(Group#14) Inverse Design of a Tunable Silicon Photonic Device using Machine Learning

Li-Yuan Chiang, Ya-Chien Chang, and Wei-Ru Lin

L.-Y. Chiang and W.-R. Lin are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92122, USA (e-mail: l1chiang@eng.ucsd.edu; w8lin@ucsd.edu).

***Abstract*—Silicon photonics has undergone rapid and significant progress in a variety of applications during the past two decades, providing solutions made by mature standard semiconductor fabrication techniques and integrable to microelectronic chips. In silicon photonic device design, size reduction of photonic devices while maintaining competitive performance is a key challenge. In this work, computational nanophotonic design using gra-dient-based learning is applied to design an ultra-compact and low-loss silicon photonic switch. The resulting design is 4 μm × 4 μm with optical loss of -0.81 dB and -0.75 dB for the two switching conditions. The results show that the inverse design method is suitable for designing active devices and building up a full device library. We also propose an unbalanced staging method to further improve the optimization algorithm without using more computation resources.**

***Index Terms*—Silicon photonics, photonic switches, machine learning**

I. Introduction

SIL ICON photonics has attracted growing interest for applications in optical communications [1], [2], light detection and ranging (LiDAR) [3], and sensors [4] during the past two decades. Utilizing the mature complementary metaloxide-semiconductor (CMOS) fabrication techniques, silicon pho-tonic chips based on the silicon-on-insulator (SOI) material platform are manufactured cost-effective and integrable to microelectronics. For photonic integrated circuit (PIC) design, having compact footprints for individual devices is favorable due to its cost-efficiency in fabrication and scalability for advanced applications requiring large-scale integrations. However, due to the physical nature of light confinement and propagation in waveguides, reducing device size usually come with higher optical loss. In the conventional design method, parameter sweep is used for device optimization through simulations. Recently, inverse design of photonic devices was proposed for designing ultra-compact and high-performance devices. The degrees of freedoms (DoF) in device parameters are extended to an unprecedented level by pixelation of device patterns. Gradient-based learning algorithms provide efficient ways to simulate and optimize device performance even with a large number of parameters. Using this concept, an inversed designed silicon wavelength demultiplexer was demonstrated with a device footprint smaller than 3 × 3 μm^2 [1], which is several orders of magnitude smaller than that of other state-ofthe-art devices with the same functionality [5].

A comprehensive device library for PIC design requires various passive and active components as building blocks. Since the inverse-designed devices demonstrated to date are mostly passive devices, it is of great interest whether the revolutionary design concept is also advantageous for active devices with tunable functionalities. In this work, we design a tunable SOI switch by implementing and modifying an opensource package SPIN-B [6].

II. Related Work

Photonic switches rely on changes of material refractive indices to modulate propagating light and achieve switching on/off. In silicon photonic cases [7], the silicon refractive index can be tuned with electro-optic (EO) effect or thermo-optic (TO) effect. By applying an electric field and modulating the electronic carrier concentration in the silicon ridge waveguide, a carrier-depletion type of EO switch was demonstrated with a device length of 5000 μm [8]. Through heating of a silicon waveguide to tune the silicon refractive index, a TO switch was demonstrated with a device length of 320 μm [9]. The two reported devices are state-of-the-art in the corresponding categories. The lengths of the devices show the bottlenecks in the pathway of keep shrinking the designed device sizes. In this work, the length of the proposed switch is only 4 μm, achieving a groundbreaking level of device compactness.

III. Dataset and Features

A 2D pattern is formed by pixels with random values between $\varepsilon_{r,SiO2}$ and $\varepsilon_{r,Si}$, where $\varepsilon_{r,SiO2}$ and $\varepsilon_{r,Si}$ are the material dielectric constant for SiO_2 and Si, respectively. The refractive indices (n) are set to be 1.444 for SiO2 and 3.48 for Si. The relation between ε_r and n is written as

$$\varepsilon_r = n^2 . \quad (1)$$

The 2D pattern is duplicated with a certain number of layer dependent on the desirable device thickness to form a 3D matrix as the dataset.

Two conditions of $\Delta n_{,Si}$ are set as the features for optimization. As $\Delta n_{,Si} = 0$, the calculated forward transmission T of the upper output waveguide is monitored and maximized, whereas $\Delta n_{,Si} = 0.1$, the T of the lower output waveguide is

Y.-C. Chang is with the Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA 92122, USA (e-mail: yac021@eng.ucsd.edu).

monitored and maximized.

IV. Method

In the traditional methods, device designers come up with certain device structures based on their knowledge in the field. Then they use simulation software based on Maxwell's equation to calculate the light propagation behaviors. Optimization is conducted through iteratively fine tuning of structural parameters and simulation of device performance. Parameter sweeping is generally used for the optimization process with 5-10 structural parameters (DoF). In contrast, in the inverse design method, a target performance is set initially. The structure is modeled with pixelization and a random value stored in each pixel (neuron). A 3D matrix is then constructed to represent the structure. The performance corresponds to the matrix is simulated with a Maxwell solver to get a testing value. The optimization process minimizes the difference between the target performance and testing performance by training the fully connected neural network formed with the 3D matrix as each layer. Compared to the traditional method, the inverse design method increases the structural DoF to several orders larger. In this work, with a grid size of 0.04 μm and 4 μm of each side for the training region, the structural DoF is larger than 10000. It is impracticable to optimize a structure with such a high DoF using traditional parameter sweeping.

We use the reported wavelength demultiplexer [1] as a basic form of our tunable device design. It consists of a cascaded pattern of an input waveguide, a square slab region with complex void patterns, and two output waveguides. The light propagation behavior in a 3D structure is simulated by a Maxwell solver. The calculated T values, between 0 and 1, of the two output waveguides are extracted as the testing results.

The design problems can be written as:

$$\min_{v} \mathcal{L}(E(\varepsilon(v)))$$
$$\text{subject to } v \in S, \qquad (2)$$

where $\mathcal{L}$ is the loss function that we want to optimize, E is the electric field distribution, ε is the permittivity distribution parameterized with vector v and S is the set of fabricable devices. For the optimization of a waveguide, the goal is maximizing transmission of the two output waveguides. Intuitively, measuring the deviation of the target 1 and the current transmission is an efficient way to optimize the parameters. A loss function is defined as $loss = -(1\ T_1)^2 + -(1\ T_2)^2$, (3) where (1-$T_1$) and (1-$T_2$) are the optical loss measured from the upper and lower output waveguide at the corresponding condition, respectively. In the learning procedure, the pattern of the central square slab region is designed to be a feedforward fully connected neural network, and the parameters are optimized through gradient descent algorithm to minimize the loss function. At the end of the optimization, the trained pattern is forced to become fully binary. The blurred part is eliminated by classifying the intermediate values with the average value of ε_{r,SiO_2} and $\varepsilon_{r,Si}$.

In the model, the simulation region for the Maxwell solver is

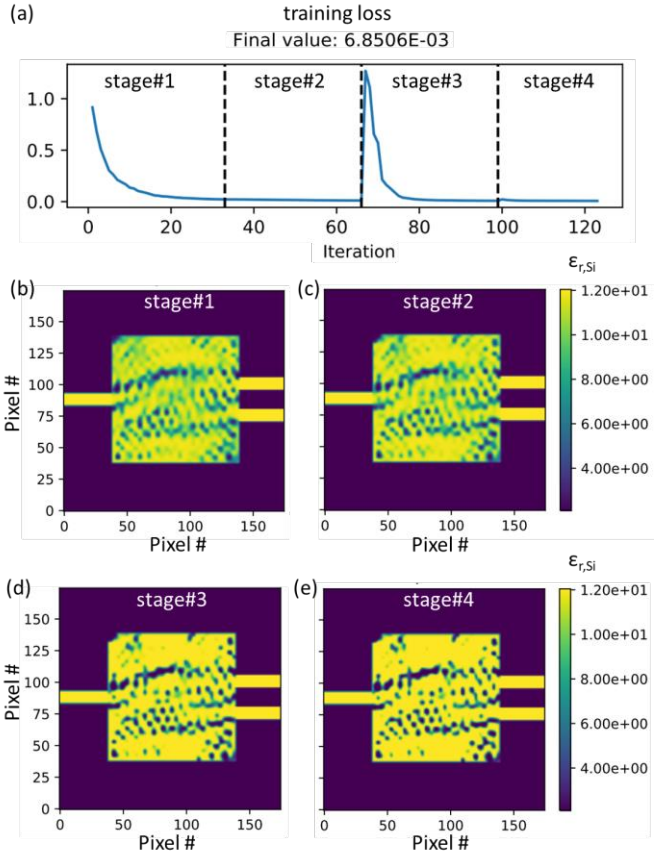

Fig. 1. The results of (a) trained loss function and (b)-(e) top-view device patterns at the end of the specified stages.

7 μm × 7 μm × 2 μm, much larger than the device region to ensure correct boundary conditions. A sample training result is shown in Fig. 1. The training loss curve is shown in Fig. 1(a). There are 120 steps and 4 stages in the training. Besides using gradient descent to optimize the 2D pattern, a sigmoid function is applied to each neuron between each stage to squash the values to either ε_{r,SiO_2} or $\varepsilon_{r,Si}$. Since only SiO_2 and Si will be used in the final device design, squashing the value of each neural to be binary helps minimize the discrepancy between testing results and validation results. The three dashed vertical lines in Fig. 1(a) indicate the timing when a sigmoid function is applied. There is a spike in loss function at the beginning of the third stage after applying the second sigmoid function. This points out that the sigmoid function applying can create an overlarge modification and degrade the testing results. While a stronger sigmoid function or more stages minimize the error between testing results and validating results, a weaker sigmoid and more steps in each stage can ensure robust loss minimization during training. Thus, sigmoid strength, number of stages, overall training steps, steps in one stage are all important

hyperparameters to be considered. Fig. 1(b)-(e) are the 2D device pattern in top view at the end of the specified stages during optimization. The pattern became apparently clearer after applying the second sigmoid function, which serve as a part of the explanation for the spike in the loss curve.

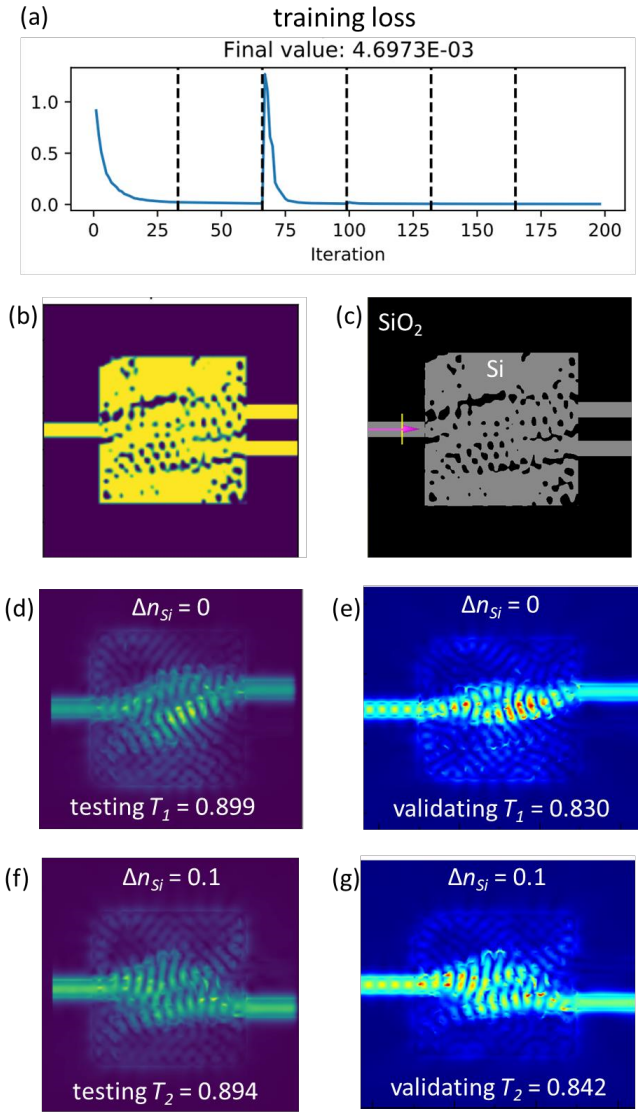

Fig. 2. The results of (a) trained loss function and (b)-(e) 2D device patterns at the end of the specified stages.

V. Results

In this section, we demonstrated that neural structure is a promising approach for inverse design. Our framework for learning and testing is based on the open-source SPINS-B. We also use a commercial software, Lumerical, to validate the trained pattern. The best results we obtained using the described 3D optimization method are shown in Fig. 2. The loss curve is shown in Fig. 2(a). There are 200 steps and 6 stages in the training, resulting in a training loss of 4.7E-3. Although the spike still emerges at the third stage, the later part of the curve is low and flat. A Geforce GTX 1660Ti GPU was used for the training. The training time was ~28 hr. The trained pattern and the fully binary pattern are shown in Fig. 2(b)-(c). Due to the reasonably high number of stages, only small blurred regions remained in Fig. 2(b). Besides the simulated results from the testing environment, we used commercial software Lumerical to validate the design with the fully binary pattern. The testing and validating light propagating behaviors for the two features

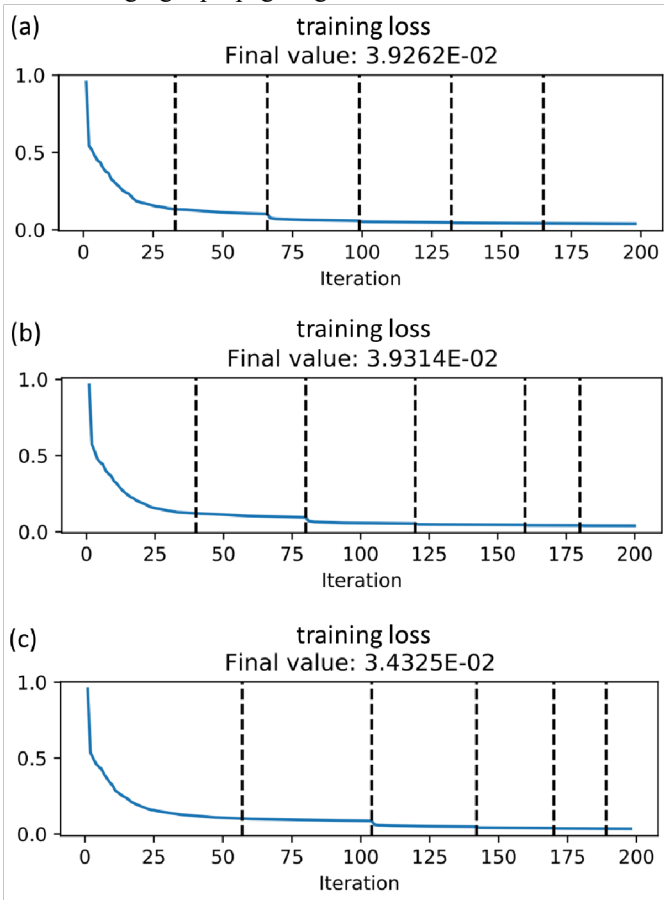

Fig. 3. The training loss curve results of (a) balanced staging, (b) unbalanced staging I, and (c) unbalanced staging II methods.

are shown in Fig. 2(d)-(g). Clear switching can be seen from both the testing and validating cases. The discrepancy between the validating and testing results can be further minimized by increasing the number of stages while maintain the number of steps in each stage, as the trained pattern can be even more discrete in that way. The resulting device design is 4 μm × 4 μm. The optical loss in the validation for both switching conditions are less than 1dB.

Since from the results we know that the hyperparameters regarding staging play an important part in the training, we tried some different staging methods on the 2D optimization to improve the tool. Three training loss curve results are shown in Fig. 3. Except for the staging method, all the other setting and parameters are the same for the three cases. A balanced staging as used in the previous results is set for the first case. The second and third cases are two different unbalanced staging methods. Firstly, we set the steps to be 40 in stage #1 to #4 and 20 in stage #5 and #6. Secondly, we set up continuous staging. From the results in Fig. 3, the first two cases have similar results, whereas

the second unbalanced staging method showed an improvement of more than 10% in the training loss result. The phenomenon can be expected since in the latter stages more neurons are saturated. The improvement by having more training steps is thus more significant in the first few stages.

During the device design, the grid size can be set smaller to allow more DoF to get more decent training results. However, it is not practical to have an infinitely small grid size due to fabrication constraints. Eventually, we want to fabricate the device for real-world applications. Each fabrication processing facility only allows certain minimum feature size in the designed pattern. Otherwise, the designed pattern will result in vanished or distorted pattern in the fabricated pattern. In order to have a more practical design, the geometry is required to be larger than a preset minimum feature size. Two requirements need to be fulfilled. Firstly, the radius of curvature must be larger than half the minimum feature size. Secondly, the smallest gap must be larger than the minimum feature size. On the other hand, if the minimum feature size constraint is too large, it hinders the optimization process. After fine tuning, a grid size of 0.04 μm and a minimum feature size of 0.08 μm were used in this work.

VI. Conclusion

We propose a compact 1 × 2 silicon photonic switch using deep neural network. The device footprint is 4 μm × 4 μm, about 2 orders of magnitude smaller than that of the state-of-the-art devices designed by traditional parameter sweeping. The switching condition is $\Delta n_{,Si} = 0.1$. The simulation results of optical loss for both conditions are -0.81 dB and -0.75 dB, respectively. An unbalanced staging algorithm is proposed to improve the optimization without using more training resources and time. Based on the results, the inverse design method with deep neural network is promising for building active device library for silicon photonics.

References

[1] A. Y. Piggott *et al.*, "Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer," *Nature Photon.*, vol. 9, pp. 374–377, May 2015. [2] D. A. B. Miller, "Optical interconnects to electronic chips," *Appl. Opt.*, vol. 49, pp. F59–F70, Jul. 2010.

[3] M. J.R. Heck, "Highly integrated optical phased arrays: photonic integrated circuits for optical beam shaping and beam steering," *Nanophotonics*, vol. 6, pp. 93-107, Jun. 2016. [4] V. S.-Y. Lin *et al.*, "A porous silicon-based optical interferometric biosensor," *Science*, vol. 278, pp. 840–843, Oct. 1997.

[5] L. Xu *et al.*, "Broadband 1310/1550 nm wavelength demultiplexer based on a multimode interference coupler with tapered internal photonic crystal for the silicon-on-insulator platform," *Opt. Lett.*, vol. 44, pp. 1770-1773, Mar. 2019. [6] L. Su *et al.*, "Nanophotonic inverse design with SPINS: Software architecture and practical considerations," *Appl. Phys. Rev.*, vol. 7, pp. 011407, Mar. 2020. [7] G. Reed, et al., "Silicon optical modulators," *Nature Photon.*, vol. 4, pp. 518–526, Jul. 2010.

[8] J. Wang, J. Zhou, L. Zhu and Q. Zhang, "Frequency- and Time-Domain Modeling and Characterization of PN Phase Shifters in All-Silicon Carrier-Depletion Modulators," *J. Light. Technol*., vol. 38, no. 16, pp. 4462-4469, Aug. 2020.

[9] M. Jacques, A. Samani, E. El-Fiky, D. Patel, Z. Xing, and D. V. Plant, "Optimization of thermo-optic phase-shifter design and mitigation of thermal crosstalk on the SOI platform," *Opt. Express*, vol. 27, pp. 10456-10471, Apr. 2019.

Critique of group 14 presentation - Inverse Design of a Tunable Silicon Photonic Device using Machine Learning

Critiques by **group 29**

This is an impressive and interesting topic.

With such limited time for the project, I am really impressed with the progress that the authors made and the level of testing they were able to perform. I'm curious to hear more about the process of designing and developing the network and test cases. What parts were original/creative/the authors own innovation, and which parts came from literature/existing work? Regardless, I think this is a great project.

I'm wondering what the inspiration for the network design was - why 7 identically structured layers? Is there precedent for such design in literature? Is Maxwell solver run for each step of training?

The mainstream of semiconductor fabrication is based on stacking of 2D layers. The 7 identical layers result in only one layer in the fabrication.

Yes, the Maxwell solver needs to run for each step to obtain a testing result during the training.

I like that consideration was given to the necessity of discretizing the pattern in order to make the fabrication of the device feasible.

Exactly, the work is aimed for real-world and fabricable device design.

Literature survey seemed a little bit lacking. Despite this, the presentation seems extremely impressive. The conclusion unfortunately was very hard to understand. This seems like pretty much a perfect application for machine learning in a physical system.

We provided the comparison of the device footprint numbers in the report. That is the selling point of this work.

Critiques by **group 1:**
Overall, we think that this project has a really strong real world application. Silicon Photonic Devices are seen around us everyday and therefore this project is very relevant in terms of physical application. After reviewing the presentation, we think that you provide a good narrative on your problem. You gave clear background reasoning on how and why your problem is significant. We also really appreciate your visualizations. You provided many good examples of your result for reference. When you explained the reasoning behind choosing specific neural network layers, we also think it made sense and you did a good job.
There were some parts of the presentation that we had trouble understanding, and we have some questions that we would like to ask. We think that answering these questions may help clarify your process and objective for your audience:

1. What is inverse design? How is this notation reflected in your project?

It is explained in detail in the first paragraph of the section IV Method in this report.

2. While we really liked your visualizations, we would like to see more explanation on them. For example on slide 10, how does each image represent progression in your training? What do the colors or the material index represent?

It is shown in Fig. 1 in this report with the corresponding explanation in the context.

3. Can you explain what values you are discretizing and how you discretize them? Are you rounding up, rounding down, binning, or other methods?

Sigmoid functions are used to squash each neuron and do the discretizing. At the end of optimization, classifying was used to form fully binary values.

4. Can you explain why discrete operation causes sharp spikes in your training and testing loss? The losses as well as the image results before and after the spike do not look like they changed much at all. Would you consider implementing an early stop before the spikes occurred?

It is explained in the last paragraph of section IV in this report.

5. Can you be more specific on how you are tuning your hyperparameters? What do you mean when you said, "used gradient descent method to optimize hyperparameters of the neural network"?

There was an error in the talk. The actual description should be using gradient descent to optimize the values in the neural network. The tuning of hyperparameters is described in this report.

6. Can you provide more details on how the Maxwell solver works?

The Maxwell solver is similar to any Maxwell solver. Based on Maxwell equation and meshing a certain structure, wave propagation can be calculated.

7. Would like to see an explanation on the significance of numbers reported. It's often difficult for people to wrap their heads around very large and very small numbers. For example, on slide 16, what do these sizes signify? How did you decide on your feature size and grid spacing?

There was a comparison in the introduction section in the slides to show the comparison of this size with other similar reported devices. The proposed device area is < 2% of other reported works.

The decision of feature size and grid spacing is based on our experience in the device fabrication constraints.

8. You mention that ML based approaches have the potential to reduce the chip size of silicon photonic devices by a huge factor. Would you briefly explain how this improvement is achieved? For example, did the researchers construct an optimization problem on the chip size with some constraints on the chip's performance, or did they use other techniques?

If every device element in the device design library for circuit designers is smaller, the same chip will be smaller.

9. On slide 7, you mention that your dataset contains input of 100*100*7. From my understanding, the first 2 dimensions (100 and 100) correspond to the width and length of the chip. What is the physical meaning of the third dimension? Are you trying to design a multi-layer chip?

The third dimension is for the thickness of the 3D design.

10. On slide 7, you mentioned that you use delta n_{si} as the feature of your input. From my understanding, we can control which output transmitter (T1 or T2) emits light by changing delta n _{si}. If that's correct, the input value (thus your dataset) should be something related to delta n_{si}. Does your 100*100*7 matrix described on slide 7 store the delta n_{si} of each part (i.e., grid) of your silicon photonic device? If yes, why do you initialize the values to be a random value between 0 and 1 while delta n_{si} is assumed to be either 0 or 0.1?

The relation between $\varepsilon_{r,Si}$ and $n_{,Si}$ is stated in the report.

The 0 and 1 means $\varepsilon_{r,SiO2}$ and $\varepsilon_{r,Si}$ for the binary notations.

11. On slide 9, you describe the loss function. From my understanding, you want T1 to be 1 when delta n_{si}=0 and T2 to be 1 when delta n_{si}=0.1. Thus, your loss function should be conditioned on the value of delta n_{si}. Why would a loss function agnostic of delta n_{si} work?

In this work the $\Delta n_{,Si}$ can be set arbitrarily. We only maximize the T_1 and T_2 at the conditions.

12. On slide 16, you describe a technique to cope with the constraint of chip's granularity in the real fabrication process. Have you applied the setup described on slide 16 before you show the results on previous slides? Or are those results acquired without considering the granularity constraint, i.e., assuming the grid spacing can be arbitrarily small?

The minimum feature size set in the work is 80 nm.

Critiques by **group 31:** Well done:
- Very specific and interesting topic.
- Seems like a great use case for machine learning if manufacturability is kept in mind. - The code seems very well documented.

Feedback:
- **Final outcome seemed a bit unclear.**

We provided the comparison of the device footprint numbers in the report. The selling point of this work is solving the challenge of designing ultra-compact device with high performance.

- **I'd like to understand more about the Neural Network structure used and what kind of tweaking or tuning was done to it. You mentioned activating each neuron using a sigmoid function. Why was that? Did you experiment with other activation functions?**

The purpose of applying sigmoid functions is described in the last paragraph of section IV in this report.

Sigmoid function is fit to this algorithm. We only need positive values in the neurons. We need to squash the values to two sides. The tuning for nonlinear function was mostly done in the strength of the sigmoid function.

- **It was a bit unclear to me the manufacturability of the design. I understand that the pattern was discretized to make it feasible but is the design something that current technology can build in an affordable manner?**

For current commercialized fabrication technology, it is easy to have feature size down below 30 nm. We set the minimum feature size to be 80 nm in this work.