NeurOLight: A Physics-Agnostic Neural Operator Enabling Parametric Photonic Device Simulation

Jiaqi Gu[1], Zhengqi Gao[2], Chenghao Feng[1], Hanqing Zhu[1], Ray T. Chen[1], Duane S. Boning[2], David Z. Pan[1]

[1]Department of ECE, UT Austin, [2]EECS MIT

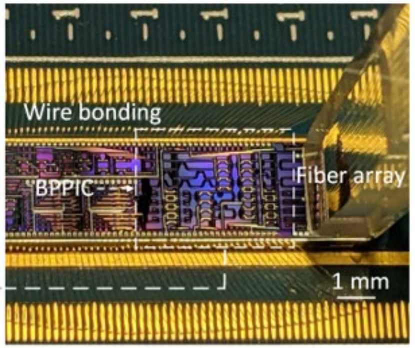
December 28, 2024

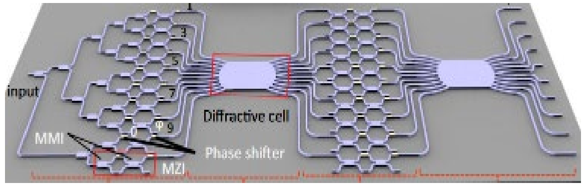
jqgu@utexas.edu

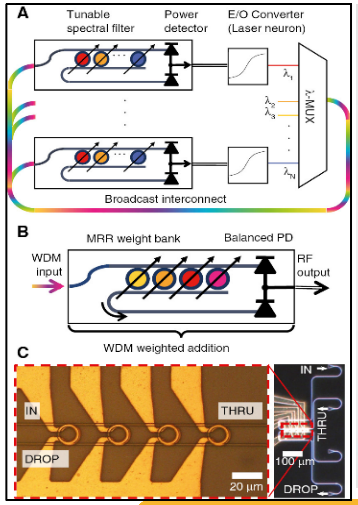
This work was supported in part by AFOSR MURI

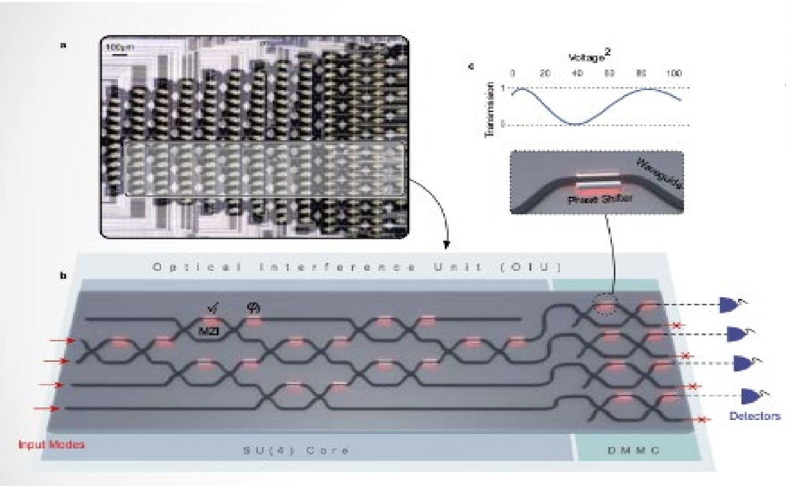
Light-AI Interaction: Photonic AI & AI for Optics

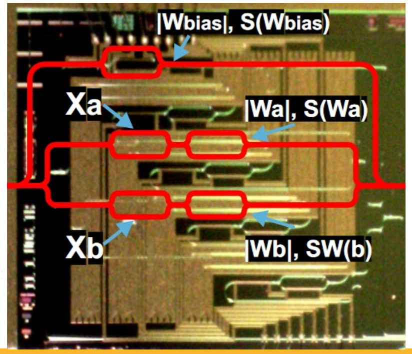
- Photonic IC for AI computing
- AI for photonic IC design

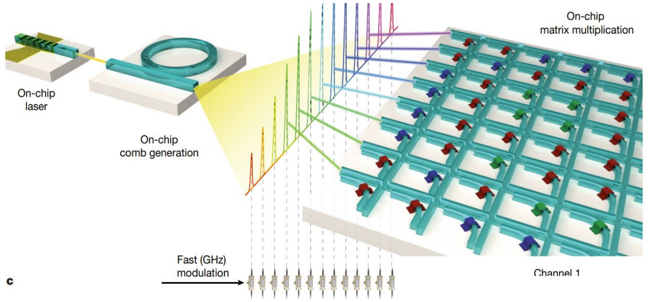












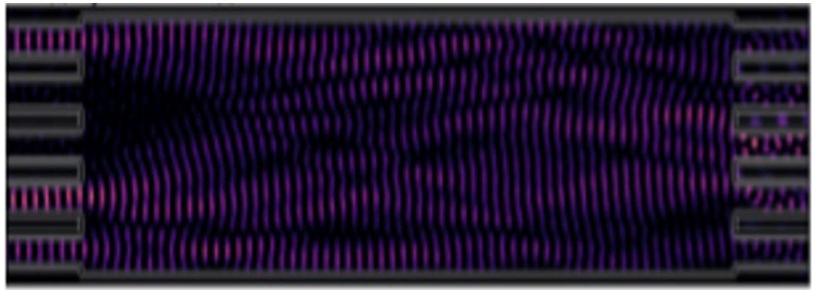
Manual Design → Automated ONN Design

Standard Devices → Customized Photonic Structure

Key step: **AI-assisted Simulation**

Motivation for AI for Optical Simulation

- Basic devices are not enough…?
- Leverage *physics* of light propagation in *customized photonic structures*
  - Customized structure can be more compact and efficient
  - Hard to get compact analytical model
  - Slow simulation and optimization loop

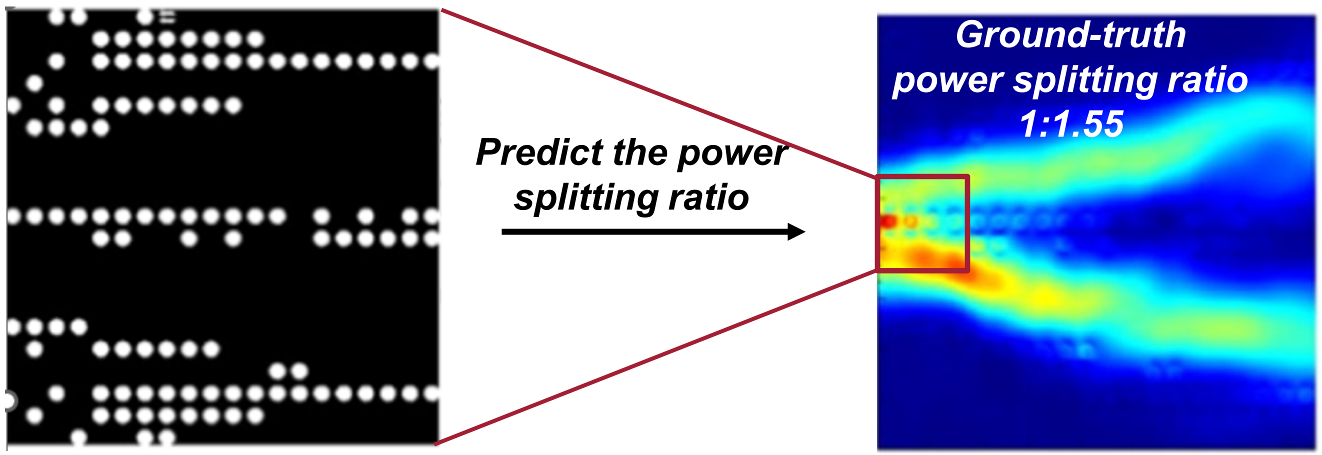


- AI accelerated optical simulation as an entry point
  - Can ML models learn the light propagation principles?

- Target
  - Query light fields of photonic structures with certain inputs and design variables
  - Ultra-fast & Parallel & Differentiable
  - **Early exploration only, do not replace commercial simulators in final validation stage**

Related Work

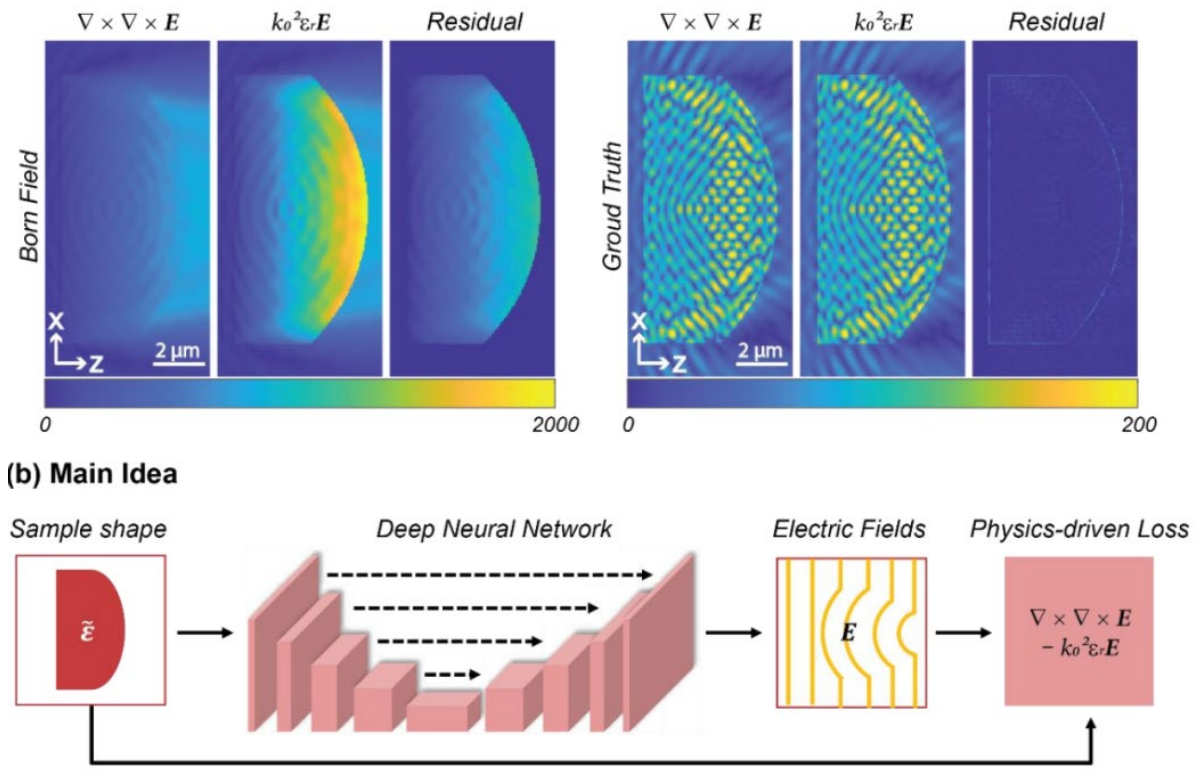
- Ad-hoc MLP model $f: \mathbb{R}^{N\times N} \rightarrow \mathbb{R}$ to fit a certain FoM
  - Boolean cavity map → power spitting ratio
- Does not learn any underlining physics principle in the device
- ***No generalization to other tasks → the fitted curve has no other usage***



SciRep'19 Mitsubishi Electric Research Laboratories
Predict power splitter transmission and inverse design
Direct prediction without physics principle

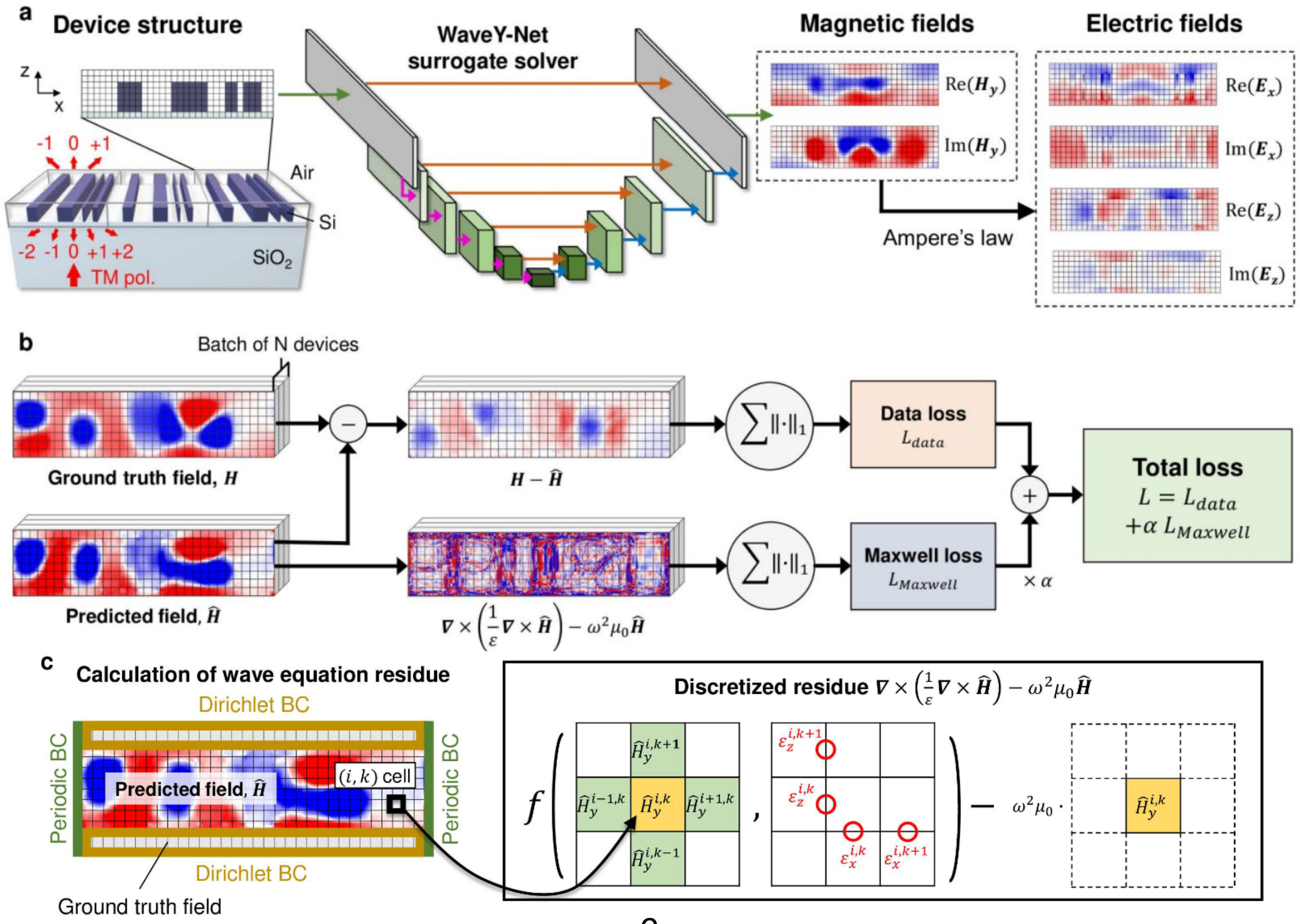
Related Work

- MaxwellNet: Physics-informed NN for Free-Space Lens [APL'22]
  - Based on maxwell equation: ***complicated***
  - Solve specific instances (fixed wavelength, fixed domain): ***limited generalization***



Related Work

- Physics-augmented NN on meta-lens [Nature'21]
  - Still need maxwell equation. Limited to specific meta-lens instance



Need Maxwell equations?

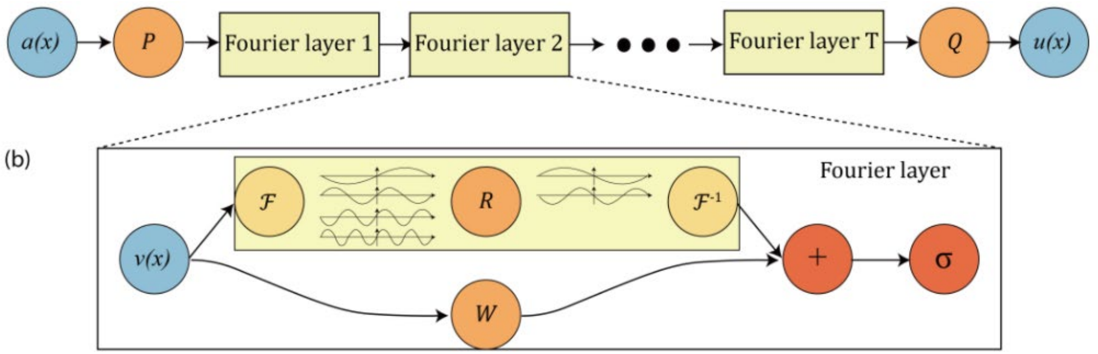
Save the efforts on PDE implementation

Only fit to specific instance?

Generalize to a family of instances

Maxwell is PDE → NN can learn to solve PDEs

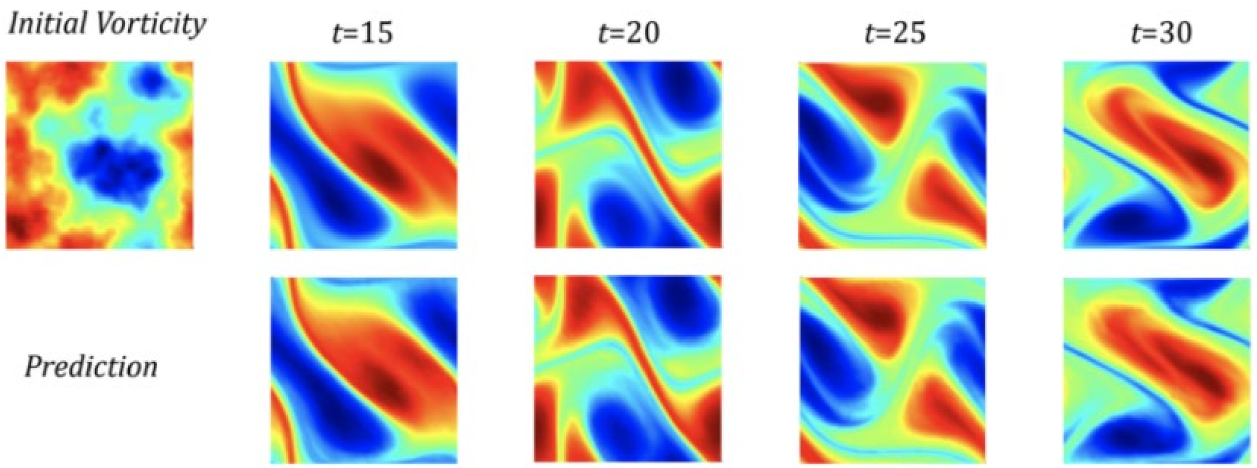
- ***Neural operator*** learns *a family of* **parametric** PDEs in a *data-driven* way
  - Not just fitting one PDE instance. Not physics-informed
- Fourier neural operator [ICLR'21]
  - › Model PDE as a series of *kernel integral*
    - » $a \to v_0 \to v_1 \to \cdots \to v_k \to u$
  - › Weather forecast, flow prediction, …



$$(\mathcal{K}v_k)(\boldsymbol{r}) = \int_\Omega \kappa(\boldsymbol{r}_1, \boldsymbol{r}_2) v_k(\boldsymbol{r}_2) \mathrm{d}v_k(\boldsymbol{r}_2), \forall \boldsymbol{r}_1 \in \Omega \quad \kappa(\boldsymbol{r}_1, \boldsymbol{r}_2) = \kappa(\boldsymbol{r}_1 - \boldsymbol{r}_2)$$

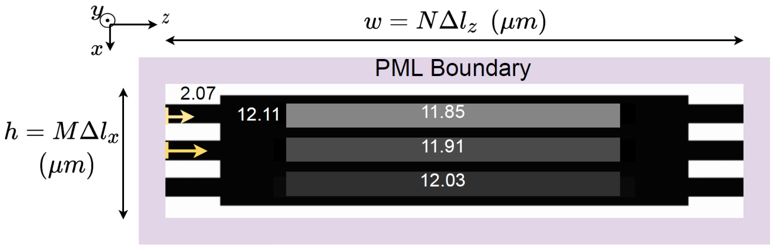
$$(\mathcal{K}v_k)(\boldsymbol{r}) = \mathcal{F}^{-1}\big(\mathcal{F}(\kappa) \cdot \mathcal{F}(v_k)\big)(\boldsymbol{r})$$

- Advantages of FNO
  - Minimum physics knowledge needed
  - Invariance to discretization
  - Fast inference
  - Good generalization



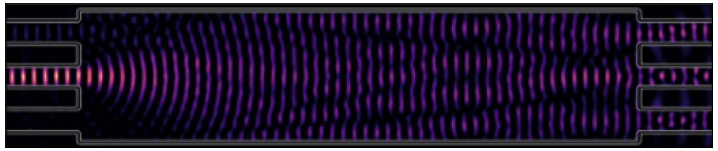
Our Proposed Idea: NeurOLight

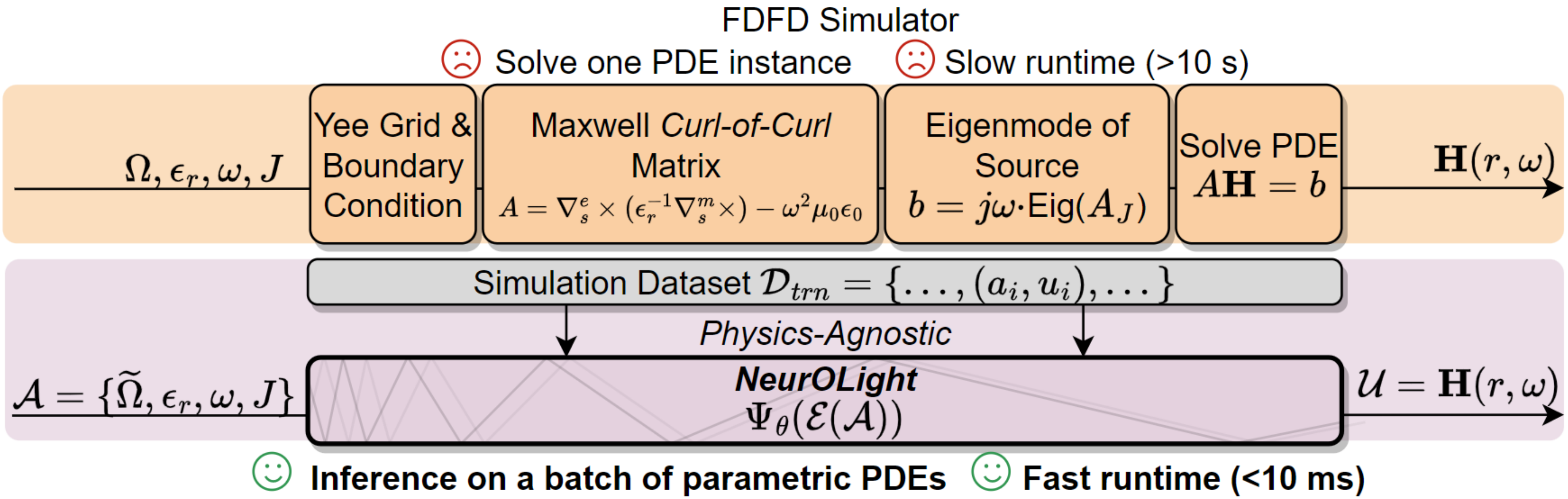
- *Physics-agnostic* simulation for photonics devices
- Slow 2-D FDFD simulation (×) → Ultra-fast surrogate NN model (√)



$$\left((\mu_0^{-1}\nabla \times \nabla\times) - \omega^2\epsilon_0\boldsymbol{\epsilon}_r(\boldsymbol{r})\right)\mathbf{E}(\boldsymbol{r}) = j\omega\mathbf{J}_e(\boldsymbol{r})$$

$$\left(\nabla \times (\boldsymbol{\epsilon}_r^{-1}(\boldsymbol{r})\nabla\times) - \omega^2\mu_0\epsilon_0\right)\mathbf{H}(\boldsymbol{r}) = j\omega\mathbf{J}_m(\boldsymbol{r})$$

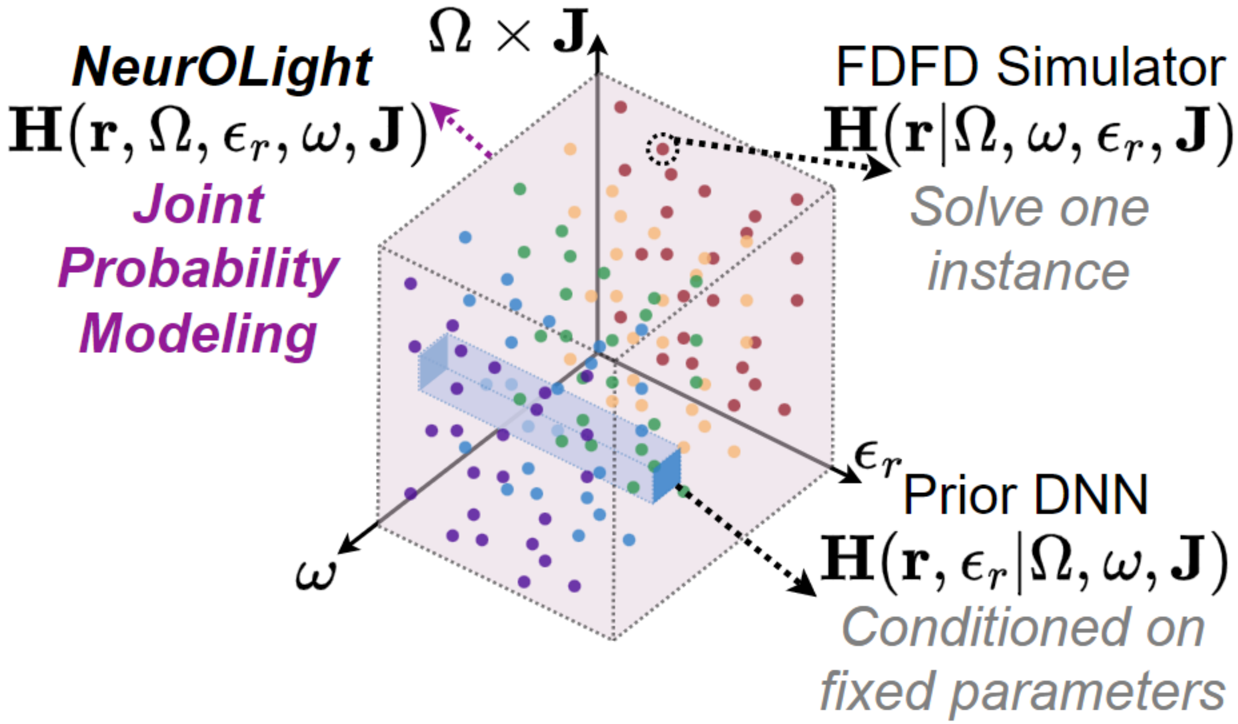




Proposed NeurOLight Framework

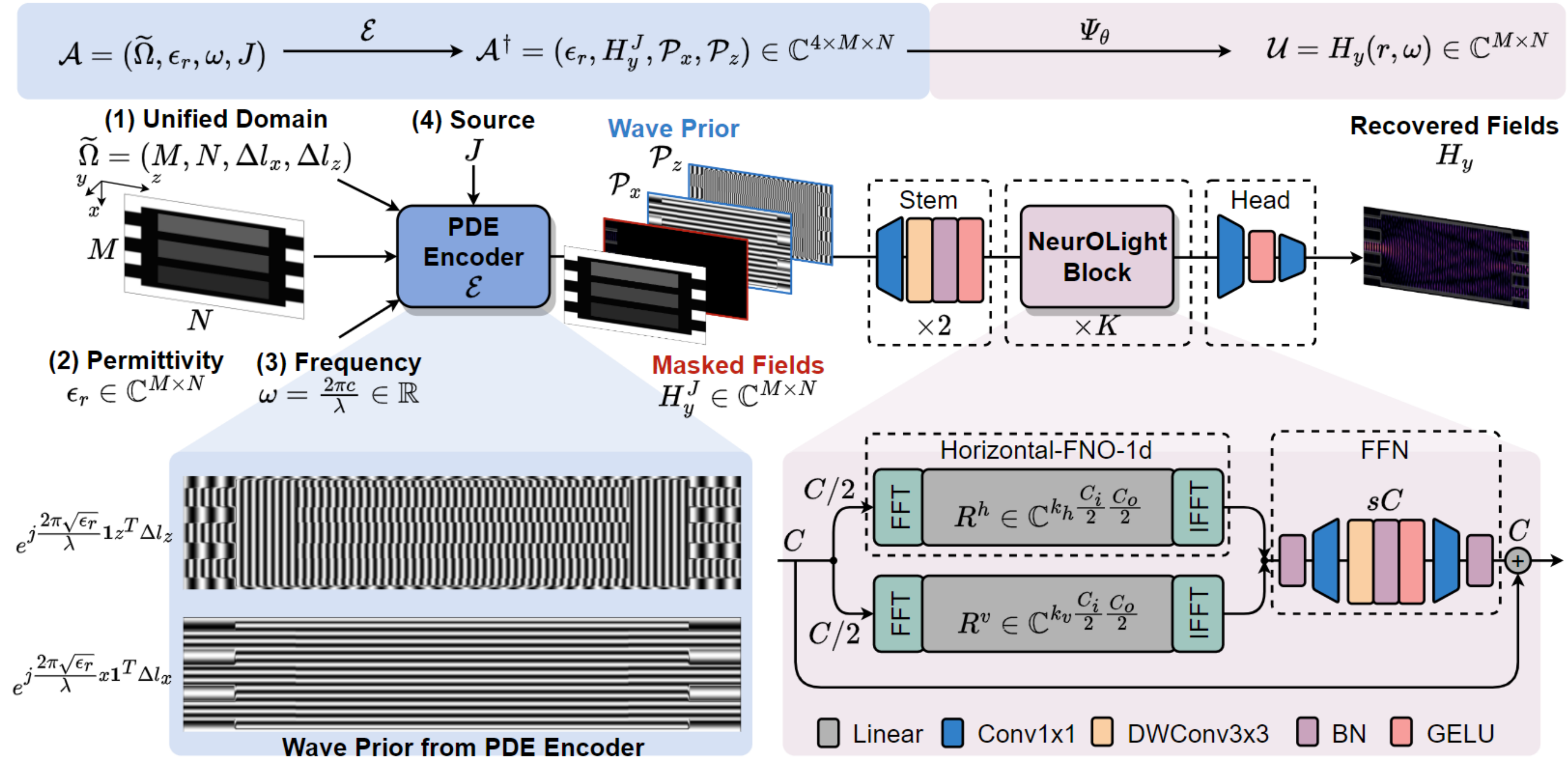
Our Proposed Idea: NeurOLight

- ***Physics-agnostic*** simulation for photonics devices
- Slow 2-D FDFD simulation (×) → Ultra-fast surrogate NN model (√)
- Limited PDE space (×) → Model the joint distribution over PDE parameters (√)



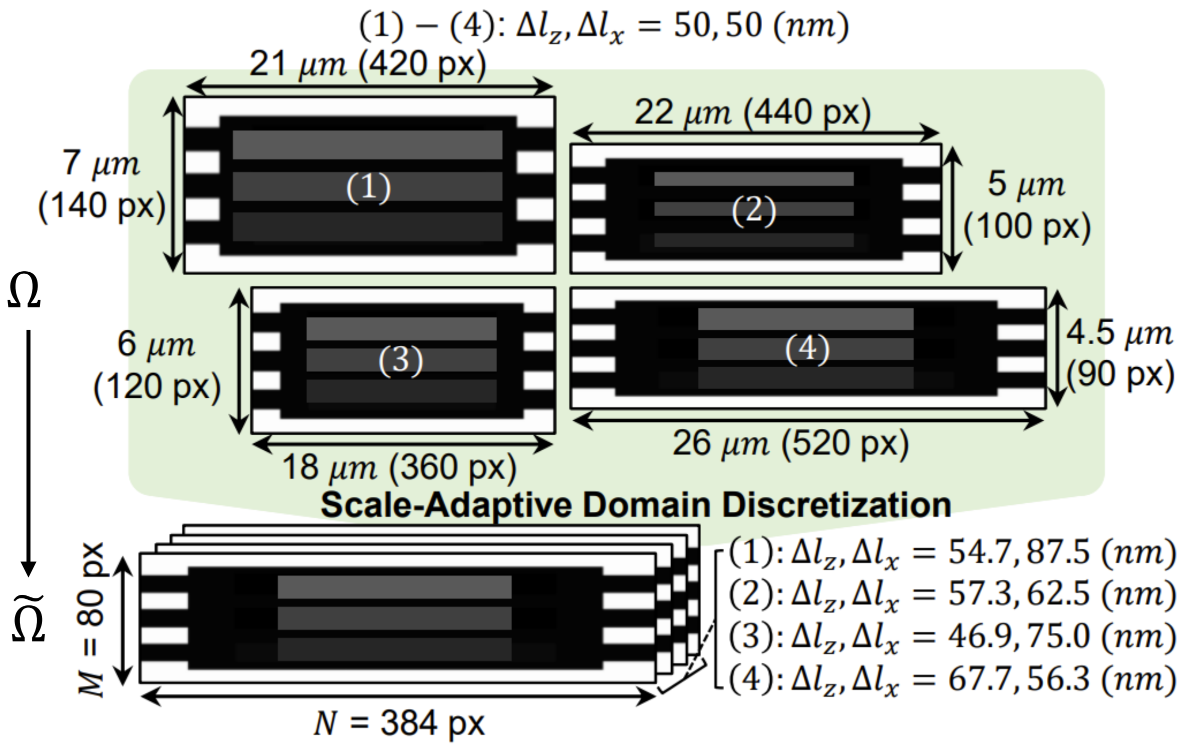
Our Proposed NeurOLight

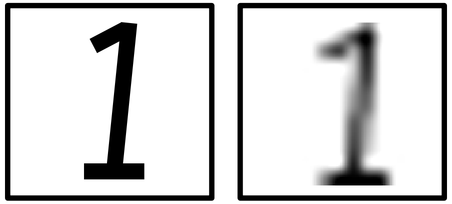
- ***Physics-agnostic*** neural operator $\theta^* = \min_{\theta} \mathbb{E}_{a\sim\mathcal{A}}\left[\mathcal{L}\left(\Psi_\theta(\mathcal{E}(a)), \Psi^*(a)\right)\right]$
- Learn the mapping: PDE observations $\mathcal{A}$ → PDE solutions $\mathcal{U}$



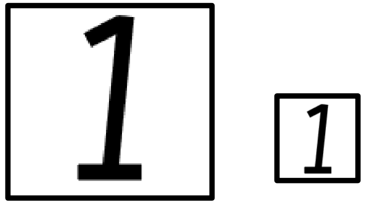
How To Generalize To Different Scales of Domains

- Scale-Adaptive Domain Discretization
  - Unified domain $\Omega \rightarrow \widetilde{\Omega}$
  - Resize different device sizes to the same image sizes
  - Mesh grid resolution $(\Delta l_x, \Delta l_z)$ is elegant representation
- Can support ***batched*** training & inference
- Do not need domain-specific retraining





Fix domain
Different resolutions



Different domain scales

How To Represent PDE Parameters

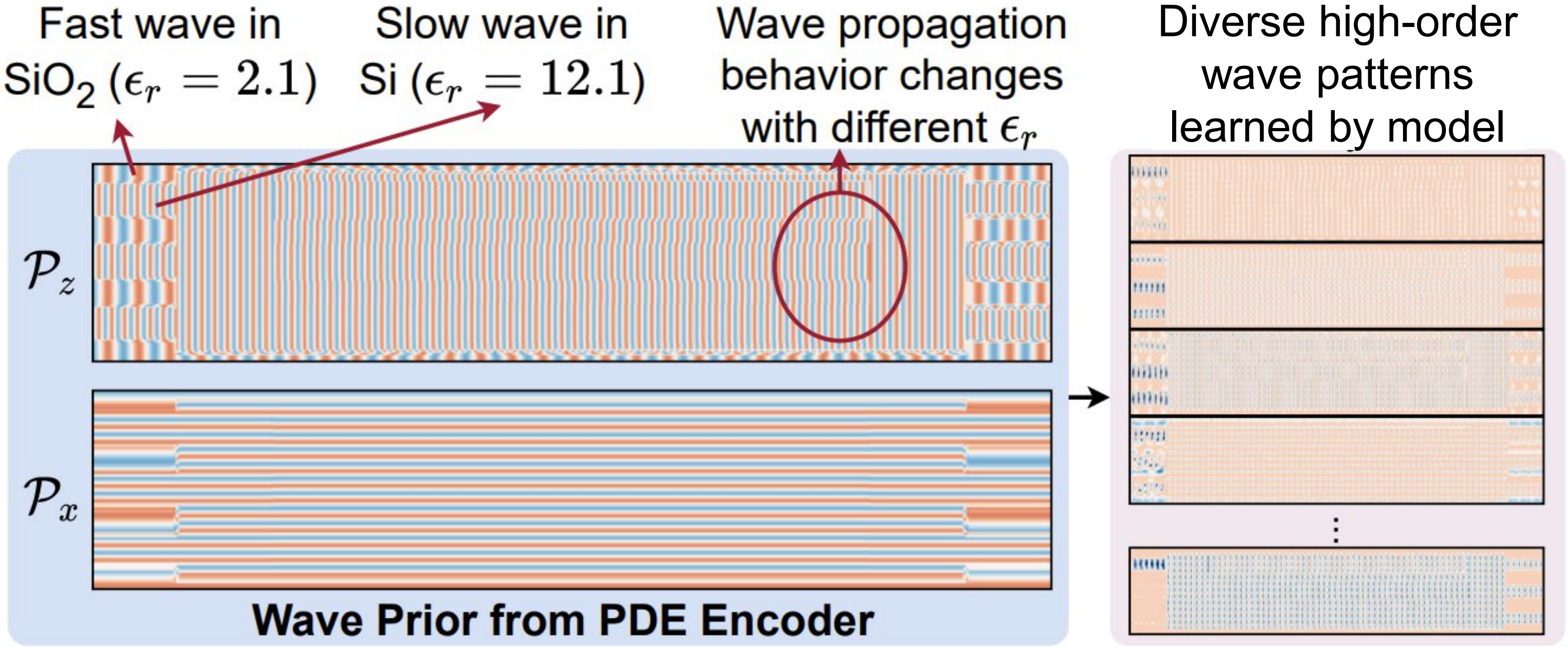
- Joint/Unified PDE Representation
  - $\widetilde{\Omega}$: tuple, ϵ_r: matrix, ω: scalar, J: vectors
  - (tuple, matrix, scalar, vectors,…) → Tensor

$$\mathcal{A} = (\widetilde{\Omega}, \epsilon_r, \omega, J) \xrightarrow{\mathcal{E}} \mathcal{A}^\dagger = (\epsilon_r, H_y^J, \mathcal{P}_x, \mathcal{P}_z) \in \mathbb{C}^{4\times M\times N}$$

- ***Wave prior*** to encode $(\Omega, \omega, \epsilon_r)$
  - Rich patterns for NN to extract features & Implicit physics prior encoding

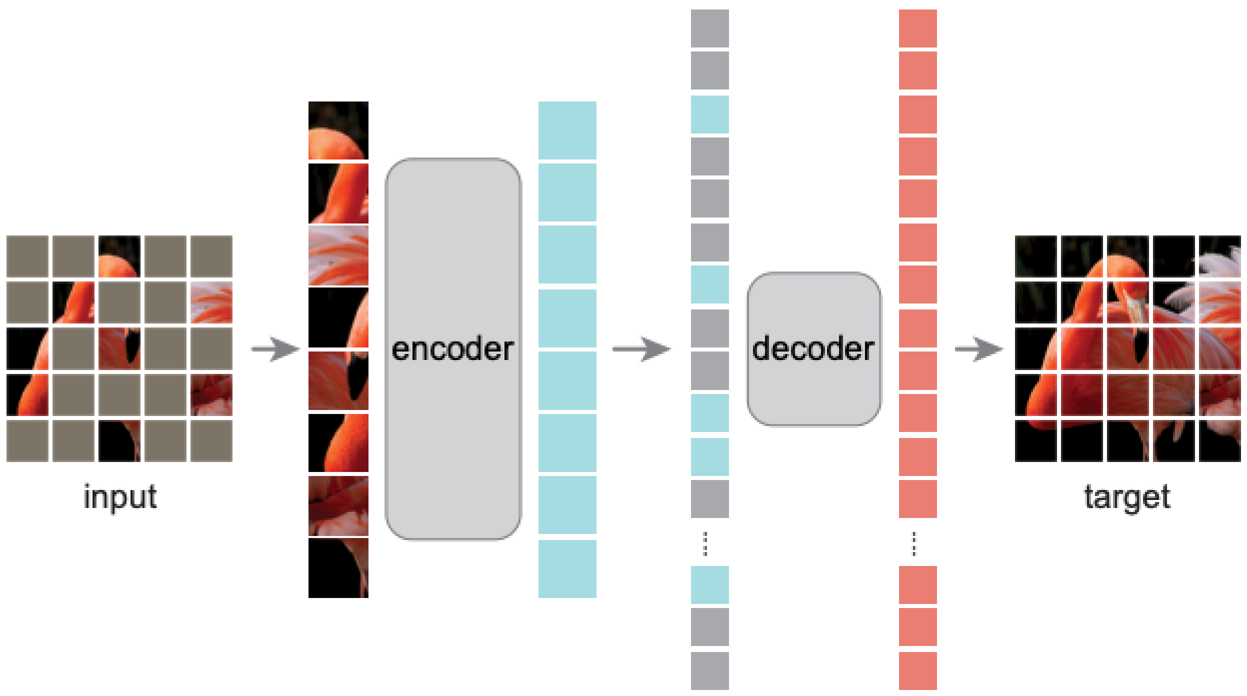
$$\mathcal{P}_z = e^{j\frac{2\pi\sqrt{\epsilon_r}}{\lambda}\mathbf{1}z^T\Delta l_z}$$

$$\mathcal{P}_x = e^{j\frac{2\pi\sqrt{\epsilon_r}}{\lambda}x\mathbf{1}^T\Delta l_x}$$

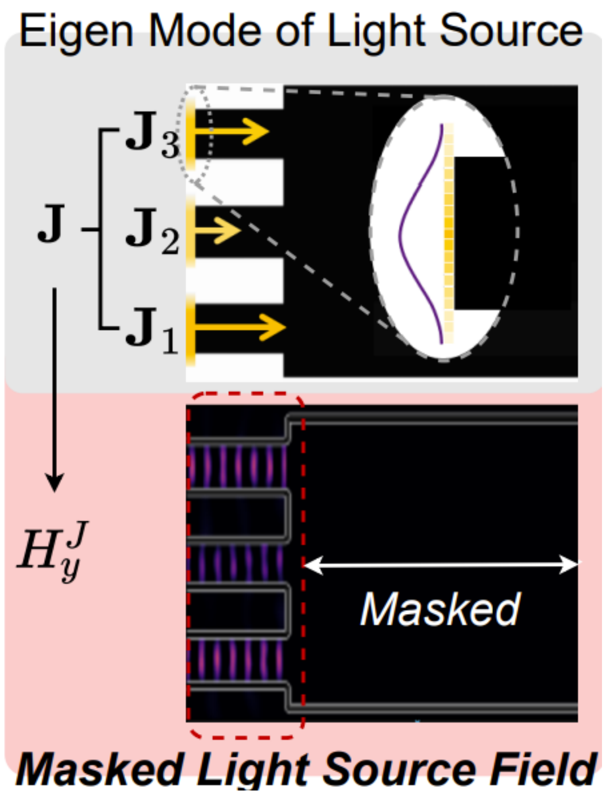


How To Represent PDE Parameters

- Light source encoding (*J*)
  - Inspired by masked image modeling (MIM) and masked autoencoder (MAE)
  - Fields in the input waveguide as a ***hint***
  - Model needs to ***restore the masked fields***

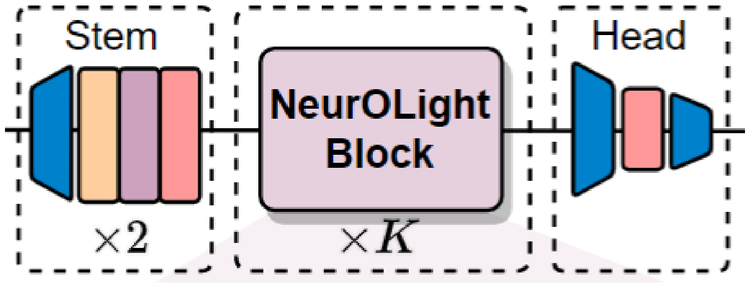


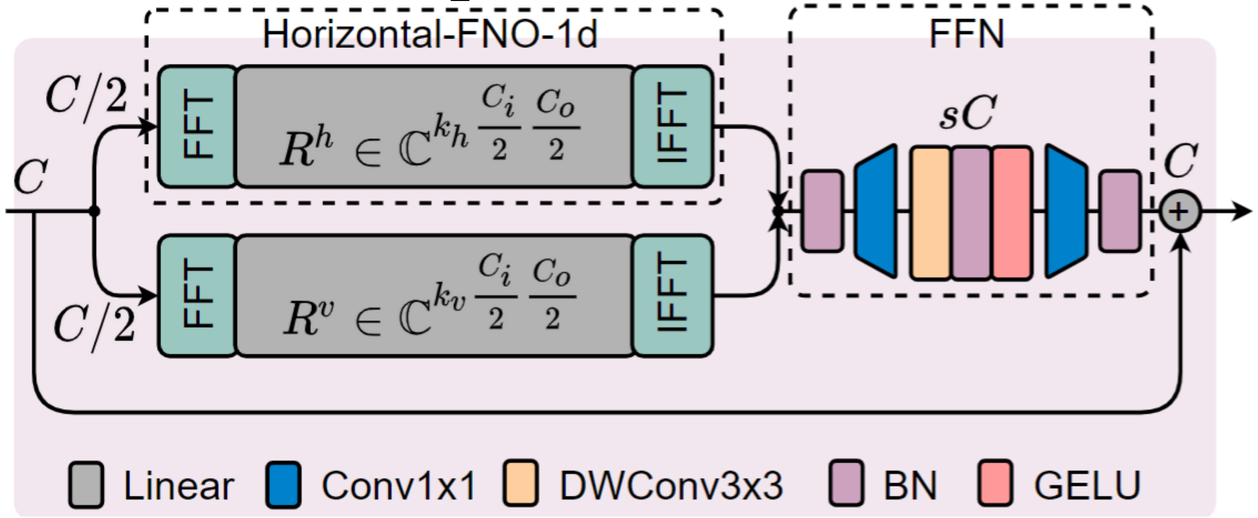
[MAE, Kaiming He+, 2021]



How To Do Efficient Kernel Integral

- Cross-shaped NeurOLight blocks
  - Orthogonal 1-D FNO: vertical / horizontal
    - Reduce #parameters: $k_v k_h C^2 \rightarrow (k_v + k_h + 8s)C^2/4$
  - Nonlinear FFN: more local feature extraction
  - More parameter-efficient and better generalization





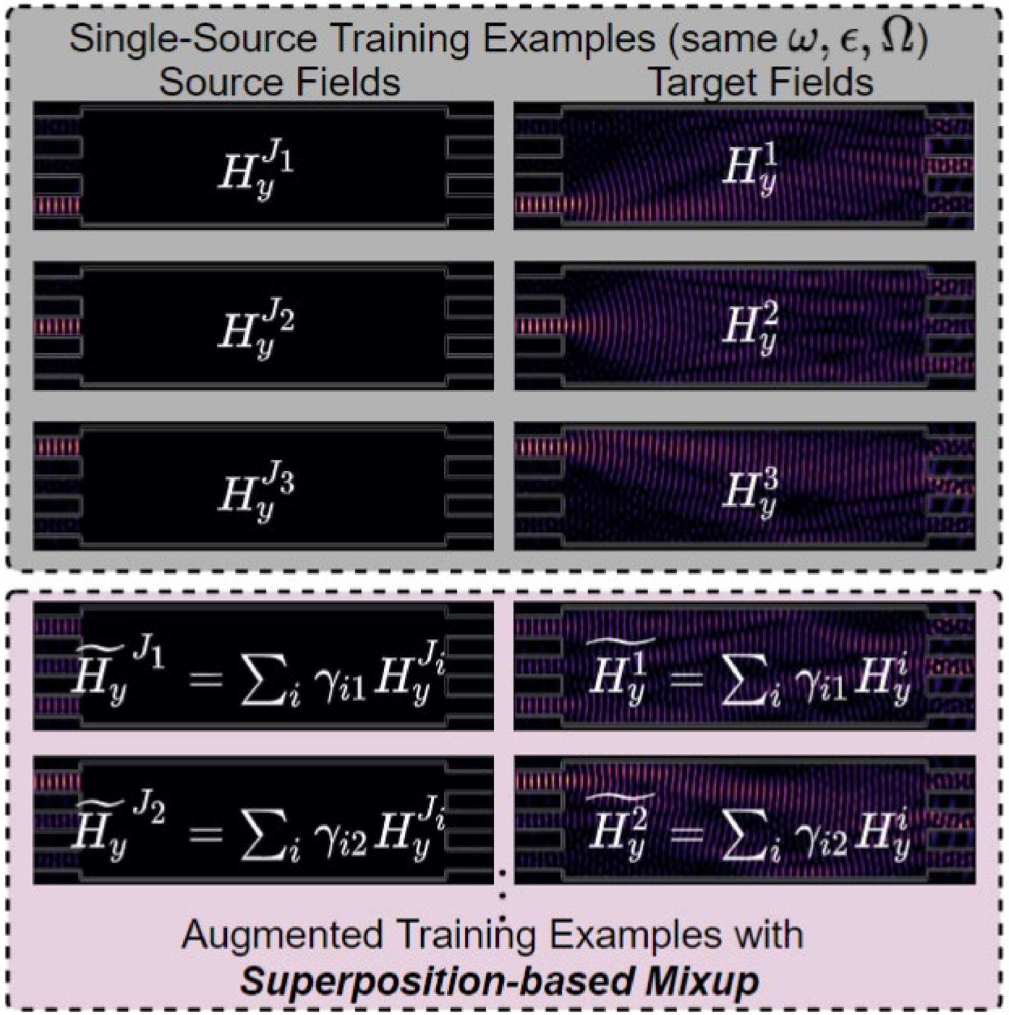
$$(\mathcal{K}^h v_k^h)(\boldsymbol{r}) = \mathcal{F}_z^{-1}\big(\mathcal{F}_z(\kappa^h) \cdot \mathcal{F}_z(v_k^h)\big)(\boldsymbol{r}) = \mathcal{F}_z^{-1}\Big(R^h(z) \cdot \mathcal{F}_z(v_k^h(\boldsymbol{r}))\Big), \forall z \in \Omega_z, \forall r \in \Omega,$$

$$(\mathcal{K}^v v_k^v)(\boldsymbol{r}) = \mathcal{F}_x^{-1}\big(\mathcal{F}_x(\kappa^v) \cdot \mathcal{F}_x(v_k^v)\big)(\boldsymbol{r}) = \mathcal{F}_x^{-1}\Big(R^v(x) \cdot \mathcal{F}_x(v_k^v(\boldsymbol{r}))\Big), \forall x \in \Omega_x, \forall r \in \Omega,$$

$$(\mathcal{K} v_k)(\boldsymbol{r}) = [(\mathcal{K}^h v_k^h)(\boldsymbol{r}); (\mathcal{K}^v v_k^v)(\boldsymbol{r})].$$

How To Improve Data Efficiency & Generalization

- Sweeping different light source combinations is costly
- Train: *single-source* simulation → Test: generalize to *multi-source inference*
- ***Proposed superposition-based mixup***
  - Dynamically superpose input modes during training
  - Multiple single-source predictions (×)
  - One-shot multi-source prediction (√)
  - Force to ***learn critical physics: interference***



Experimental Settings

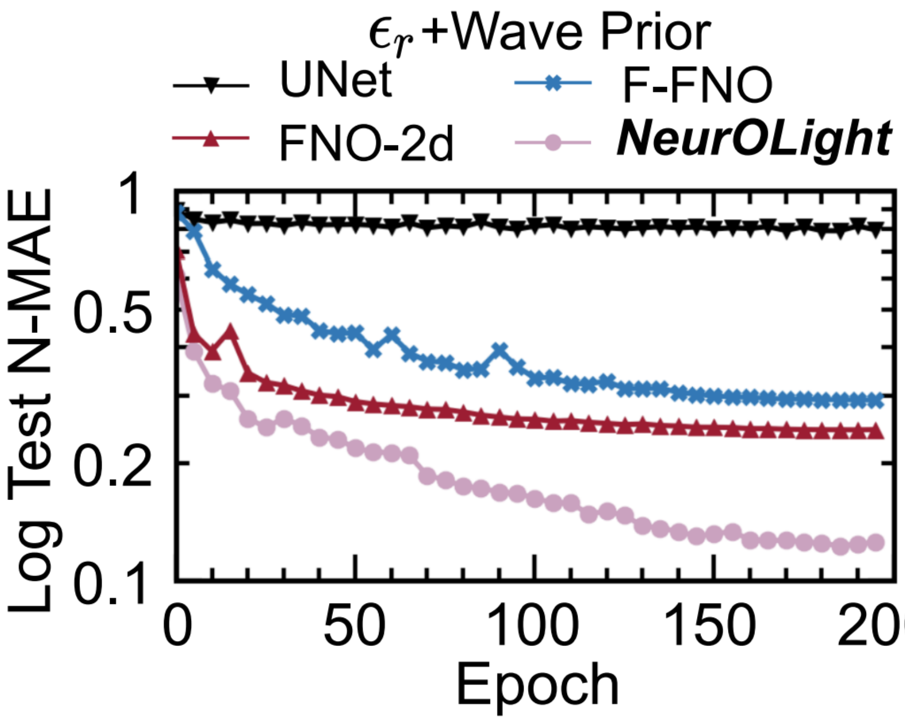
- Device benchmarks
  - Randomly generated (1) Tunable MMIs and (2) Etched MMIs. Simulate with ***angler***

Variables	Value/Distribution		Unit
	$\lvert\mathbf{J}\rvert \times \lvert\mathbf{J}\rvert$ Tunable MMI	$\lvert\mathbf{J}\rvert \times \lvert\mathbf{J}\rvert$ Etched MMI	
Length	$\mathcal{U}(20, 30)$	$\mathcal{U}(20, 30)$	μm
Width	$\mathcal{U}(5.5, 7)$	$\mathcal{U}(5.5, 7)$	μm
Port Length	3	3	μm
Port Width	$\mathcal{U}(0.8, 1.1)$	$\mathcal{U}(0.8, 1.1)$	μm
Border Width	0.25	0.25	μm
PML Width	1.5	1.5	μm
Pad Length	$\mathcal{U}(0.7, 0.9)\times$Length	$\mathcal{U}(0.7, 0.9)\times$Length	μm
Pad Width	$\mathcal{U}(0.4, 0.65)\times$Width/$\lvert\mathbf{J}\rvert$	$\mathcal{U}(0.4, 0.65)\times$Width/$\lvert\mathbf{J}\rvert$	μm
Wavelengths λ	$\mathcal{U}(1.53, 1.565)$	$\mathcal{U}(1.53, 1.565)$	μm
Cavity Ratio	-	$\mathcal{U}(0.05, 0.1)$	-
Cavity Size	-	0.027 Length $\times$ 0.114 Width	μm^2
Relative Permittivity ϵ_r	$\mathcal{U}(11.9, 12.3)$	$\{2.07, 12.11\}$	-

- Comparison models (comparable #Params)
  - UNet-2d [APL'22]
  - 5-layer FNO-2d ($\#Mode_z$=32, $\#Mode_x$=10) [ICLR'21]
  - 12-layer Factorized FNO (F-FNO) [NeurIPS workshop'21]

Main Results

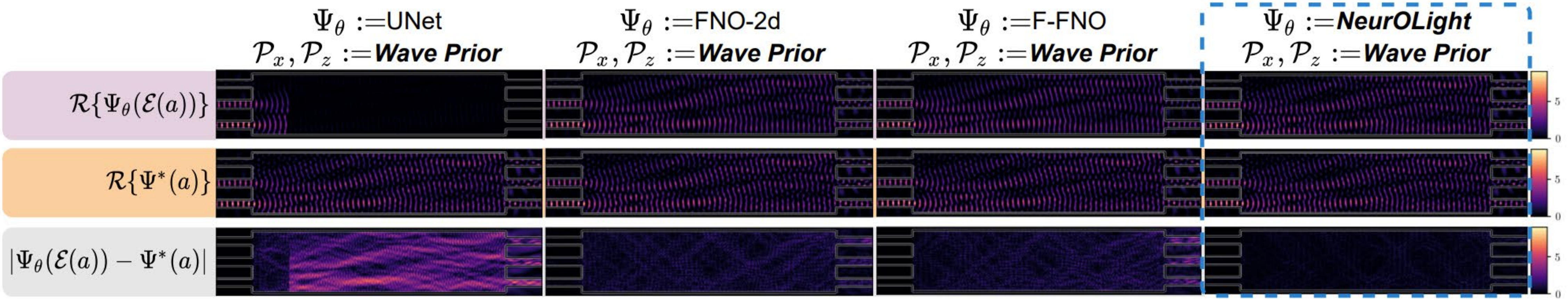
- NeurOLight is more parameter-efficient
- 53.8% lower error and 44.2% fewer parameters



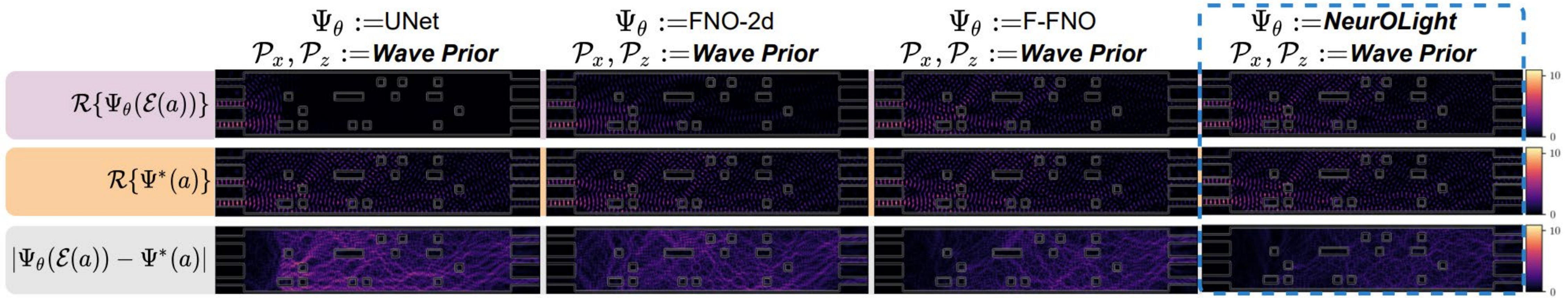
Benchmarks	Model	#Params (M) ↓	Test Err ↓
Tunable MMI	UNet [22, 2]	3.47	0.801
	FNO-2d [21]	3.29	0.244
	F-FNO [34]	3.16	0.292
	NeurOLight	**1.58**	**0.122**
Etched MMI	UNet [22, 2]	3.47	0.792
	FNO-2d [21]	3.29	0.648
	F-FNO [34]	3.16	0.525
	NeurOLight	**2.11**	**0.387**
Average Improvement		**-44.2%**	**-53.8%**

Main Results: Visualization

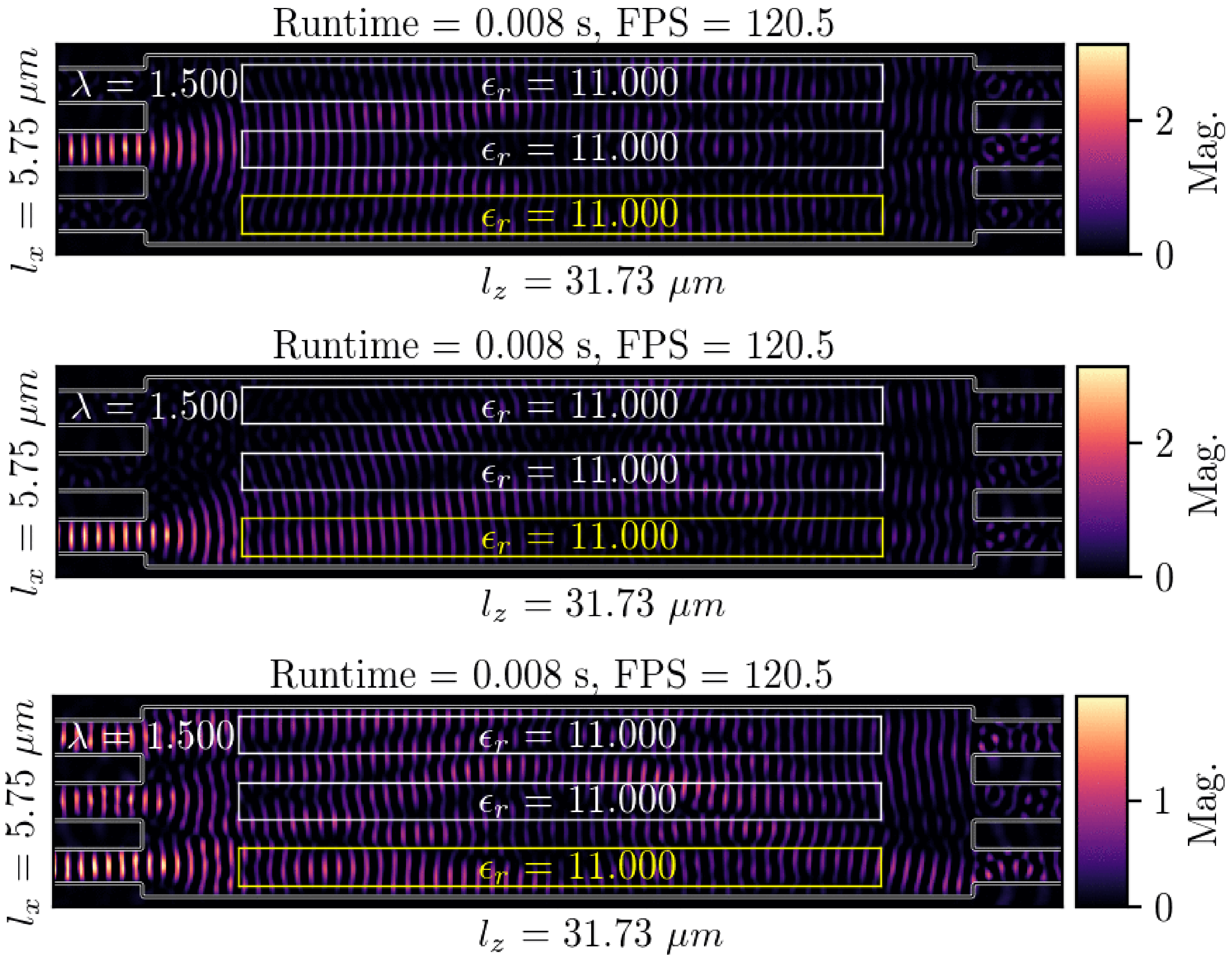
- Tunable MMI



- Etched MMI

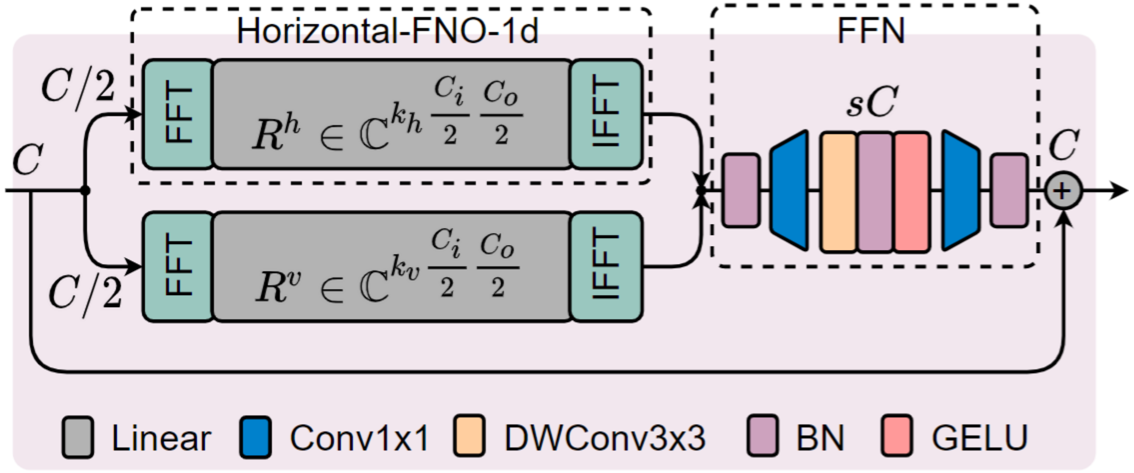


Animation



Runtime = 0.008 s, FPS = 120.5
$\lambda = 1.500$
$\epsilon_r = 11.000$
$\epsilon_r = 11.000$
$\epsilon_r = 11.000$
$l_x = 5.75\ \mu m$
$l_z = 31.73\ \mu m$
Mag.
2
0
Runtime = 0.008 s, FPS = 120.5
$\lambda = 1.500$
$\epsilon_r = 11.000$
$\epsilon_r = 11.000$
$\epsilon_r = 11.000$
$l_x = 5.75\ \mu m$
$l_z = 31.73\ \mu m$
Mag.
2
0
Runtime = 0.008 s, FPS = 120.5
$\lambda = 1.500$
$\epsilon_r = 11.000$
$\epsilon_r = 11.000$
$\epsilon_r = 11.000$
$l_x = 5.75\ \mu m$
$l_z = 31.73\ \mu m$
Mag.
1
0

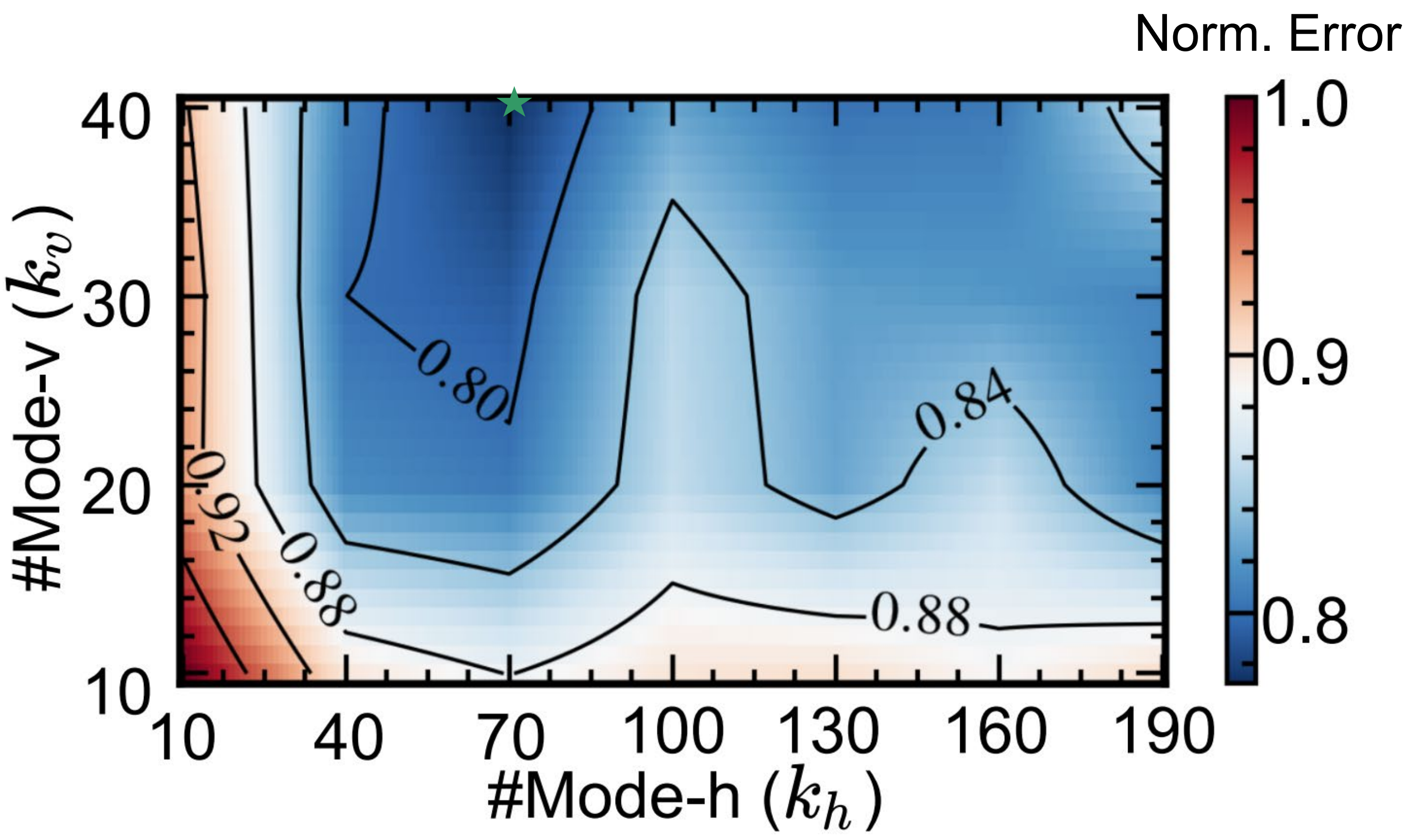
Ablation Studies on Model Designs



Variants	#Params (M)↓	Test Err ↓	Runtime (ms) ↓
NeurOLight	**1.58**	**0.122**	**12.1**
ConvStem → Lifting	1.58	0.134	11.9
Extra Parallel Conv Path	1.64	0.129	14.5
FFN → BN-GELU	1.37	0.446	6.3
Remove DWConv in FFN	1.57	0.144	10.6
Extra GELU After FNO	1.58	0.148	12.4
Remove DropPath	1.58	0.136	12.1

Ablation Studies on Frequency Components (#Mode)

- Full mode is overparametrized
- (40,70) is the best setting



Ablations on PDE Representation

- Raw PDE parameters are not helpful
- Different from positional encoding in Transformer
- ϵ_r + *wave prior* is the best setting

ϵ_r	λ	$\widetilde{\Omega}$	$\mathcal{P}_x$	$\mathcal{P}_z$	$\lvert\Psi_\theta(\mathcal{E}(a)) - \Psi^*(a)\rvert$	Test N-MAE
✓						0.165
✓			$x\mathbf{1}^T/N$	$\mathbf{1}z^T/M$		0.176
✓	✓	✓				0.220
✓	✓	✓	***Wave Prior***			0.152
			Wave Prior			0.149
✓			***Wave Prior***			**0.122**

Ablations on Superposition Mixedup

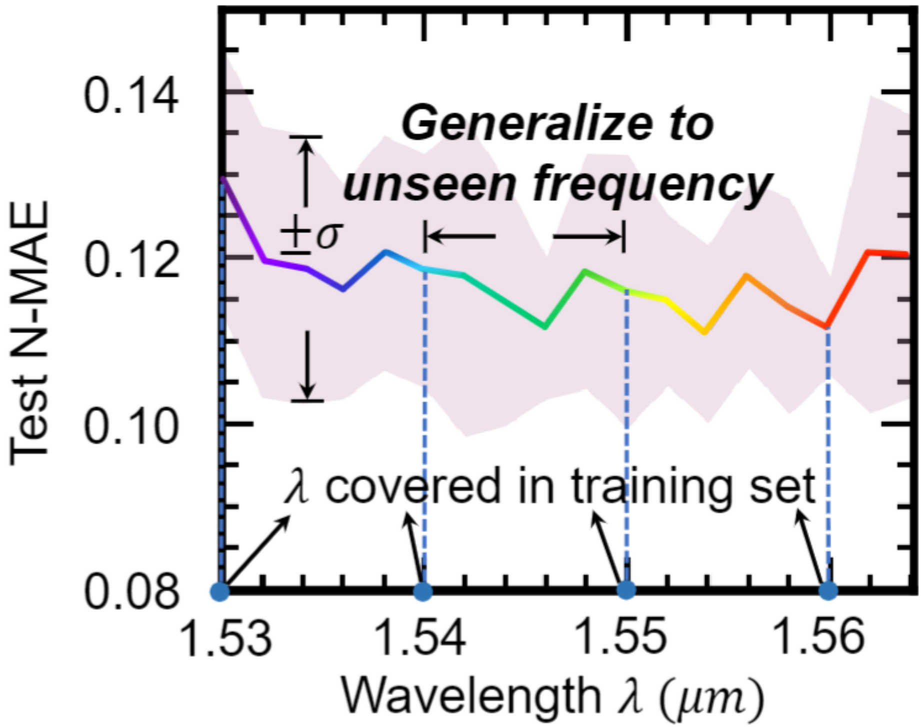
- **Single-source simulation**: save dataset acquisition cost
- **Multi-source training**: significantly boost generalization
- **Multi-source test**: fast one-shot prediction
- Low runtime cost & high data efficiency & good generalization

Train Augmentation	Inference Mode	#Train Examples (K) 1.4	4.1	6.9	9.7	12.4	Runtime (ms)
None	Single-Source	0.346	0.257	0.202	0.198	0.194	23.8
	Multi-Source	0.892	0.882	0.880	0.865	0.873	8.3
Superposition Mixup	Single-Source	0.229	0.205	0.204	0.200	0.199	23.8
	Multi-Source	**0.230**	**0.208**	**0.206**	**0.202**	**0.202**	**8.3**

Spectrum Analysis: Generalize to Wavelengths

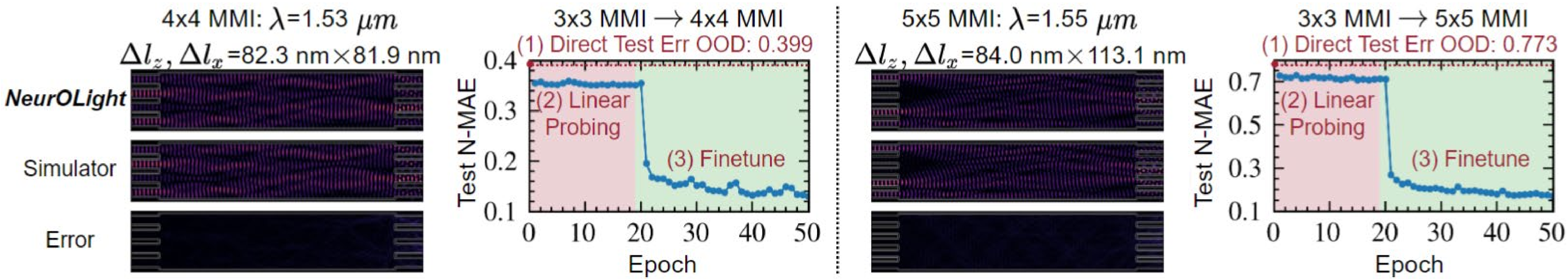
- Train only sees 5 wavelengths
- Sweep over C-band, 2-nm step
- Generalize to unseen devices with unseen wavelengths

FDFD: >1 min* v.s. *NeurOLight: <150 ms



Device Adaptation

- Adapt from 3x3 MMIs to 4x4 and 5x5 MMIs
- Finetuning to close the out-of-distribution generalization gap
  - Small number of new data
  - Short tuning steps



Conclusion

- NeurOLight framework to predict light fields for photonic devices
- New neural operator model: parameter-efficient & data-efficient
- ***2-order-of-magnitude*** faster runtime than numerical solver
- Generalize to large design space
- 53.8% better prediction fidelity and 44.2% less parameter cost

- Preprint: https://arxiv.org/abs/2209.10098
- Open-source codes: https://github.com/JeremieMelo/NeurOLight