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

Anonymous Author(s)
Affiliation
Address
email

Abstract

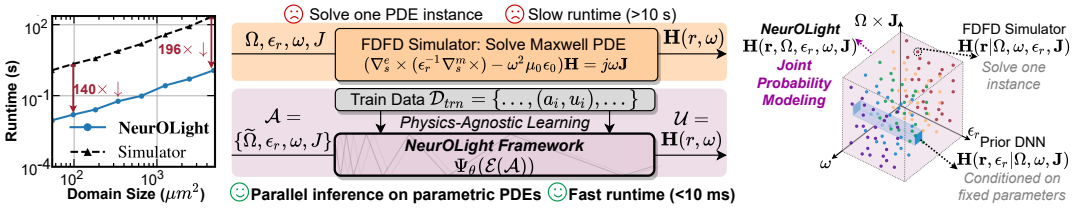
Optical computing is an emerging technology for next-generation efficient artificial intelligence (AI) due to its ultra-high speed and efficiency. Electromagnetic field simulation is critical to the design, optimization, and validation of photonic devices and circuits. However, costly numerical simulation significantly hinders the scalability and turn-around time in the photonic circuit design loop. Recently, physics-informed neural networks have been proposed to predict the optical field solution of a single instance of a partial differential equation (PDE) with predefined parameters. Their complicated PDE formulation and lack of efficient parametrization mechanisms limit their flexibility and generalization in practical simulation scenarios. In this work, *for the first time*, a physics-agnostic neural operator-based framework, dubbed `NeurOLight`, is proposed to learn a family of frequency-domain Maxwell PDEs for ultra-fast parametric silicon-photonic device simulation. We balance the efficiency and generalization of `NeurOLight` via several novel techniques. Specifically, we discretize different devices into a unified domain, represent parametric PDEs with a compact wave prior, and encode the incident light via masked source modeling. We design our model with parameter-efficient cross-shaped `NeurOLight` blocks and adopt superposition-based augmentation for data-efficient learning. With these synergistic approaches, `NeurOLight` generalizes to a large space of unseen simulation settings, demonstrates *2-orders-of-magnitude* faster simulation speed than numerical solvers, and outperforms prior neural network models by $\sim$54% lower prediction error with $\sim$44% fewer parameters. Our code will be publicly available.

1 Introduction

With recent advances in silicon-photonics technology, optical deep learning represents a new paradigm in next-generation efficient artificial intelligence (AI) [27, 26, 3]. An increasing number of co-design efforts have been made to enable synergistic *light-AI interaction*. We see extensive developments for photonic AI with rapidly evolving optical neural network (ONN) hardware accelerator designs [27, 7, 38, 5, 22, 28] and various circuit-algorithm co-optimization methodologies [7, 8, 6, 29, 9, 13]. However, *applying AI for optics* is much less explored. Complementary to prior AI-assisted architectural exploration [18, 10], a natural question is *whether AI can assist in the lower-level device simulation* that requires a deep understanding of the physical nature of optics.

AI-assisted photonic device simulation is a critical step to closing the synergistic loop of *light-AI interaction*. Besides using standard devices that already have a compact transfer matrix [27, 7, 31],

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) (b) (c)

Figure 1: (a) Compare FDFD simulation and our `NeurOLight` framework. (b) `NeurOLight` (1 Quadro RTX 6000 GPU) runs 140×-200× faster than FDFD simulator (8-core i7-9700 CPUs) across different domain sizes (50 nm grid step). (c) Different methods cover different solution space.

modern optical AI shows a trend to exploit customized photonic structures for scalable optical computing [6, 29, 39, 35]. Unfortunately, because customized devices do not have analytical transfer functions, understanding their behavior heavily relies on numerical simulators [15] to solve Maxwell partial differential equations (PDEs) to obtain the optical field distribution. Even solving a single 2-dimensional (2-D) simulation instance on a 5×20 μm^2 region can cost nearly 4 s on 8 CPUs, as shown in Figure 1(a), which significantly hinders scalable circuit-level simulation and optimization. Hence, our target is to propose a surrogate model that learns the *light propagation principle* and efficiently approximates the field solutions to new simulation instances while running *orders-of-magnitude faster* than the numerical solver, as shown in Figure 1(b).

Several prior works attempt to leverage neural networks (NNs) [30, 34] and physics-informed NNs (PINNs) [32, 2, 21] with physics-augmented residual loss to predict electromagnetic field solutions. However, previous methods have three major limitations. First, as illustrated in Figure 1(c), they only model the field distribution conditioned on pre-defined solving domains, input sources, and frequency. In other words, their models only learn the solution of a certain PDE instance with fixed parameters. Second, they all belong to the category of PINNs [24] that require an explicit description of the PDE as well as strict initial/boundary conditions. Constructing complicated Maxwell equation-based residual loss is no easier than re-implementing a numerical solver [23, 19, 20]. Third, their CNN-based models show inadequate modeling capacity with too small receptive fields to learn important light propagation, scattering, and interference effects.

To learn *a family of parametric Maxwell PDEs* that models the *joint probability* of different design variables as shown in Figure 1(c), we propose a physics-agnostic light field prediction framework `NeurOLight` that consists of a joint PDE encoder and an efficient cross-shaped neural operator backbone. The main contributions of the work are as follows:

- *For the first time*, an AI-based photonic device simulation framework is proposed to *learn a family of parametric* Maxwell PDEs for ultra-fast optical field prediction.
- We propose a novel joint PDE encoder for compact PDE representation and an efficient cross-shaped Fourier neural operator backbone for end-to-end optical field prediction, over 2-order-of-magnitude faster than numerical simulators.
- We propose a superposition-based mixup technique that dynamically boosts the data efficiency and generalization during the training of `NeurOLight`.
- On different photonic device benchmarks, our `NeurOLight` achieves the best prediction fidelity, generalizability, and transferability, outperforming UNet and state-of-the-art (SoTA) Fourier neural operators by an average of 53.8% lower prediction error with 44.2% fewer parameters.
- To our best knowledge, this is *the first* AI-based framework that can learn the terahertz light propagation inside photonic devices that generalizes to different domains, permittivities, input sources, and frequencies. We will open-source our framework.

2 Related Work

Optical field simulation with machine learning. Finite difference frequency domain (FDFD) simulation is a widely adopted method to analyze silicon-photonic devices. Numerical simulators are used to solve frequency-domain Maxwell PDEs to obtain electromagnetic field distributions of an optical component with terahertz incident light sources. To accelerate this time-consuming process, NNs have been utilized as surrogate models for fast optical simulation approximation. A multi-layer perceptron was used to map the design variables to a scalar performance metric of a power splitter [30]. NNs were also utilized to predict intermediate values to accelerate the convergence of the numerical solver [34]. MaxwellNet [21] was proposed to train a UNet with physics-informed loss to predict the scattered field based on the material permittivity of the free-space lens. WaveYNet [2] also adopted a 2-D UNet as the model trained with both data-driven supervision loss and Maxwell-equation-based physics-augmented loss to predict the optical field of silicon meta-lens. However, prior NN-based methods require explicit physics knowledge of the Maxwell PDEs and only learn a small field solution space conditioned on fixed parameters.

Learning PDEs via neural operators. Recently, neural operators have been proposed as new NN models that learn a family of parametric PDEs in the infinite-dimensional function space in a mesh-free and purely data-driven fashion. The Fourier neural operator (FNO) [20] approximates the nonlinear mapping from PDE observations to solutions through Fourier-domain kernel integral operations, achieving record-breaking performance on a wide range of challenging applications. Several variants have been proposed to improve the performance and efficiency of the original FNO models, e.g., Factorized FNO [33], U-FNO [36], and multiwavelet-based neural operator [11].

3 Neural operator-based optical field simulation framework `NeurOLight`

3.1 Understanding optical field simulation for photonic devices

Silicon waveguides can confine the incident laser beam and allow the optical signals to propagate and interfere with each other. Various optical components, e.g., couplers, shifters, and multi-mode interference (MMI) devices [4], can create phase shifts, magnitude modulation, and interference. Given a linear isotropic optical component, we will shine time-harmonic continuous-wave light on its input ports and analyze the steady-state electromagnetic field distributions $\mathbf{E} = \hat{\mathbf{x}}\mathbf{E}_x + \hat{\mathbf{y}}\mathbf{E}_y + \hat{\mathbf{z}}\mathbf{E}_z$ and $\mathbf{H} = \hat{\mathbf{x}}\mathbf{H}_x + \hat{\mathbf{y}}\mathbf{H}_y + \hat{\mathbf{z}}\mathbf{H}_z$ in it, each of which includes horizontal (x), vertical (y), and longitudinal (z) components. We can solve the steady-state optical field $\mathbf{E}(\boldsymbol{r})$ and $\mathbf{H}(\boldsymbol{r})$ from the frequency-domain *curl-of-curl* Maxwell PDE under absorptive boundary conditions [15] (details in Appendix A1),

$$\big((\mu_0^{-1}\nabla\times\nabla\times) - \omega^2\epsilon_0\boldsymbol{\epsilon}_r(\boldsymbol{r})\big)\mathbf{E}(\boldsymbol{r}) = j\omega\mathbf{J}_e(\boldsymbol{r}),\ \big(\nabla\times(\boldsymbol{\epsilon}_r^{-1}(\boldsymbol{r})\nabla\times) - \omega^2\mu_0\epsilon_0\big)\mathbf{H}(\boldsymbol{r}) = j\omega\mathbf{J}_m(\boldsymbol{r}) \quad (1)$$

where $\nabla\times$ is the curl operator of a vector function, μ_0 is the vacuum magnetic permeability, ϵ_0 and $\boldsymbol{\epsilon}_r$ are the vacuum and relative electric permittivity, and $\mathbf{J}_m$ and $\mathbf{J}_e$ are the magnetic and electric current sources. FDFD simulation discretizes the continuous domain into an $M \times N$ mesh grid and solves the above linear equation $\mathbf{AX} = \boldsymbol{b}$ to obtain the fields. Solving for these optical fields exactly with a sparse matrix $\mathbf{A} \in \mathbb{C}^{MN\times MN}$ is prohibitively expensive and not scalable to large photonic devices. A fast surrogate model that predicts optical fields with high fidelity is of tremendous interest.

3.2 The proposed `NeurOLight` framework

As shown in Figure 2, `NeurOLight` models the optical field simulation problem as an infinite-dimensional-space mapping from Maxwell PDE observations $\mathcal{A} \in \mathbb{C}^{\Omega\times d_a}$ to the optical field solution $\mathcal{U} \in \mathbb{C}^{\Omega\times d_u}$. Here, Ω is the continuous 2-D physical solving domain, $\Omega = (l_x, l_z)$, typically in units of micrometers (μm), where the photonic device-of-interest can be tightly located. $\mathcal{A}$ and $\mathcal{U}$ take values with d_a and d_u dimensions, respectively. To learn the ground truth nonlinear mapping $\Psi^* : \mathcal{A} \to \mathcal{U}$, we construct `NeurOLight` with a PDE encoder $\mathcal{E}$ followed by an efficient parametric approximator Ψ_θ that minimizes the empirical error on discrete observable samples $a \sim \mathcal{A}$,

$$\theta^* = \min_\theta \mathbb{E}_{a\sim\mathcal{A}}\big[\mathcal{L}\big(\Psi_\theta(\mathcal{E}(a)), \Psi^*(a)\big)\big]. \quad (2)$$

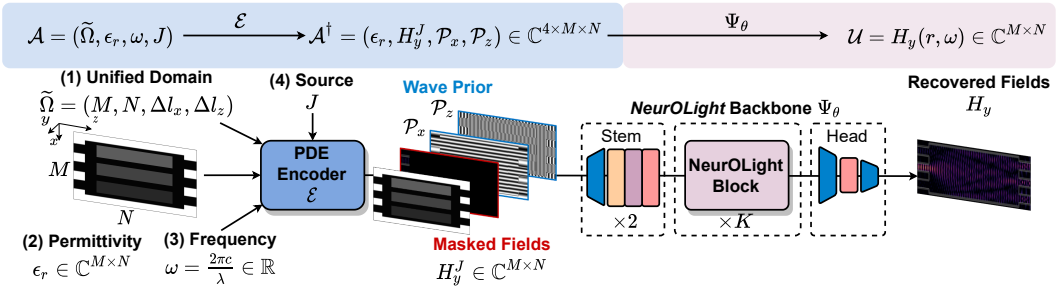


Figure 2: `NeurOLight` framework for optical field simulation. Real part is plotted for complex fields.

3.2.1 Scale-adaptive domain discretization: $\Omega \to \widetilde{\Omega}$

To generalize to PDEs in *different physical domains* and support batched parallel inference, we adopt an $M \times N$ discrete unified domain $\widetilde{\Omega} = (M, N, \Delta l_x, \Delta l_z)$ with an *adaptive mesh granularity*, i.e., with grid steps $\Delta l_x = l_x/M$ and $\Delta l_z = l_z/N$. As shown in Figure. 3, multiple photonic devices with different physical dimensions are normalized to the same $\widetilde{\Omega}$. Their original physical dimensions can be elegantly encoded into the re-calculated mesh granularities. This unified discrete domain gives `NeurOLight` the flexibility to handle parallel inference on different physical domain dimensions, unlike prior work [21, 2] that requires time-consuming model retraining once the physical domain changes.

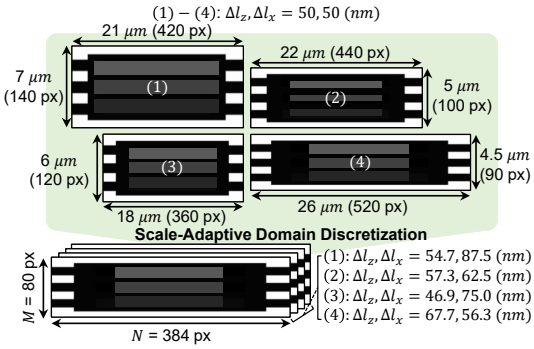


Figure 3: Scale-adaptive domain discretization enables generalization to different solving domain dimensions and efficient batched processing.

3.2.2 Joint PDE representation: $\mathcal{A} \to \mathcal{A}^\dagger$

After we define a unified solving domain, we need to construct effective PDE representations that describe the raw observations $\mathcal{A} = (\widetilde{\Omega}, \boldsymbol{\epsilon}_r, \omega, \mathbf{J})$. The relative permittivity distribution can be simply represented by $\boldsymbol{\epsilon}_r \in \mathbb{C}^{M\times N}$. However, how to compactly encode other parameters, i.e., $(\widetilde{\Omega}, \omega, \mathbf{J})$, remains a non-trivial challenge. Let us first consider what makes a good representation. First, it needs to be compatible with the model input, i.e., it can be fused with the 2-D image representation in a *compact* way. Second, it is preferred to reveal the physical essence of the parameters and inject useful *prior knowledge* that helps model generalization. Based on the above considerations, we propose a PDE encoder $\mathcal{E} : \mathcal{A} \to \mathcal{A}^\dagger$ that converts the raw observations to a joint PDE representation $\mathcal{A}^\dagger = (\boldsymbol{\epsilon}_r, \mathbf{H}_y^J, \mathcal{P}_x, \mathcal{P}_z)$.

Encoding $(\widetilde{\Omega}, \boldsymbol{\epsilon}_r, \omega)$ via wave prior. The intuition behind the wave prior design is that the vacuum angular frequency $\omega = \frac{2\pi c}{\lambda}$ and electric permittivity $\boldsymbol{\epsilon}_r$ together decide the physical light wavelength inside the material, i.e., $\lambda' = \lambda/\sqrt{\boldsymbol{\epsilon}_r}$. The mesh granularity determines how many pixels can depict a wave period along both directions, i.e., $(\lambda'/\Delta l_x, \lambda'/\Delta l_z)$. Therefore, as shown in Figure 4, we construct artificial wave patterns, named *wave prior*, as $\mathcal{P}_z = e^{j\frac{2\pi\sqrt{\boldsymbol{\epsilon}_r}}{\lambda}\mathbf{1}z^T\Delta l_z}$ and $\mathcal{P}_x = e^{j\frac{2\pi\sqrt{\boldsymbol{\epsilon}_r}}{\lambda}x\mathbf{1}^T\Delta l_x}$, where $x = (0, 1, \cdots, M-1)$ and

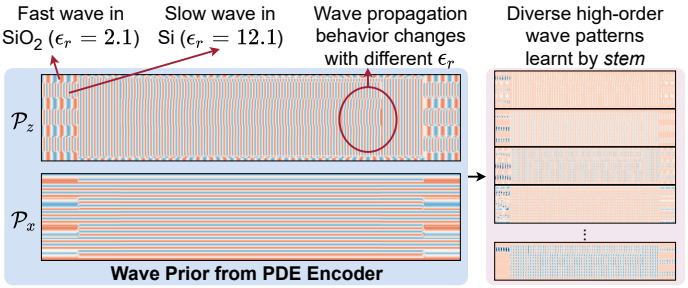


Figure 4: Wave prior as joint PDE representations.

$z = (0, 1, \cdots, N-1)$. The wave prior jointly translates the $(\widetilde{\Omega}, \boldsymbol{\epsilon}_r, \omega)$ pair to a unified representation with strong prior knowledge, significantly reducing the learning difficulty on overly-abstract raw observations. We note that the `NeurOLight` stem learns complex combinations of the wave prior and generates diverse high-order wave patterns for later feature transformation.

Masked image modeling-inspired light source (J) encoding. In the optical simulation, *light source* $\mathbf{J}$ will be injected by shining light on the input waveguides of the photonic devices as stimuli to the system. $\mathbf{J}$ will have a vacuum angular frequency ω and a polarization mode. For example, in the transverse magnetic (TM) mode, we have $\mathbf{H}_x = \mathbf{H}_z = 0$. Thus we focus on the prediction of $\mathbf{H}_y$. However, as shown in Figure 5, $\mathbf{J}$ is a combination of multiple length-w 1-D vectors, where w is the input port width, placed at the input waveguides, which is hard to be encoded in the image prediction flow. Therefore, we borrow the idea of *masked image modeling* [1] to light source encoding. In the source representation $\mathbf{H}^J$, we only maintain the fields in the input waveguides before entering the key region of the device, which is easy to obtain and irrelevant to the structure it enters into, and mask out all the fields after. In this way, the field prediction task translates to a *masked field restoration* task conditioned on the input light source as a *hint*.

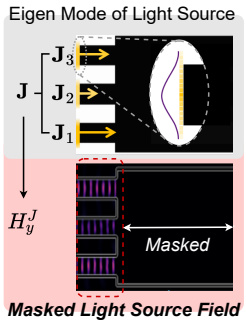


Figure 5: Masked light source modeling.

3.3 Efficient `NeurOLight` model architecture: Ψ_θ

Convolutional stem. The proposed `NeurOLight` architecture starts with a convolutional stem $\mathcal{S}$: $a^\dagger(\boldsymbol{r}) \to v_0(\boldsymbol{r}), \forall \boldsymbol{r} \in \Omega$ that encodes each complex-valued observation sample $a^\dagger(\boldsymbol{r}) \in \mathbb{C}^{4\times M\times N}$ to a real-valued representation $v_0(\boldsymbol{r}) \in \mathbb{R}^{C\times M\times N}$. Lightweight blueprint convolutions [12] are used to perform local wave pattern transformation with a low hardware cost.

Cross-shaped `NeurOLight` block. In the projected C-dimensional space, we place K cascaded `NeurOLight` blocks to gradually restore the complex light field in the frequency domain as $v_0(\boldsymbol{r}) \to v_1(\boldsymbol{r}) \to \cdots \to v_K(\boldsymbol{r})$. Each `NeurOLight` block is formulated as

$$v_{k+1}(\boldsymbol{r}) := \texttt{FFN}\big((\mathcal{K}v_k)(\boldsymbol{r})\big) + v_k,\ \forall \boldsymbol{r} \in \Omega; \quad (\mathcal{K}v_k)(\boldsymbol{r}) = \int_\Omega \kappa(\boldsymbol{r}_1, \boldsymbol{r}_2) v_k(\boldsymbol{r}_2) \mathrm{d}v_k(\boldsymbol{r}_2), \forall \boldsymbol{r}_1 \in \Omega, \tag{3}$$

where $\mathcal{K}$ is a learnable kernel integral transform, and $\texttt{FFN}(\cdot)$ is a feedforward network. When the kernel satisfies $\kappa(\boldsymbol{r}_1, \boldsymbol{r}_2) = \kappa(\boldsymbol{r}_1 - \boldsymbol{r}_2)$, the above integral kernel operator is equivalent to a spatial-domain 2-D convolution, which can be efficiently computed by using Fourier transform $\mathcal{F}(\cdot)$ [20]. A clear downside of the original 2-D FNO is the huge parameter cost, i.e., $\mathcal{F}(\kappa)(\boldsymbol{r}) \in \mathbb{C}^{k_v\times k_h\times C\times C}$, and the resultant severe overfitting issues. To improve the model *efficiency and generalization* simultaneously, we introduce a *cross-shaped Fourier neural operator*, shown in Figure 6. The input feature is first bi-partitioned along the channel dimension into two chunks $v_k(\boldsymbol{r}) = [v_k^h(\boldsymbol{r}); v_k^v(\boldsymbol{r})]$, representing horizontal and vertical patterns, and 1-D FNO is applied to both directions,

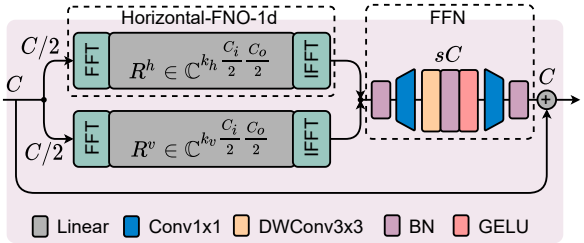


Figure 6: `NeurOLight` backbone model design.

$$\begin{aligned}
(\mathcal{K}^h v_k^h)(\boldsymbol{r}) &= \mathcal{F}_z^{-1}\big(\mathcal{F}_z(\kappa^h) \cdot \mathcal{F}_z(v_k^h)\big)(\boldsymbol{r}) = \mathcal{F}_z^{-1}\big(R^h(z) \cdot \mathcal{F}_z(v_k^h(\boldsymbol{r}))\big), \forall z \in \Omega_z, \forall r \in \Omega, \\
(\mathcal{K}^v v_k^v)(\boldsymbol{r}) &= \mathcal{F}_x^{-1}\big(\mathcal{F}_x(\kappa^v) \cdot \mathcal{F}_x(v_k^v)\big)(\boldsymbol{r}) = \mathcal{F}_x^{-1}\big(R^v(x) \cdot \mathcal{F}_x(v_k^v(\boldsymbol{r}))\big), \forall x \in \Omega_x, \forall r \in \Omega, \\
(\mathcal{K}v_k)(\boldsymbol{r}) &= [(\mathcal{K}^h v_k^h)(\boldsymbol{r}); (\mathcal{K}^v v_k^v)(\boldsymbol{r})].
\end{aligned} \tag{4}$$

We parametrize the Fourier kernels as lightweight complex-valued tensors $R^h(z) \in \mathbb{C}^{k_z\times\frac{C}{2}\times\frac{C}{2}}$ and $R^v(x) \in \mathbb{C}^{k_x\times\frac{C}{2}\times\frac{C}{2}}$. These orthogonal 1-D kernel operations intrinsically perform spatial and channel-wise feature aggregation in an interleaved way that are able to provide global receptive fields to achieve long-distance modeling.

To increase nonlinearity and enhance local information interaction, we append an `FFN` block after the cross-shaped FNO. Inspired by the MixFFN designs in SoTA vision transformers [37], our `FFN` expands the channels by s times, performs local information aggregation via 3×3 depth-wise convolution (DWConv), activates using the GELU function, and projects it back to C channels.

Projection head. At the end, two point-wise convolutional layers are used to project $v_K(\boldsymbol{r})$ to the light field space $u(\boldsymbol{r}) = \mathcal{Q}(v_K(\boldsymbol{r}))$. Dropout layer is inserted to mitigate overfitting issues.

Loss function. Even with normalized light source power, optical fields tend to have distinct statistics. To balance the optimization effort among different fields, we adopt the normalized mean absolute error (N-MAE) as the objective $\mathcal{L}\big(\Psi_\theta(\mathcal{E}(a)), \Psi^*(a)\big) = (\|\Psi_\theta(\mathcal{E}(a)) - \Psi^*(a)\|_1)/\|\Psi^*(a)\|_1$.

3.4 Toward better data efficiency and generalization via superposition-based mixup

The PDE observations $\mathcal{A}$ can cover a huge design space. Hence, the data efficiency and generalization of pure data-driven models raise a concern. Simply drawing large numbers of random training examples with all possible light sources has an intractable data acquisition cost. Standard augmentation techniques are effective in improving data efficiency and generalization on tasks with *natural images*; however, their direct application is *not compatible with PDE simulation*. Since $\Psi^*(a_i)$ is a highly nonlinear function of a_i and closely related to the boundary conditions, simultaneously augmenting $\boldsymbol{\epsilon}_r$, Ω, ω, and $\mathbf{H}$, e.g., via cropping, distortion, or resizing, leads to *invalid* field solutions.

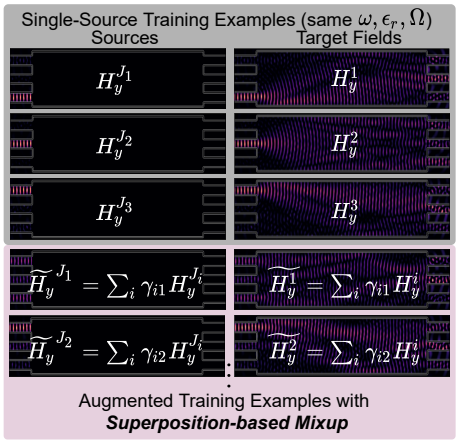


Figure 7: Data augmentation with superposition-based mixup. Only real part is plotted for each field.

Interestingly, we notice that the photonic system satisfies the *superposition* property w.r.t. the light source,

$$\widetilde{\mathbf{H}} = \Psi^*(\widetilde{\mathbf{H}}^J) = \Psi^*(\sum_{i=1}^{|\mathbf{J}|} \gamma_i \mathbf{H}^{J_i}) = \sum_{i=1}^{|\mathbf{J}|} \gamma_i \Psi^*(\mathbf{H}^{J_i}). \quad (5)$$

Based on this, we only involve *single-input simulation data* in the training set and dynamically mix multiple ($|\mathbf{J}|$) input sources via *Superposition-based Mixup*, as shown in Figure 7,

$$\begin{pmatrix} \widetilde{\mathbf{H}}^{J_1} & \dots & \widetilde{\mathbf{H}}^{J_{|\mathbf{J}|}} \\ \widetilde{\mathbf{H}}^1 & \dots & \widetilde{\mathbf{H}}^{|\mathbf{J}|} \end{pmatrix}^T = \boldsymbol{\Gamma} \begin{pmatrix} \mathbf{H}^{J_1} & \dots & \mathbf{H}^{J_{|\mathbf{J}|}} \\ \mathbf{H}^1 & \dots & \mathbf{H}^{|\mathbf{J}|} \end{pmatrix}^T, \ \boldsymbol{\Gamma} \in \mathbb{C}^{|\mathbf{J}|\times|\mathbf{J}|}, \|\boldsymbol{\Gamma}_j\|_2 = 1, \phi(\gamma_{j1}) = 0, \forall j \in [|\mathbf{J}|]. \quad (6)$$

At each iteration, we randomly generate the mixup coefficient matrix $\boldsymbol{\Gamma}$ and make it have a unit row-wise ℓ_2-norm for power normalization. The global phase of the complex-valued optical field is normalized by forcing the first input port always to have a phase that equals 0. In this way, `NeurOLight` learns how multiple incident light sources interfere with one another. Once `NeurOLight` can generalize to arbitrary light sources, multi-source simulation only needs *a one-shot inference with superposed source fields* instead of explicitly accumulating $|\mathbf{J}|$ single-source inference results.

4 Results

4.1 Experiment setup

Datasets. We focus on widely applied multi-mode interference (MMI) photonic devices. We select MMIs with rectangular tunable control pads (*Tunable MMI*). The permittivity of the tuning region can be programmed by external signals, so this family of devices exemplifies photonic structures with reconfigurable transmissions [17]. We also evaluate MMIs with rectangular etched cavities (*Etched MMI*) [30], which exemplifies optical devices with fixed yet highly discrete permittivity distributions. We use an open-source 2-D FDFD simulator angler [15] to simulate optical fields for randomly generated MMIs as our dataset. Details on dataset generation is in Appendix A2.

4.2 Main results

In Table 1, we compare four models: (1) UNet [21, 2], (2) a 5-layer FNO-2d [20], (3) a 12-layer factorized FNO (F-FNO) [33], and (4) our `NeurOLight`. Detailed training settings and model architectures can be found in Appendix A3 and Appendix A4, respectively. On these benchmarks, `NeurOLight` outperforms UNet and prior SoTA FNO variants by **53.8%** lower test error with **44.2%** fewer parameters on average.

Results on tunable MMI. On tunable MMI, `NeurOLight` achieves the best prediction error with only half the parameter cost. Figure 8 visualizes the field prediction for one test MMI. UNet is significantly limited by its small receptive field and lack of long-distance modeling capability, thus failing to predict the full field even with the hint of wave prior. As representative neural operators, FNO-2d and factorized FNO (F-FNO) manifest the superior advantages of the Fourier-domain kernel integral operations, showing considerably lower prediction errors than their CNN counterparts. However, given the parameter budget (∼3 M), the 5-layer FNO-2d only has 10 modes in the x-direction and 32-modes in the z-direction, which may not be enough to extract high-frequency waves. The 12-layer F-FNO adopts factorized 1-D Fourier kernel to save parameters; however, its modeling capability is limited by the lack of local feature extractors. Our `NeurOLight` blocks benefit from the global view of the cross-shaped 1-D kernel and local feature aggregation from convolutional `FFN` blocks. In the training curves in Figure 10(a), `NeurOLight` achieves the fastest convergence and best generalization among all models. We animate the *real-time* prediction process of `NeurOLight` in Appendix A5.

Table 1: Comparison of parameter count, train error, and test error on two benchmarks among four different models.

Benchmarks	Model	#Params (M) ↓	Train Err ↓	Test Err ↓
Tunable MMI	UNet [21, 2]	3.47	0.776	0.801
	FNO-2d [20]	3.29	0.231	0.244
	F-FNO [33]	3.16	0.272	0.292
	`NeurOLight`	**1.58**	**0.145**	**0.122**
Etched MMI	UNet [21, 2]	3.47	0.779	0.792
	FNO-2d [20]	3.29	0.601	0.648
	F-FNO [33]	3.16	0.411	0.525
	`NeurOLight`	**2.11**	**0.376**	**0.387**
Average Improvement		**-44.2%**	**-49.1%**	**-53.8%**

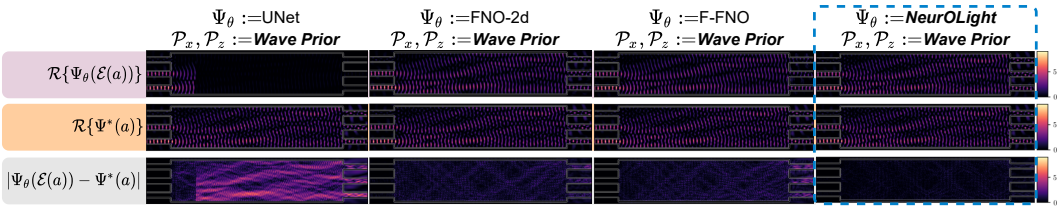


Figure 8: Visualization on one test tunable MMI. ($\Delta l_x = 83.1\ nm, \Delta l_z = 70.8\ nm, \lambda = 1.54\ \mu m$).

Results on etched MMI. Compared with tunable MMIs, predicting the field in etched MMIs, even with 2× more training examples, is a much harder task given the complicated scattering at the cavity-silicon interface and the considerably larger and highly discrete design space, shown in Figure 9. Hence, we increase the model capacity of `NeurOLight` by using 16 layers. Among all prediction models, `NeurOLight` achieves the best results with 42% lower error while still saving 36% parameters on average.

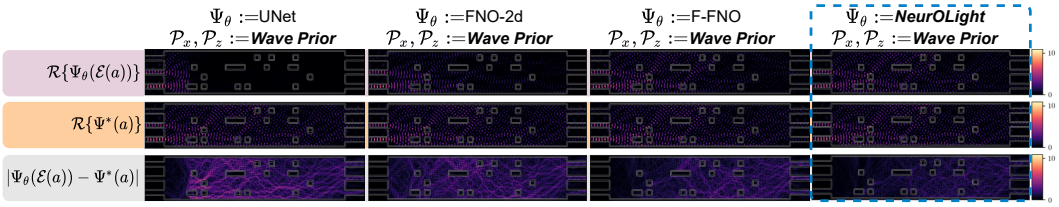
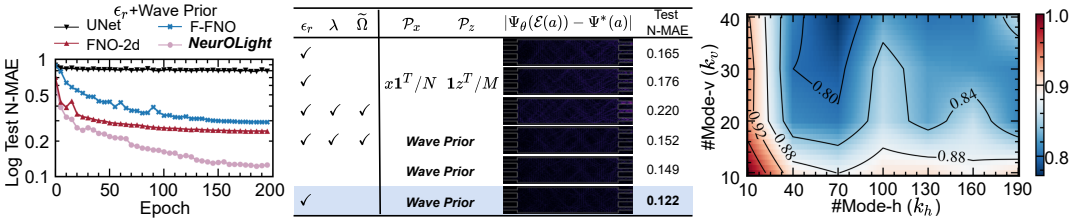


Figure 9: Visualization on one test etched MMI. ($\Delta l_x = 91.3\ nm, \Delta l_z = 89.1\ nm, \lambda = 1.55\ \mu m$).



(a) (b) (c)

Figure 10: (a) Test N-MAE curves of four models. (b) Our PDE encoder achieves the lowest error. (c) Normalized test error contour of a 8-layer `NeurOLight` with different # of Fourier modes.

4.3 Ablation studies

PDE encoding. In Figure 10(b), we extensively compare different combinations of PDE encoding methods. The first two methods only model the distribution over $\boldsymbol{\epsilon}_r$ conditioned on fixed wavelength and domain sizes like prior work [21, 2], which fail to generalize to examples in larger design space. With raw PDE parameters ($\boldsymbol{\epsilon}_r, \lambda, \tilde{\Omega}$), the model finds it difficult to learn a generalizable representation, thus showing large errors on the test dataset. The last three combinations validate that permittivity and our wave prior are compact and effective encodings in our joint PDE representation method, while extra raw wavelength and domain information are redundant and even harmful.

Fourier modes. As shown in Figure 10(c), we perform a fine-grained exploration in the Fourier mode space to find the most suitable configuration. Unlike the flow prediction tasks evaluated in FNO [20] that only require a few modes, using inadequate Fourier modes fails to learn the terahertz high-frequency optical fields in the photonic device simulation task. However, using all Fourier series is not necessary and makes the model prone to overfitting issues. (k_h, k_v)=(70, 40) is the best setting that balances expressiveness and efficiency in our `NeurOLight`.

Cross-shaped `NeurOLight` block. In Table 2, we *independently* change one technique in the full `NeurOLight` model to verify each individual contribution. Our proposed essential techniques *synergistically* boost the modeling capacity and generalization. Compared to the linear lifting in FNO-2d that only performs point-wise projection, our lightweight convolution stem can extract complex high-order wave patterns with negligible runtime overhead. Similar to U-FNO [36], we append an additional parallel convolution path alongside the cross-shaped FNO block; however, the extra 20% runtime penalty does not pay off. The proposed convolutional `FFN` significantly improves the nonlinearity and local feature extraction ability of `NeurOLight`. Changing it to a simple BatchNorm-GELU causes significant degradation. Different from the MLP-based FFN in F-FNO [33], our extra depthwise CONV is critical to local feature extraction and can reduce the test error by 16%. Note that an extra GELU after the FNO block will distort the feature in the low-dimensional space and have a negative impact on the performance [25]. Besides the dropout in the head, stochastic network depth [14] in the residual `NeurOLight` block is also effective in mitigating overfitting.

Table 2: Ablation on proposed techniques. Each entry changes one technique independently. Runtime is averaged over multiple runs on 1 NVIDIA Quadro RTX 6000 GPU.

Variants	#Params (M)↓	Train Err ↓	Test Err ↓	Runtime (ms) ↓
NeurOLight	**1.58**	**0.145**	**0.122**	**12.1**
ConvStem → Lifting	1.58	0.156	0.134	11.9
Extra Parallel Conv Path	1.64	0.149	0.129	14.5
FFN → BN-GELU	1.37	0.469	0.446	6.3
Remove DWConv in FFN	1.57	0.164	0.144	10.6
Extra GELU After FNO	1.58	0.164	0.148	12.4
Remove DropPath	1.58	0.131	0.136	12.1

4.4 Discussion

Superposition-based mixup. As shown in Table 3, without augmentation, `NeurOLight` only sees single-source training examples, thus failing to generalize when multiple sources are fused as a unified input source for *fast one-shot* prediction, named multi-source inference mode. A simple

work-around would be to perform single-source prediction on $|\mathbf{J}|$ ports and superpose the resultant $|\mathbf{J}|$ fields, named single-source inference mode. When training on a large enough training set, this method indeed works. However, it quickly deteriorates as the training set reduces with a $|\mathbf{J}|$ times higher runtime cost for a $|\mathbf{J}|$-port device. With our dynamic superposition-based mixup, `NeurOLight` works well both in single-source and multi-source inference modes with superior generalizability even with only 10% training data.

Table 3: Test N-MAE of an 8-layer `NeurOLight` with different number of training examples. Multi-source inference mode has similar performance as the single-source method but shows 3× faster runtime on 3×3 MMIs.

Train Augmentation	Inference Mode	#Train Examples (K)					Runtime (ms)
		1.4	4.1	6.9	9.7	12.4	
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. Spectroscopy is an important approach to understanding the broadband response of a photonic device. A traditional FDFD simulator has to sweep the spectrum with multiple simulations. In contrast, `NeurOLight` models the joint probability over wavelengths, and thus only needs to perform parallel inference with different $\omega = \frac{2\pi}{\lambda}$ values *at one shot*. Figure 11 demonstrates that, though `NeurOLight` is only trained with five selected wavelengths, it can generalize to unseen devices with unseen wavelengths. Sweeping in the standard C-band (1550 nm-1565 nm) with a 2 nm granularity, `NeurOLight` can finish within 150 ms, achieving 450× speedup over the FDFD simulator.

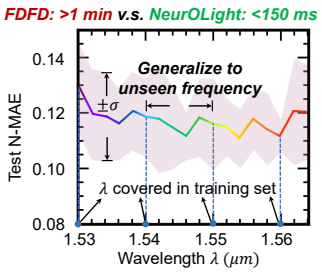


Figure 11: `NeurOLight` can generalize to unseen devices and wavelengths.

Device adaptation. We evaluate the transferability of `NeurOLight` via device adaptation. In Figure 12, we transfer `NeurOLight` trained on 3-port MMIs to larger MMIs with 4 and 5 ports. Directly predicting new devices shows unsatisfying test error out of distribution (OOD). We adapt the model with 20-epoch fast linear probing and 30-epoch finetuning [16] on 3.7 K 4-port MMI examples and 4.6 K 5-port MMI examples. The model quickly transfers to new photonic devices with good prediction fidelity.

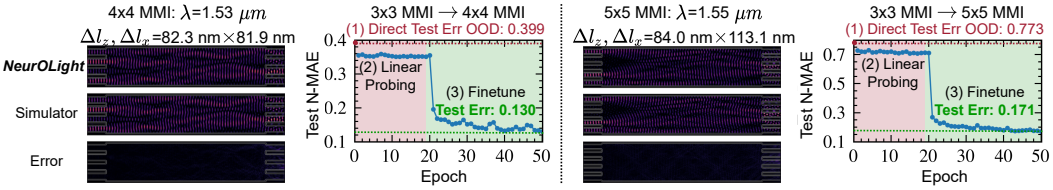


Figure 12: Device adaptation from 3-port to 4-/ 5-port MMI via linear probing and finetuning.

5 Conclusion

In this work, *for the first time*, a physics-agnostic neural operator, named `NeurOLight`, is proposed for ultra-fast parametric photonic device simulation. We propose a joint PDE encoder with wave prior and masked source modeling for compact PDE representation. Our lightweight cross-shaped `NeurOLight` backbone design achieves a superior balance between modeling capability and parameter efficiency. In addition, our novel superposition-based mixup technique significantly boosts the data efficiency and model generalizability. Experiments show that `NeurOLight` outperforms prior DNN models with 53.8% better prediction fidelity and 44.2% less parameter cost, serving as an over 100× faster surrogate model to the numerical solvers in photonic device simulation. Currently, our model focuses on device-level simulation. As a future direction, we look forward to exploring the circuit-level simulation and utilizing our model to streamline the optimization loop for efficient AI-assisted optical circuit design automation.

References

[1] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proc. NeurIPS*, 2020.

[2] Mingkun Chen, Robert Lupoiu, Chenkai Mao, Der-Han Huang, Jiaqi Jiang, Philippe Lalanne, and Jonathan Fan. Physics-augmented deep learning for high-speed electromagnetic simulation and optimization. *Nature*, 2021.

[3] Q. Cheng, J. Kwon, M. Glick, M. Bahadori, L. P. Carloni, and K. Bergman. Silicon Photonics Codesign for Deep Learning. *Proceedings of the IEEE*, 2020.

[4] Kieran Cooney and Frank H. Peters. Multimode Interference Couplers with Tunable Power Splitting Ratios. *Optics Express*, 2016.

[5] Johannes Feldmann, Nathan Youngblood, Maxim Karpov, Helge Gehring, Xuan Li, Maik Stappers, Manuel Le Gallo, Xin Fu, Anton Lukashchuk, Arslan Raja, Junqiu Liu, David Wright, Abu Sebastian, Tobias Kippenberg, Wolfram Pernice, and Harish Bhaskaran. Parallel convolutional processing using an integrated photonic tensor core. *Nature*, 2021.

[6] Jiaqi Gu, Chenghao Feng, Zheng Zhao, Zhoufeng Ying, et al. Squeezelight: Towards scalable optical neural networks with multi-operand ring resonators. In *Proc. DATE*, 2021.

[7] Jiaqi Gu, Zheng Zhao, Chenghao Feng, et al. Towards area-efficient optical neural networks: an FFT-based architecture. In *Proc. ASPDAC*, 2020.

[8] Jiaqi Gu, Zheng Zhao, Chenghao Feng, Hanqing Zhu, Ray T. Chen, and David Z. Pan. ROQ: A noise-aware quantization scheme towards robust optical neural networks with low-bit controls. In *Proc. DATE*, 2020.

[9] Jiaqi Gu, Hanqing Zhu, Chenghao Feng, Zixuan Jiang, et al. L2ight: Enabling on-chip learning for optical neural networks via efficient in-situ subspace optimization. In *Proc. NeurIPS*, 2021.

[10] Jiaqi Gu, Hanqing Zhu, Chenghao Feng, Zixuan Jiang, et al. Adept: Automatic differentiable design of photonic tensor cores. In *Proc. DAC*, 2022.

[11] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In *Proc. NeurIPS*, 2021.

[12] Daniel Haase and Manuel Amthor. Rethinking Depthwise Separable Convolutions: How Intra-Kernel Correlations Lead to Improved MobileNets. In *Proc. CVPR*, pages 14588–14597, 2020.

[13] Ryan Hamerly, Liane Bernstein, Alexander Sludds, et al. Large-scale optical neural networks based on photoelectric multiplication. *Phys. Rev. X*, 2019.

[14] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep Networks with Stochastic Depth. In *Proc. ECCV*, 206.

[15] Tyler W. Hughes, Momchil Minkov, Ian A. D. Williamson, and Shanhui Fan. Adjoint method and inverse design for nonlinear nanophotonic devices. *ACS Photonics*, 2018.

[16] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *Proc. ICLR*, 2022.

[17] Juerg Leuthold and Charles H. Joyner. Multimode Interference Couplers with Tunable Power Splitting Ratios. *J. Lightwave Technol.*, 2001.

[18] Mengquan Li, Zhongzhi Yu, Yongan Zhang, Yonggan Fu, and Yingyan Lin. O-has: Optical hardware accelerator search for boosting both acceleration performance and development speed. In *Proc. ICCAD*, 2021.

[19] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differ- ential equations. *arXiv preprint, arXiv:2003.03485*, 2020.

[20] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *Proc. ICLR*, 2021.

[21] Joowon Lim and Demetri Psaltis. Maxwellnet: Physics-driven deep neural network training based on maxwell's equations. *Appl. Phys. Lett.*, 2022.

[22] W. Liu, W. Liu, Y. Ye, Q. Lou, Y. Xie, and L. Jiang. Holylight: A nanophotonic accelerator for deep learning in data centers. In *Proc. DATE*, 2019.

[23] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint, arXiv:1910.03193*, 2019.

[24] M. Raissia, P. Perdikarisb, and G. E. Karniadakisa. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comp. Phys.*, 2019.

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proc. CVPR*, 2018.

[26] Bhavin J. Shastri, Alexander N. Tait, T. Ferreira de Lima, Wolfram H. P. Pernice, Harish Bhaskaran, C. D. Wright, and Paul R. Prucnal. Photonics for artificial intelligence and neuromorphic computing. *Nature Photonics*, 2021.

[27] Yichen Shen, Nicholas C. Harris, Scott Skirlo, et al. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 2017.

[28] Kyle Shiflett, Dylan Wright, Avinash Karanth, and Ahmed Louri. PIXEL: Photonic Neural Network Accelerator. In *Proc. HPCA*, pages 474–487, 2020.

[29] Febin Sunny, Asif Mirza, Mahdi Nikdast, and Sudeep Pasricha. Crosslight: A cross-layer optimized silicon photonic neural network accelerator. In *Proc. DAC*, 2021.

[30] Mohammad H. Tahersima, Keisuke Kojima, Toshiaki Koike-Akino, Devesh Jha, BingnanWang, and Chungwei Lin. Deep neural network inverse design of integrated photonic power splitters. *Sci. Rep.*, 2019.

[31] Alexander N. Tait, Thomas Ferreira de Lima, Ellen Zhou, et al. Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.*, 2017.

[32] Yingheng Tang, Jichao Fan, Xinwei Li, Jianzhu Ma, Minghao Qi, Cunxi Yu, and Weilu Gao. Physics-guided and physics-explainable recurrent neural network for time dynamics in optical resonances. *Nat. Compu. Sci.*, 2022.

[33] Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *NeurIPS Workshop*, 2021.

[34] Rahul Trivedi, Logan Su, Jesse Lu, Martin F. Schubert, and JelenaVuckovic. Data-driven acceleration of photonic simulations. *Sci. Rep.*, 2019.

[35] Zi Wang, Lorry Chang, Feifan Wang, Tiantian Li, and Tingyi Gu. Integrated photonic metasystem for image classifications at telecommunication wavelength. *preprint*, 2021.

[36] Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-fno: an enhanced fourier neural operator based-deep learning model for multiphase flow. *arXiv preprint, arXiv:2109.03697*, 2021.

[37] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *Proc. NeurIPS*, 2021.

[38] Xingyuan Xu, Mengxi Tan, Bill Corcoran, Jiayang Wu, Andreas Boes, Thach G. Nguyen, Sai T. Chu, Brent E. Little, Damien G. Hicks, Roberto Morandotti, Arnan Mitchell, and David J. Moss. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature*, 2021.

[39] H. H. Zhu, J. Zou, H. Zhang, Y. Z. Shi, S. B. Luo, et al. Space-efficient optical computing with an integrated chip diffractive neural network. *Nature Communications*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please see Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We attached codes in the supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Appendix A3 and Appendix A4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Since the training process is deterministic and not sensitive to random seeds. Also computing resources are limited.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see Appendix A3
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The new dataset generation code is in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]