Generative Neural Network Based Non-Convex Optimization Using Policy Gradients with an Application to Electromagnetic Design

Anonymous Author(s)
Affiliation
Address
email

Abstract

A generative neural network based non-convex optimization algorithm using a one-step implementation of the policy gradient method is introduced and applied to electromagnetic design. We demonstrate state-of-the-art performance of electromagnetic devices called grating couplers, with key advantages over local gradient-based optimization via the adjoint method.

1 Introduction

There has been a recent surge in research syncretizing electromagnetics (EM) and machine learning (ML), including photonic analog accelerators [1–8], physics emulators [9–15], and ML-enhanced EM design techniques [15–35]. While local gradient-based optimization with the adjoint method has been successfully applied to many EM design problems [36–60], EM design leveraging ML techniques promises superior computational performance, advanced data analysis and insight, and improved device optimization. Pursuing the lattermost topic, we propose a generative neural network (NN) based optimization method leveraging a policy gradient update, called PHORCED = PHotonic Optimization using REINFORCE Criteria for Enhanced Design.

While the target application discussed below is EM design, the method underlying PHORCED is applicable to non-convex optimization in general and is relevant to a broad class of science and engineering disciplines, such as fluid mechanics [61]. Indeed, PHORCED does not require a gradient evaluation of the target physical performance function and is compatible with any external physics solver (with or without built-in capability for the adjoint method or automatic differentiation). Furthermore, we will show, through the example of grating coupler optimization (key components in integrated photonics systems), that PHORCED exhibits fewer overall simulation evaluations (the limiting computational bottleneck in many engineering design problems) relative to competing generative NN based approaches previously proposed for EM design [24, 25].

Our contributions

1. We demonstrate state-of-the-art grating coupler performance after optimization with PHORCED, with resilience to choice of initial condition (in contrast to local gradient-based optimization).

2. We show that PHORCED is amenable to transfer learning, reducing overall simulation requirements for similar EM design optimizations by a factor $10\times$ relative to the original training.

3. We contrast PHORCED with a related algorithm called GLOnet [24–26]. Related work in alternative disciplines may be found in Refs. [61–64].

Submitted to NeurIPS 2021 AI for Science Workshop.

2 Neural network generators for optimization: GLOnet vs. PHORCED

Let us assume we need to optimize a function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to its n-dimensional input vector $\mathbf{p} = [p_1, p_2, ..., p_{n-1}, p_n]^T \in \mathbb{R}^n$, that is, we need to find $\mathbf{p}^* = \text{argmax}_{\mathbf{p}}\, f(\mathbf{p})$.

2.1 GLOnet

GLOnet [24, 25] uses a NN generator $h_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^n$ to produce a candidate solution $\mathbf{p} = h_{\boldsymbol{\theta}}(\mathbf{z})$ from a noise vector $z \sim \mathcal{D}$ sampled from a noise distribution $\mathcal{D}$. The NN generator reward function $F(\boldsymbol{\theta})$ is the expected value of $f(\mathbf{p})$ when $\mathbf{z}$ is drawn from $\mathcal{D}$. $F(\boldsymbol{\theta})$ and $\frac{\partial}{\partial \theta_j} F(\boldsymbol{\theta})$ (derived in the appendix) are:

$$F(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\mathbf{p}=h_{\boldsymbol{\theta}}(\mathbf{z}),\, \mathbf{z}\sim\mathcal{D}}[f(\mathbf{p})] = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D}}[(f \circ h_{\boldsymbol{\theta}})(\mathbf{z})] \tag{1}$$

$$\frac{\partial}{\partial \theta_j} F(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D}} \left[\left. \frac{\partial f(\mathbf{p})}{\partial \mathbf{p}} \right|_{\mathbf{p}=h_{\boldsymbol{\theta}}(\mathbf{z})} \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial \theta_j} \right] \tag{2}$$

These reward gradients are used to update the NN parameters $\boldsymbol{\theta}$ iteratively. We note that one needs to compute the derivatives of the original objective function, $\frac{\partial f(\mathbf{p})}{\partial \mathbf{p}}$, in this method.

2.2 PHORCED

We propose PHORCED, which uses a NN generator $\pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})$ that again takes a noise vector $\mathbf{z} \sim \mathcal{D}$ as input but instead produces as output the mean and variance of a Gaussian distribution over candidate solutions. The NN generator reward function $G(\boldsymbol{\theta})$ is the expected value of $f(\mathbf{p})$ over samples of $\mathbf{z}$ and $\mathbf{p}$. $G(\boldsymbol{\theta})$ and $\frac{\partial}{\partial \theta_j} G(\boldsymbol{\theta})$ (derived in the appendix) are:

$$G(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D},\mathbf{p}\sim\pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})}[f(\mathbf{p})] \tag{3}$$

$$\frac{\partial}{\partial \theta_j} G(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D},\mathbf{p}\sim\pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})} \left[f(\mathbf{p}) \frac{\partial \log \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})}{\partial \theta_j} \right] \tag{4}$$

These reward gradients are used to update the NN parameters $\boldsymbol{\theta}$ iteratively. A key advantage of this method is that we **do not need to compute the derivatives** of the original objective function, $\frac{\partial f(\mathbf{p})}{\partial \mathbf{p}}$.

2.3 Advantages of PHORCED

1. PHORCED works even in situations where it is infeasible to compute gradients of the merit function with respect to its arguments.
2. In problems where gradients can be computed but require significant effort, PHORCED can reduce the computational load because it avoids computing them altogether. This is demonstrated by application to EM design in the next section.

3 Electromagnetic optimization example: grating coupler design

Here, we optimize a grating coupler (Fig. 1(a)), an integrated photonic device used in scientific research and high-speed data communications, which diffracts light (1550nm central wavelength) from an on-chip waveguide to an external optical fiber. The grating coupler consists of periodically spaced corrugations on the silicon waveguide in the center of Fig. 1(a). The EM objective function is light coupling efficiency into a Gaussian beam mode (diameter 10.4μm) [44, 52, 65], propagating 8° relative to the waveguide normal. The grating coupler has 60 designable parameters: the width of and spacing between 30 waveguide corrugations (Fig. 1(a)). Individual grating coupler simulations of this geometry require ~4 seconds with 30 concurrent MPI processes (where EMopt [43, 66] was employed for forward and adjoint simulations).

We compare BFGS, GLOnet, and PHORCED with two initial designs, Seed 1 (Fig. 1(b)), and Seed 2 (Fig 1(c)). Seed 1 has a length-wise linearly increasing corrugation duty cycle; Seed 2 has a uniform duty cycle of 90%. Both designs have pitch that satisfies the grating equation [44, 52] for 8° scattering.

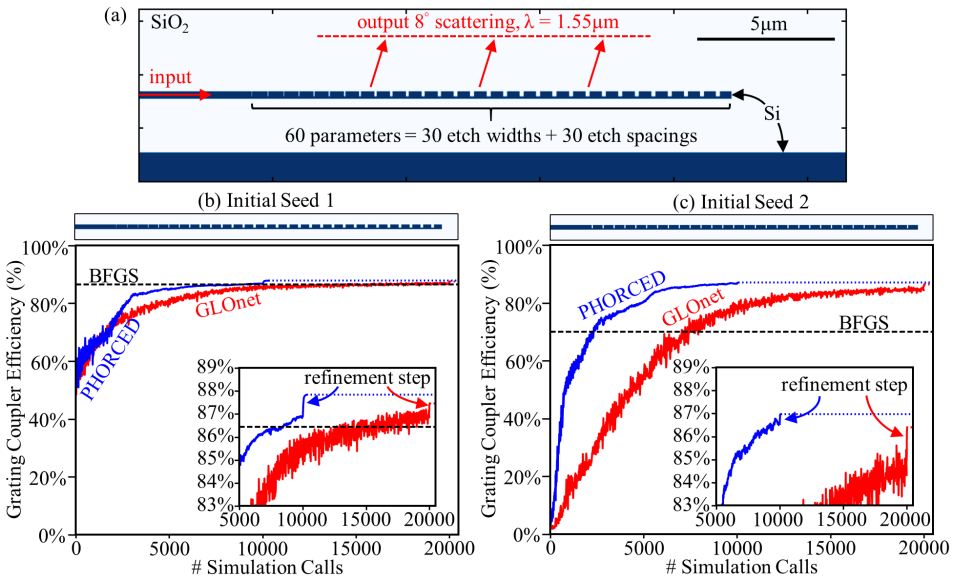

Figure 1: PHORCED and GLOnet outperform conventional gradient-based optimization, with different simulation evaluation requirements. The grating coupler simulation geometry is given in (a). Optimizations were performed on Seed 1 (b) and Seed 2 (c), illustrated above the plots. Insets depict zoomed-in views of the peak efficiencies attained by PHORCED and GLOnet after BFGS refinement.

Table 1: Optimized results

Optimization	# of EM simulations	Final efficiency
BFGS, seed 1	144	86.4%
GLOnet, seed 1	20,023	87.4%
PHORCED, seed 1	10,327	**87.8%**
BFGS, seed 2	214	69.9%
GLOnet, seed 2	20,029	86.4%
PHORCED, seed 2	10,057	**87.0%**

These seeds were chosen to explore algorithm robustness to "good" and "poor" initialization. Seed 1 satisfies physical intuition and already has an efficiency of 56%, while Seed 2 has a low efficiency of 1%. We use these two cases as a proof-of-concept to demonstrate that PHORCED and GLOnet can boost EM performance even when the initial condition is not inspired by physical intuition, as is common in high-dimensionality problems.

The standard approach to optimize EM designs is local gradient-based optimization via the adjoint method where the EM merit function gradient, $\frac{\partial f(\mathbf{p})}{\partial \mathbf{p}}$, can be calculated with just two EM simulations regardless of the dimension of $\mathbf{p} \in \mathbb{R}^n$. We implemented this approach using BFGS with default settings from the SciPy optimize module. GLOnet and PHORCED were implemented in PyTorch. GLOnet uses a convolutional NN and an exponentially-weighted EM performance function originally recommended by Jiang and Fan [24, 25]. PHORCED uses a fully-connected NN that outputs the parameters of a multivariate, isotropic Gaussian distribution. GLOnet was initialized by adding the design vector representing Seed 1/Seed 2 to the direct neural network output. PHORCED was initialized by adding the design vector to the mean of the output Gaussian. In both methods, we ran 1,000 total optimizer iterations, with 10 devices sampled per iteration. GLOnet requires an adjoint simulation for each device but PHORCED does not. The # of EM simulations in the table is computed as: # of optimizer iterations $\times$ # of devices sampled per iteration $\times$ # of simulations per device $+$ # refinement simulations. Hyperparameters were individually tuned for GLOnet and PHORCED, but our hyperparameter search was limited by the prohibitive EM simulation computation requirements (training required ~12 and ~36 hours for PHORCED and

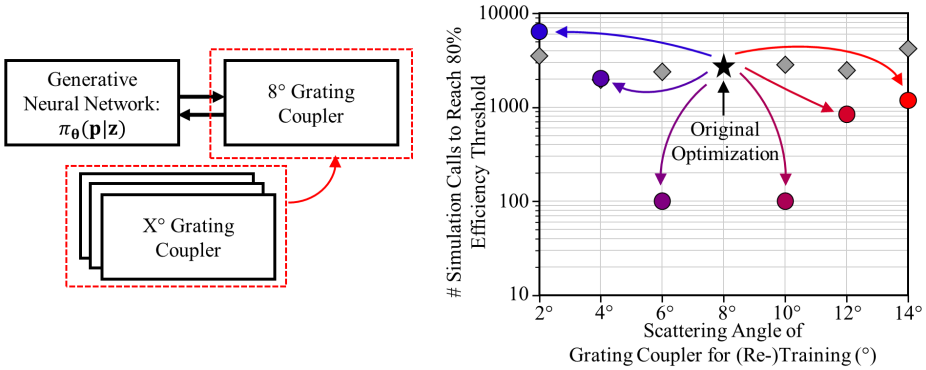

Figure 2: Transfer learning applied on original grating coupler (8°, black star) to nearby scattering angles (color-coded circles) can drastically lower overall simulations needed to match performance of control cases (gray diamonds).

GLOnet respectively). The outputs of the GLOnet and PHORCED optimizations were further refined using BFGS. The results are shown in Table 1 and plotted in Fig. 1(b) and Fig. 1(c).

PHORCED and GLOnet clearly outperform BFGS for both initial conditions. In fact, PHORCED approaches an estimated theoretical upper bound of 88% as suggested by an optical thin-film calculation [55, 67]. While GLOnet had comparable performance to PHORCED, PHORCED needed $2\times$ fewer EM simulations because it does not require adjoint gradient calculations.

Transfer learning

Unlike alternative optimizers, NN generators can be subjected to transfer learning to optimize related problems with accelerated convergence. As a proof-of-concept, in Fig. 2, we use transfer learning to re-train the generator originally obtained for the $8°$ case using Seed 1 to instead produce devices at nearby scattering angles, $\{2°, 4°, 6°, 10°, 12°, 14°\}$. We characterize the improvement of transfer learning relative to the original optimization and control cases in terms of # simulation calls to reach 80% grating coupler efficiency, a desirable benchmark in commercial photonics. We observe that devices with similar physics to the original optimization ($6°$ and $10°$) exhibit over $10\times$ reduction in required simulation evaluations, while there is only temperate improvement for $12°$ and $14°$, and no improvement for $2°$ and $4°$. These latter two cases are expected because grating couplers become plagued by parasitic back-reflection at small diffraction angles [44, 52], and it is possible the optimizer struggles to adapt to these previously un-encountered physics. In another work [68], A. Anonymous showed that transfer learning can instead be applied adiabatically, resulting in both improved computational speed and grating coupler efficiency for all the above scattering angles.

4 Analysis and Conclusion

Perhaps the most important result in Fig. 1 is that both PHORCED and GLOnet proved to be robust to choice of initialization. Indeed, while BFGS provided a competitive result for Seed 1, it failed for Seed 2. PHORCED and GLOnet, on the other hand, attained results within 1% coupler efficiency for both seeds. Thus, we claim generative NN based optimization offers the possibility of global optimization effort in EM design problems, with resiliency to missing human intuition, at the cost of additional simulations. Moreover, we demonstrated that PHORCED can reduce overall simulation requirements compared to competing global optimizers by (1) removing gradient calculations and (2) applying transfer learning to physically similar design problems (Fig. 2). A. Anonymous showed elsewhere [68] that transfer learning further improves speed when the computational stringency of a device optimization is turned on adiabatically. Furthermore, we believe that our implementation of PHORCED has significant room for improvement. For example, one could leverage concepts from reinforcement learning such as importance sampling [63, 69], or complementary model-based methods that utilize a physics surrogate/inverse model to reduce the number of costly full simulations needed for training [13, 19, 28, 32]. We anticipate that further cross-pollination of physical design and reinforcement learning could open the floodgates for new possibilities in scientific optimization.

References

[1] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, "Deep Learning with Coherent Nanophotonic Circuits," *Nature Photonics*, vol. 11, no. June, 2016. DOI: 10.1038/nphoton.2017.93.

[2] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K.-i. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue, "A coherent Ising machine for 2000-node optimization problems," *Science*, vol. 4243, no. October, 2016. DOI: 10.1126/science.aah4243.

[3] N. C. Harris, G. R. Steinbrecher, M. Prabhu, Y. Lahini, J. Mower, D. Bunandar, C. Chen, F. N. Wong, T. Baehr-Jones, M. Hochberg, S. Lloyd, and D. Englund, "Quantum transport simulations in a programmable nanophotonic processor," *Nature Photonics*, vol. 11, no. 7, pp. 447–452, 2017. DOI: 10.1038/nphoton.2017.95.

[4] Y. Yamamoto, K. Aihara, T. Leleu, K.-i. Kawarabayashi, S. Kako, M. Fejer, K. Inoue, and H. Takesue, "Coherent Ising machines—optical neural networks operating at the quantum limit," *npj Quantum Information*, vol. 3, no. 1, pp. 1–15, Dec. 2017. DOI: 10.1038/s41534-017-0048-9.

[5] E. Khoram, A. Chen, D. Liu, L. Ying, Q. Wang, M. Yuan, and Z. Yu, "Nanophotonic media for artificial neural inference," *Photonics Research*, vol. 7, no. 8, 2018. DOI: 10.1364/prj.7.000823.

[6] T. W. Hughes, I. A. D. Williamson, M. Minkov, and S. Fan, "Wave physics as an analog recurrent neural network," *Science Advances*, vol. 5, no. 12, 2019. DOI: 10.1126/sciadv.aay6946.

[7] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. P. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, "Photonics for artificial intelligence and neuromorphic computing," *Nature Photonics*, vol. 15, no. 2, pp. 102–114, Feb. 2021. DOI: 10.1038/s41566-020-00754-y.

[8] X. Xu, M. Tan, B. Corcoran, J. Wu, A. Boes, T. G. Nguyen, S. T. Chu, B. E. Little, D. G. Hicks, R. Morandotti, A. Mitchell, and D. J. Moss, "11 TOPS photonic convolutional accelerator for optical neural networks," *Nature*, vol. 589, no. 7840, pp. 44–51, Jan. 2021. DOI: 10.1038/s41586-020-03063-0.

[9] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. DOI: https://doi.org/10.1016/j.jcp.2018.10.045.

[10] D. Gostimirovic and W. N. Ye, "An Open-Source Artificial Neural Network Model for Polarization-Insensitive Silicon-on-Insulator Subwavelength Grating Couplers," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 25, no. 3, pp. 1–5, May 2019. DOI: 10.1109/JSTQE.2018.2885486.

[11] Y. Guo, X. Cao, B. Liu, and M. Gao, "Solving Partial Differential Equations Using Deep Learning and Physical Constraints," *Applied Sciences*, vol. 10, no. 17, p. 5917, Jan. 2020. DOI: 10.3390/app10175917.

[12] Y. Chen, L. Lu, G. E. Karniadakis, and L. D. Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials," *Optics Express*, vol. 28, no. 8, pp. 11 618–11 633, Apr. 2020. DOI: 10.1364/OE.384875.

[13] R. Pestourie, Y. Mroueh, T. V. Nguyen, P. Das, and S. G. Johnson, "Active learning of deep surrogates for PDEs: Application to metasurface design," *npj Computational Materials*, vol. 6, no. 1, pp. 1–7, Oct. 2020. DOI: 10.1038/s41524-020-00431-2.

[14] A. Ghosh, D. J. Roth, L. H. Nicholls, W. P. Wardley, A. V. Zayats, and V. A. Podolskiy, "Machine Learning-Based Diffractive Image Analysis with Subwavelength Resolution," *ACS Photonics*, vol. 8, no. 5, pp. 1448–1456, May 2021. DOI: 10.1021/acsphotonics.1c00205.

[15] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, "Physics-informed neural networks with hard constraints for inverse design," *arXiv:2102.04626 [physics]*, Feb. 2021.

[16] R. Trivedi, L. Su, J. Lu, M. F. Schubert, and J. Vuckovic, "Data-driven acceleration of photonic simulations," *Scientific Reports*, vol. 9, no. 1, pp. 1–7, 2019. DOI: 10.1038/s41598-019-56212-5.

[17] Y. Qu, L. Jing, Y. Shen, M. Qiu, and M. Soljačić, "Migrating Knowledge between Physical Scenarios Based on Artificial Neural Networks," *ACS Photonics*, vol. 6, no. 5, pp. 1168–1174, May 2019. DOI: 10.1021/acsphotonics.8b01526.

[18] D. Melati, Y. Grinberg, M. Kamandar Dezfouli, S. Janz, P. Cheben, J. H. Schmid, A. Sánchez-Postigo, and D. X. Xu, "Mapping the global design space of nanophotonic components using machine learning pattern recognition," *Nature Communications*, vol. 10, no. 1, pp. 1–9, 2019. DOI: 10.1038/s41467-019-12698-1.

[19] M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, "Deep Neural Network Inverse Design of Integrated Photonic Power Splitters," *Scientific Reports*, vol. 9, no. 1, p. 1368, Feb. 2019. DOI: 10.1038/s41598-018-37952-2.

[20] A. Demeter-Finzi and S. Ruschin, "S-matrix absolute optimization method for a perfect vertical waveguide grating coupler," *Optics Express*, vol. 27, no. 12, pp. 16 713–16 718, Jun. 2019. DOI: 10.1364/OE.27.016713.

[21] M. K. Dezfouli, Y. Grinberg, D. Melati, J. H. Schmid, P. Cheben, S. Janz, and D.-X. Xu, "Design of fully apodized and perfectly vertical surface grating couplers using machine learning optimization," in *Integrated Optics: Devices, Materials, and Technologies XXV*, vol. 11689, International Society for Optics and Photonics, Mar. 2021, 116890J. DOI: 10.1117/12.2576945.

[22] M. M. R. Elsawy, S. Lanteri, R. Duvigneau, G. Brière, M. S. Mohamed, and P. Genevet, "Global optimization of metasurface designs using statistical learning methods," *Scientific Reports*, vol. 9, no. 1, p. 17 918, Nov. 2019. DOI: 10.1038/s41598-019-53878-9.

[23] A. M. Hammond and R. M. Camacho, "Designing integrated photonic devices using artificial neural networks," *Optics Express*, vol. 27, no. 21, pp. 29 620–29 638, Oct. 2019. DOI: 10.1364/OE.27.029620.

[24] J. Jiang and J. A. Fan, "Global Optimization of Dielectric Metasurfaces Using a Physics-Driven Neural Network," *Nano Letters*, vol. 19, no. 8, pp. 5366–5372, Aug. 2019. DOI: 10.1021/acs.nanolett.9b01857.

[25] J. Jiang and J. A. Fan, "Simulator-based training of generative neural networks for the inverse design of metasurfaces," *Nanophotonics*, vol. 9, no. 5, pp. 1059–1069, 2020. DOI: doi:10.1515/nanoph-2019-0330.

[26] J. Jiang and J. A. Fan, "Multiobjective and categorical global optimization of photonic structures based on ResNet generative neural networks," *Nanophotonics*, vol. 10, no. 1, pp. 361–369, Sep. 2020. DOI: doi:10.1515/nanoph-2020-0407.

[27] J. Jiang, M. Chen, and J. A. Fan, "Deep neural networks for the evaluation and design of photonic devices," *Nature Reviews Materials*, pp. 1–22, Dec. 2020. DOI: 10.1038/s41578-020-00260-1.

[28] R. S. Hegde, "Deep learning: A new tool for photonic nanostructure design," *Nanoscale Advances*, vol. 2, no. 3, pp. 1007–1023, 2020. DOI: 10.1039/c9na00656g.

[29] M. Minkov, I. A. Williamson, L. C. Andreani, D. Gerace, B. Lou, A. Y. Song, T. W. Hughes, and S. Fan, "Inverse Design of Photonic Crystals through Automatic Differentiation," *ACS Photonics*, vol. 7, no. 7, pp. 1729–1741, 2020. DOI: 10.1021/acsphotonics.0c00327.

[30] S. So, T. Badloe, J. Noh, J. Rho, and J. Bravo-Abad, "Deep learning enabled inverse design in nanophotonics," *Nanophotonics*, vol. 2234, pp. 1–17, 2020. DOI: 10.1515/nanoph-2019-0474.

[31] Z. Ma and Y. Li, "Parameter extraction and inverse design of semiconductor lasers based on the deep learning and particle swarm optimization method," *Optics Express*, vol. 28, no. 15, p. 21 971, 2020. DOI: 10.1364/oe.389474.

[32] K. Kojima, M. H. Tahersima, T. Koike-Akino, D. K. Jha, Y. Tang, Y. Wang, and K. Parsons, "Deep Neural Networks for Inverse Design of Nanophotonic Devices," *Journal of Lightwave Technology*, vol. 39, no. 4, pp. 1010–1019, Feb. 2021. DOI: 10.1109/JLT.2021.3050083.

[33] W. Ma, Z. Liu, Z. A. Kudyshev, A. Boltasseva, W. Cai, and Y. Liu, "Deep learning for the design of photonic structures," *Nature Photonics*, vol. 15, no. 2, pp. 77–90, 2021. DOI: 10.1038/s41566-020-0685-y.

[34] D. Melati, M. Kamandar Dezfouli, Y. Grinberg, J. H. Schmid, R. Cheriton, S. Janz, P. Cheben, and D. X. Xu, "Design of Compact and Efficient Silicon Photonic Micro Antennas with Perfectly Vertical Emission," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 27, no. 1, 2021. DOI: 10.1109/JSTQE.2020.3013532.

[35] R. Hegde, "Sample-efficient deep learning for accelerating photonic inverse design," *OSA Continuum*, vol. 4, no. 3, pp. 1019–1033, Mar. 2021. DOI: 10.1364/OSAC.420977.

[36] J. S. Jensen and O. Sigmund, "Topology optimization for nano-photonics," *Laser and Photonics Reviews*, vol. 5, no. 2, pp. 308–321, 2011. DOI: 10.1002/lpor.201000014.

[37] J. Lu and J. Vučković, "Objective-first design of high-efficiency, small-footprint couplers between arbitrary nanophotonic waveguide modes," *Optics Express*, vol. 20, no. 7, pp. 7221–7236, Mar. 2012. DOI: 10.1364/OE.20.007221.

[38] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch, "Adjoint shape optimization applied to electromagnetic design," *Optics Express*, vol. 21, no. 18, pp. 21 693–21 701, Sep. 2013. DOI: 10.1364/OE.21.021693.

[39] Y. Elesin, B. S. Lazarov, J. S. Jensen, and O. Sigmund, "Time domain topology optimization of 3D nanophotonic devices," *Photonics and Nanostructures - Fundamentals and Applications*, vol. 12, no. 1, pp. 23–33, Feb. 2014. DOI: 10.1016/j.photonics.2013.07.008.

[40] A. Y. Piggott, J. Lu, K. G. Lagoudakis, J. Petykiewicz, T. M. Babinec, and J. Vučković, "Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer," *Nature Photonics*, vol. 9, no. 6, pp. 374–377, Jun. 2015. DOI: 10.1038/nphoton.2015.69.

[41] L. F. Frellsen, Y. Ding, O. Sigmund, and L. H. Frandsen, "Topology optimized mode multiplexing in silicon-on-insulator photonic wire waveguides," *Optics Express*, vol. 24, no. 15, pp. 16 866–16 873, Jul. 2016. DOI: 10.1364/OE.24.016866.

[42] L. Su, R. Trivedi, N. V. Sapra, A. Y. Piggott, D. Vercruysse, and J. Vučković, "Fully-automated optimization of grating couplers," *Optics Express*, vol. 26, no. 4, pp. 2614–2617, 2017. DOI: 10.1364/OE.26.004023.

[43] A. Michaels and E. Yablonovitch, "Leveraging continuous material averaging for inverse electromagnetic design," *Optics Express*, vol. 26, no. 24, pp. 31 717–31 737, Nov. 2018. DOI: 10.1364/OE.26.031717.

[44] A. Michaels and E. Yablonovitch, "Inverse design of near unity efficiency perfectly vertical grating couplers," *Optics Express*, vol. 26, no. 4, pp. 4766–4779, Feb. 2018. DOI: 10.1364/OE.26.004766.

[45] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, "Inverse design in nanophotonics," *Nature Photonics*, vol. 12, no. 11, pp. 659–670, Nov. 2018. DOI: 10.1038/s41566-018-0246-9.

[46] T. W. Hughes, M. Minkov, I. A. D. Williamson, and S. Fan, "Adjoint Method and Inverse Design for Nonlinear Nanophotonic Devices," *ACS Photonics*, vol. 5, no. 12, pp. 4781–4787, Dec. 2018. DOI: 10.1021/acsphotonics.8b01522.

[47] Y. Liu, W. Sun, H. Xie, N. Zhang, K. Xu, Y. Yao, S. Xiao, and Q. Song, "Very sharp adiabatic bends based on an inverse design," *Opt. Lett.*, vol. 43, no. 11, pp. 2482–2485, 2018. DOI: 10.1364/OL.43.002482.

[48] N. M. Andrade, S. Hooten, S. A. Fortuna, K. Han, E. Yablonovitch, and M. C. Wu, "Inverse design optimization for efficient coupling of an electrically injected optical antenna-LED to a single-mode waveguide," *Optics Express*, vol. 27, no. 14, pp. 19 802–19 814, Jul. 2019. DOI: 10.1364/OE.27.019802.

[49] D. Vercruysse, N. V. Sapra, L. Su, R. Trivedi, and J. Vučković, "Analytical level set fabrication constraints for inverse design," *Scientific Reports*, vol. 9, no. 1, pp. 1–7, 2019. DOI: 10.1038/s41598-019-45026-0.

[50] Y. Augenstein and C. Rockstuhl, "Inverse Design of Nanophotonic Devices with Structural Integrity," *ACS Photonics*, vol. 7, no. 8, pp. 2190–2196, 2020. DOI: 10.1021/acsphotonics.0c00699.

[51] E. Bayati, R. Pestourie, S. Colburn, Z. Lin, S. G. Johnson, and A. Majumdar, "Inverse Designed Metalenses with Extended Depth of Focus," *ACS Photonics*, vol. 7, no. 4, pp. 873–878, Apr. 2020. DOI: 10.1021/acsphotonics.9b01703.

[52] S. Hooten, T. Van Vaerenbergh, P. Sun, S. Mathai, Z. Huang, and R. G. Beausoleil, "Adjoint Optimization of Efficient CMOS-Compatible Si-SiN Vertical Grating Couplers for DWDM Applications," *Journal of Lightwave Technology*, vol. 38, no. 13, pp. 3422–3430, Jul. 2020. DOI: 10.1109/JLT.2020.2969097.

[53] W. Jin, S. Molesky, Z. Lin, K. M. C. Fu, and A. W. Rodriguez, "Inverse Design of Compact Multimode Cavity Couplers," *Optics Express*, vol. 26, no. 20, pp. 26 713–26 721, 2018. DOI: 10.1364/oe.26.026713.

[54] W. Jin, W. Li, M. Orenstein, and S. Fan, "Inverse Design of Lightweight Broadband Reflector for Relativistic Lightsail Propulsion," *ACS Photonics*, vol. 7, no. 9, pp. 2350–2355, Sep. 2020. DOI: 10.1021/acsphotonics.0c00768.

[55] A. Michaels, M. C. Wu, and E. Yablonovitch, "Hierarchical Design and Optimization of Silicon Photonics," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 2, pp. 1–12, 2020. DOI: 10.1109/JSTQE.2019.2935299.

[56] P. Sun, T. Van Vaerenbergh, M. Fiorentino, and R. Beausoleil, "Adjoint-method-inspired grating couplers for CWDM O-band applications," *Optics Express*, vol. 28, no. 3, pp. 3756–3767, Feb. 2020. DOI: 10.1364/OE.382986.

[57] Z. Lin, C. Roques-Carmes, R. Pestourie, M. Soljačić, A. Majumdar, and S. G. Johnson, "End-to-end nanophotonic inverse design for imaging and polarimetry," *Nanophotonics*, vol. 10, no. 3, pp. 1177–1187, 2021. DOI: 10.1515/nanoph-2020-0579.

[58] Z. Omair, S. M. Hooten, and E. Yablonovitch, "Broadband mirrors with >99% reflectivity for ultra-efficient thermophotovoltaic power conversion," in *Energy Harvesting and Storage: Materials, Devices, and Applications XI*, vol. 11722, International Society for Optics and Photonics, Apr. 2021, p. 1 172 208. DOI: 10.1117/12.2588738.

[59] D. Vercruysse, N. V. Sapra, K. Y. Yang, and J. Vučković, "Inverse-Designed Photonic Crystal Devices for Optical Beam Steering," *arXiv:2102.00681 [physics]*, Feb. 2021.

[60] Z. Zeng, P. K. Venuthurumilli, and X. Xu, "Inverse Design of Plasmonic Structures with FDTD," *ACS Photonics*, vol. 8, no. 5, pp. 1489–1496, May 2021. DOI: 10.1021/acsphotonics.1c00260.

[61] H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, and E. Hachem, "Single-step deep reinforcement learning for open-loop control of laminar and turbulent flows," *Physical Review Fluids*, vol. 6, no. 5, p. 053 902, May 2021. DOI: 10.1103/PhysRevFluids.6.053902.

[62] L. Faury, F. Vasile, C. Calauzènes, and O. Fercoq, "Neural Generative Models for Global Optimization with Gradients," *arXiv:1805.08594 [cs]*, May 2018.

[63] L. Faury, C. Calauzenes, O. Fercoq, and S. Krichen, "Improving Evolutionary Strategies with Generative Neural Networks," *arXiv:1901.11271 [cs]*, Jan. 2019.

[64] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution Strategies as a Scalable Alternative to Reinforcement Learning," *arXiv:1703.03864 [cs, stat]*, Sep. 2017.

[65] T. Watanabe, M. Ayata, U. Koch, Y. Fedoryshyn, and J. Leuthold, "Perpendicular Grating Coupler Based on a Blazed Antiback-Reflection Structure," *Journal of Lightwave Technology*, vol. 35, no. 21, pp. 4663–4669, 2017. DOI: 10.1109/JLT.2017.2755673.

[66] A. Michaels, *EMopt*, May 2019. [Online]. Available: https://github.com/anstmichaels/emopt.

[67] A. Michaels, *GCSlab*, Sep. 2019. [Online]. Available: https://github.com/anstmichaels/gcslab.

[68] A. Anonymous, "Inverse Design of Grating Couplers Using the Policy Gradient Method from Reinforcement Learning," *arXiv:2107.00088 [physics]*, Jul. 2021. [Online]. Available: http://arxiv.org/abs/2107.00088.

[69] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

Appendix for: Generative Neural Network Based Non-Convex Optimization Using Policy Gradients with an Application to Electromagnetic Design

In this Appendix, we discuss how generative neural networks can be used for non-convex optimization, and how the choice of deterministic-output neural networks (GLOnet) or probabilistic-output neural networks (PHORCED) for generation of design vectors affects the associated expressions for backpropagation.

Let us assume we need to optimize a function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to its n-dimensional input vector $\mathbf{p} = [p_1, p_2, ..., p_{n-1}, p_n]^T \in \mathbb{R}^n$, that is, we need to find $\mathbf{p}^* = \text{argmax}_{\mathbf{p}}\, f(\mathbf{p})$. Please note that this is an oversimplification – in EM optimization the mapping from geometrical parameters to the objective must satisfy Maxwell's Equations. Please refer to Refs. [1–4] for a more extensive discussion of the adjoint method applied to electromagnetic device optimization.

Deterministic-output (GLOnet)

Let $\mathbf{z} \sim \mathcal{D}$ be a random vector drawn from a (d-dimensional) distribution $\mathcal{D}$. This noise vector serves as an input for a neural network that deterministically generates geometrical parameter vectors ($\mathbf{p}$) for simulation, based on programmable weights denoted by $\boldsymbol{\theta}$. Specifically, let $h_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^n$ represent this NN generator such that for given $\boldsymbol{\theta}$ and sampled $\mathbf{z} \sim \mathcal{D}$ we have:

$$\mathbf{p} = h_{\boldsymbol{\theta}}(\mathbf{z}) \tag{1}$$

By virtue of $\mathbf{z}$ being a random variable, $\mathbf{p}$ is also random but drawn from an unknown distribution parameterized by $\boldsymbol{\theta}$. Consequently, the NN reward function for GLOnet training becomes:

$$\mathop{\mathbb{E}}_{\mathbf{p}=h_{\boldsymbol{\theta}}(\mathbf{z}),\, \mathbf{z}\sim\mathcal{D}}[f(\mathbf{p})] = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D}}[(f \circ h_{\boldsymbol{\theta}})(\mathbf{z})] \tag{2}$$

where $\mathbb{E}[\cdot]$ is the expected value operator and "$\circ$" denotes function composition. In the right-hand side we simplified the expression by replacing $\mathbf{p} = h_{\boldsymbol{\theta}}(\mathbf{z})$. Intuitively, our intention is to optimize the average value of the EM merit function over the distribution of devices generated by the neural network. Note that because of this exchange, the variable(s) that may be explicitly controlled by the user or optimization algorithm are the programmable weights, $\boldsymbol{\theta}$. Consequently, the optimization problem for GLOnet may be written:

$$\text{GLOnet Optimization: } \boldsymbol{\theta}^* = \mathop{\text{argmax}}_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D}}[(f \circ h_{\boldsymbol{\theta}})(\mathbf{z})] \tag{3}$$

Optimizing Eq. 3 requires the partial derivatives of the expected value with respect to the neural networks weights.

Let the probability density function of $\mathbf{z}$ be denoted $\text{pdf}(\mathbf{z})$. Then, we may write:

$$\mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D}}[(f \circ h_{\boldsymbol{\theta}})(\mathbf{z})] = \int (f \circ h_{\boldsymbol{\theta}})(\mathbf{z})\text{pdf}(\mathbf{z})d\mathbf{z} \tag{4}$$

Furthermore, let $\mathbf{p} = h_{\boldsymbol{\theta}}(\mathbf{z})$ be the output of the neural network for given input $\mathbf{z}$. Then for scalar weight $\theta_j \in \boldsymbol{\theta}$ in our neural network, the partial derivative of the reward is given by:

$$\frac{\partial}{\partial\theta_j} \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D}}[(f \circ h_{\boldsymbol{\theta}})(\mathbf{z})] = \int \frac{\partial\left[f\left(h_{\boldsymbol{\theta}}(\mathbf{z})\right)\right]}{\partial\theta_j}\text{pdf}(\mathbf{z})d\mathbf{z} \tag{5}$$

$$= \int \left.\frac{\partial f(\mathbf{p})}{\partial\mathbf{p}}\right|_{\mathbf{p}=h_{\boldsymbol{\theta}}(\mathbf{z})} \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial\theta_j}\text{pdf}(\mathbf{z})d\mathbf{z} \tag{6}$$

$$\frac{\partial}{\partial\theta_j} \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{D}}[(f \circ h_{\boldsymbol{\theta}})(\mathbf{z})] = \mathop{\mathbb{E}}_{\mathbf{z}}\left[\left.\frac{\partial f(\mathbf{p})}{\partial\mathbf{p}}\right|_{\mathbf{p}=h_{\boldsymbol{\theta}}(\mathbf{z})} \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial\theta_j}\right] \tag{7}$$

where in the second step we applied the chain rule, and in the last step we re-wrote the integral as an expected value. Note that we are using $\partial f/\partial\mathbf{p}$ as equivalent notation for $\nabla_{\mathbf{p}} f(\mathbf{p})$, and the "$\cdot$" operator is the vector dot product. This concludes the derivation. Observe that $\partial f(\mathbf{p})/\partial\mathbf{p}$ is the adjoint method gradient of the electromagnetic performance function, and therefore requires two simulations to compute per output vector $\mathbf{p} = h_{\boldsymbol{\theta}}(\mathbf{z})$ of the neural network. The additional term $\partial h_{\boldsymbol{\theta}}(\mathbf{z})/\partial\boldsymbol{\theta}$ may be evaluated using automatic differentiation.

Note that in their original GLOnet paper, Jiang and Fan [5, 6] emphasized taking the exponential value of the electromagnetic merit function, $f \to \exp\{f/\sigma\}$, where σ is a hyperparameter; for the GLOnet results in Fig. 1 we used this exponential form of the objective with hyperparameter $\sigma = 0.6$.

Probabilistic-output (PHORCED)

As mentioned in the main text, PHORCED is qualitatively similar to GLOnet in the sense that we wish to use a generative neural network to suggest geometrical degrees-of-freedom. However, by contrast to GLOnet where $\mathbf{p}$ is provided deterministically from the neural network, in PHORCED we treat $\mathbf{p}$ as a random vector sampled from a conditional probability distribution, $\pi_{\boldsymbol{\theta}}$:

$$\mathbf{p} \sim \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z}) \tag{8}$$

where $\mathbf{z} \sim \mathcal{D}$ is once again an input vector, which in this case conditions the distribution. Note that, for given $\mathbf{z}$, the distribution defined by $\pi_{\boldsymbol{\theta}}$ is not static and is in fact programmable by virtue of the neural network weights $\boldsymbol{\theta}$ that determine its statistical parameters. In the main text of this paper we chose $\pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})$ to be a multivariate Gaussian, with mean and covariance matrix defined by a generative neural network with weights $\boldsymbol{\theta}$.

Because $\mathbf{p}$ is now explicitly a random vector (not merely random just by virtue of $\mathbf{z}$ being random) the reward of PHORCED is modified relative to GLOnet (from Eq. (3)):

$$\text{PHORCED Optimization: } \boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \underset{(\mathbf{p},\mathbf{z})\sim \text{pdf}(\mathbf{p},\mathbf{z})}{\mathbb{E}}[f(\mathbf{p})] \tag{9}$$

where the subtle difference is we now wish to improve the expected value of the electromagnetic merit function under the joint probability of sampling random vectors $\mathbf{p}$ and $\mathbf{z}$ [1]. Noting that the joint probability density function of $(\mathbf{p}, \mathbf{z})$ can be written as

$$\text{pdf}(\mathbf{p}, \mathbf{z}) = \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})\text{pdf}(\mathbf{z}), \tag{10}$$

we have,

$$\underset{(\mathbf{p},\mathbf{z})}{\mathbb{E}}[f(\mathbf{p})] = \iint f(\mathbf{p})\pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})\text{pdf}(\mathbf{z})d\mathbf{p}d\mathbf{z} \tag{11}$$

Then for scalar weight $\theta_j \in \boldsymbol{\theta}$ in our neural network, the partial derivatives of this expected value are given by:

$$\frac{\partial}{\partial \theta_j} \underset{(\mathbf{p},\mathbf{z})}{\mathbb{E}}[f(\mathbf{p})] = \iint f(\mathbf{p})\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})}{\partial \theta_j}\text{pdf}(\mathbf{z})d\mathbf{p}d\mathbf{z} \tag{12}$$

where we applied the Leibniz integral rule followed by the product rule. Observe the very important fact that neither $\mathbf{p}$ nor f are explicitly related to θ_j, and therefore a term of the form $\frac{\partial f}{\partial \theta_j}$ is zero in the product rule derivative. Indeed, only the policy distribution, $\pi_{\boldsymbol{\theta}}$, is modeled by the neural network, and therefore is the only quantity subject to the derivative. Moreover, we note that by the "log trick" we may write:

$$\frac{\partial \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})}{\partial \theta_j} = \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})\frac{\partial}{\partial \theta_j}\log \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z}) \tag{13}$$

Upon substitution of Eq. 13 into Eq. (12), we obtain:

$$\frac{\partial}{\partial \theta_j} \underset{(\mathbf{p},\mathbf{z})}{\mathbb{E}}[f(\mathbf{p})] = \underset{(\mathbf{p},\mathbf{z})}{\mathbb{E}}\left[f(\mathbf{p})\frac{\partial \log \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z})}{\partial \theta_j}\right] \tag{14}$$

which concludes the derivation. The informed reader should recognize Eq. 14 as a one-step implementation of REINFORCE, with continuous action- and state-spaces represented by $\mathbf{p}$ and random vector $\mathbf{z}$ respectively. Eq. 14 should be contrasted with the Eq. 7, in that our optimization reward in the two cases was identical but the gradients required for backpropagation are drastically different. We observe that PHORCED requires no evaluation of the gradient of the electromagnetic performance function: $\partial f/\partial \mathbf{p}$. Indeed, the first term within the expectation, $f(\mathbf{p})$, requires only "forward simulation" evaluations while the latter term, $\partial \log \pi_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}$, may be computed using automatic differentiation.

Note that to produce the results in Fig. 1 of the main text, we included a "baseline subtraction" term, where we subtract the sample average of the electromagnetic performance function, $\text{average}_{\mathbf{p}}[f(\mathbf{p})]$, from the electromagnetic function each iteration of the optimization routine. This heuristic is well-known in the reinforcement learning to reduce model variance without affecting bias in expectation (Ref. [7]).

[1] In the general reinforcement learning literature, the merit function within the expected value may be a function of both $\mathbf{p}$ and $\mathbf{z}$, but this situation was excluded because it is non-applicable here.

Equivalence of methods

We now show that GLOnet optimization (Eq. 7) is equivalent to PHORCED (Eq. 14) if PHORCED is restricted to outputting Dirac delta distributions:

$$\text{GLOnet and PHORCED equivalence condition: } \pi_{\boldsymbol{\theta}}(\mathbf{p}|\mathbf{z}) = \delta(\mathbf{p} - h_{\boldsymbol{\theta}}(\mathbf{z})) \tag{15}$$

where δ is the Dirac delta distribution, and $h_{\boldsymbol{\theta}}(\mathbf{z})$ is the direct output of the neural network given $\mathbf{z} \sim \mathcal{D}$. Substituting Eq. 15 into Eq. 12, we find:

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} \mathop{\mathbb{E}}_{(\mathbf{p},\mathbf{z})}[f(\mathbf{p})] &= \iint f(\mathbf{p}) \frac{\partial \delta(\mathbf{p} - h_{\boldsymbol{\theta}}(\mathbf{z}))}{\partial \theta_j} \text{pdf}(\mathbf{z}) d\mathbf{p} d\mathbf{z} && (16)\\
&= \iint f(\mathbf{p}) \frac{\partial \delta(\mathbf{p} - h_{\boldsymbol{\theta}}(\mathbf{z}))}{\partial h_{\boldsymbol{\theta}}(\mathbf{z})} \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial \theta_j} \text{pdf}(\mathbf{z}) d\mathbf{p} d\mathbf{z} && (17)\\
&= \iint -f(\mathbf{p}) \frac{\partial \delta(\mathbf{p} - h_{\boldsymbol{\theta}}(\mathbf{z}))}{\partial \mathbf{p}} \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial \theta_j} \text{pdf}(\mathbf{z}) d\mathbf{p} d\mathbf{z} && (18)\\
&= \int \left(\int -f(\mathbf{p}) \frac{\partial \delta(\mathbf{p} - h_{\boldsymbol{\theta}}(\mathbf{z}))}{\partial \mathbf{p}} d\mathbf{p} \right) \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial \theta_j} \text{pdf}(\mathbf{z}) d\mathbf{z} && (19)\\
&= \int \left(\int \delta(\mathbf{p} - h_{\boldsymbol{\theta}}(\mathbf{z})) \frac{\partial f(\mathbf{p})}{\partial \mathbf{p}} d\mathbf{p} \right) \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial \theta_j} \text{pdf}(\mathbf{z}) d\mathbf{z} && (20)\\
&= \int \left. \frac{\partial f(\mathbf{p})}{\partial \mathbf{p}} \right|_{\mathbf{p}=h_{\boldsymbol{\theta}}(\mathbf{z})} \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial \theta_j} \text{pdf}(\mathbf{z}) d\mathbf{z} && (21)\\
\frac{\partial}{\partial \theta_j} \mathop{\mathbb{E}}_{(\mathbf{p},\mathbf{z})}[f(\mathbf{p})] &= \mathop{\mathbb{E}}_{\mathbf{z}} \left[\left. \frac{\partial f(\mathbf{p})}{\partial \mathbf{p}} \right|_{\mathbf{p}=h_{\boldsymbol{\theta}}(\mathbf{z})} \cdot \frac{\partial h_{\boldsymbol{\theta}}(\mathbf{z})}{\partial \theta_j} \right] && (22)
\end{aligned}$$

where we expanded the derivative of the delta function using the chain rule (Eq. (17)), re-expressed the delta derivative in terms of a derivative with respect to $\mathbf{p}$ (Eq. (18)), clubbed the $\mathbf{p}$ terms together (Eq. (19)), integrated by parts with respect to $\mathbf{p}$ to move from the product of f and the gradient of δ to the product of δ and the gradient of f (Eq. (20)), and evaluated the inner $\mathbf{p}$ integral using the delta function (Eq. (21)). Equation (22) recognizes that the integral in Eq. (21) is an expectation value. This concludes the proof.

References

[1] J. Lu and J. Vučković, "Objective-first design of high-efficiency, small-footprint couplers between arbitrary nanophotonic waveguide modes," *Optics Express*, vol. 20, no. 7, pp. 7221–7236, Mar. 2012. DOI: 10.1364/OE.20.007221.

[2] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch, "Adjoint shape optimization applied to electromagnetic design," *Optics Express*, vol. 21, no. 18, pp. 21 693–21 701, Sep. 2013. DOI: 10.1364/OE.21.021693.

[3] Y. Elesin, B. S. Lazarov, J. S. Jensen, and O. Sigmund, "Time domain topology optimization of 3D nanophotonic devices," *Photonics and Nanostructures - Fundamentals and Applications*, vol. 12, no. 1, pp. 23–33, Feb. 2014. DOI: 10.1016/j.photonics.2013.07.008.

[4] A. Michaels and E. Yablonovitch, "Leveraging continuous material averaging for inverse electromagnetic design," *Optics Express*, vol. 26, no. 24, pp. 31 717–31 737, Nov. 2018. DOI: 10.1364/OE.26.031717.

[5] J. Jiang and J. A. Fan, "Global Optimization of Dielectric Metasurfaces Using a Physics-Driven Neural Network," *Nano Letters*, vol. 19, no. 8, pp. 5366–5372, Aug. 2019. DOI: 10.1021/acs.nanolett.9b01857.

[6] J. Jiang and J. A. Fan, "Simulator-based training of generative neural networks for the inverse design of metasurfaces," *Nanophotonics*, vol. 9, no. 5, pp. 1059–1069, 2020. DOI: doi:10.1515/nanoph-2019-0330.

[7] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.