

This item is likely protected under Title 17 of the U.S. Copyright Law. Unless on a Creative Commons license, for uses protected by Copyright Law, contact the copyright holder or the author.

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

Designing Photodetectors with Machine Learning

Ergun Simsek, Ishraq Md Anjum, Thomas F. Carruthers, and Curtis R. Menyuk

Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD 21250, USA

Author e-mail addresses: simsek@umbc.edu, ianjum1@umbc.edu, tcarruth@umbc.edu, menyuk@umbc.edu

Abstract: Modern optimization techniques are used to design high-performance photodetectors. All the designs created during these optimization studies are then used to train a physics-inspired, two-stage neural network to obtain even better-performing devices. © 2022 The Author(s)

1. Introduction

Highly linear, high-current photodetectors (PDs) with extremely low phase noise are key components in systems that are used in many photonic and RF applications [1] to achieve high link gain, high-speed operation, low signal distortion, and a large spurious-free dynamic range. Designing a high-performance PD or even improving the performance of an already existing design is a challenging task due to the required computation time, difficulties in estimating the sensitivity of the device to the design parameters, and the existence of design constraints, e.g., the electrical breakdown potential of the materials used in the device. From number of layers to thickness of each layer, material types to doping levels, there are several variables that should be considered to design a highly efficient PD meeting the desired specifications for a given task. Today's modern numerical optimization and machine learning algorithms can be used to efficiently determine those variables [2]. In this work, we first use both traditional and modern numerical optimization (NO) techniques to design high-performance PDs operating at 1550 nm. All the designs, both the good and bad ones, are saved to be used later to train a two-stage neural network (NN) for additional improvement. The layers of these NNs and the connections among neurons are chosen to mimic the current flow and pulse propagation inside the PD.

2. Numerical Optimization

To simulate and to determine the performance metrics (phase noise, P_n , average output current, I_a , and maximum value of the impulse response, H_m) of PDs, we use numerical models that we developed that are based on the drift-diffusion equations [3]. These models include many important real-world effects and have demonstrated excellent agreement with experiments when these photodetectors operate in both continuous-wave and pulsed modes. We use the modified uni-traveling carrier PD proposed in [4] as the starting point in our optimization study. As shown in Fig. 1(a), this PD has 17 semiconducting layers. With varying material types, thicknesses, dopants, and doping densities, this PD is designed to block the motion of photogenerated holes, so that only electrons contribute to the photocurrent. Its phase noise is -178 dBc/Hz. All the design parameters and the details of our numerical model can be found in [4]. For the NO, we assume that material and doping types are fixed. Hence, there are 34 variables to be optimized: the doping density and thickness of each layer. We use four different NO techniques (Nelder-Mead simplex method [5], particle swarm optimization, genetic algorithm, and surrogate optimization) to design PDs that (i) produce lower phase noise, (ii) can handle stronger excitations, and (iii) lead to higher output currents. Eventually all the methods successfully reduce this multi-objective cost function, but there are major differences among them in terms of convergence rate, number of successful designs generated, and memory usage. The pros and cons of these algorithms will be discussed at the conference in detail. All the designs generated during the optimization studies, both high- and low-performing ones, are saved to be used as a training dataset in our NN implementation.

3. Machine Learning

Our NN implementation consists of two separate networks. The forward NN predicts the device performance metrics from the device parameters (layer thicknesses, t_i , and doping concentrations, d_i , for $i = 1, 2, 3, \dots, 17$) and the backward NN recommends a new set of parameters to obtain a new PD with the desired performance metrics. In order to achieve most efficient learning, both NNs are designed in a unique way to mimic the current flow and wave propagation occurring inside the device. For example, as depicted in Fig. 1(b), we first utilize a few NN layers to pair the parameters of each PD layer. By doing so, we tell the NN that these are not just 34 uncorrelated numbers, there are two distinct groups. Then we use another group of NN layers to inform our network about the interfaces separating PD layers. The third group of NN layers is designed to mimic the current transport among neighboring PD layers. The fourth group aims to teach the NN that there is a certain order among these 17 paired inputs. At the end, we have additional NN layers to do the regression of the device performance metrics.

The Backward NN consists of two sub-NNs and it is trained and run to predict the t_i - d_i pairs for each PD layer separately. The first sub-NN is a fully connected one and it generates some rough estimates of t_i - d_i pairs. The second sub-NN takes performance metrics (P_n , I_a , H_m) and the estimates of all the other PD layers as inputs to calculate a

better estimate of t_i - d_i for one layer only, in an iterative fashion. This procedure is repeated to obtain t_i - d_i estimates for all of the 17 layers. Then, the design recommended by the backward NN is input to the forward NN. If the predicted performance metrics are close to the initial (desired) performance metrics, then the NN-recommended design is accepted. Otherwise, a secondary refinement is applied as follows. The PD, whose performance metrics has the closest Euclidean distance to the predicted performance metrics, is determined from the training data. Similar layers of these two PDs are identified assuming those are the layers causing performance degradation. New pairs of t_i - d_i values are chosen after a grid-based search, which uses the forward NN as a device simulator. If the grid-based search does not lead to an improvement in the performance metrics, then device design with the NN is halted.

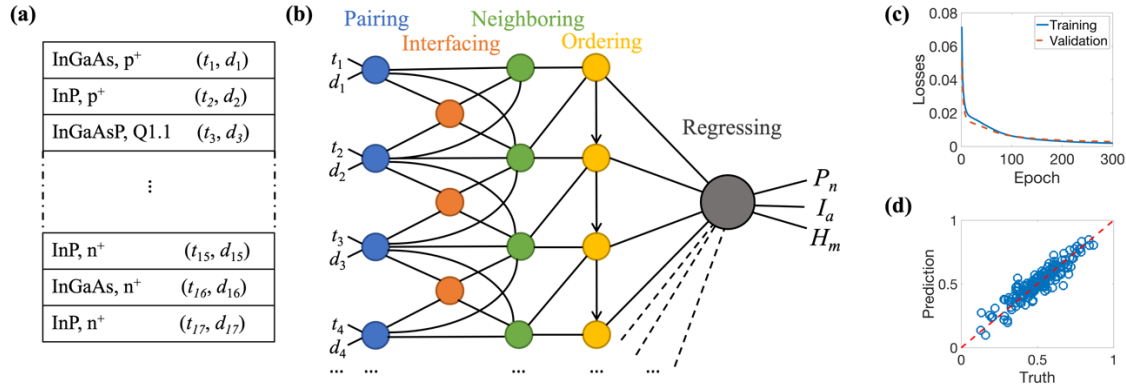


Figure 1 (a) A schematic of the MUTC photodetector with varying layer thicknesses and doping concentrations, t_i and d_i , respectively, for $i = 1, 2, 3, \dots, 17$, (b) An illustration of the neural network specifically designed to learn from device parameters to predict performance metrics (P_n , I_a , H_m), (c) Training loss (blue curve) and validation loss (red dashed curve) vs. epoch number, and (d) ground truth vs. predicted I_a values.

4. Numerical Results

The best design obtained at the end of NO has a phase noise floor of -184 dBc/Hz. It is two times faster and it outputs 50% more current than the one proposed in [4]. The forward NN, which is designed to mimic the physical mechanisms, has a higher accuracy compared to the traditional NN used in [5], i.e., 99.8% vs. 97.53%. Figure 1(c) shows how the losses decay during the training. Figure 1(d) compares the calculated vs. predicted I_a values. We use our drift-diffusion equation solver to simulate the design recommended by the backward NN and we find out that it has a phase noise of -185 dBc/Hz. The parameters of this new design are given in Table 1.

Table 1. Thickness and doping concentrations of the design recommended by the NN.

L. No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
t_i (nm)	65	29	24	19	85	48	67	201	392	39	38	34	1170	61	1058	32	296
$d_i / 10^{16}$ (cm ⁻³)	381	397	108	131	7480	1550	88.8	10.3	0.16	0.5	0.12	6.85	0.1	70.1	1560	8140	1290

5. Conclusions

With the availability of affordable processing units specifically designed for scientific computing, modern numerical optimization methods bring a remarkable efficiency in improving the performance of existing devices. Even though a relatively small improvement is achieved in this work, our results clearly demonstrate that NNs can design photodetectors that perform better than all the samples used for training.

6. References

- [1] Z. Li, H. Pan, H. Chen, A. Beling and J. C. Campbell, "High-Saturation-Current Modified Uni-Traveling-Carrier Photodiode with Cliff Layer," IEEE J. Quantum Electron. **46** (5), 626-632 (2010).
- [2] E. Simsek, S. E. J. Mahabadi, I. Md Anjum, and C. R. Menyuk, "Thinner and Faster Photodetectors Producing Lower Phase Noise," IEEE Photonics Conference 2021, Online, 18-21 October 2021.
- [3] E. Simsek, S. E. J. Mahabadi, I. Md Anjum, and C. R. Menyuk, "A Robust Drift-Diffusion Equations Solver Enabling Accurate Simulation of Photodetectors," PIERS 2021, Hangzhou, 21-25 November 2021.
- [4] S. E. J. Mahabadi, S. Wang, T. F. Carruthers, C. R. Menyuk, F. J. Quinlan, M. N. Hutchinson, J. D. McKinney, and K. J. Williams, "Calculation of the impulse response and phase noise of a high-current photodetector using the drift-diffusion equations," Opt. Express **27** (3), 3717-3730 (2019).
- [5] E. Simsek, S. E. J. Mahabadi, T. F. Carruthers, and C. R. Menyuk, "Photodetector Performance Prediction with Machine Learning," Frontiers in Optics (2021).