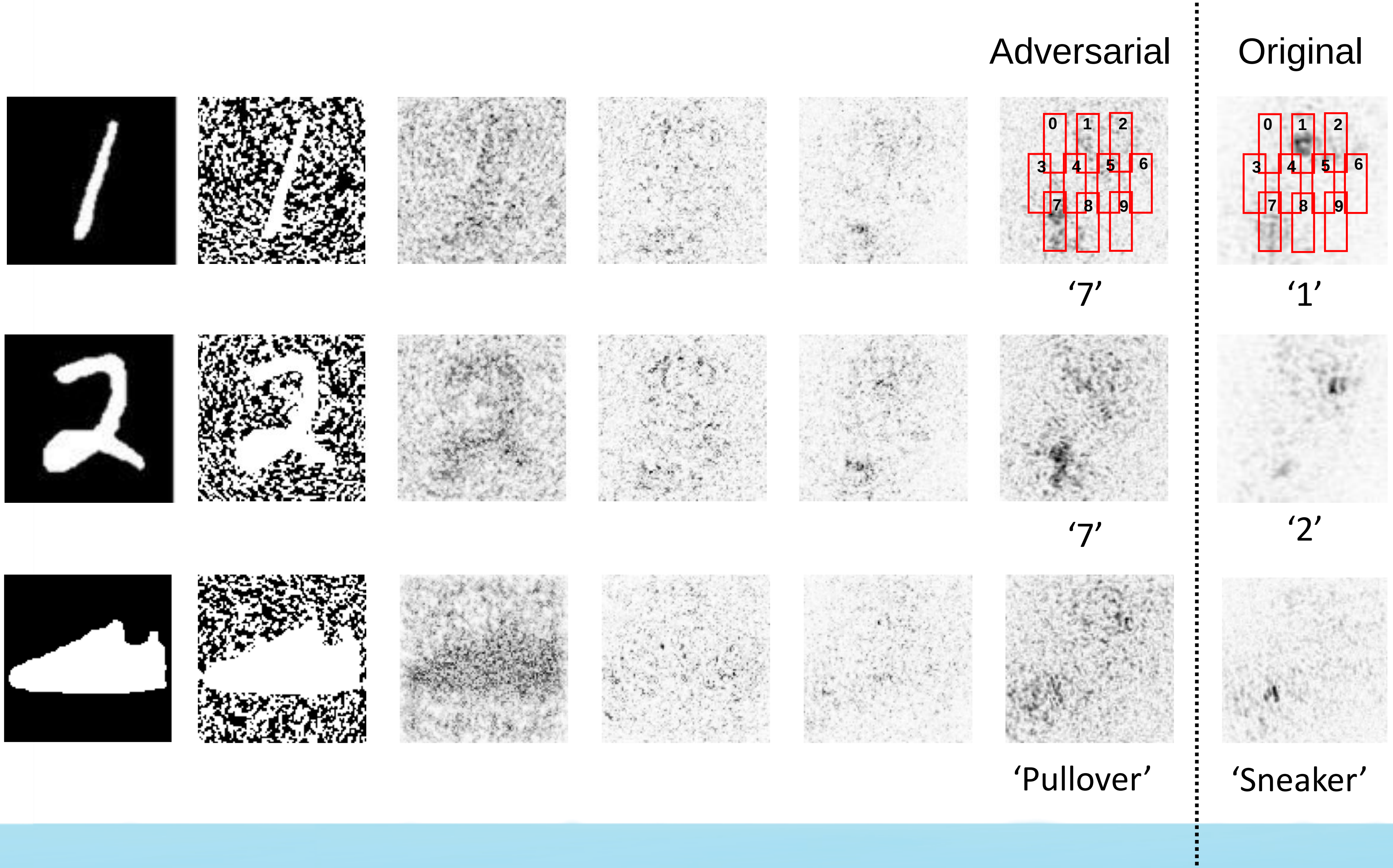
Adversaries of Light



- Domain-specific generation of adversarial examples
  - Restricted space w.r.t physics meanings
  - Perturbation engineering needs to be considered in the attack phase

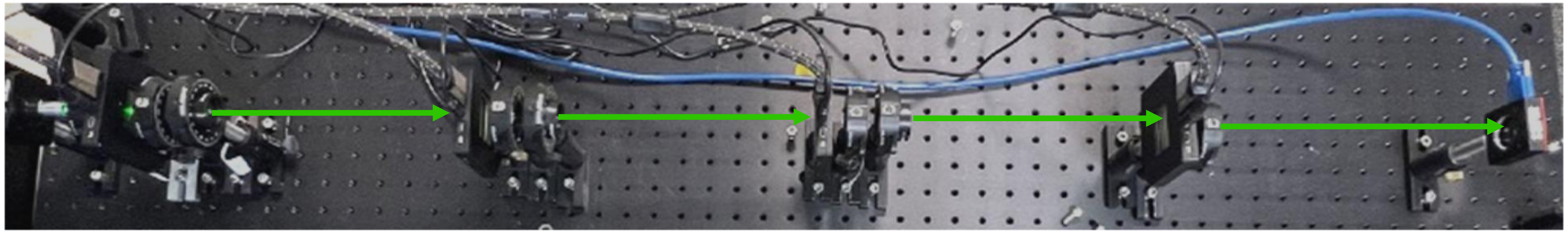
- **C-FGSM**: Complex fast-gradient-signed-method
  - Gumbel-Softmax guided co-design and perturbation engineering

Evaluations of C-FGSM

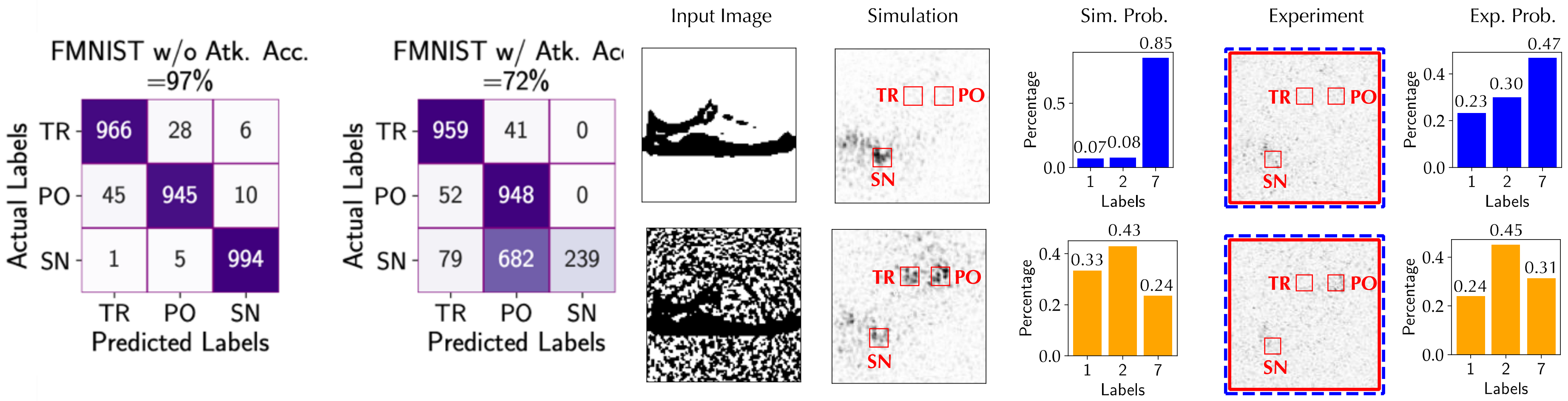


Physical Experimental Validation

Input Image Diffractive Layer1 Diffractive Layer2 Diffractive Layer3 Camera



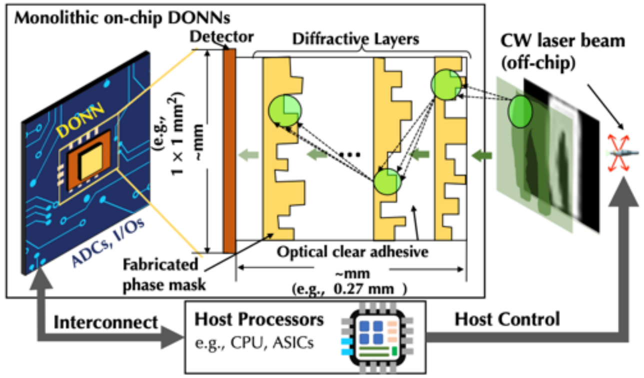
- Vulnerability exist and experimentally demonstrated
  - Natural counter-measurements - the miscorrelation and device noise

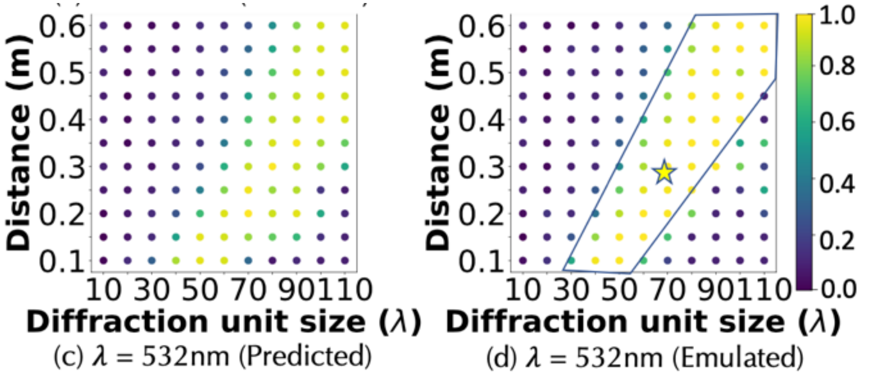


Chen, Ruiyang*, Yingjie Li*, Minhan Lou, Jichao Fan, Yingheng Tang, Berardi Sensale-Rodriguez, **Cunxi Yu**, and Weilu Gao. "Physics-Aware Machine Learning and Adversarial Attack in Complex-Valued Reconfigurable Diffractive All-Optical Neural Network." *Laser & Photonics Reviews* (2022).

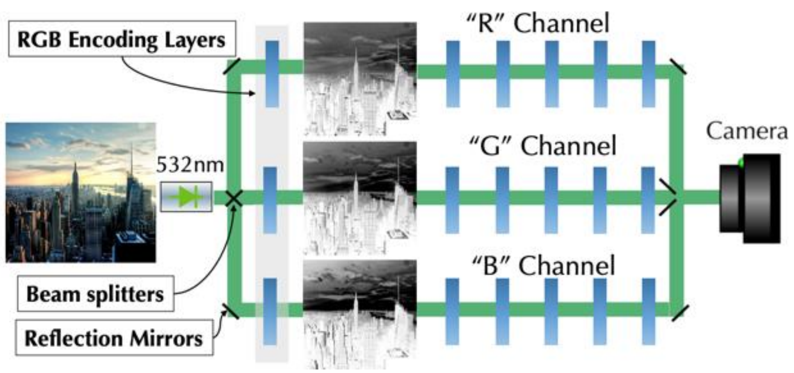
Other Features

ML-assisted DSE

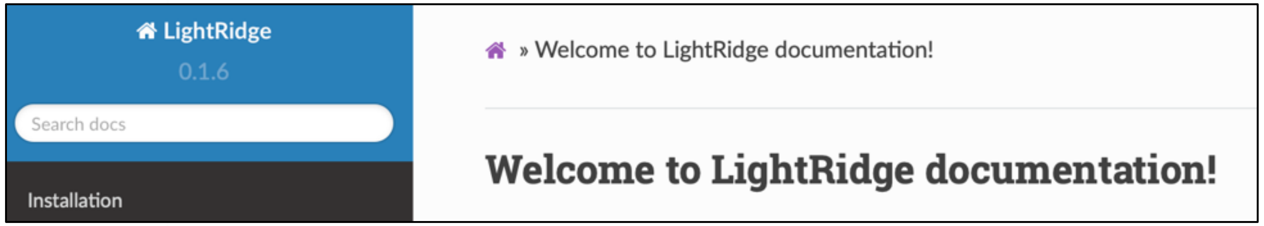




Monolithic Integration



Advanced Architectures & Multi-task Learning



Conclusions

- *A Computer Engineering* journey to **Diffractive Optical Neural Networks**

[DAC’21]
[OSCAR@ISCA’22]
[ASPLOS’24]

Programming Infrastructure

[ICCAD’22][1]
[Laser & Photonics Reviews’22][1]

End-to-end Co-design

[ICCAD’22][1]
[DAC’23]

Domain-specific Algorithms

Physics ⟺ **LightRidge Design Flow** ⟺ Real-world System

Adversaries of Light

[DAC’21]
[Laser & Photonics Reviews’22][1]
[CLEO’22][1]

Architectures & Integration

[Scientific Report’21]
[IJCAI’23]

Application-driven Material/Device

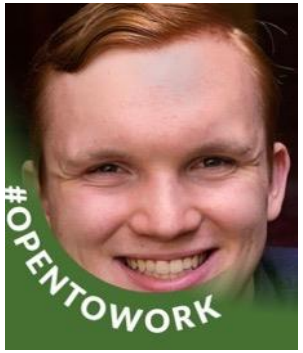
[Optics Letters’23]
[CLEO’22][2]
[Laser & Photonics Reviews’22][2]

Research Group

PI Dr. Cunxi Yu

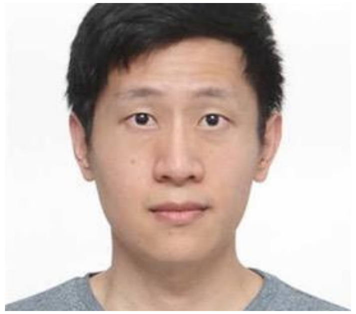
Yingjie Li
PhD (S20 -)
(at DAC'23)

Jiaqi Yin
PhD (F20 -)
(at DAC'23)



Daniel Robinson
B.S. NSF REU (F21 -)
(at DAC'23)

Mingju Liu
PhD (F22 -)
(at DAC'23)



Dr. Minhan Lou
Postdoc

Co-advised PhD
Dr. Walter Lau Neto

BS/MS (thesis)
Tara Zamani

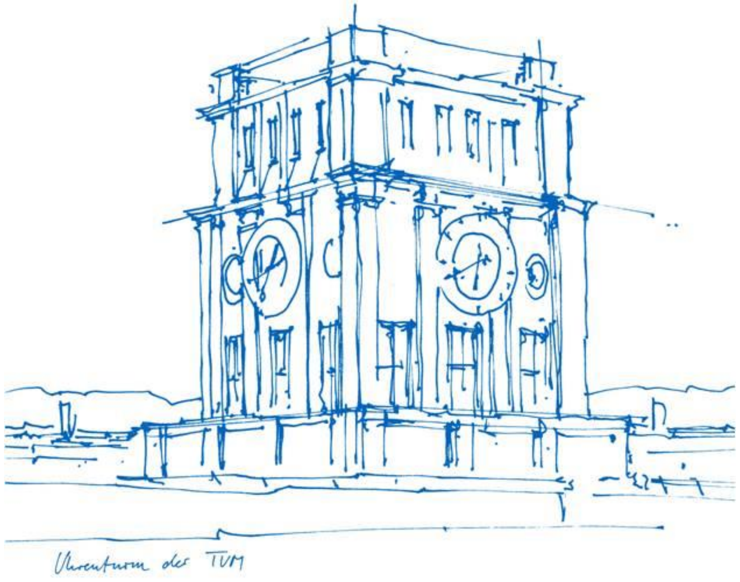
NSF REU
Nicolas Taylor

Tutorial III: Topology and physical layout optimization of photonic networks-on-chip and PIC variation analysis

Ulf Schlichtmann,

Technical University of Munich

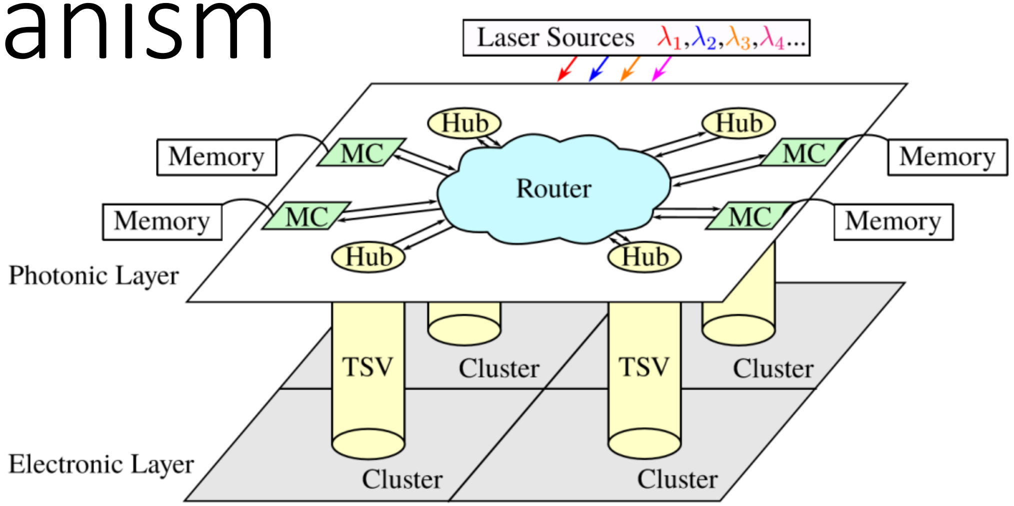
Topology and physical layout optimization of photonic networks-on-chip

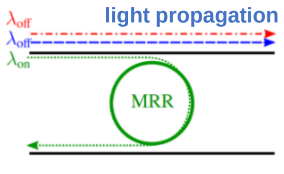


WRONoC – Wavelength-Routed ONoC

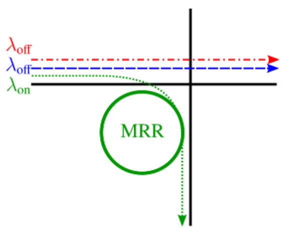
WRONoC Working Mechanism

- Usage of microring resonators (MRRs) for multiplexing and demultiplexing
- Dedicated signal path determined in design phase for each tuple (initiator, target, wavelength)
- Main constraint: No path overlap between signals with the same wavelength

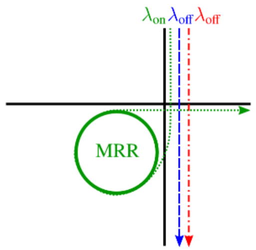




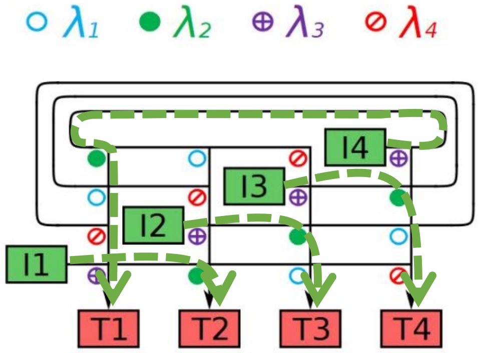
180° direction change



90° direction change

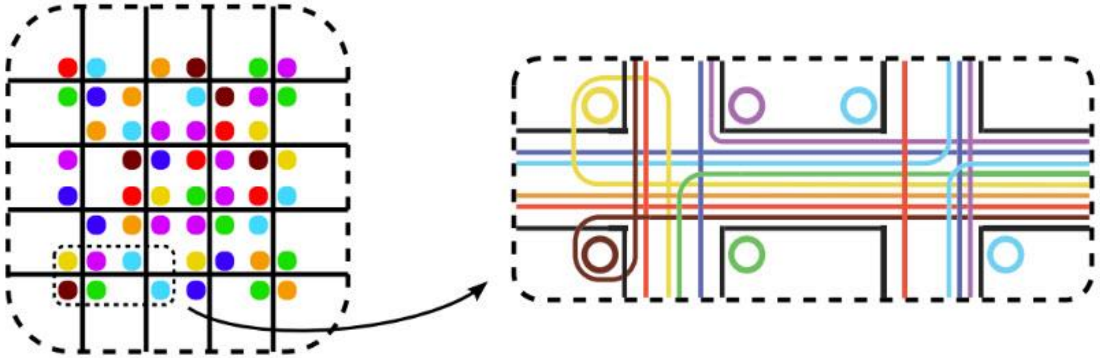


270° direction change



WRONoC Pros and Cons

- Advantages:
  - No control resource
  - No scheduling effort
  - No congestion control
  - No signal path construction → no uncertain signal delay
- Disadvantages:
  - Extensive usage of MRRs (1 MRR serves constant #paths) → **Scalability issue!** → suitable for application-specific usage → need design optimization to save resources



WRONoC Design Features

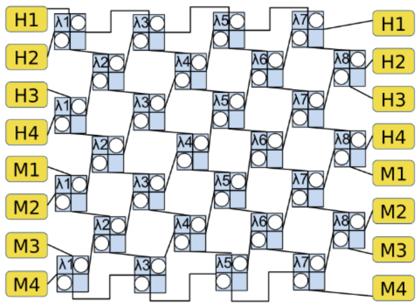
Topological features:

- Waveguide connection structure
- MRR topological locations
- MRR resonant wavelengths
- Signal wavelength assignment
- Signal path routing

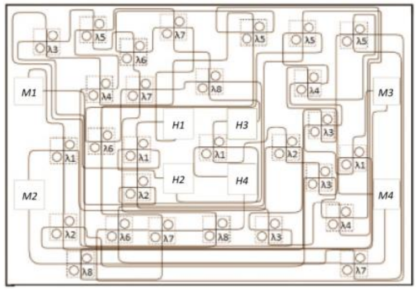
Physical design features:

- Waveguide routing
- MRR placement

Manual topology



Manual layout



- All these need to be done during the design phase

  → **Challenges of efficiency! & beyond human capability!**

***Sources**:*

1) Engineering a Bandwidth-Scalable Optical Layer for a 3D Multi-core Processor with Awareness of Layout Constraints, NOCS'12, Luca Ramini et al.

WRONoC Research at TUM — Since 2018

- Router design and synthesis:
  - Topology synthesis
    - *CustomTopo (ICCAD’18)*
    - *FAST (DATE’21, TCAD‘22)*
  - Topology design
    - *Light (ASP-DAC’21)*
  - Physical synthesis
    - *ToPro (ICCAD’21)*
  - Topology synthesis + physical synthesis
    - *PSION (ISPD’19, TCAD’20, ICCAD’20)*
- Bandwidth maximization: *MaxBW (ASP-DAC’20)*

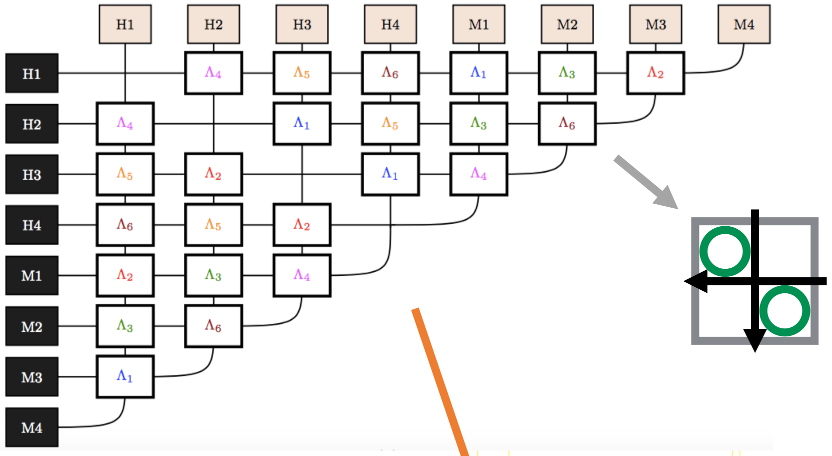
WRONoC Research at TUM

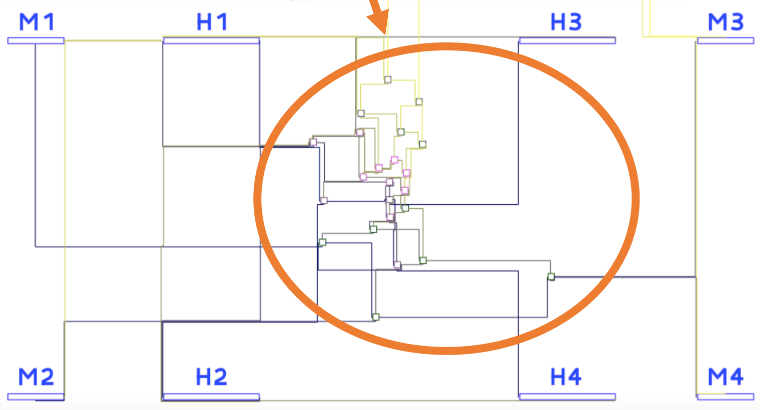
- **Router design and synthesis:**
  - **Topology synthesis**
    - ***CustomTopo (ICCAD’18)***
    - ***FAST (DATE’21, TCAD‘22)***
  - Topology design
    - *Light (ASP-DAC’21)*
  - Physical synthesis
    - *ToPro (ICCAD’21)*
  - Topology synthesis + physical synthesis
    - *PSION (ISPD’19, TCAD’20, ICCAD’20)*
- Bandwidth maximization: *MaxBW (ASP-DAC’20)*

Separate Design Steps

- Topology generation and then physical design
- Advantages:
    - Natural problem partitioning
    - Observable intermediate solution, i.e. topology
    - Fast
- Disadvantages:
    - Node position not considered → long waveguide detours and crossings

automatically-synthesized topology



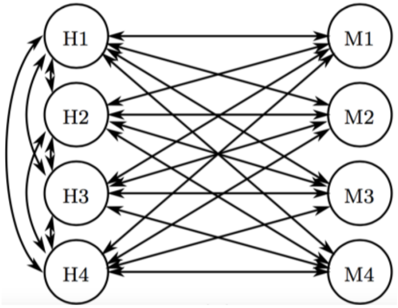


automatically-synthesized layout

Sources:
1) *CustomTopo: A Topology Generation Method for Application-Specific Wavelength-Routed Optical NoCs, ICCAD'18*, Mengchu Li et al.

Topology Synthesis by CustomTopo

Input: communication graph



communication matrix

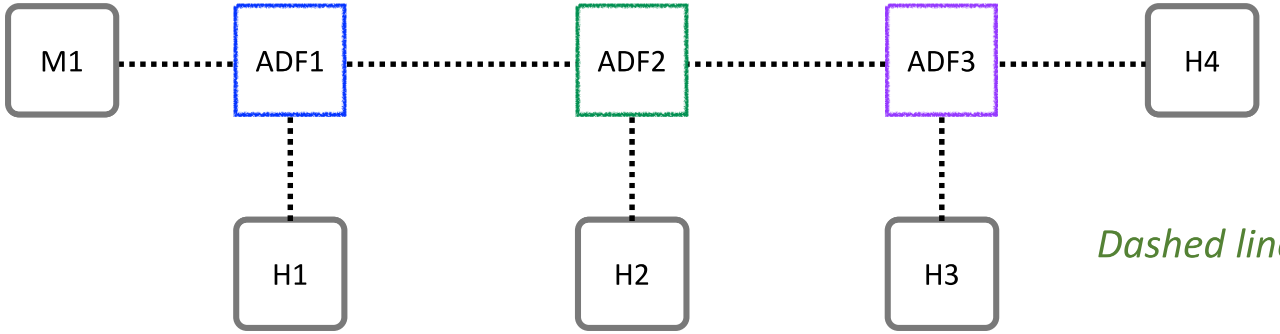
$$\begin{array}{cc} & \begin{matrix} H1 & H2 & H3 & H4 & M1 & M2 & M3 & M4 \end{matrix} \\ \begin{matrix} H1 \\ H2 \\ H3 \\ H4 \\ M1 \\ M2 \\ M3 \\ M4 \end{matrix} & \begin{pmatrix} & 4 & 5 & 6 & 1 & 3 & 2 & 0 \\ 4 & & 1 & 5 & 3 & 6 & 0 & 2 \\ 5 & 2 & & 1 & 4 & 0 & 6 & 3 \\ 6 & 5 & 2 & & 0 & 4 & 3 & 1 \\ 2 & 3 & 4 & 0 & & & & \\ 3 & 6 & 0 & 4 & & & & \\ 1 & 0 & 6 & 3 & & & & \\ 0 & 1 & 3 & 2 & & & & \end{pmatrix} \end{array}$$

wavelength for each message determined
wavelength for each ADF determined
#ADF-sharing structures maximized

Sources:
1) *CustomTopo: A Topology Generation Method for Application-Specific Wavelength-Routed Optical NoCs, ICCAD'18*, Mengchu Li et al.

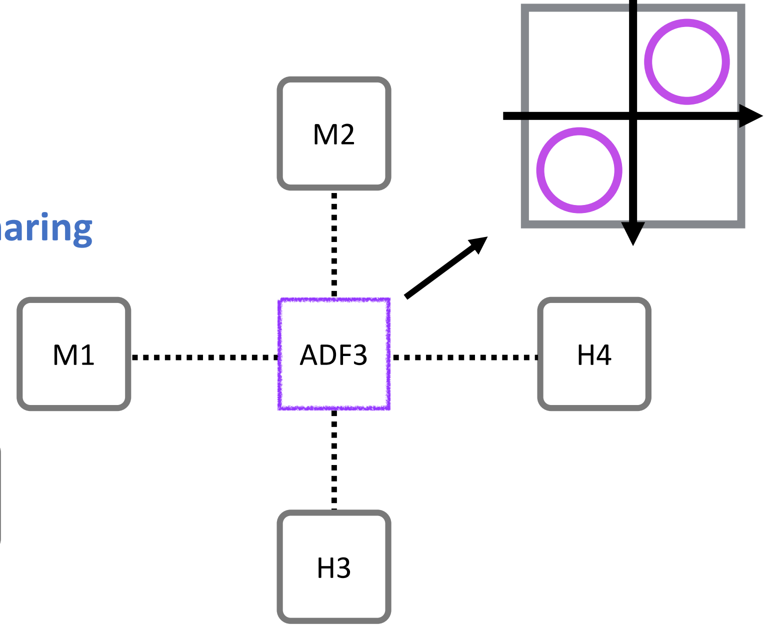
Information in the Communication Matrix

	H1	H2	H3	H4	M1	M2	M3	M4
H1		4	5	6	1	3	2	0
H2	4		1	5	3	6	0	2
H3	5	2		1	4	0	6	3
H4	6	5	2		0	4	3	1
M1	2	3	4	0				
M2	3	6	0	4				
M3	1	0	6	3				
M4	0	1	3	2				



Add-drop filter (ADF) with 2 MRRs

ADF sharing

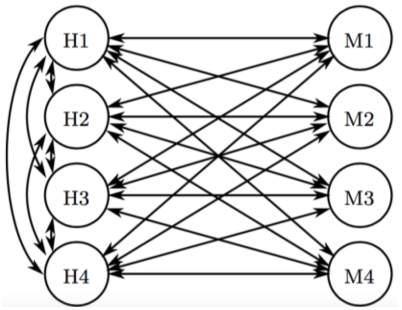


Dashed lines: components may not be directly connected

Sources:
1) *CustomTopo: A Topology Generation Method for Application-Specific Wavelength-Routed Optical NoCs, ICCAD'18*, Mengchu Li et al.

Topology Synthesis by CustomTopo

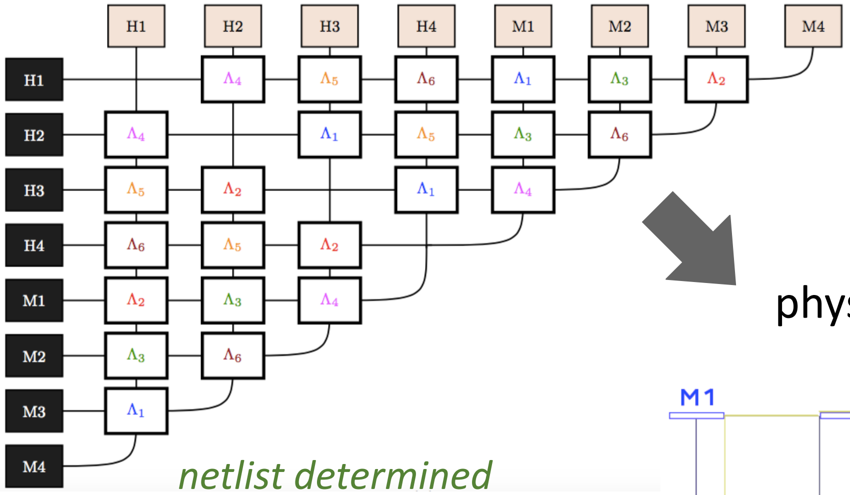
Input: communication graph



communication matrix

	H1	H2	H3	H4	M1	M2	M3	M4
H1		4	5	6	1	3	2	0
H2	4		1	5	3	6	0	2
H3	5	2		1	4	0	6	3
H4	6	5	2		0	4	3	1
M1	2	3	4	0				
M2	3	6	0	4				
M3	1	0	6	3				
M4	0	1	3	2				

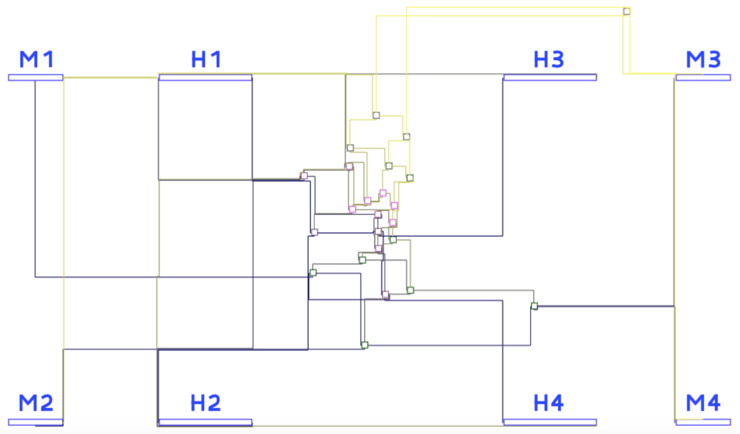
topology



netlist determined

wavelength for each message determined
wavelength for each ADF determined
#ADF-sharing structures maximized

physical design (by PROTON)



Sources:

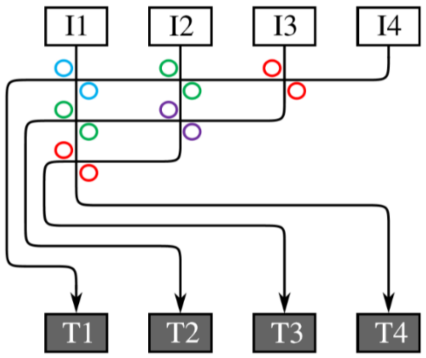
1) *CustomTopo: A Topology Generation Method for Application-Specific Wavelength-Routed Optical NoCs, ICCAD'18*, Mengchu Li et al.
2) *PROTON+: A Placement and Routing Tool for 3D Optical Networks-on-Chip with a Single Optical Layer, JETC'15, Anja von Beuningen et al.*

Topology Synthesis by FAST

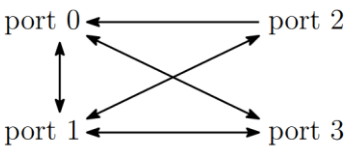
- Reduced from Snake topology

Results comparable with CustomTopo, but much faster

4×4 Snake



Comm. Graph



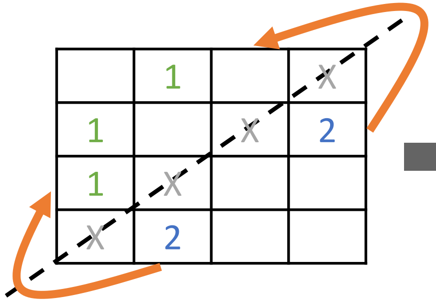
Comm. Matrix

	R0	R1	R2	R3
S0		1		1
S1	1		1	1
S2	1	1		
S3	1	1		

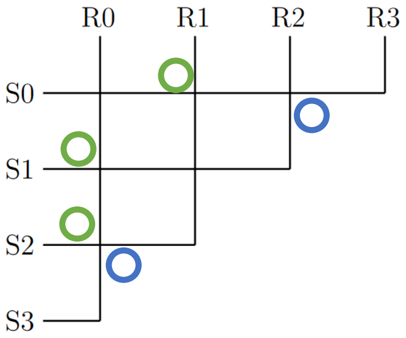
Entry Revision

	1		X
1		X	2
1	X		
X	2		

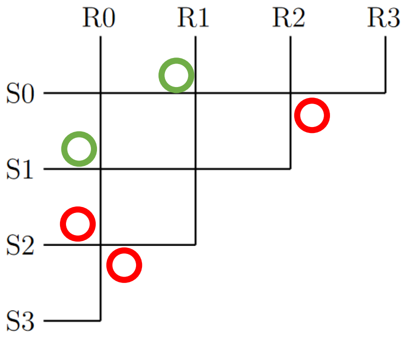
Folding



Resulting Topology



Wavelength Assignment



Sources:

1) *Contrasting Wavelength-Routed Optical NoC Topologies for Power-Efficient 3D-stacked Multicore Processors using Physical-Layer Analysis, DATE'13*, Luca Ramini et al.
2) *FAST: A Fast Automatic Sweeping Topology Customization Method for Application-Specific Wavelength-Routed Optical NoCs, DATE'21*, Moyuan Xiao et al.
3) *Crosstalk-Aware Automatic Topology Customization and Optimization for Wavelength-Routed Optical NoCs, IEEE TCAD'22*, Moyuan Xiao et al.

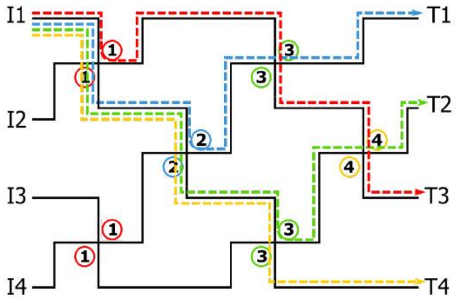
WRONoC Research at TUM

- **Router design and synthesis:**
  - Topology synthesis
    - *CustomTopo (ICCAD'18)*
    - *FAST (DATE'21, TCAD'22)*
  - **Topology design**
    - ***Light (ASP-DAC'21)***
  - Physical synthesis
    - *ToPro (ICCAD'21)*
  - Topology synthesis + physical synthesis
    - *PSION (ISPD'19, TCAD'20, ICCAD'20)*
- Bandwidth maximization: *MaxBW (ASP-DAC'20)*

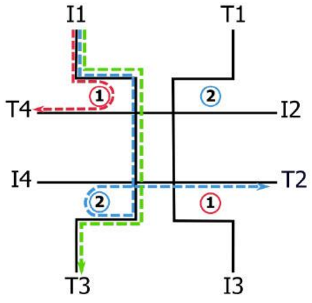
Light: $n \times (n-1)$ WRONoC Router

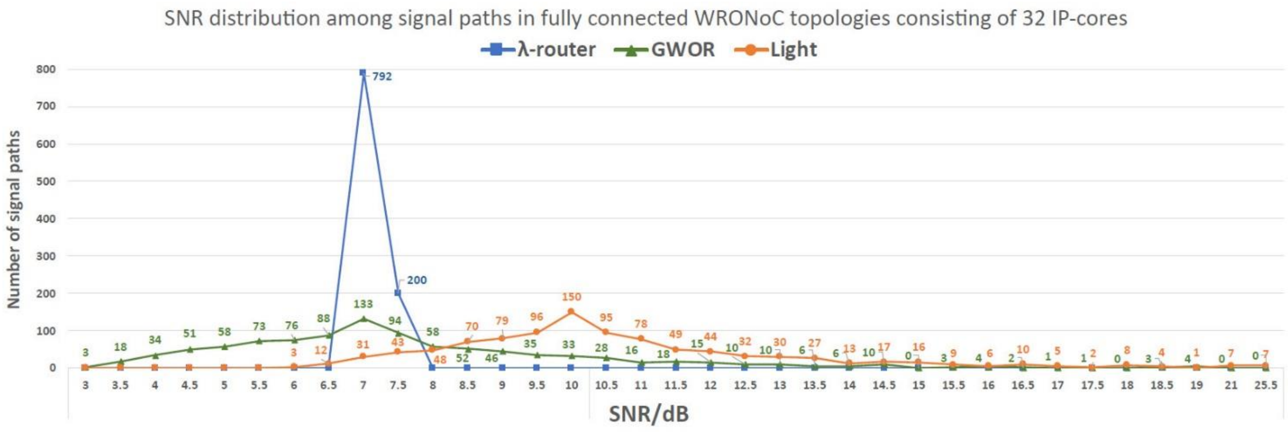
- Physical-design-aware
- A wide range of signal-to-noise ratio (SNR) distribution — good potential for signal path binding

4×4 λ-router

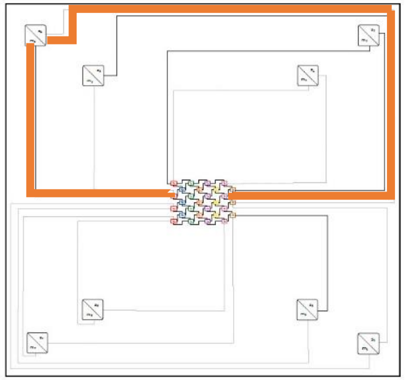


4×3 Light

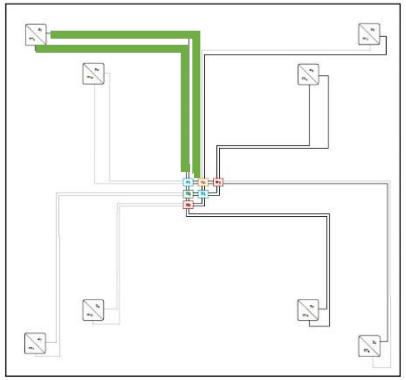




8×8 λ-router



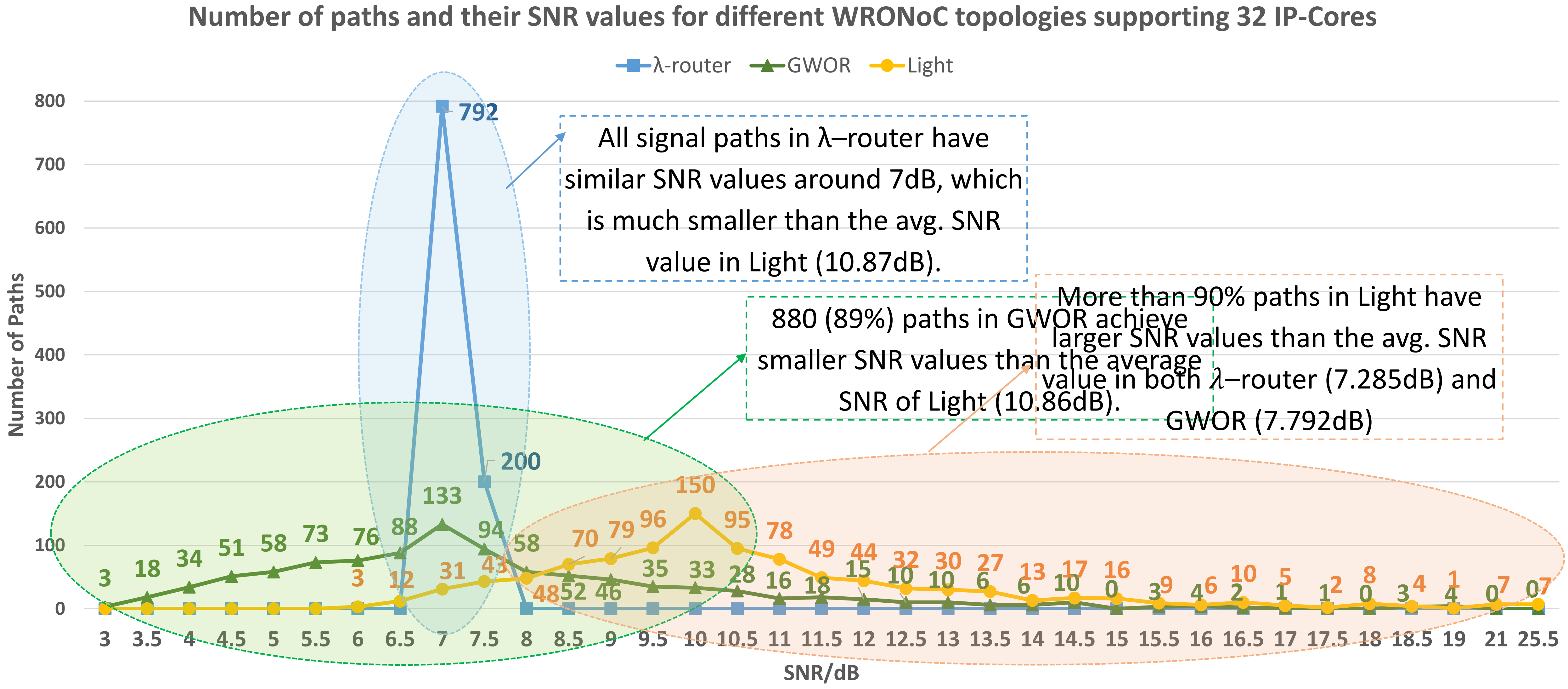
8×7 Light



Sources:

1) *Light: A Scalable and Efficient Wavelength-Routed Optical Networks-On-Chip Topology, ASP-DAC'21*, Zhidan Zheng et al.

Light: Results in detail



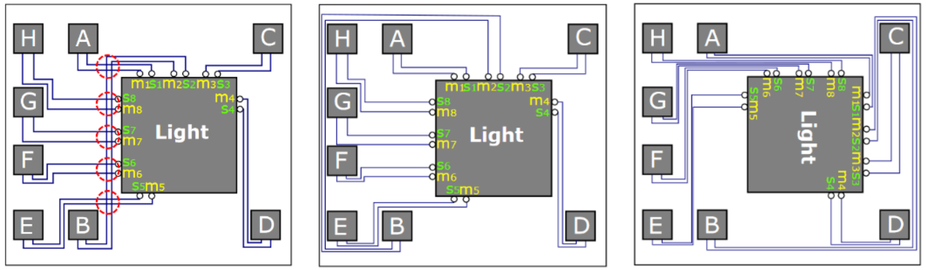
WRONoC Research at TUM

- **Router design and synthesis:**
  - Topology synthesis
    - *CustomTopo (ICCAD'18)*
    - *FAST (DATE'21, TCAD'22)*
  - Topology design
    - *Light (ASP-DAC'21)*
  - **Physical synthesis**
    - ***ToPro (ICCAD'21)***
  - Topology synthesis + physical synthesis
    - *PSION (ISPD'19, TCAD'20, ICCAD'20)*
- Bandwidth maximization: *MaxBW (ASP-DAC'20)*

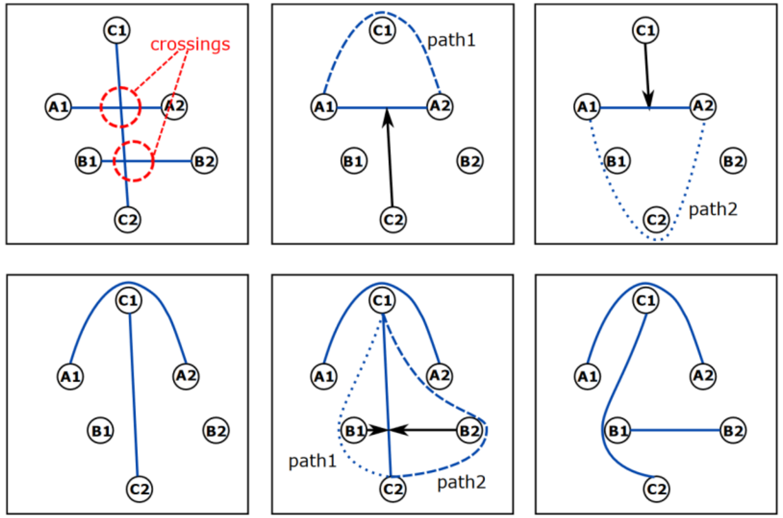
ToPro: Waveguide Router

- Steps:
  1. Project a physical-design-aware topology, e.g. Light, onto the center of the routing plane
  2. Route shortest paths
  3. Crossing resolution by path pushing
- Zero-crossing waveguide routing from router to nodes
- Minimize insertion loss & Maximize SNR

Router rotation and flip



Crossing resolution by path pushing



Sources:
1) *ToPro: A Topology Projector and Waveguide Router for Wavelength-Routed Optical Networks-on-Chip, ICCAD'21*, Zhidan Zheng et al.
2) *Topological routing to maximize routability for package substrate*, *DAC'08,* Shenghua Liu et al.

WRONoC Research at TUM

- **Router design and synthesis:**
  - Topology synthesis
    - *CustomTopo (ICCAD'18)*
    - *FAST (DATE'21, TCAD'22)*
  - Topology design
    - *Light (ASP-DAC'21)*
  - Physical synthesis
    - *ToPro (ICCAD'21)*
  - **Topology synthesis + physical synthesis**
    - ***PSION (ISPD'19, TCAD'20, ICCAD'20)***
- Bandwidth maximization: *MaxBW (ASP-DAC'20)*

PSION: Template-Based Synthesis

Topological features:

- Waveguide connection structure
- MRR topological locations
- MRR resonant wavelengths
- Signal wavelength assignment
- Signal path routing

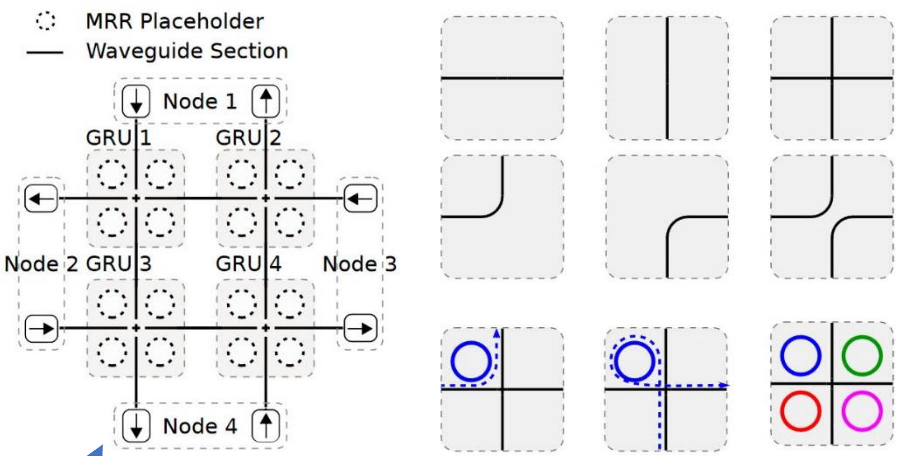
to be determined

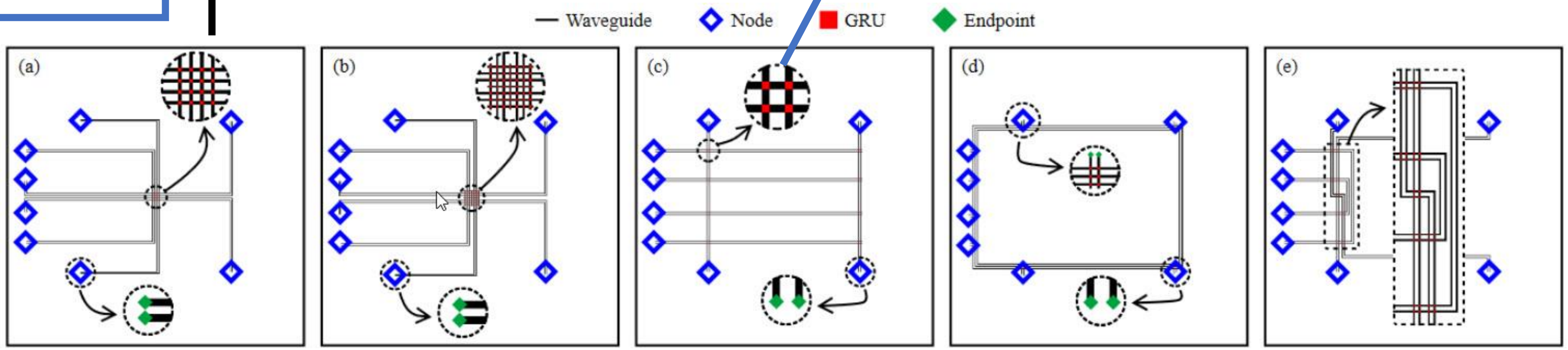
Physical design features:

- Waveguide routing
- MRR placement

fixed

placeholders



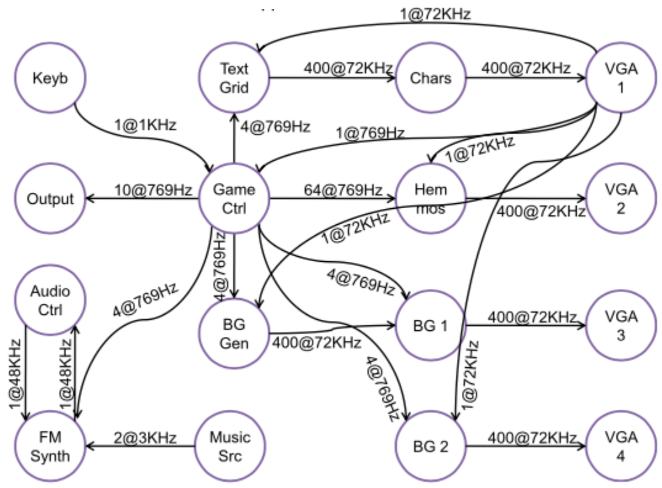


Sources:

1) *PSION: Combining logical topology and physical layout optimization for Wavelength-Routed ONoCs, ISPD'19,* Alexandre Truppel et al.
2) *PSION+: Combining logical topology and physical layout optimization for Wavelength-Routed ONoCs, IEEE TCAD 39(12) 2020,* Alexandre Truppel et al.
3) *PSION 2: Optimizing Physical Layout of Wavelength-Routed ONoCs for Laser Power Reduction,* Alexandre Truppel et al.

WRONoC Synthesis by PSION

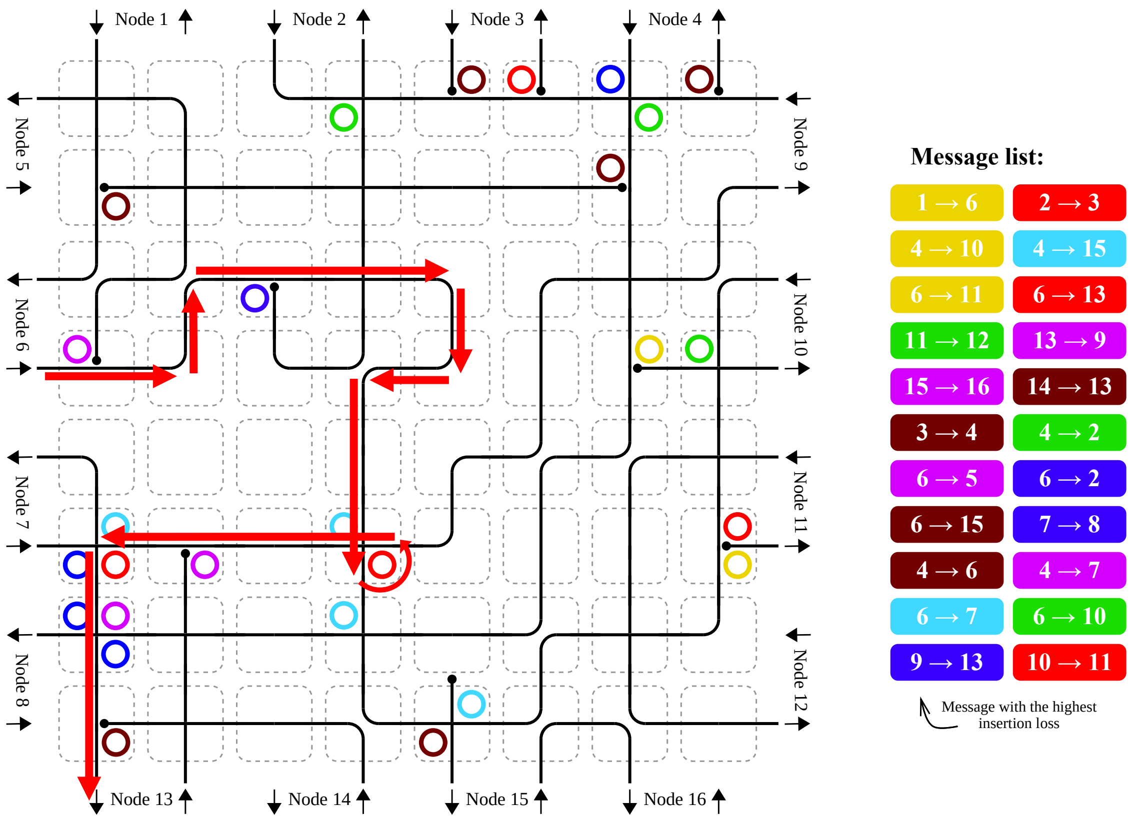
a “Screen Savor” multimedia application



16 nodes, 22 messages

- Full CM would have 240 messages, **240 MRRs** required for Lambda-router
- Here only **27 MRRs** are used

WRONoC router synthesized by PSION



Sources:

1) *PSION: Combining logical topology and physical layout optimization for Wavelength-Routed ONoCs, ISPD’19,* Alexandre Truppel et al.
2) *A scalable, non-interfering, synthesizable Network-on-chip monitor — extended version, Microprocessors and Microsystems’13*, Antti Alhonen et al.

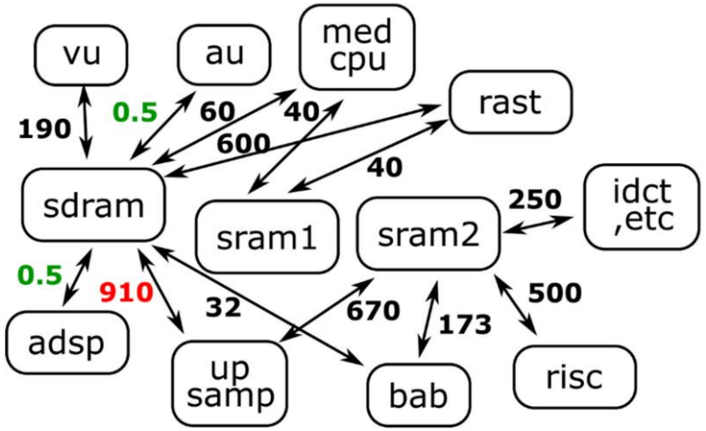
WRONoC Research at TUM

- Router design and synthesis:
  - Topology synthesis
    - *CustomTopo (ICCAD'18)*
    - *FAST (DATE'21, TCAD'22)*
  - Topology design
    - *Light (ASP-DAC'21)*
  - Physical synthesis
    - *ToPro (ICCAD'21)*
  - Topology synthesis + physical synthesis
    - *PSION (ISPD'19, TCAD'20, ICCAD'20)*
- **Bandwidth maximization: *MaxBW (ASP-DAC'20)***

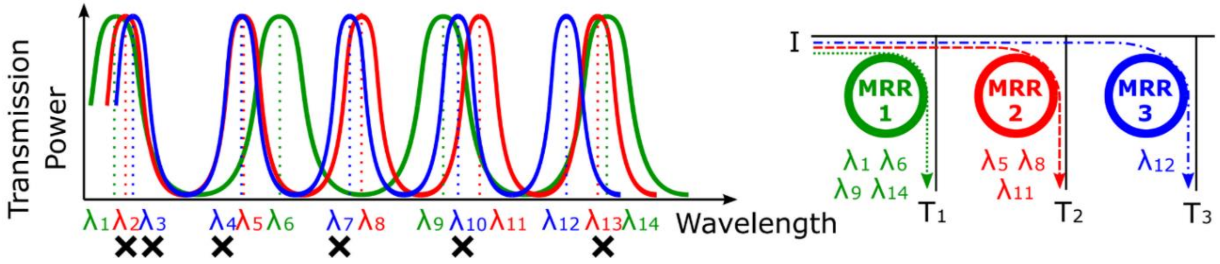
Bandwidth Maximization

- Convention: “1-bit communication”
- Periodic transmission spectrum of microring resonators
- Input: a WRONoC topology
- Output: the same topology with maximized communication parallelism

Bandwidth requirement (unit: MB/s)
of an MPEG-4 decoder application



Transmission spectra of three MRRs.



Sources:
1) *Maximizing the Communication Parallelism for Wavelength-Routed Optical Networks-on-Chips, ASP-DAC’20*, Mengchu Li et al.
2) *NoC synthesis flow for customized domain specific multiprocessor systems-on-chip, IEEE TPDS 16(2) 2008,* Davide Bertozzi et al.

WRONoC Research at TUM — Since 2018

- Router design and synthesis:
  - Topology synthesis
    - *CustomTopo (ICCAD'18)*
    - *FAST (DATE'21, TCAD accepted)*
  - Topology design
    - *Light (ASP-DAC'21)*
  - Physical synthesis
    - *ToPro (ICCAD'21)*
  - Topology synthesis + physical synthesis
    - *PSION (ISPD'19, TCAD'20, ICCAD'20)*
- Bandwidth maximization: *MaxBW (ASP-DAC'20)*

Many thanks to the researchers and students working with me:
Tsun-Ming Tseng,
Mengchu Li, Alexandre Truppel, Zhidan Zheng, Moyuan Xiao
and to my collaborators:

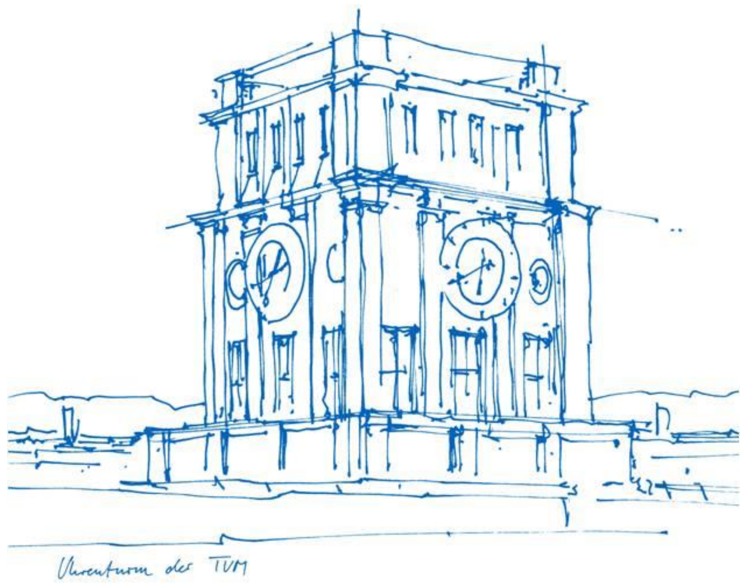
- ***Prof. Davide Bertozzi (University of Ferrara, Italy)***
- ***Dr. Mahdi Tala (University of Ferrara, Italy)***
- ***Prof. Mahdi Nikdast (Colorado State University, USA)***

PIC variation analysis

Thanks to

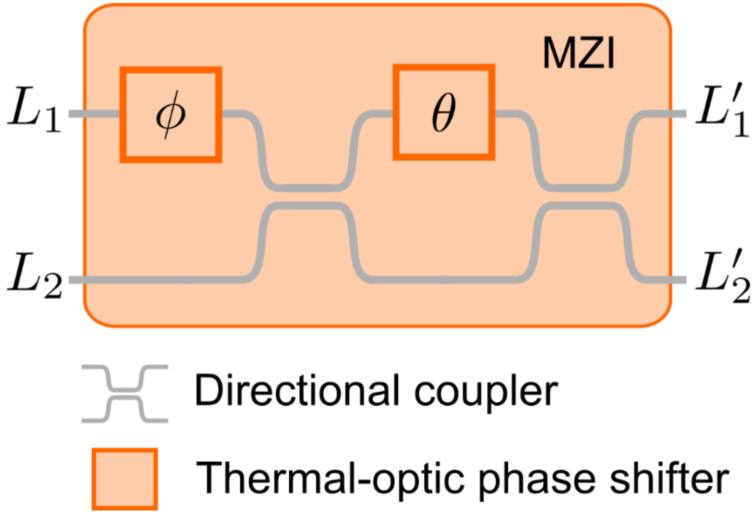
Ying Zhu, Bing Li, Grace Li Zhang (TUM)

and to our collaborators
Xunzhao Yin, Cheng Zhuo (Zhejiang),
Huaxi Gu (Xidian),
Tsung-Yi Ho (CUHK)



Mach-Zehnder Interferometer (MZI)

- Component for light signal transformation
- Behavior of optical signals:
  - Directional coupler (beam splitter): split signal by 50:50; append π/2 in phases of diagonal transmission
  - Phase shifter: thermally controllable phases for programming



transformation matrices of phase shifters

$$\begin{bmatrix} L_1'^c \\ L_2'^c \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L_1^c \\ L_2^c \end{bmatrix} = ie^{i\theta/2} \begin{bmatrix} e^{i\phi}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \\ e^{i\phi}\cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \end{bmatrix} \begin{bmatrix} L_1^c \\ L_2^c \end{bmatrix} = \boxed{\mathbf{T} \begin{bmatrix} L_1^c \\ L_2^c \end{bmatrix}}$$

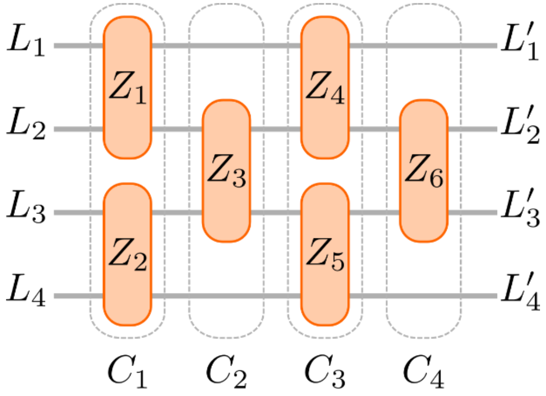
matrix-vector multiplication

transformation matrices of directional couplers

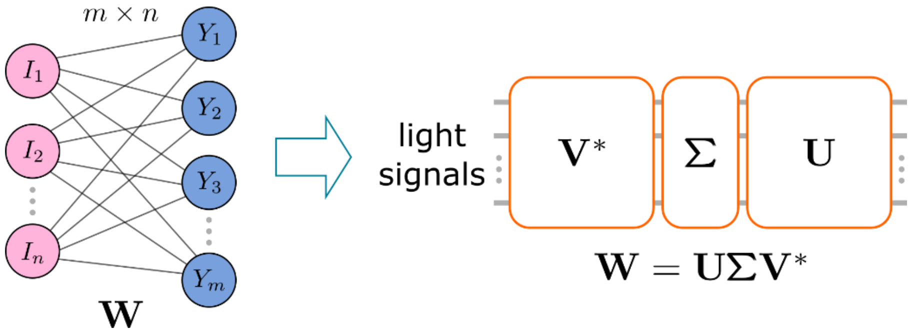
MZI Network as Neural Network

- MZIs can be connected to transform more signals simultaneously

$\mathbf{T}=\mathbf{T}_{C_4}\mathbf{T}_{C_3}\mathbf{T}_{C_2}\mathbf{T}_{C_1}$: multiplication of column matrices formed from the matrices of MZIs



- Neural networks can be mapped onto MZI networks by matrix decomposition

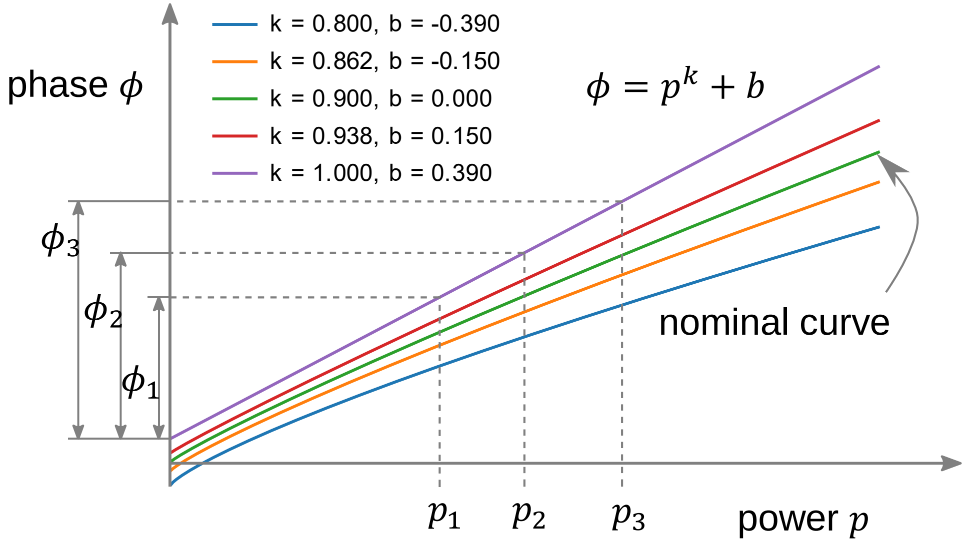


Y. Shen, N. C. Harris, S. Skirlo, et al. *Deep learning with coherent nanophotonic circuits*. naturephotonics, 2017.

Process Variations of MZIs

- Same thermal power results in different phase changes in different MZIs due to process variations

- Smaller MZI phases have smaller deviations.



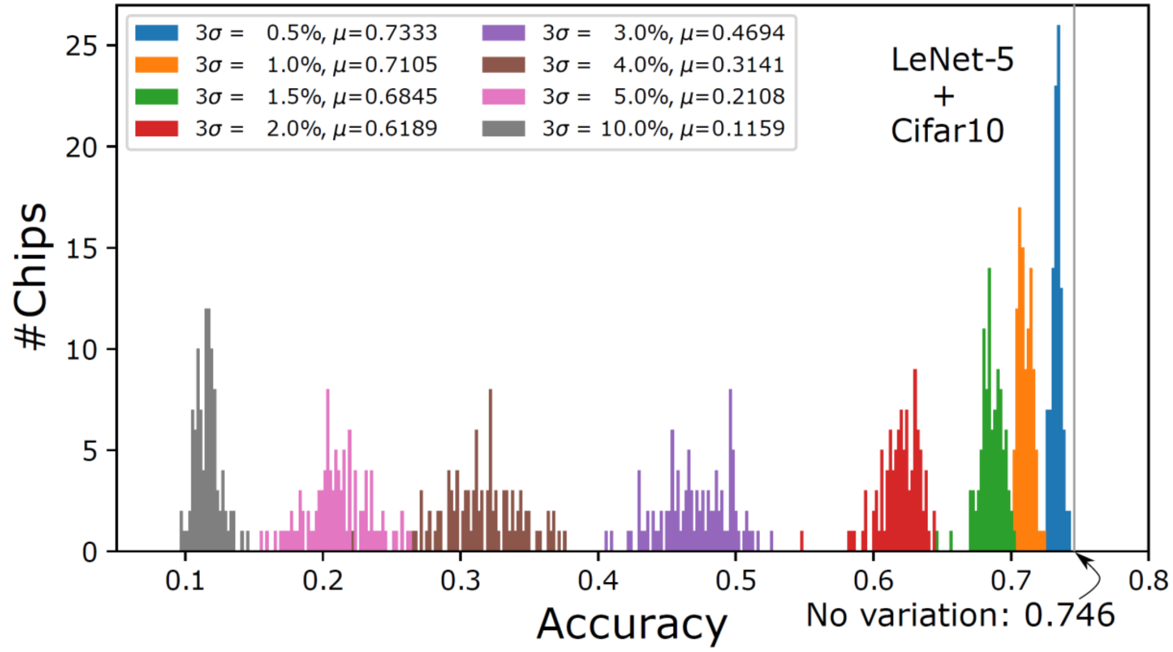
Phase changes *vs* applied power: characteristic curves of five MZIs under process variations

Ma, J. Mower et al. *Efficient, compact and low loss thermo-optic phase shifter in silicon.* Opt. Express, 2014.

Accuracy Degradation of Neural Networks due to Process Variations

- LeNet-5 + Cifar10
- Obvious accuracy drop with 0.5%–1% random variations in the MZI phases
- With beyond 3% variations the optical network becomes unusable.

3σ: variation setting, μ: mean value of accuracy



Ying Zhu, Grace Li Zhang, Bing Li, Xunzhao Yin, Cheng Zhuo, Huaxi Gu, Tsung-Yi Ho, Ulf Schlichtmann. *Countering Variations and Thermal Effects for Accurate Optical Neural Networks*. ICCAD, 2020

Variation Extraction from MZI Network

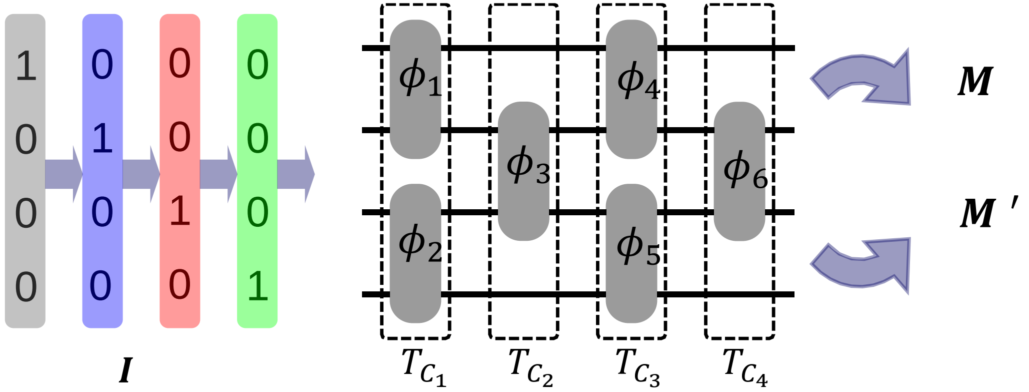
- Input test pattern: Identity matrix I

  $\rightarrow M = T_{C_4}T_{C_3}T_{C_2}T_{C_1}\, I$

- Change MZI phases in column four

  $\rightarrow M' = T'_{C_4}T_{C_3}T_{C_2}T_{C_1} I$

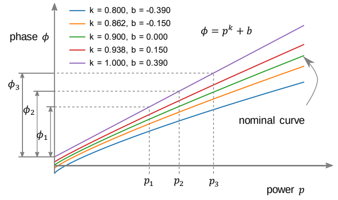
- Determine MZI variations by curve matching and column-wise iterative test



$$M'M^{-1} = (T'_{C_4}T_{C_3}T_{C_2}T_{C_1})(T_{C_4}T_{C_3}T_{C_2}T_{C_1})^{-1} = T'_{C_4}T_{C_4}^{-1}$$

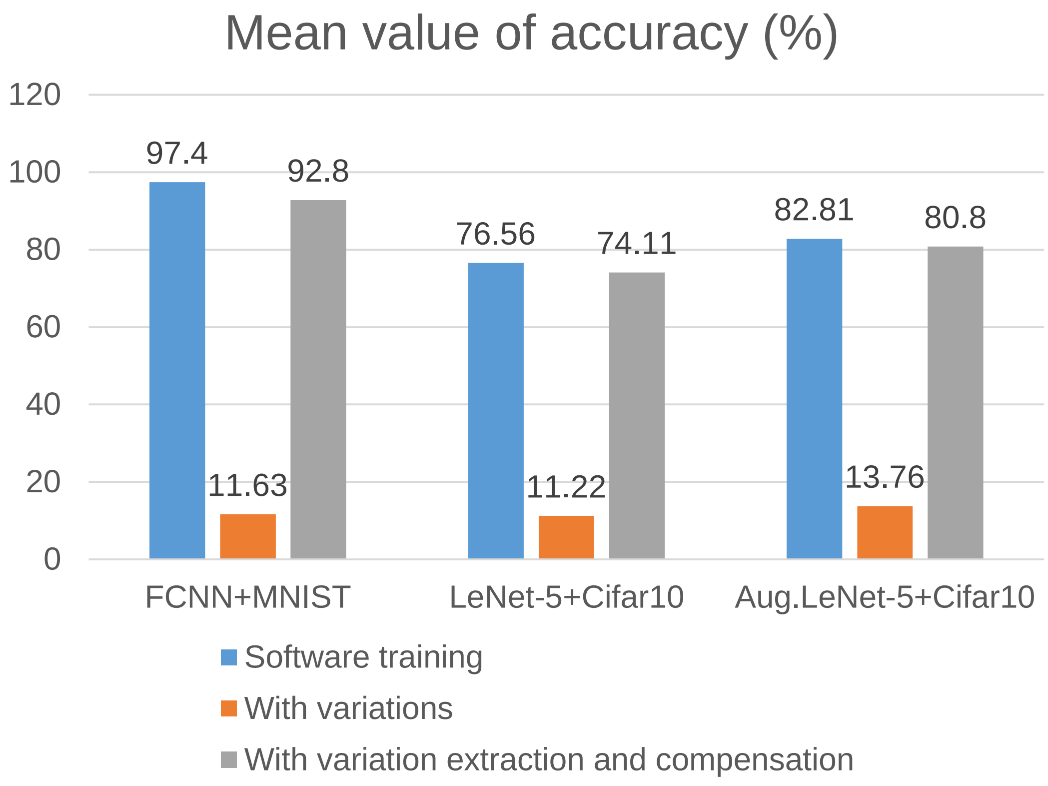
$$= \begin{bmatrix} 1 & \mathbf{0} & 0 \\ \mathbf{0} & T'_6 & \mathbf{0} \\ 0 & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} & 0 \\ \mathbf{0} & T^*_6 & \mathbf{0} \\ 0 & \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} & 0 \\ \mathbf{0} & T'_6T^*_6 & \mathbf{0} \\ 0 & \mathbf{0} & 1 \end{bmatrix}$$

Phase changes in the corresponding MZI



Curve matching to determine variations

Accuracy Enhancement in Variation-aware Design

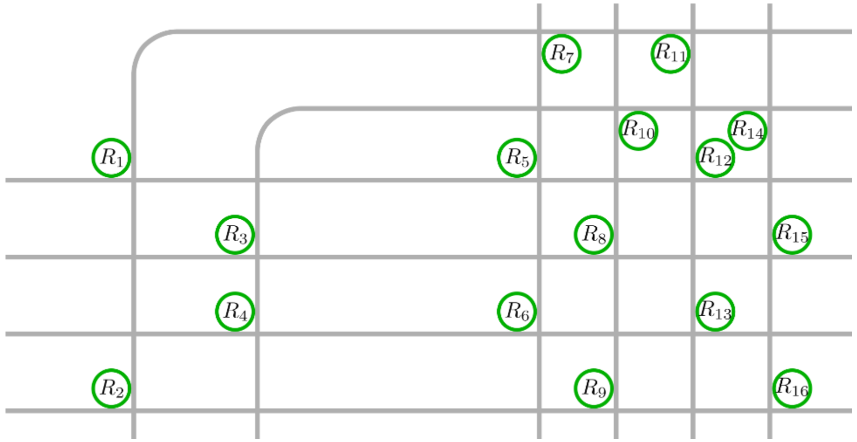


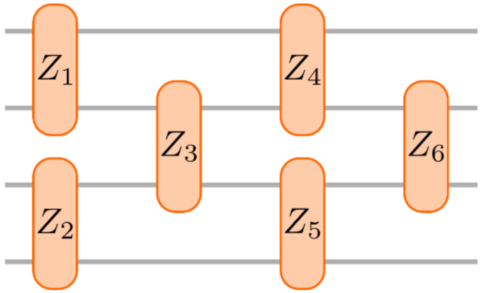
#sampled ONNs: 100; 3σ of the phases at 2π: 20%; Aug. LeNet-5: LeNet-5 with more convolutional layers

Ying Zhu, Grace Li Zhang, Bing Li, Xunzhao Yin, Cheng Zhuo, Huaxi Gu, Tsung-Yi Ho, Ulf Schlichtmann. *Countering Variations and Thermal Effects for Accurate Optical Neural Networks*. ICCAD, 2020

Future Challenges of Optical Systems

- Design and test of optical networks
  - Fault test
  - Variation characterization of complex MRR and MZI networks
- Fusion of optical interconnects and computing components
  - Optical interconnects can create test paths and enable fault tolerance.
  - Overlapped design allows more flexible MZI network structures.
- Computing in the optical domain
  - More functions can be integrated into the optical domain → optic-electro conversion as late as possible
  - Codesign of optical and electrical systems





Tutorial IV: Integrated Programmable Photonic Circuits

Zhengqi Gao, Duane S. Boning

Department of EECS, MIT

July 10th, San Francisco

Terminology

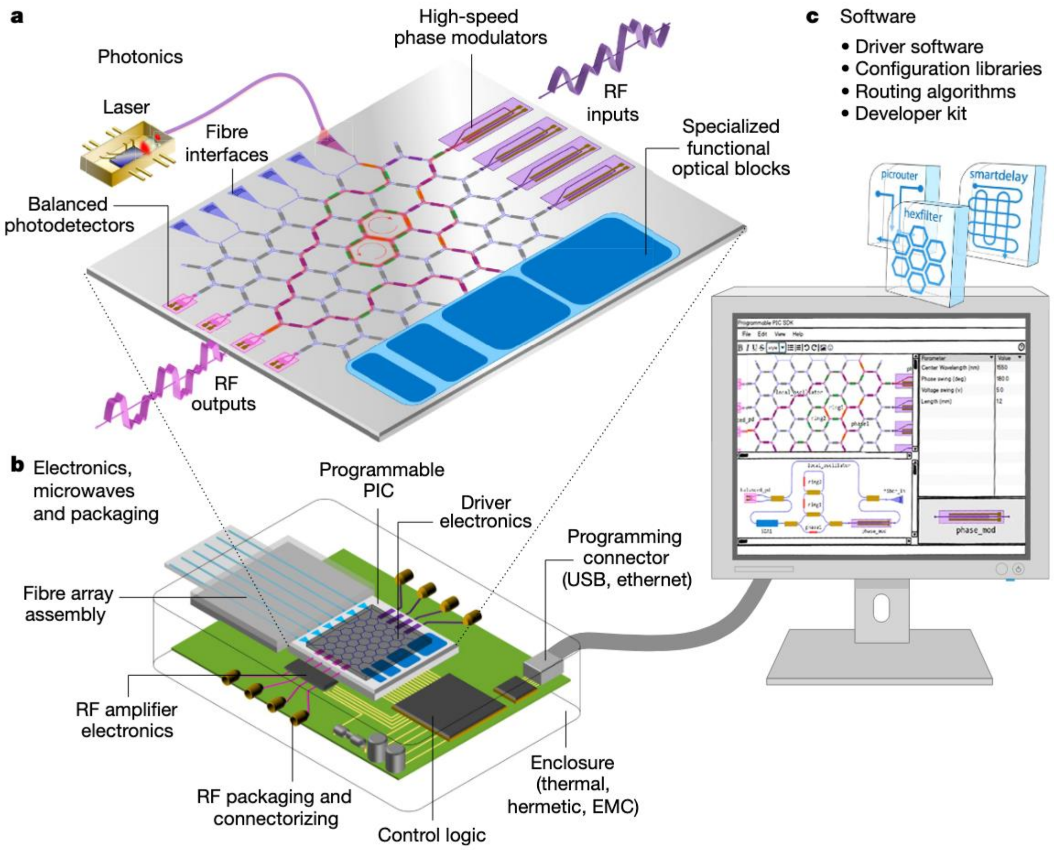


Figure credit: Wim Bogaerts et al., *Nature*, 2020.

- Contrast to bulk optics (which are individual, discretized)
- Integrate multiple optical functions onto a single chip
- Several platforms, mostly used: silicon-based CMOS

Integrated Programmable Photonic Circuit

- Active photonic devices (thermal-optic phase shifter)
- Exploit run-time reconfigurability
- Analogy to the concept of FPGA

- Manipulate light (EM wave), instead of electric signal
- Physical abstraction is $\{\mathbf{E}, \mathbf{H}\}$, instead of $\{\mathbf{I}, \mathbf{V}\}$.
- Simulation more complicated (PDE, Maxwell Equations)

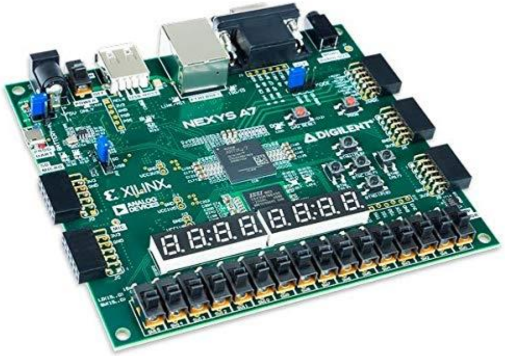
FPGA Reconfigurability

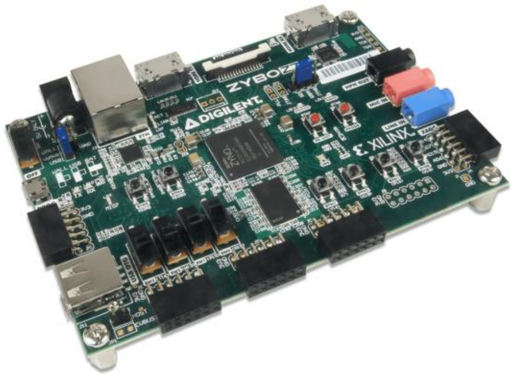
- A large number of logic blocks (e.g., lookup tables, flip-flops, multiplexers)
- An interconnect routing network, which can be programmed

Hardware side

- Program FPGA with HDL (e.g., VHDL, Verilog)

Software side





What about an integrated programmable photonic circuit?

Hardware Side

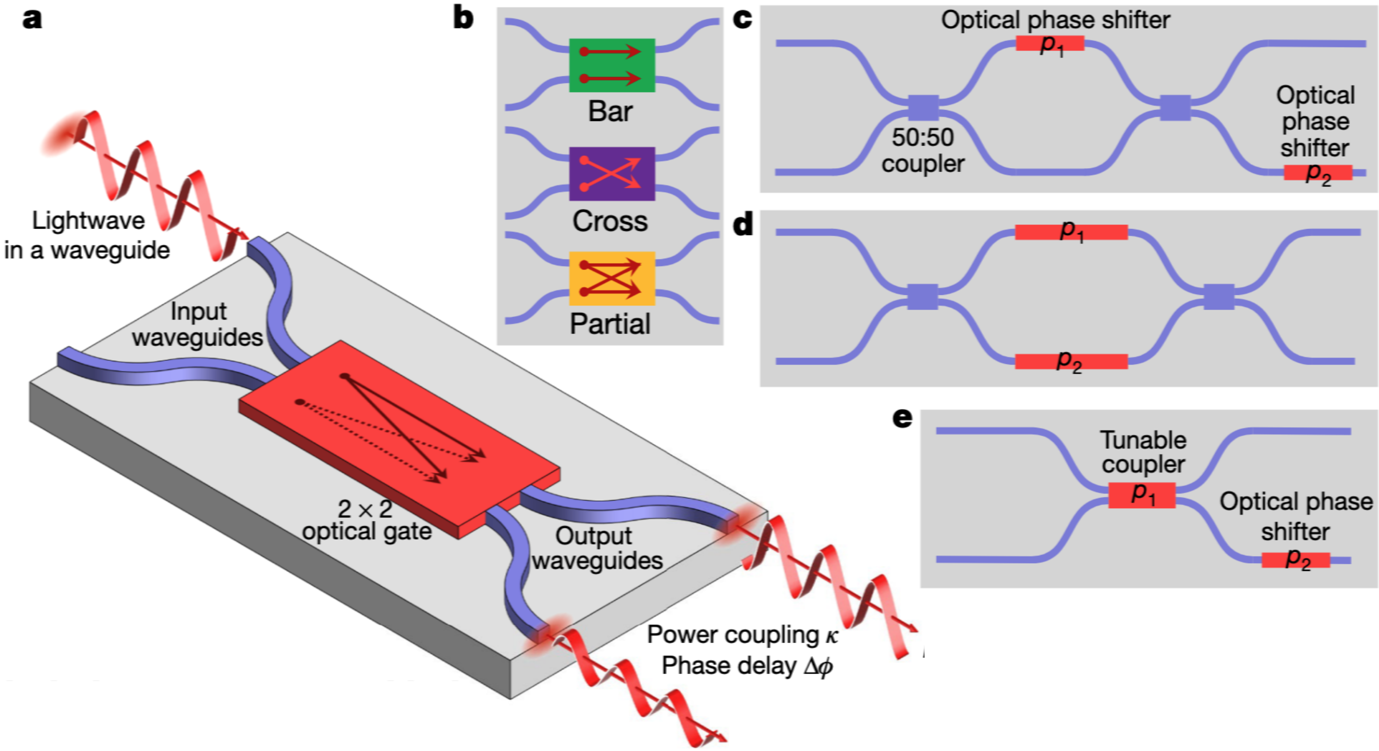


Figure credit: Wim Bogaerts et al., *Nature*, 2020.

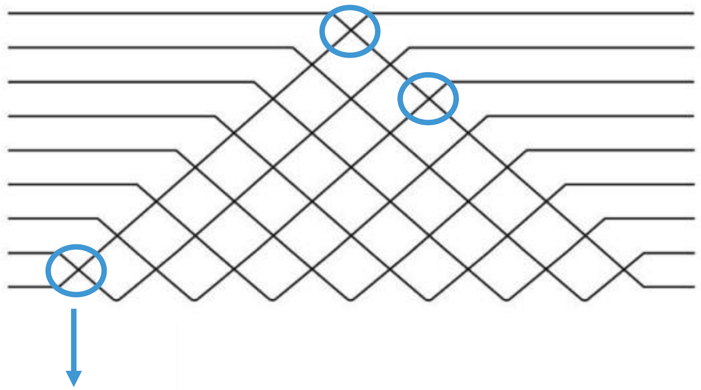
Tunable Basic Unit (TBU)

- An active 2x2 MZI device
- Two degrees of freedom
- Thermal/electric-optical phase shifters
- Three states: bar, cross, partial
- Several implementations (figs. c, d, e)

Remarks: (i) analog computing, (ii) topology difference

Topology I: Feedforward Mesh

Reck's design



One MZI

VOLUME 73, NUMBER 1 PHYSICAL REVIEW LETTERS 4 JULY 1994

Experimental Realization of Any Discrete Unitary Operator

Michael Reck and Anton Zeilinger
Institut für Experimentalphysik, Universität Innsbruck, Technikerstrasse 25, A-6020 Innsbruck, Austria

Herbert J. Bernstein and Philip Bertani
Hampshire College and ISIS, Amherst, Massachusetts 01002
(Received 11 February 1994)

An algorithmic proof that any discrete finite-dimensional unitary operator can be constructed in the laboratory using optical devices is given. Our recursive algorithm factorizes any $N \times N$ unitary matrix into a sequence of two-dimensional beam splitter transformations. The experiment is built from the corresponding devices. This also permits the measurement of the observable corresponding to any discrete Hermitian matrix. Thus optical experiments with any type of radiation (photons, atoms, etc.) exploring higher-dimensional discrete quantum systems become feasible.

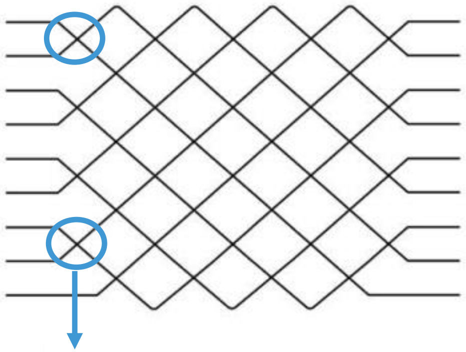
Remarks: (i) Reck's design could implement any complex unitary N-by-N matrix with $N(N\text{-}1)/2$ MZIs.

(ii) Feedforward: light only propogate from left to right, or vice versa; no loops.

Massachusetts Institute of Technology

Topology I: Feedforward Mesh

Clement's design



One MZI

Optimal design for universal multiport interferometers

William R. Clements,* Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley

Clarendon Laboratory, Department of Physics, University of Oxford, Oxford OX1 3PU, UK
**Corresponding author: william.clements@physics.ox.ac.uk*

Received 23 May 2016; revised 7 October 2016; accepted 7 October 2016 (Doc. ID 266897); published 6 December 2016

Universal multiport interferometers, which can be programmed to implement any linear transformation between multiple channels, are emerging as a powerful tool for both classical and quantum photonics. These interferometers are typically composed of a regular mesh of beam splitters and phase shifters, allowing for straightforward fabrication using integrated photonic architectures and ready scalability. The current, standard design for universal multiport interferometers is based on work by Reck *et al.* [Phys. Rev. Lett. 73, 58 (1994)]. We demonstrate a new design for universal multiport interferometers based on an alternative arrangement of beam splitters and phase shifters, which outperforms that by Reck *et al.* Our design requires half the optical depth of the Reck design and is significantly more robust to optical losses.

Remarks: (i) Similar to Reck's: $N(N\text{-}1)/2$ MZIs needed; feedforward.

(ii) Difference: better tolerance to error; more compact.

Topology I: Feedforward Mesh

Singular value decomposition (SVD): $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}$

$\mathbf{M}$ is a complex (real) matrix => $\mathbf{U}$ and $\mathbf{V}$ are unitary (orthogonal) matrices

Motivates a novel DL hardware accelerator: Optical Neural Network

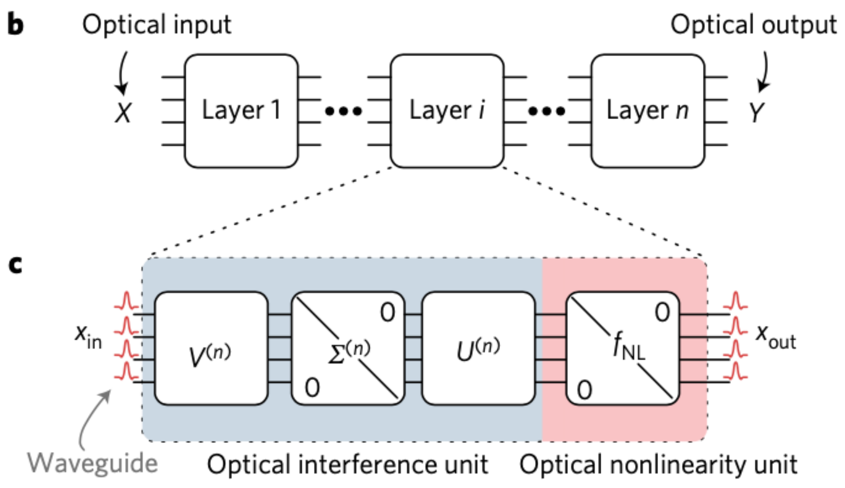
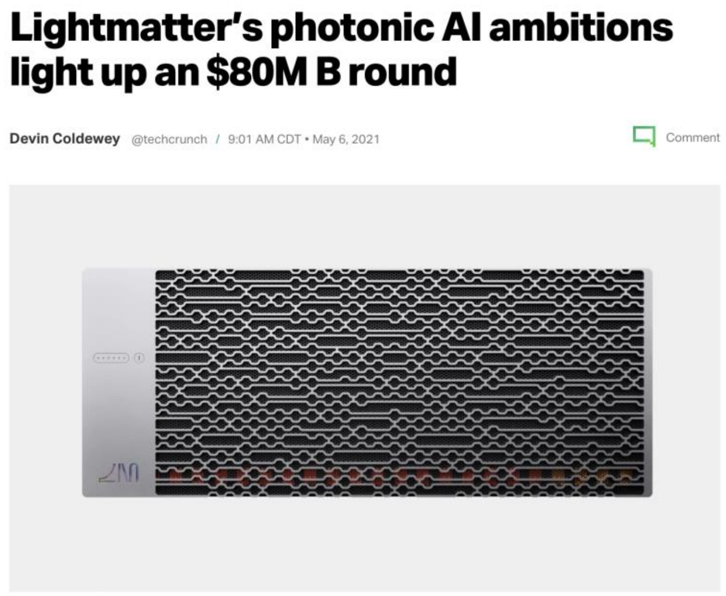


Figure credit: Yichen Shen et al., *Nature*, 2017.

LIGHTMATTER

LIGHTELLIGENCE



Topology I: Feedforward Mesh

Q1: Let's implement an optical ring resonator on a feedforward mesh!

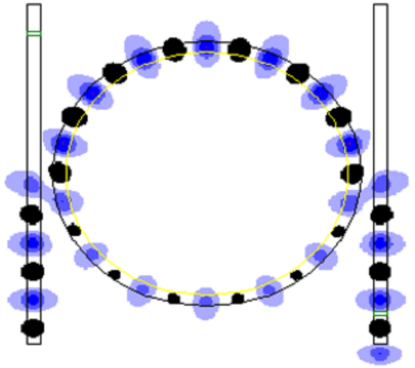
---- We cannot... No closed loops.

Q2: Let's implement an IIR filter on a feedforward mesh!

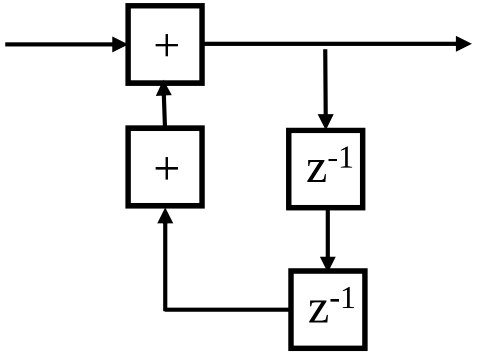
---- Again, we cannot… No closed loops.

Remark: Feedforward mesh is thus more specialized as DL accelator.

For a general optical application?



Ring resonator



IIR filter

Topology II: Recirculating Mesh (Main Focus)

Common realizations: Square, hexagonal, triangular mesh

A “photonic FPGA”: Fast prototyping integrated silicon photonic circuits

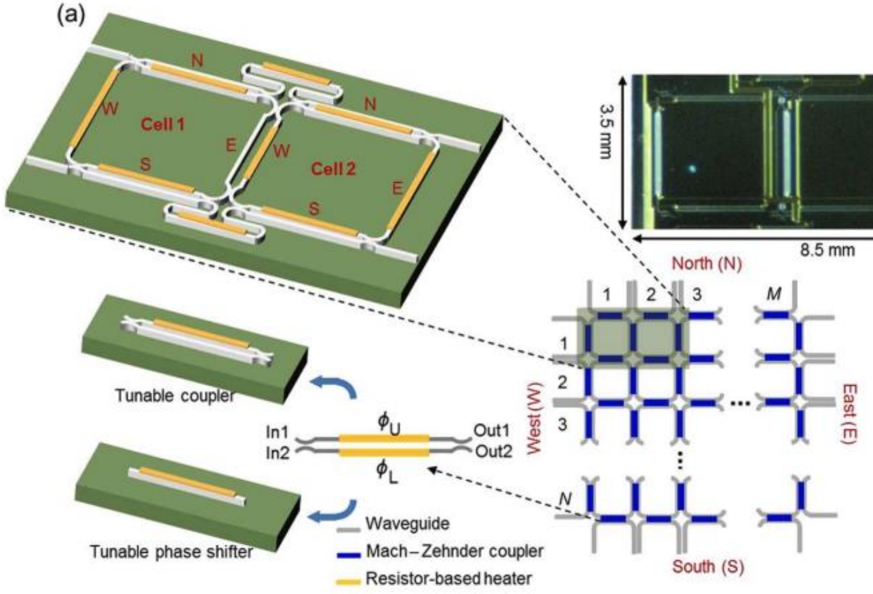


Figure credit: Leimeng Zhuang et al., *Optica*, 2015.

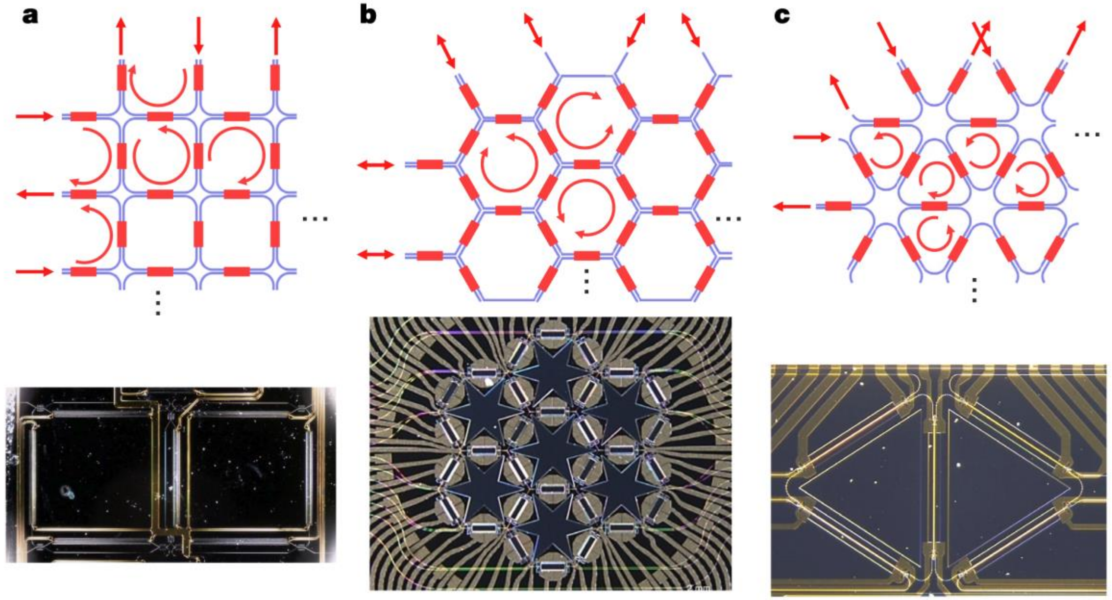


Figure credit: Wim Bogaerts et al., *Nature*, 2020.

Go back to Software Side

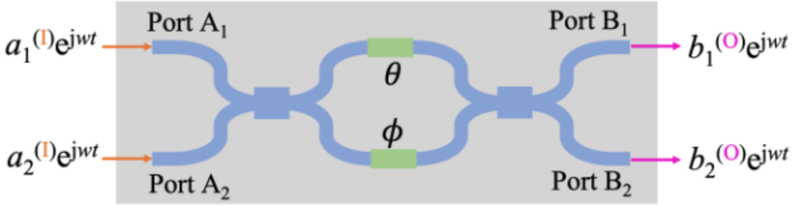
- How do we program it?
  - Recall: mature tools for electronic FPGA; digital.
  - But for programmable photonic circuits, it’s analog computing.
- Feedforward mesh (Reck’s and Clement’s) has analytical solution
- This tutorial will focus on recirculating mesh (less touched)
  - No analytical solution available
  - Take mathematical and algorithmical perspectives

Remark: In a nutshell, we are doing synthesis.

Modeling and Simulation

A time-harmonic chromatic optical signal is represented by: ae^{jwt} (a is complex)

Scattering matrix relation for a TBU

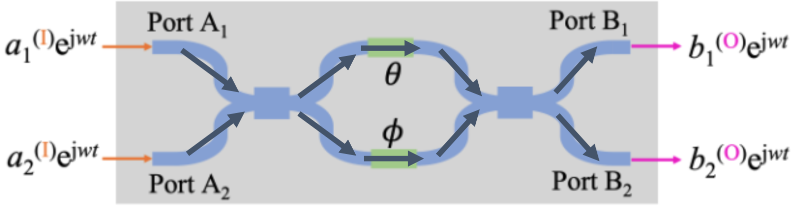


$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} \stackrel{?}{=} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

Modeling and Simulation

A time-harmonic chromatic optical signal is represented by: $a e^{jwt}$ (a is complex)

Scattering matrix relation for a TBU



$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -\mathrm{j} \\ -\mathrm{j} & 1 \end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} e^{-\mathrm{j}\theta} & 0 \\ 0 & e^{-\mathrm{j}\phi} \end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -\mathrm{j} \\ -\mathrm{j} & 1 \end{bmatrix}}_{\text{left DC}} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

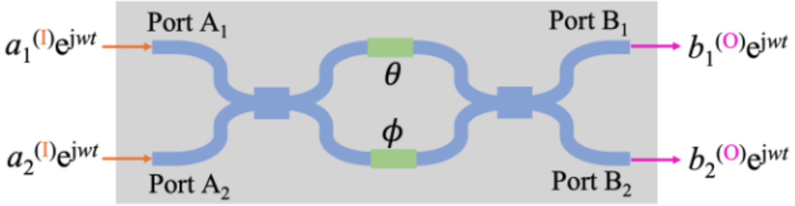
Remarks: (i) This is the form usually used in a feedforward case

(ii) But more careful treatment needs to be done in a recirculating case

Modeling and Simulation

A time-harmonic chromatic optical signal is represented by: ae^{jwt} (a is complex)

Scattering matrix relation for a TBU



$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{left DC}} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

Further take waveguide into consideration:

$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \boxed{\underbrace{\alpha \cdot e^{-j\omega \frac{n_{\text{eff}} L}{c}}}_{\text{waveguide}}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{left DC}} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

$\{\theta, \phi\}$: tunable phase shifts (design variable)

c: light speed in vaccum

$n_{\text{eff}}(w)$: effective index of propogating mode

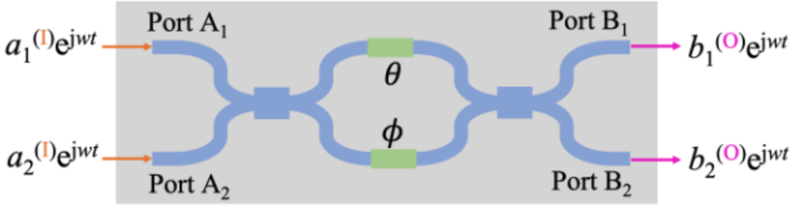
α: tunable basic unit (TBU) loss

L: length of waveguide in the TBU

Modeling and Simulation

A time-harmonic chromatic optical signal is represented by: $a\mathrm{e}^{\mathrm{j}wt}$ (a is complex)

Scattering matrix relation for a TBU



$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \underbrace{\alpha \cdot \mathrm{e}^{-\mathrm{j}\omega \frac{n_{\mathrm{eff}} L}{c}}}_{\text{waveguide}} \underbrace{\frac{\sqrt{2}}{2} \begin{bmatrix} 1 & -\mathrm{j} \\ -\mathrm{j} & 1 \end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} \mathrm{e}^{-\mathrm{j}\theta} & 0 \\ 0 & \mathrm{e}^{-\mathrm{j}\phi} \end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2} \begin{bmatrix} 1 & -\mathrm{j} \\ -\mathrm{j} & 1 \end{bmatrix}}_{\text{left DC}} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

Remark I: Bar, cross, and partial state

Bar state: $|\theta - \phi| = \pi$ Example: $\theta = 0, \ \phi = \pi$ $\Rightarrow$

$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \alpha \mathrm{e}^{-\mathrm{j}\omega \frac{n_{\mathrm{eff}} L}{c}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

Cross state: $\theta = \phi$ Example: $\theta = \phi = -0.5\pi$ $\Rightarrow$

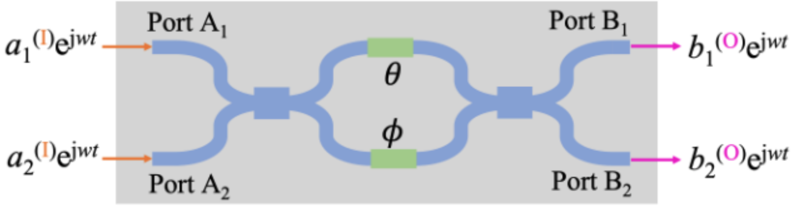
$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \alpha \mathrm{e}^{-\mathrm{j}\omega \frac{n_{\mathrm{eff}} L}{c}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

Other cases are referred to as the partial state

Modeling and Simulation

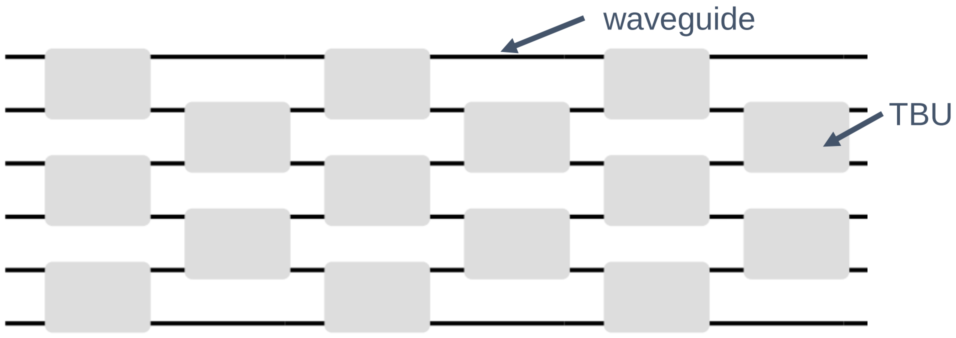
A time-harmonic chromatic optical signal is represented by: ae^{jwt} (a is complex)

Scattering matrix relation for a TBU



$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \underbrace{\alpha \cdot e^{-j\omega \frac{n_{\text{eff}} L}{c}}}_{\text{waveguide}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -j \\ -j & 1\end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi}\end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -j \\ -j & 1\end{bmatrix}}_{\text{left DC}} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)}\end{bmatrix}$$

Remark II: Why it doesn't matter in a feedforward case?

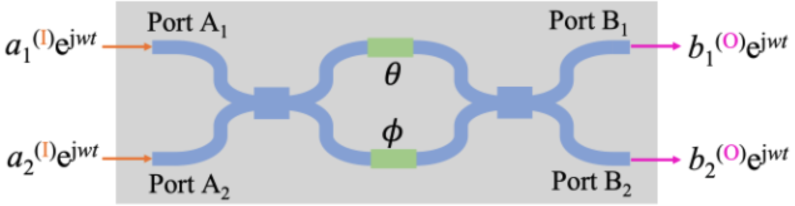


Credit: Saumil Bandyopadhyay et al., *Optica*, 2021.

Modeling and Simulation

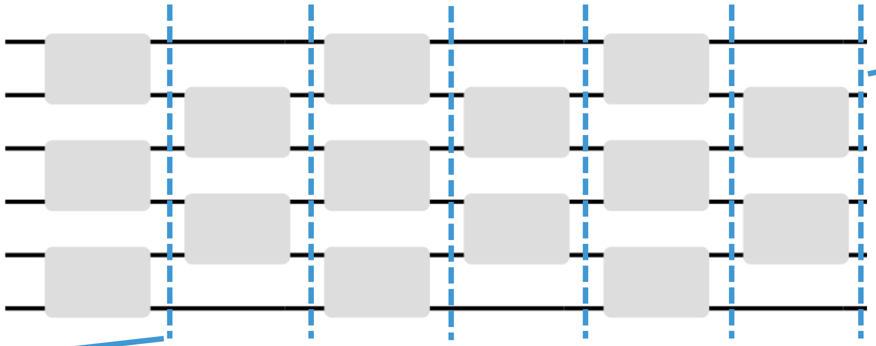
A time-harmonic chromatic optical signal is represented by: $a e^{jwt}$ (a is complex)

Scattering matrix relation for a TBU



$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \underbrace{\alpha \cdot e^{-j\omega \frac{n_{\mathrm{eff}}L}{c}}}_{\text{waveguide}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{left DC}} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

Remark II: Why it doesn't matter in a feedforward case?



All signals $\propto e^{-j\omega\frac{6n_{\mathrm{eff}}L}{c}}$

where 6 is the number of columns

Thus, we could omit the impact of waveguide

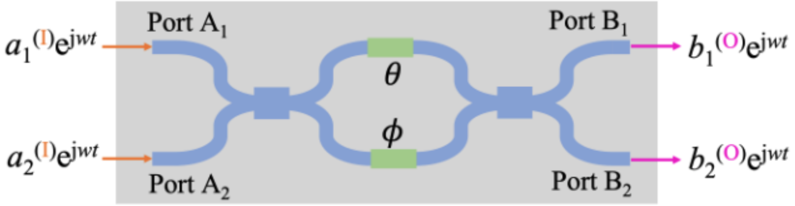
All signals $\propto e^{-j\omega\frac{n_{\mathrm{eff}}L}{c}}$

Figure credit: Saumil Bandyopadhyay et al., *Optica*, 2021.

Modeling and Simulation

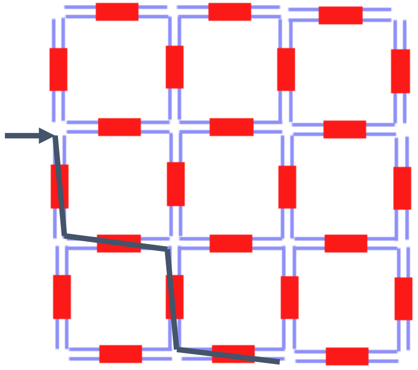
A time-harmonic chromatic optical signal is represented by: $a\mathrm{e}^{\mathrm{j}wt}$ (a is complex)

Scattering matrix relation for a TBU

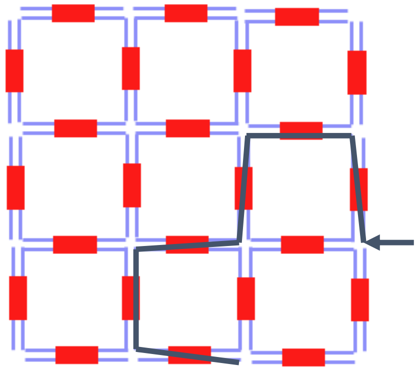


$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \underbrace{\alpha \cdot \mathrm{e}^{-\mathrm{j}\omega\frac{n_{\mathrm{eff}}L}{c}}}_{\text{waveguide}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -\mathrm{j} \\ -\mathrm{j} & 1 \end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} \mathrm{e}^{-\mathrm{j}\theta} & 0 \\ 0 & \mathrm{e}^{-\mathrm{j}\phi} \end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -\mathrm{j} \\ -\mathrm{j} & 1 \end{bmatrix}}_{\text{left DC}} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

Remark III: Why it does matter in a recirculating case?



$\mathrm{e}^{-\mathrm{j}\omega\frac{4n_{\mathrm{eff}}L}{c}}$ dependence

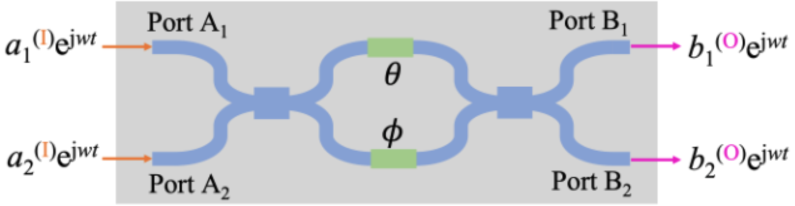


$\mathrm{e}^{-\mathrm{j}\omega\frac{6n_{\mathrm{eff}}L}{c}}$ dependence

Modeling and Simulation

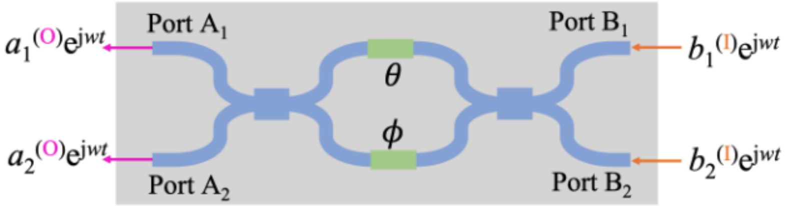
A time-harmonic chromatic optical signal is represented by: $a e^{jwt}$ (a is complex)

Scattering matrix relation for a TBU



$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \mathbf{F} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix} = \underbrace{\alpha \cdot e^{-j\omega \frac{n_{\text{eff}} L}{c}}}_{\text{waveguide}} \underbrace{\frac{\sqrt{2}}{2} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{left DC}} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

Remark IV: TBU is a bi-directional device:



$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \\ a_1^{(O)} \\ a_2^{(O)} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \\ b_1^{(I)} \\ b_2^{(I)} \end{bmatrix}$$

$\{\theta, \phi\}$: tunable phase shifts (design variable)

c: light speed in vaccum

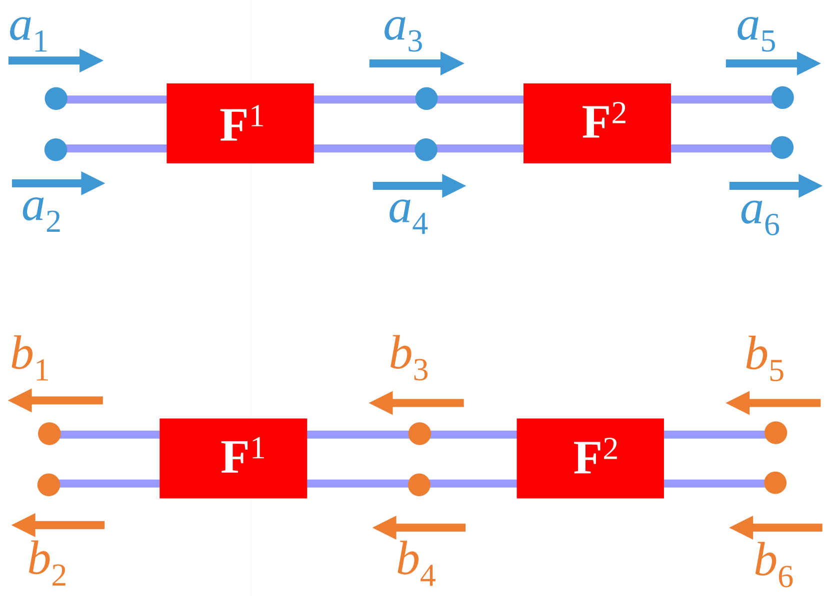
$n_{\text{eff}}(w)$: effective index of propogating mode

α: tunable basic unit (TBU) loss

L: length of waveguide in the TBU

Modeling and Simulation

Frequency-domain scattering matrix simulation



$$\begin{bmatrix} a_3 \\ a_4 \end{bmatrix} = \mathbf{F}^1 \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} F^1_{11} & F^1_{12} \\ F^1_{21} & F^1_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

$$\begin{aligned} F^1_{11} b_3 + F^1_{12} b_4 - b_1 &= 0 \\ F^1_{21} b_3 + F^1_{22} b_4 - b_2 &= 0 \end{aligned}$$

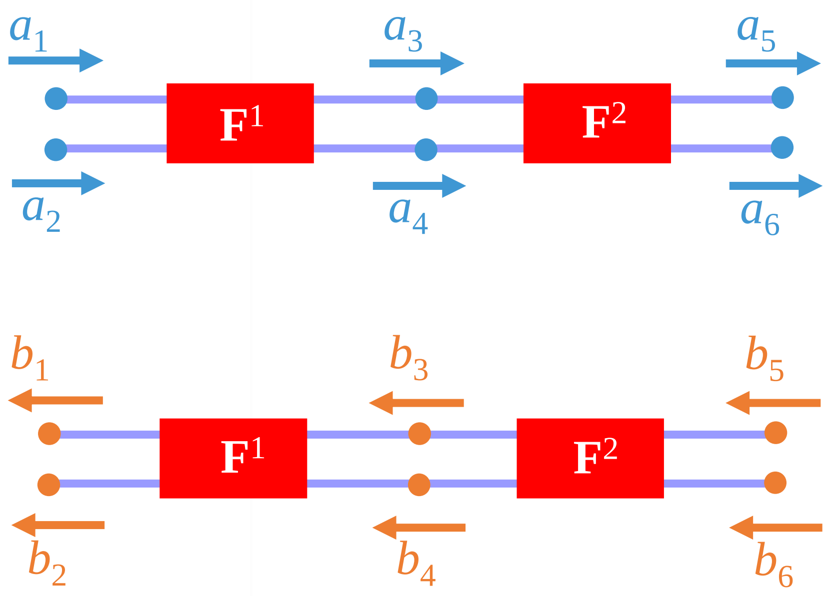
$$\begin{aligned} F^1_{11} a_1 + F^1_{12} a_2 - a_3 &= 0 \\ F^1_{21} a_1 + F^1_{22} a_2 - a_4 &= 0 \end{aligned}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \mathbf{F}^1 \begin{bmatrix} b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} F^1_{11} & F^1_{12} \\ F^1_{21} & F^1_{22} \end{bmatrix} \begin{bmatrix} b_3 \\ b_4 \end{bmatrix}$$

A system of linear equations!

Modeling and Simulation

Frequency-domain scattering matrix simulation



$$F^1_{11}a_1 + F^1_{12}a_2 - a_3 = 0$$

$$F^1_{21}a_1 + F^1_{22}a_2 - a_4 = 0$$

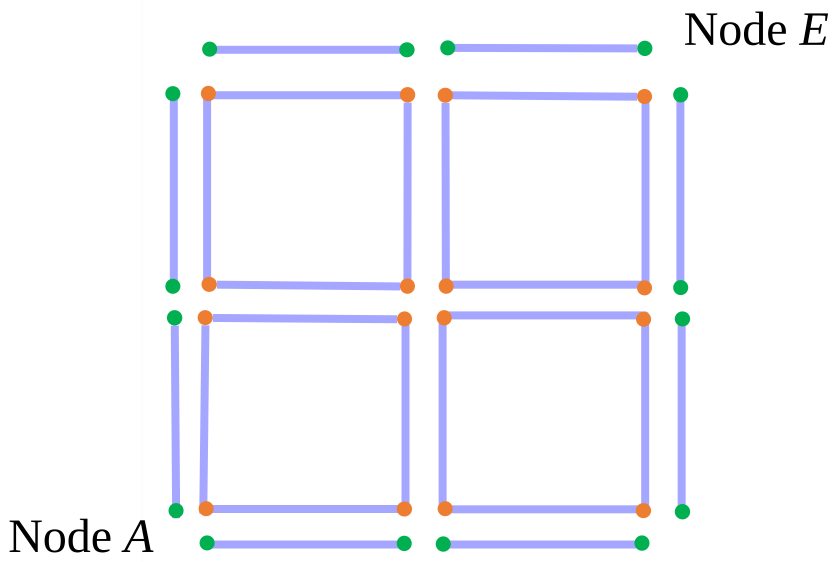
$$F^1_{11}b_3 + F^1_{12}b_4 - b_1 = 0$$

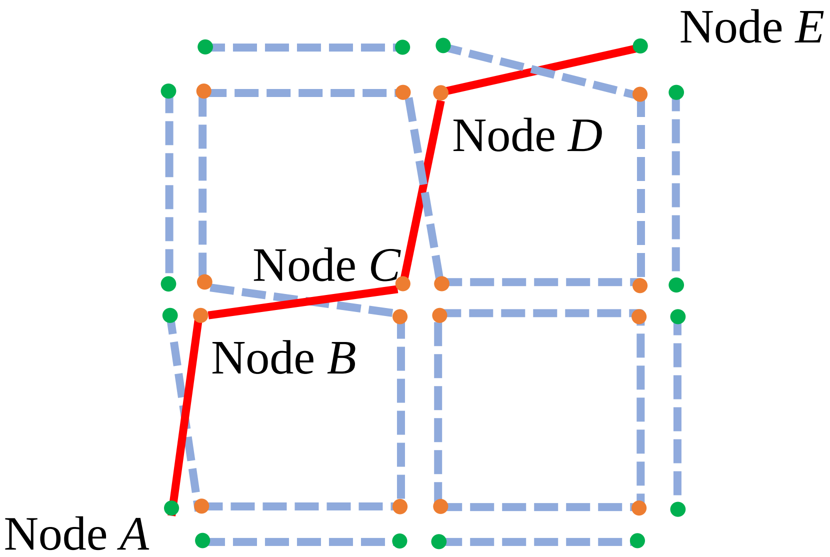
$$F^1_{21}b_3 + F^1_{22}b_4 - b_2 = 0$$

$$\begin{bmatrix} F^1_{11} & 0 & F^1_{12} & 0 & -1 & 0 & 0 & 0 & \cdots \\ F^1_{21} & 0 & F^1_{22} & 0 & 0 & -1 & 0 & 0 & \cdots \\ 0 & -1 & 0 & 0 & 0 & F^1_{11} & 0 & F^1_{12} & \cdots \\ 0 & 0 & 0 & -1 & 0 & F^1_{21} & 0 & F^1_{22} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \\ a_3 \\ b_3 \\ a_4 \\ b_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \mathbf{u} \end{bmatrix}$$

Routing Analysis

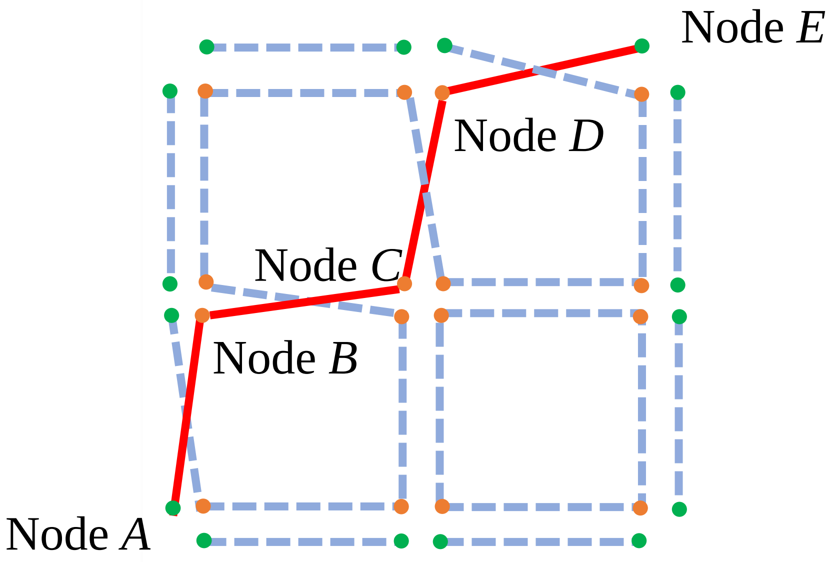
- Reasonable assumption: all TBUs in bar or cross states because of ‘routing’.
- Example: How to route an optical signal from node A to node E?   -- Fairly easy





Routing Analysis

- Reasonable assumption: all TBUs in bar or cross states because of ‘routing’.
- Example: How to route an optical signal from node A to node E? -- Fairly easy



Recall the S-matrix of a TBU in bar/cross state:

$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \alpha e^{-j\omega \frac{n_{\text{eff}} L}{c}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \alpha e^{-j\omega \frac{n_{\text{eff}} L}{c}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}$$

The frequency response of is: $(\alpha e^{-j\omega \frac{n_{\text{eff}} L}{c}})^4$

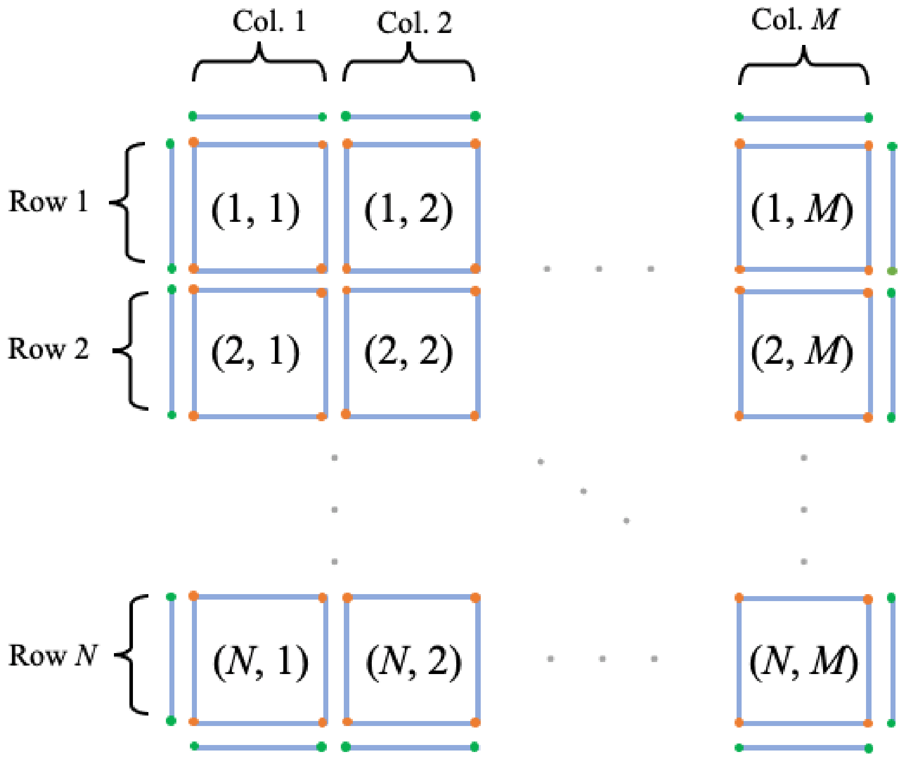
‘4’ represents the number of TBUs the trajectory bypasses

Routing Analysis

- Reasonable assumption: all TBUs in bar or cross states because of ‘routing’.
- Define Path length = #TBUs bypassed
- Analyzing path length is very important
  - It determines the frequency response (previous page)
  - Application I: N signals, goes through the programmable photonic circuit, maintaining phases
    - Realize N paths with the same path length.
  - Application II: Work as time delay element for filtering
    - Realize paths with length constructing arithmetic sequence, e.g., {1,3,5,7,…}.

Routing Analysis

Conclusion I (warm up)



TBUs = $N(M+1) + M(N+1)$

Configs = $2^{N(M+1)+M(N+1)}$

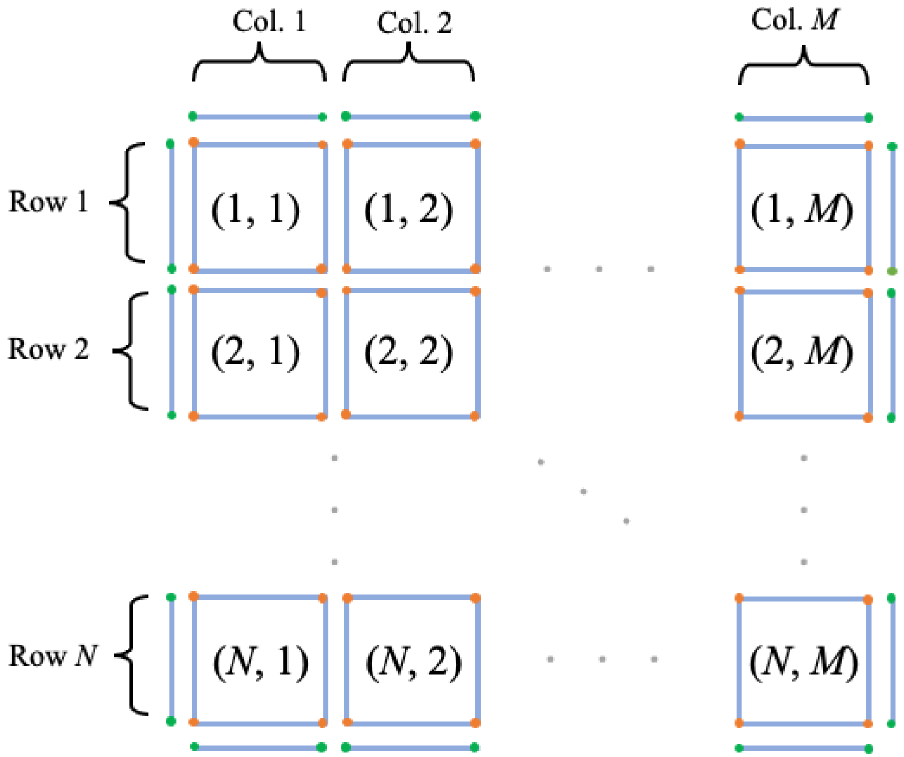
floating nodes = $4N + 4M$

non-floating nodes = $4NM$

undirected optical path = $2N + 2M$

Routing Analysis

Conclusion II: maximum path length = $4NM + 1$



Intuition: a path starts and ends both at a floating node, with non-floating nodes in the middle.

Path length = #Nodes - 1

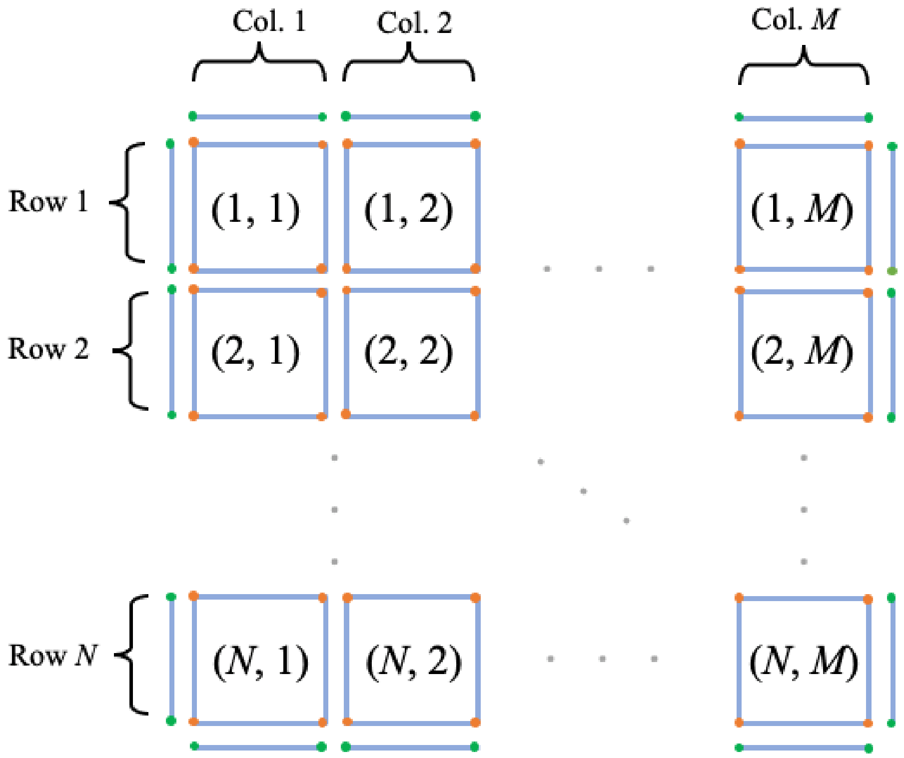
where #Nodes = 2 + #Non-floating Nodes

=> Max #Nodes = $4NM + 2$

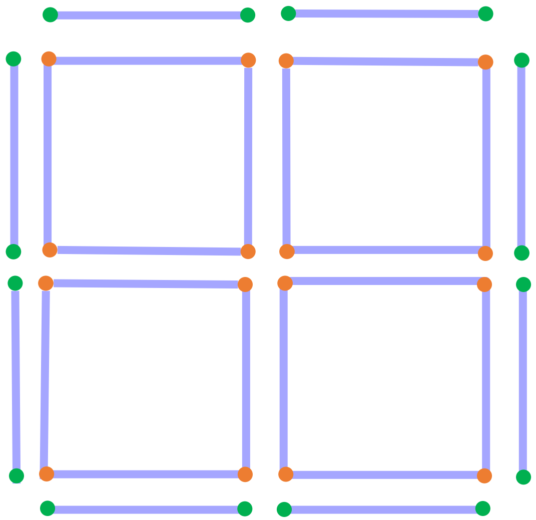
=> Max Path length = $4NM + 1$

Routing Analysis

Conclusion III: Is any path length x in $[1, 4NM+1]$ realizable on a N-by-M square mesh? Unluckily, no….



Example: Try $x = 3$ on this 2-by-2 squre mesh

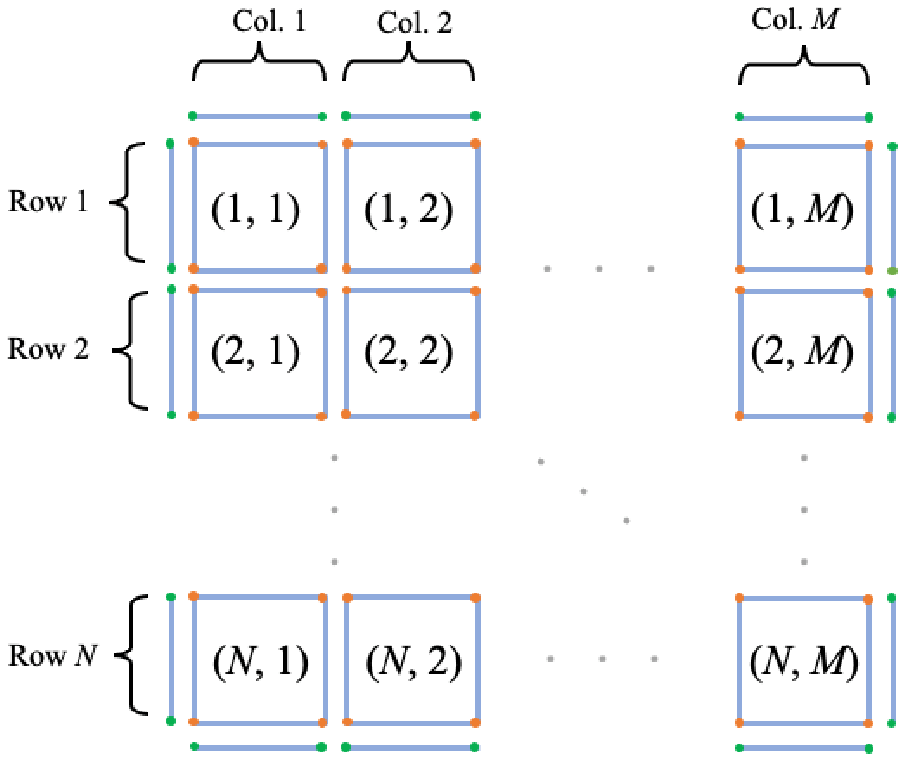


Our finding: If both N and M are even,

Any $x = 0, 1, 2 \pmod 4$ in $[1, 4NM+1]$ is realizable

Routing Analysis

Conclusion III: Is any path length x in $[1, 4NM+1]$ realizable on a N-by-M square mesh?



Our findings:

If both N and M are even,

Any $x = 0, 1, 2 \pmod 4$ in $[1, 4NM+1]$ is realizable

If both N is even and M is odd,

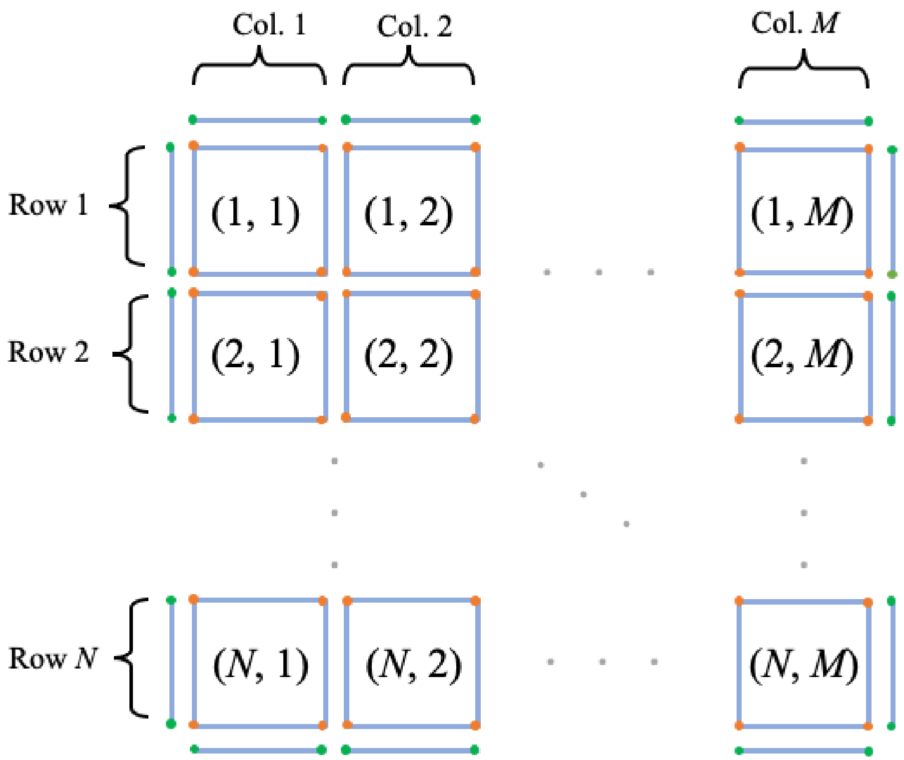
Any $x = 0, 1, 2 \pmod 4$ in $[1, 4NM+1]$ is realizable

Any $x = 3 \pmod 4$ in $[2M + 1, 4NM+1- 2M]$ is realizable

Other cases….

Single path reliazability

Routing Analysis



Other questions when multiple paths considered:

- Recall there are $(2N + 2M)$ paths in total, what are their sum and standard deviation?
- Given a N-by-M square mesh, and a desired path length x, how many paths could we realize?

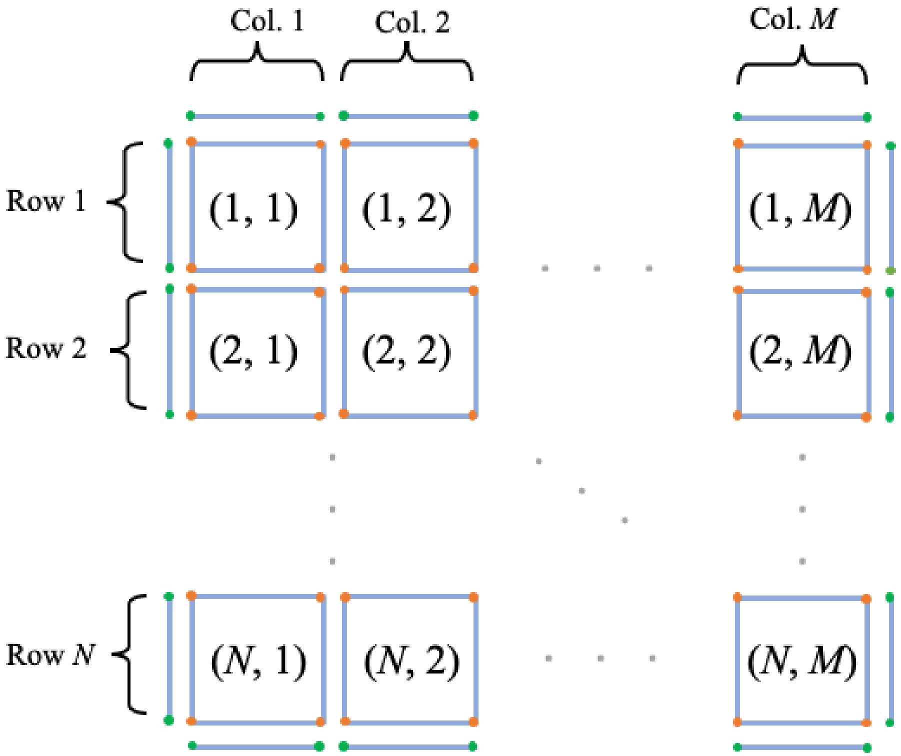
Refer to: https://arxiv.org/abs/2306.12607

Functional Synthesis

- Route several signals on this programmable photonic circuits?
  - Preliminary investigation published in the literature (See Aitor Lopez et al., OE, 2020)
  - Our view: still an open problem, missing strict analysis
  - Graph theory might be helpful

- Besides routing, we also want other functions, e.g., splitting, filtering, WDM
  - Most demos are hand crafted: ✗ size goes up, realize several functions.
  - Can we automatically synthesize light processing function?
  - Use analytical synthesis? --- No closed form for recirculating structure

Functional Synthesis



\# TBUs = $N(M + 1) + M\,(N + 1)$

\# Phase shifts = $2N(M + 1) + 2M(N + 1) \sim 4NM$

We want to adjust phase shifts, to realize a desired function.

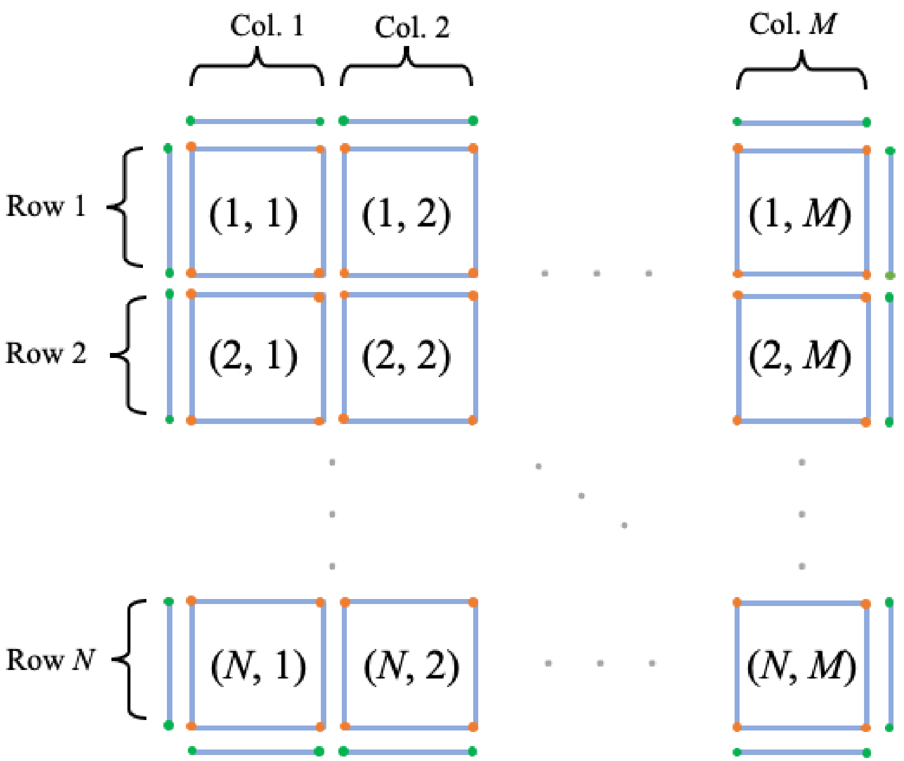
Formulate as an optimization problem!

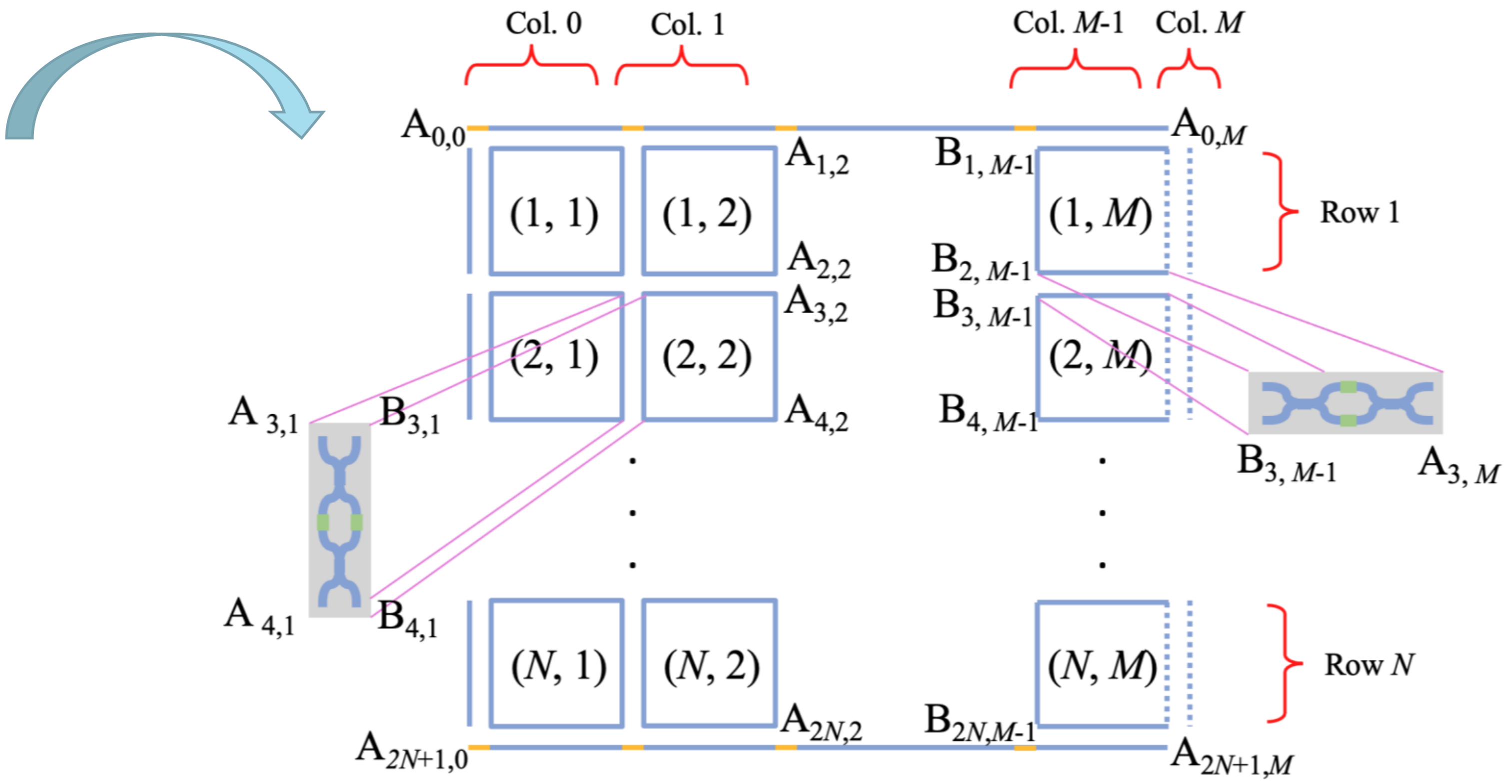
Challenge: high-dimensional space

Efficient solution: Gradient descent w/ analytical gradients

Functional Synthesis

We do a simplification

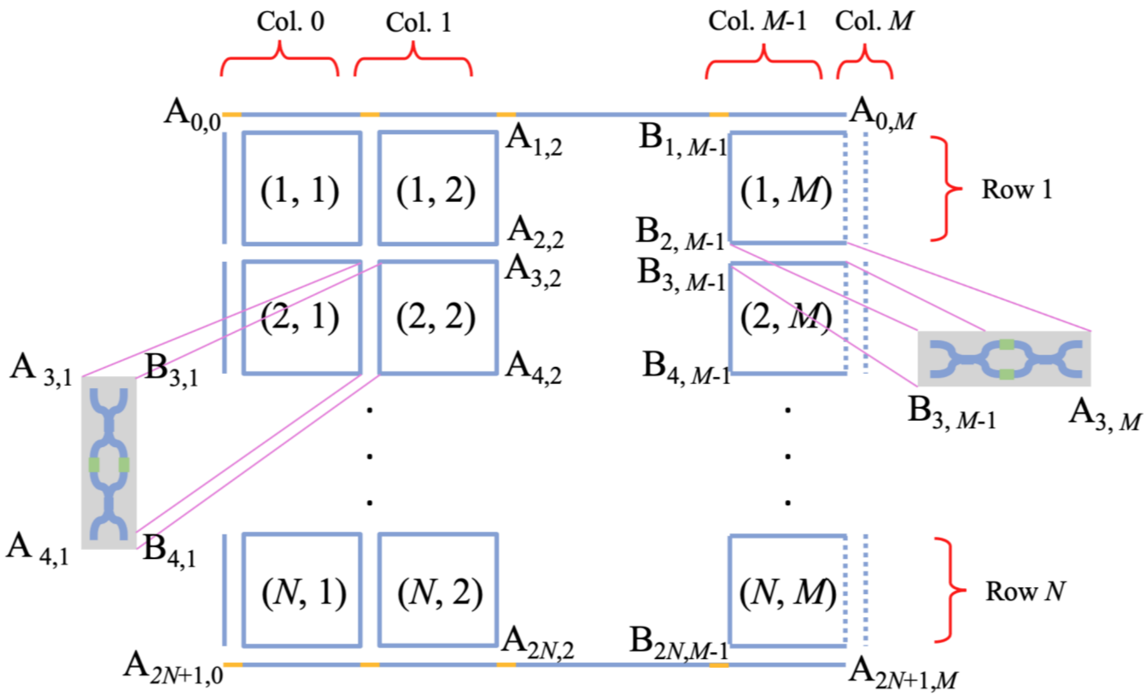


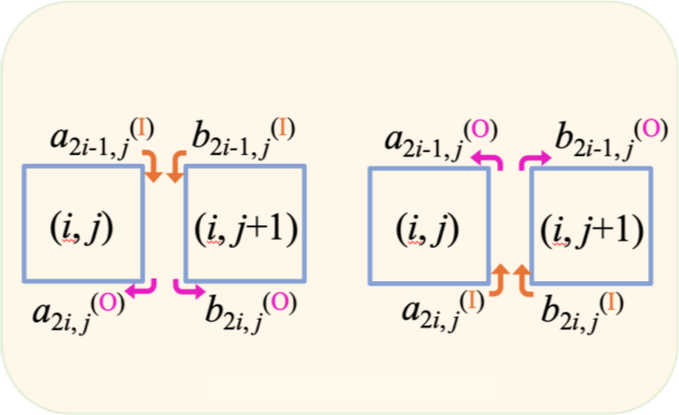


Remark: We consider this simplified case, so that we could derive the transfer function analytically

Functional Synthesis

V matrix: Scattering matrix relation for a vertical TBU

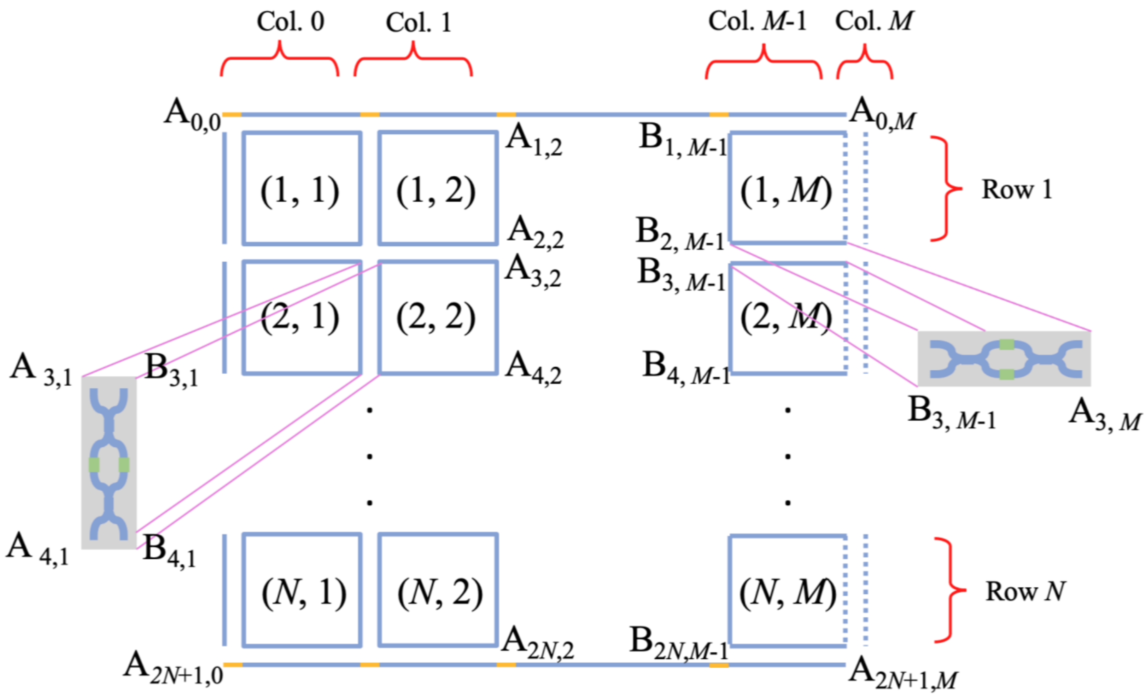


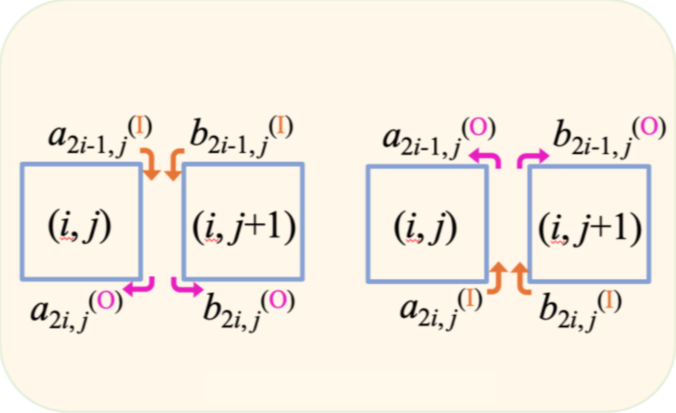


$$\begin{bmatrix} a^{(O)}_{2i,j} \\ b^{(O)}_{2i,j} \\ a^{(O)}_{2i-1,j} \\ b^{(O)}_{2i-1,j}\end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}\end{bmatrix}\begin{bmatrix} a^{(I)}_{2i-1,j} \\ b^{(I)}_{2i-1,j} \\ a^{(I)}_{2i,j} \\ b^{(I)}_{2i,j}\end{bmatrix}$$

Functional Synthesis

V matrix: Scattering matrix relation for a vertical TBU

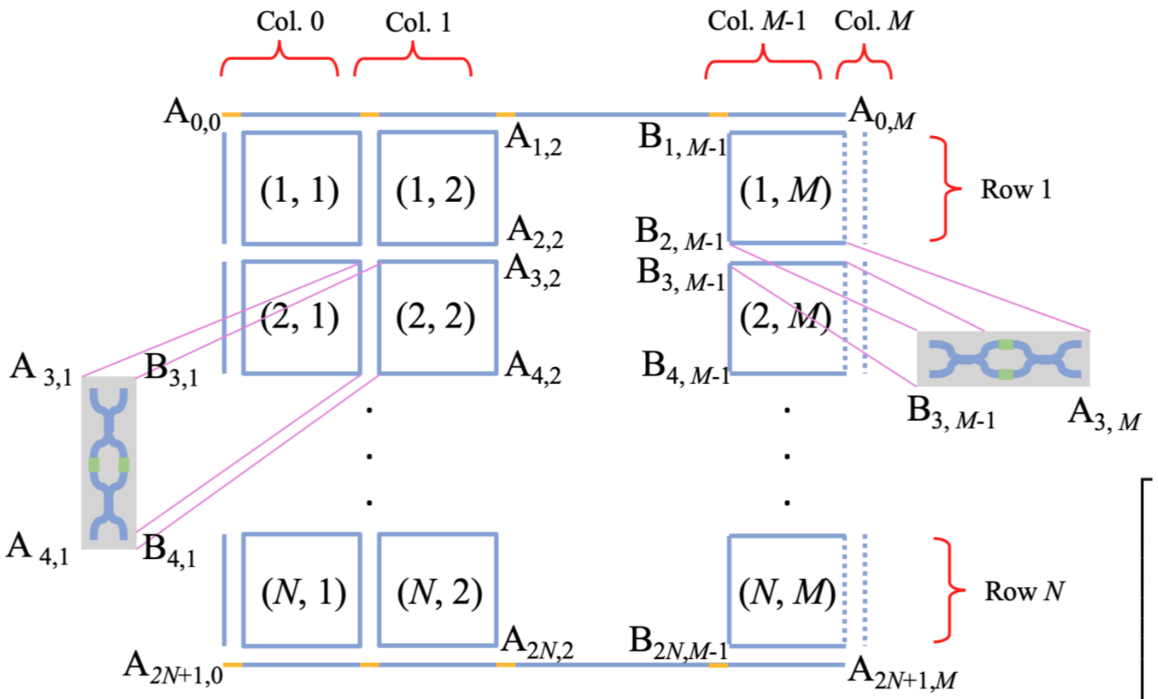


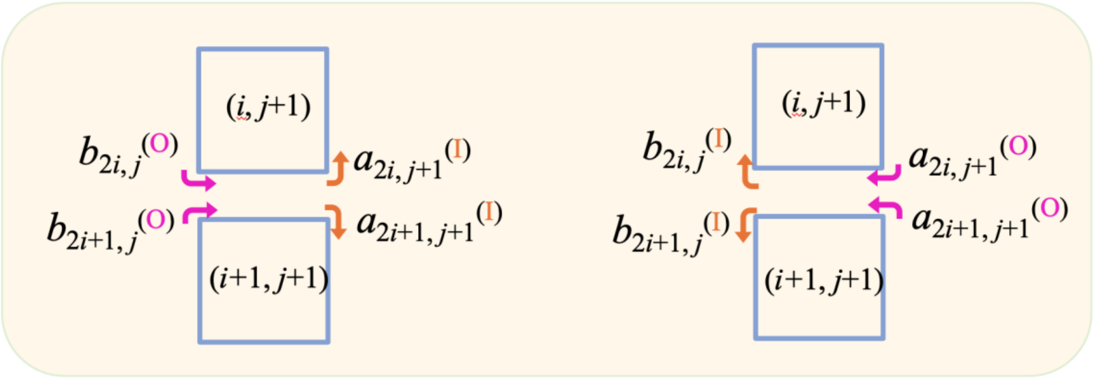


$$\begin{bmatrix} a^{(O)}_{2i,j} \\ b^{(O)}_{2i,j} \\ a^{(O)}_{2i-1,j} \\ b^{(O)}_{2i-1,j} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} a^{(I)}_{2i-1,j} \\ b^{(I)}_{2i-1,j} \\ a^{(I)}_{2i,j} \\ b^{(I)}_{2i,j} \end{bmatrix} \longrightarrow \begin{bmatrix} b^{(I)}_{2i-1,j} \\ b^{(O)}_{2i-1,j} \\ b^{(I)}_{2i,j} \\ b^{(O)}_{2i,j} \end{bmatrix} = \mathbf{V} \begin{bmatrix} a^{(I)}_{2i-1,j} \\ a^{(O)}_{2i-1,j} \\ a^{(I)}_{2i,j} \\ a^{(O)}_{2i,j} \end{bmatrix}$$

Functional Synthesis

H matrix: Scattering matrix relation for a horizontal TBU





$$
\begin{bmatrix} a^{(I)}_{2i,j+1} \\ a^{(I)}_{2i+1,j+1} \\ b^{(I)}_{2i,j} \\ b^{(I)}_{2i+1,j} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} b^{(O)}_{2i,j} \\ b^{(O)}_{2i+1,j} \\ a^{(O)}_{2i,j+1} \\ a^{(O)}_{2i+1,j+1} \end{bmatrix} \longrightarrow \begin{bmatrix} a^{(I)}_{2i,j+1} \\ a^{(O)}_{2i,j+1} \\ a^{(I)}_{2i+1,j+1} \\ a^{(O)}_{2i+1,j+1} \end{bmatrix} = \mathbf{H} \begin{bmatrix} b^{(I)}_{2i,j} \\ b^{(O)}_{2i,j} \\ b^{(I)}_{2i+1,j} \\ b^{(O)}_{2i+1,j} \end{bmatrix}
$$

Functional Synthesis

Build the overall scattering matrix iteratively (the j-th to the j+1-th column)

$$\begin{bmatrix} a^{(I)}_{2i,j+1} \\ a^{(O)}_{2i,j+1} \\ a^{(I)}_{2i+1,j+1} \\ a^{(O)}_{2i+1,j+1} \end{bmatrix} = \mathbf{H} \begin{bmatrix} b^{(I)}_{2i,j} \\ b^{(O)}_{2i,j} \\ b^{(I)}_{2i+1,j} \\ b^{(O)}_{2i+1,j} \end{bmatrix} \qquad \begin{bmatrix} b^{(I)}_{2i-1,j} \\ b^{(O)}_{2i-1,j} \\ b^{(I)}_{2i,j} \\ b^{(O)}_{2i,j} \end{bmatrix} = \mathbf{V} \begin{bmatrix} a^{(I)}_{2i-1,j} \\ a^{(O)}_{2i-1,j} \\ a^{(I)}_{2i,j} \\ a^{(O)}_{2i,j} \end{bmatrix}$$

=> $$\begin{bmatrix} a^{(I)}_{0,j+1} \\ a^{(O)}_{0,j+1} \\ \vdots \\ a^{(I)}_{2N+1,j+1} \\ a^{(O)}_{2N+1,j+1} \end{bmatrix} = \mathbf{T}^j \begin{bmatrix} a^{(I)}_{0,j} \\ a^{(O)}_{0,j} \\ \vdots \\ a^{(I)}_{2N+1,j} \\ a^{(O)}_{2N+1,j} \end{bmatrix}$$

where

$$\mathbf{T}^j = \mathrm{Diag}(\overbrace{\mathbf{H}, \cdots, \mathbf{H}}^{(N+1)}) \times \mathrm{Diag}\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \underbrace{\mathbf{V}, \cdots, \mathbf{V}}_{N}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right)$$

Functional Synthesis

Build the overall scattering matrix iteratively (the 0-th to the M-th column)

$$\begin{bmatrix} a^{(I)}_{2i,j+1} \\ a^{(O)}_{2i,j+1} \\ a^{(I)}_{2i+1,j+1} \\ a^{(O)}_{2i+1,j+1} \end{bmatrix} = \mathbf{H} \begin{bmatrix} b^{(I)}_{2i,j} \\ b^{(O)}_{2i,j} \\ b^{(I)}_{2i+1,j} \\ b^{(O)}_{2i+1,j} \end{bmatrix} \qquad \begin{bmatrix} b^{(I)}_{2i-1,j} \\ b^{(O)}_{2i-1,j} \\ b^{(I)}_{2i,j} \\ b^{(O)}_{2i,j} \end{bmatrix} = \mathbf{V} \begin{bmatrix} a^{(I)}_{2i-1,j} \\ a^{(O)}_{2i-1,j} \\ a^{(I)}_{2i,j} \\ a^{(O)}_{2i,j} \end{bmatrix}$$

=> $$\begin{bmatrix} a^{(I)}_{0,j+1} \\ a^{(O)}_{0,j+1} \\ \vdots \\ a^{(I)}_{2N+1,j+1} \\ a^{(O)}_{2N+1,j+1} \end{bmatrix} = \mathbf{T}^{j} \begin{bmatrix} a^{(I)}_{0,j} \\ a^{(O)}_{0,j} \\ \vdots \\ a^{(I)}_{2N+1,j} \\ a^{(O)}_{2N+1,j} \end{bmatrix}$$

=> $$\begin{bmatrix} a^{(I)}_{0,M} \\ a^{(O)}_{0,M} \\ \vdots \\ a^{(I)}_{2N+1,M} \\ a^{(O)}_{2N+1,M} \end{bmatrix} = \mathbf{T} \begin{bmatrix} a^{(I)}_{0,0} \\ a^{(O)}_{0,0} \\ \vdots \\ a^{(I)}_{2N+1,0} \\ a^{(O)}_{2N+1,0} \end{bmatrix}$$

where

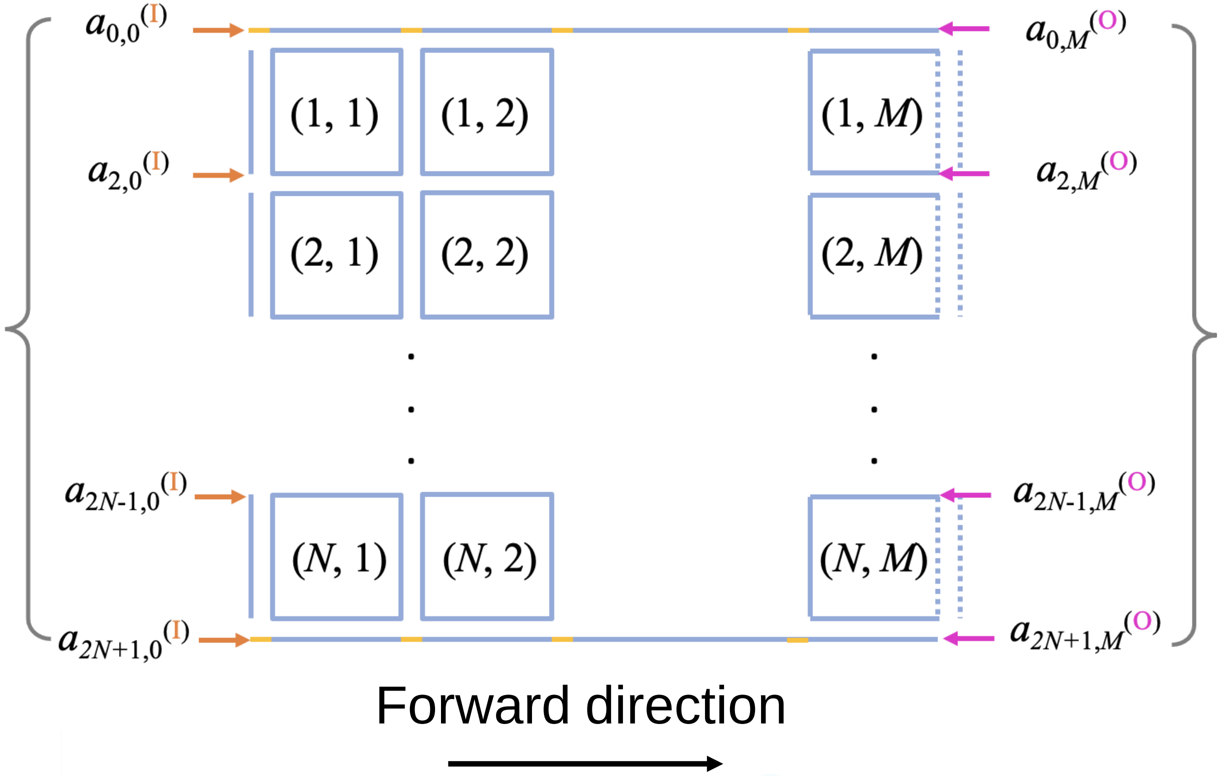
$$\mathbf{T} = \mathbf{T}^{M-1} \cdots \mathbf{T}^{1}\mathbf{T}^{0}$$

Functional Synthesis

We know how to build matrix $\mathbf{T}$, and all operations invovled are differentiable

Define input and output and use a cost function: $$Cost_{LogMag} = \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^{N} r_k \left| \ln |a_{2n,M}^{(I)}(\omega_k)| - \ln U_n(\omega_k) \right|^2$$

Forward input $\mathbf{a}_0^{(I)}$

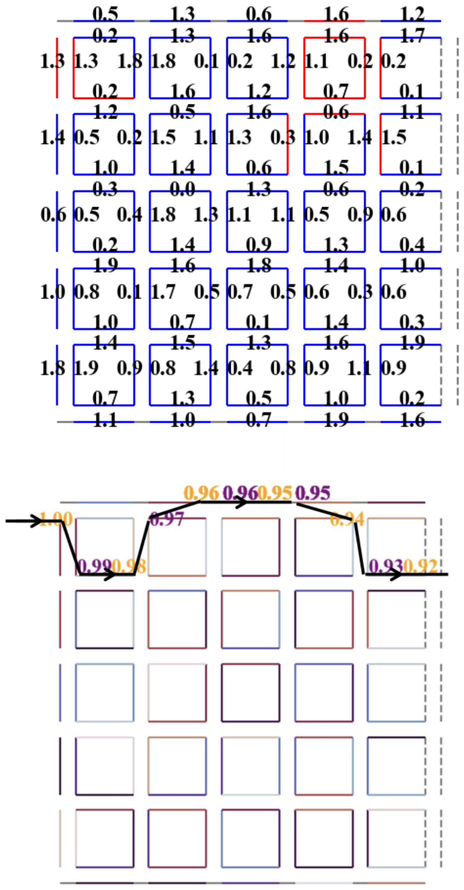


Backward input $\mathbf{a}_M^{(O)}$

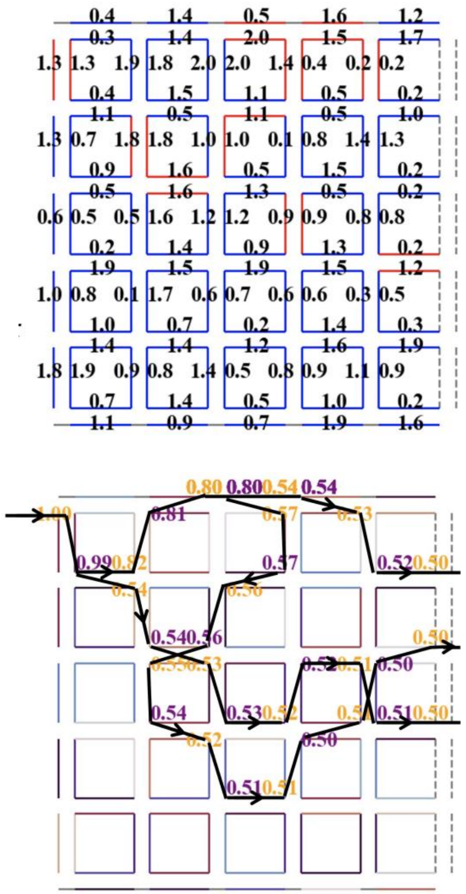
$$\begin{bmatrix} a_{0,M}^{(I)} \\ a_{0,M}^{(O)} \\ \vdots \\ a_{2N+1,M}^{(I)} \\ a_{2N+1,M}^{(O)} \end{bmatrix} = \mathbf{T} \begin{bmatrix} a_{0,0}^{(I)} \\ a_{0,0}^{(O)} \\ \vdots \\ a_{2N+1,0}^{(I)} \\ a_{2N+1,0}^{(O)} \end{bmatrix}$$

Functional Synthesis

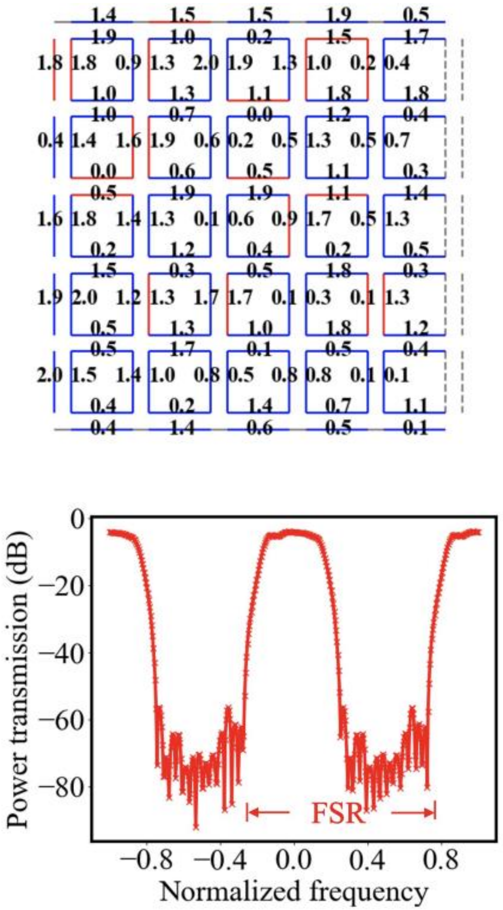
Routing



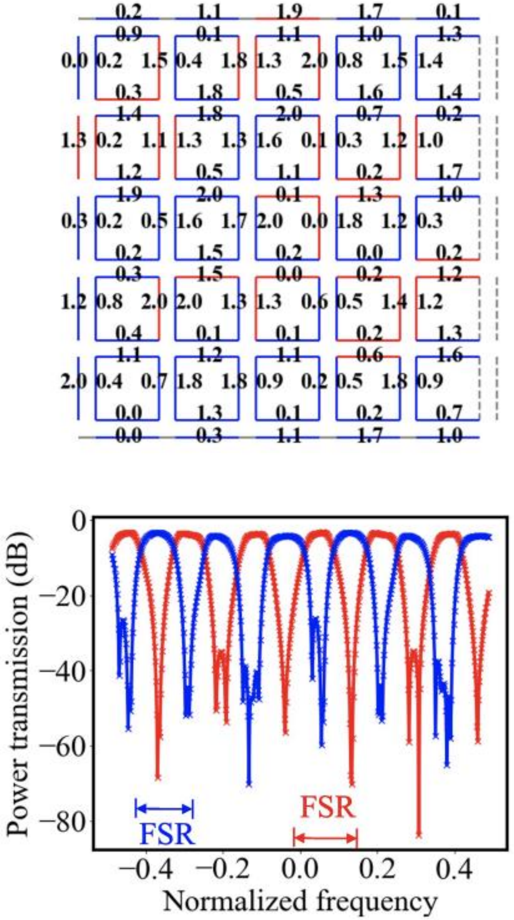
Splitting



Filtering

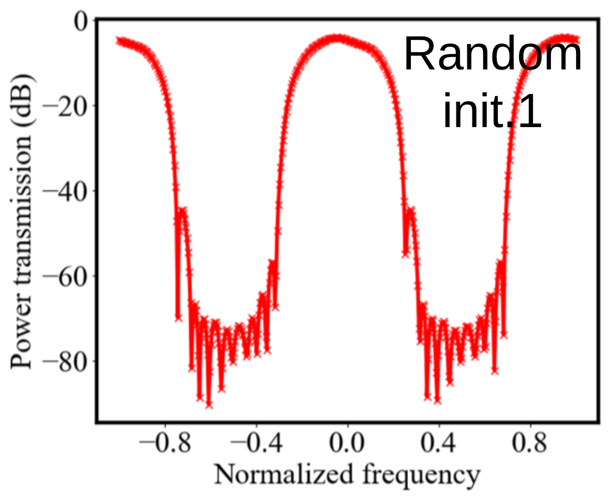


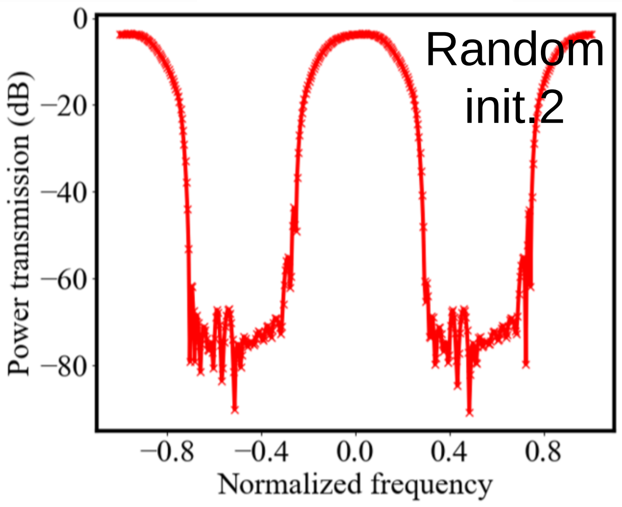
WDM

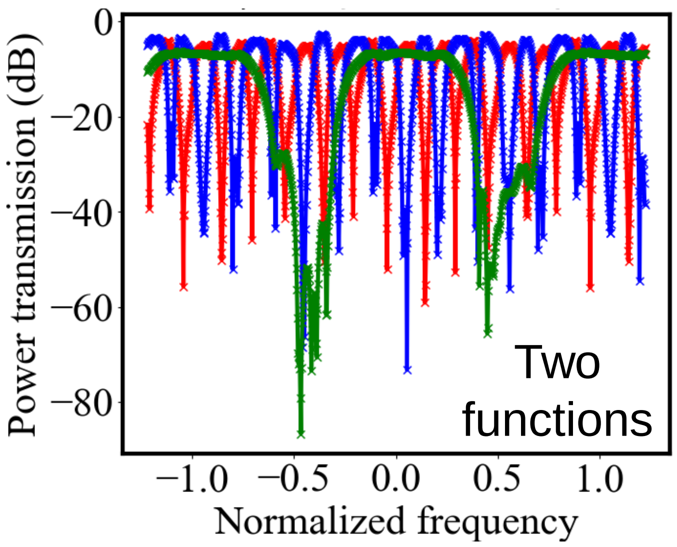


Functional Synthesis

- Local minimum is accpetable
- Random initialization doesn’t impact the synthesized results much
- Even could realize two functions at the same time







Ref: Zhengqi Gao et al., *Photonics Res.* 2023. (**highlighted as Editor’s pick**)

Online Demo

In previous page, we show how to derive gradients analyticall in a simplified square mesh

What about gradient calculation in any topology (hexagonal, triangular, even mix)?

=> A light-weight Python package, Spode, specialized for programmable photonic circuit

SPODE

A simulator with programmable photonics and differentiability emphasis

https://colab.research.google.com/drive/1lLw5831I-cmhHSIQOWGuc7vPmQEsNKoq?usp=sharing

Remark: The package is for ease of research; integrate simulation, visualization, circuit generator.

Discussions and Future Directions

- Impact of the photodetector
- Error cascading (See Saumil's Optica 2021 paper)
- Provable routing algorithm
- Hardware demonstration

Further Reading

1. Wim Bogaerts, 'Tutorial: Programmable Phototonics,' *OFC*, 2021.
2. Wim Bogaerts et al., 'Programmable Phototonic Circuits,' *Nature*, 2020.
3. Zhengqi Gao et al., 'Automatic synthesis of light-processing functions for programmable photonics: theory and realization,' *Photonics Res.*, 2020.
4. Saumil Bandyopadhyay et al., 'Hardware error correction for programmable photonics,' Optica, 2021.
5. Daniel Perez-Lopez et al., 'Multipurpose self-configuration of programmable photonic circuits', Nat. Comm., 2020.
6. William R. Clements et al., '"Optimal design for universal multiport interferometers,' Optica, 2016.
7. Aitor Lopez, el al., 'Auto-routing algorithm for field-programmable photonic gate arrays,' Optica Express, 2020.
8. Michael Reck et al., 'Experimental realization of any discrete unitary operator,' Phys. Rev. Lett., 1994,