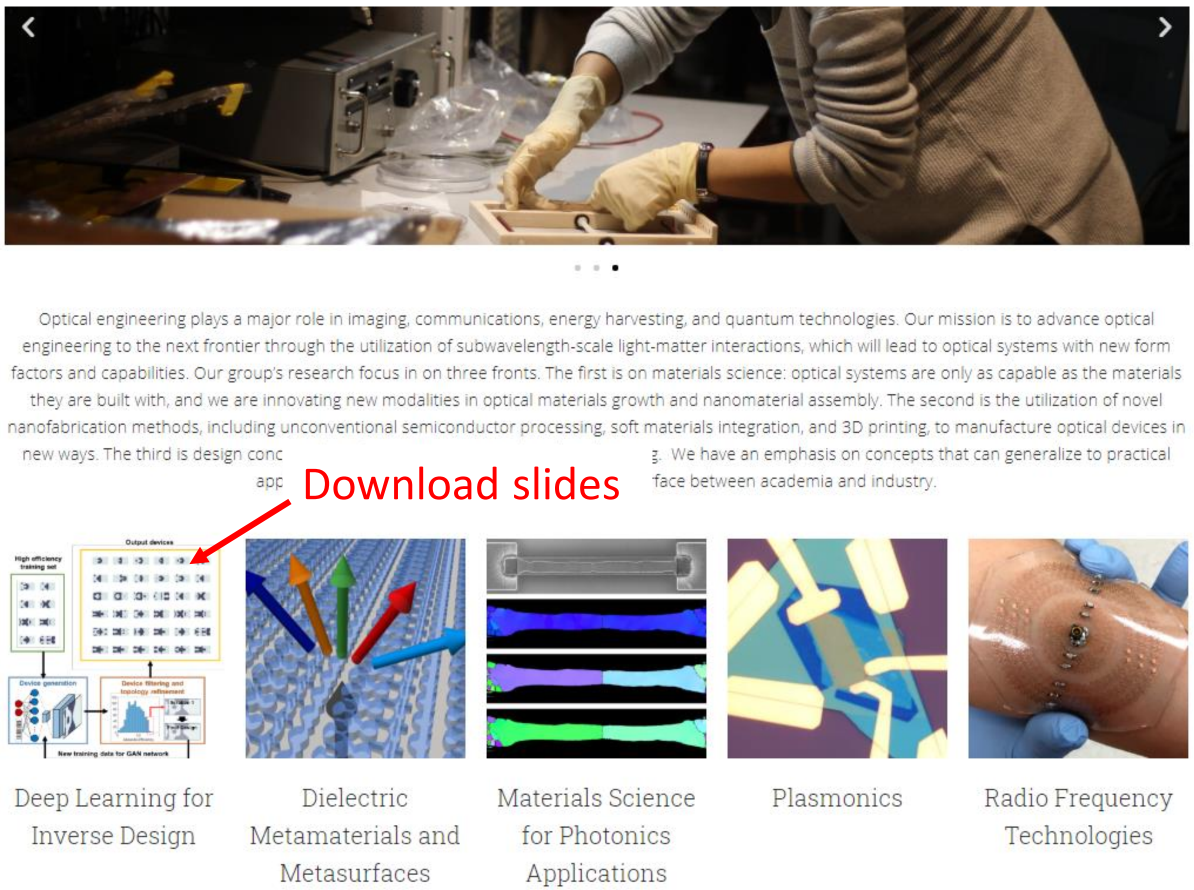
A tutorial on the simulation and design of photonic structures using deep neural networks

Jonathan Fan
Department of Electrical Engineering
Stanford University

Stanford | ENGINEERING
Electrical Engineering

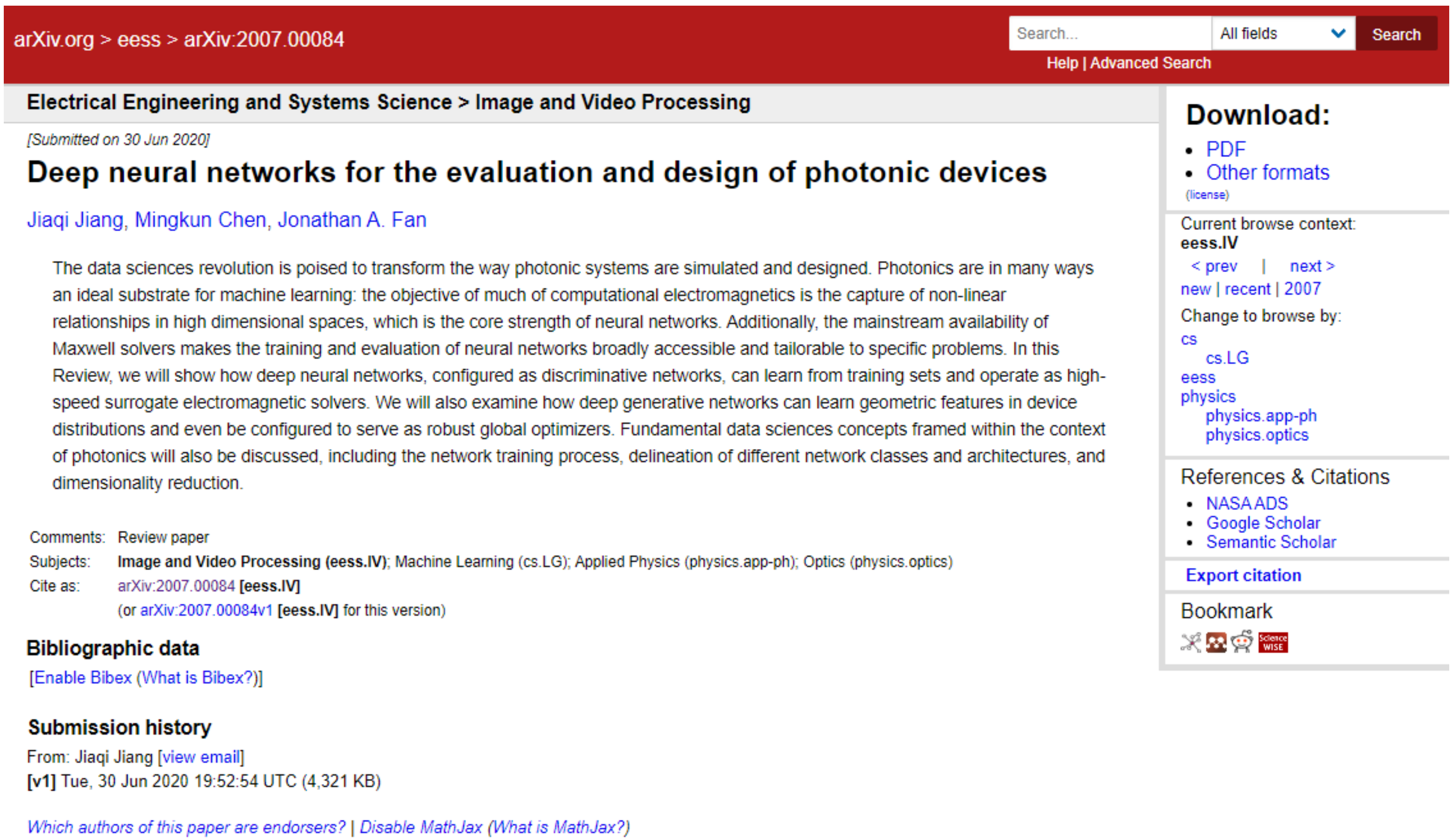
Copy of slides

Go to my website: **http://fanlab.stanford.edu**



Review paper reference

A more detailed discussion of neural networks for simulation and design is here: **arXiv:2007.00084**


arXiv.org > eess > arXiv:2007.00084

Search... | All fields | Search

Help | Advanced Search

Electrical Engineering and Systems Science > Image and Video Processing

[Submitted on 30 Jun 2020]

Deep neural networks for the evaluation and design of photonic devices

Jiaqi Jiang, Mingkun Chen, Jonathan A. Fan

The data sciences revolution is poised to transform the way photonic systems are simulated and designed. Photonics are in many ways an ideal substrate for machine learning: the objective of much of computational electromagnetics is the capture of non-linear relationships in high dimensional spaces, which is the core strength of neural networks. Additionally, the mainstream availability of Maxwell solvers makes the training and evaluation of neural networks broadly accessible and tailorable to specific problems. In this Review, we will show how deep neural networks, configured as discriminative networks, can learn from training sets and operate as high-speed surrogate electromagnetic solvers. We will also examine how deep generative networks can learn geometric features in device distributions and even be configured to serve as robust global optimizers. Fundamental data sciences concepts framed within the context of photonics will also be discussed, including the network training process, delineation of different network classes and architectures, and dimensionality reduction.

Comments: Review paper

Subjects: **Image and Video Processing (eess.IV)**; Machine Learning (cs.LG); Applied Physics (physics.app-ph); Optics (physics.optics)

Cite as: arXiv:2007.00084 **[eess.IV]**

(or arXiv:2007.00084v1 **[eess.IV]** for this version)

Bibliographic data

[Enable Bibex (What is Bibex?)]

Submission history

From: Jiaqi Jiang [view email]

[v1] Tue, 30 Jun 2020 19:52:54 UTC (4,321 KB)

Which authors of this paper are endorsers? | Disable MathJax (What is MathJax?)

Download:

- PDF
- Other formats

(license)

Current browse context:
eess.IV
< prev | next >
new | recent | 2007

Change to browse by:
cs
cs.LG
eess
physics
physics.app-ph
physics.optics

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

Export citation

Bookmark

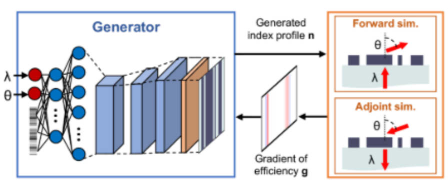
Download and install software

If you would like to participate in the live demonstrations, please go to http://metanet.stanford.edu/

Metanet Contribute Code About

Click on “Code” tab

Code and Publications



GLOnets

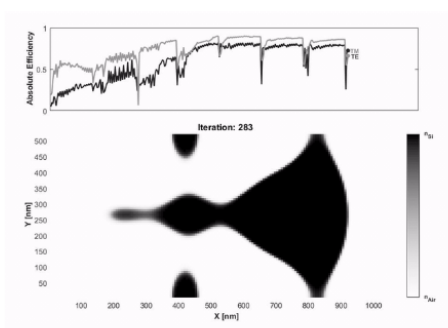
Jonathan Fan, Stanford University

GLobal Optimization NETwork or GLOnet is a global optimizer, based on a generative neural network, which can output ensembles of highly efficient topology-optimized metasurfaces.

J. Jiang and J. A. Fan

software package | arXiv | paper

Download GLOnets code



Metagrating Topology Optimization

Jonathan Fan, Stanford University

Basic topology optimization codebase for simple periodic metasurface deflectors or metagratings. This package utilizes adjoint-based gradient descent in order to generate devices with freeform geometries. Such devices are physically complex and demonstrate ultra-high efficiencies.

D. Sell, J. Yang, S. Doshay, R. Yang, and J. A. Fan

software package | Reticolo RCWA solver | paper

Outline

- Network classes and mathematical formulation
- Discriminative networks
- Generative networks
- Dataless training of networks for optimization
  - Demonstration (http://metanet.stanford.edu/)

Outline

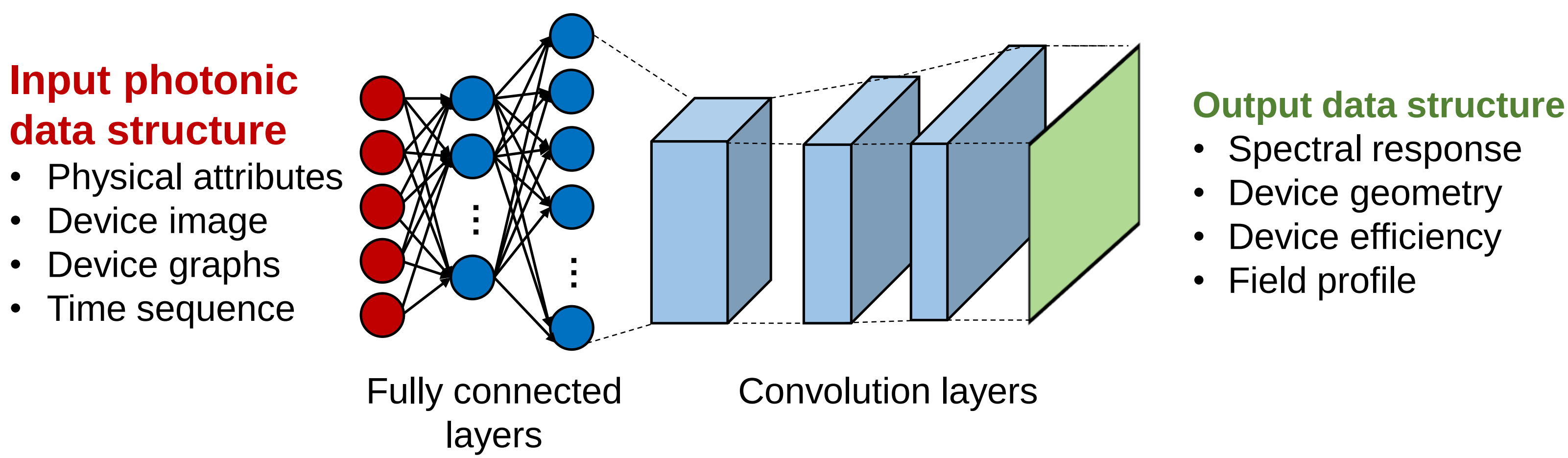
- Network classes and mathematical formulation
  - Discriminative versus generative models
  - Network building blocks and training
  - Data structures and network architectures
- Discriminative networks
- Generative networks
- Dataless training of networks for optimization

Deep neural networks

A deep neural network can model the nonlinear relationships between input and output patterns.

- Highly nontrivial relationships can be specified by performing a series of nonlinear computations.
- Accurate correlations between input and output patterns can be achieved by a training process, which uses training data to specify network weights.

Network architecture

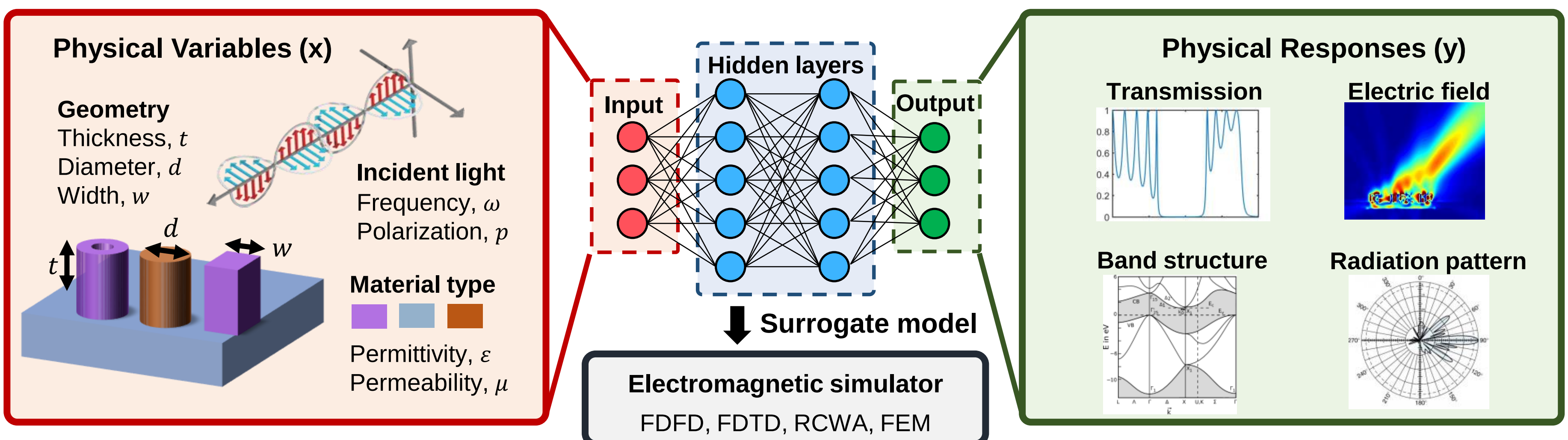


Network building blocks

Discriminative models

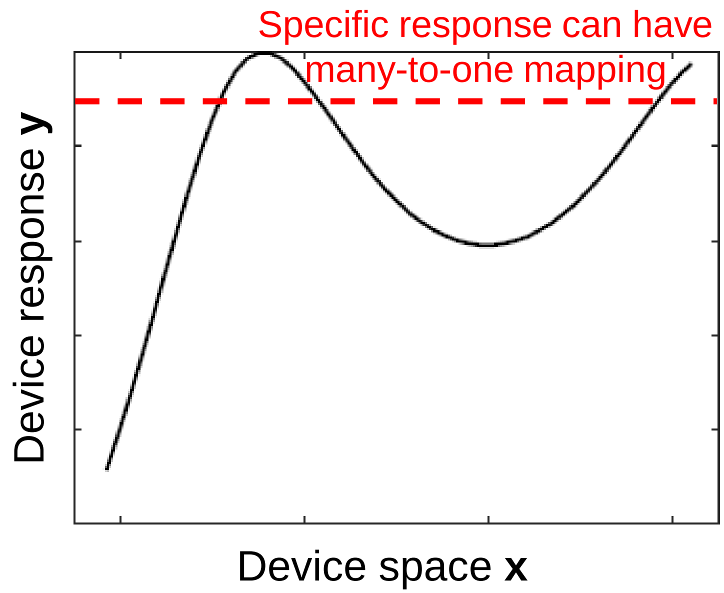
Discriminative models infer knowledge from training data to perform classification and regression tasks.

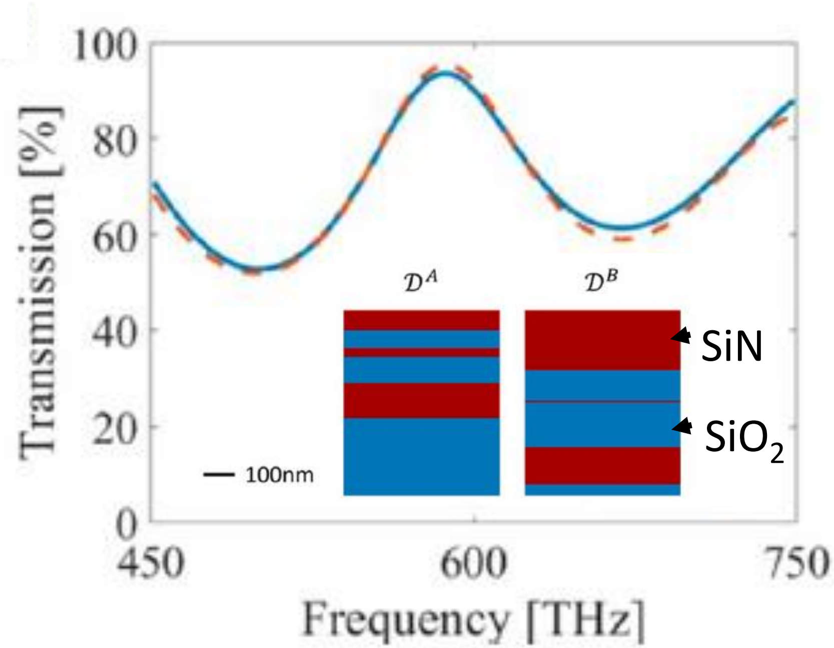
- Generally maps data as: **y** = f(**x**).
- For many optics problems, models are used for regression. **x**: physical variables; **y**: physical responses.
- Has many forms: support vector machines, naïve Bayes classifiers, neural networks, etc.



Discriminative models 2

With the functional form **y** = f(**x**), discriminative models can perform one-to-one and many-to-one mappings but not one-to-many mappings.



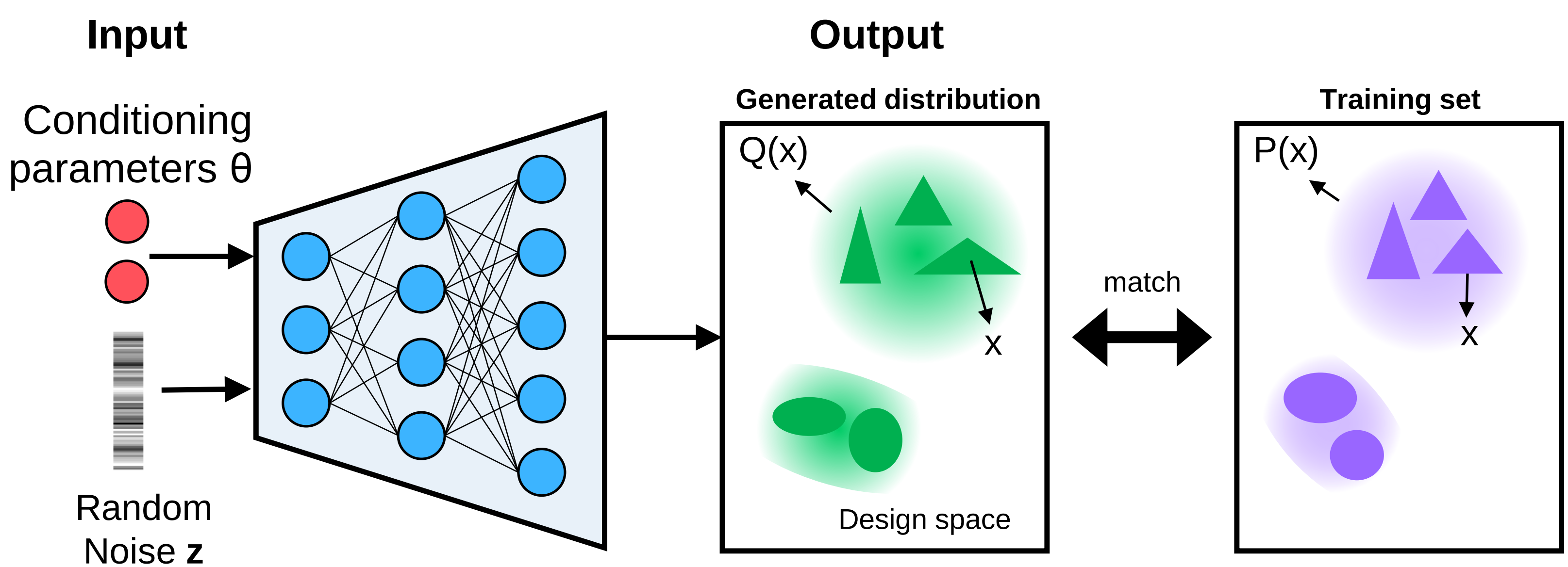


Z. Yu, *ACS Phot 5*, 1365 (2018)

Generative models

Generative models have latent space (random noise) inputs that can be sampled to produce a distribution of outputs.

- Can perform one-to-many mappings.
- Can be conditioned with device labels including physical variables and physical responses.



Generative models 2

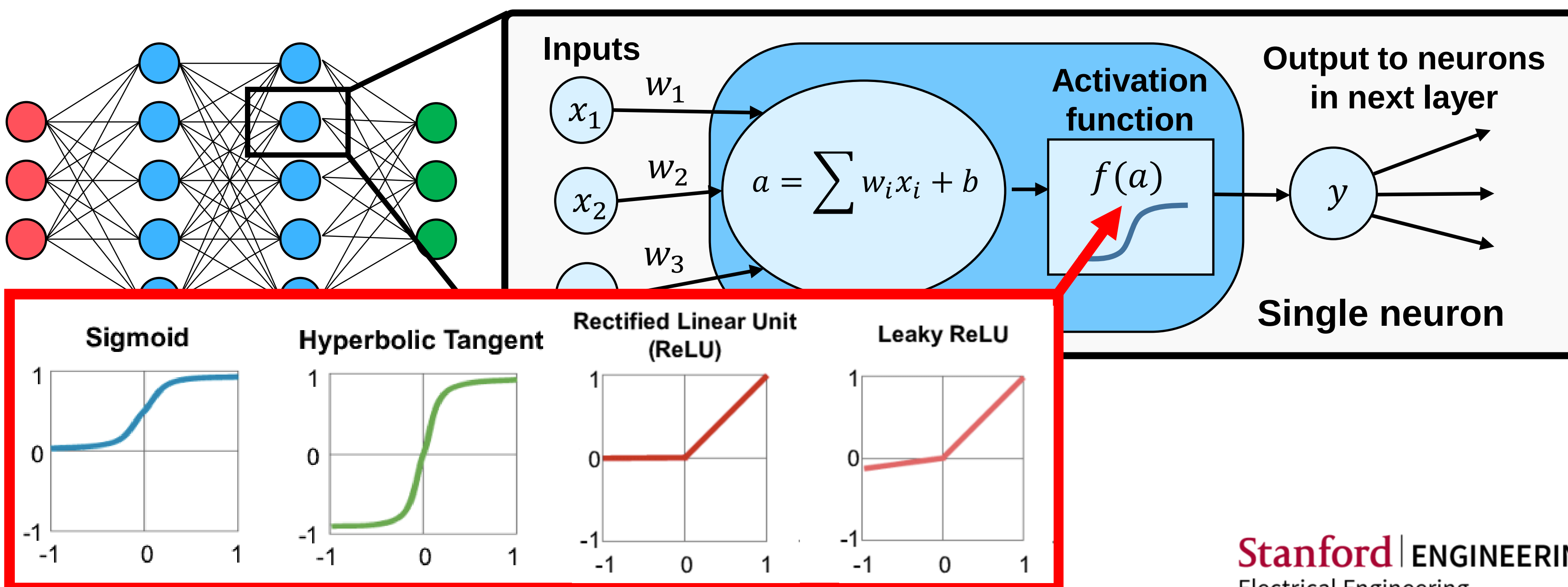
Generative models can produce a wide range of data structures that mimic the training data distribution.

Karras, ICLR 2018

Network building blocks: neurons

The basic building block for many deep network layers is the neuron.

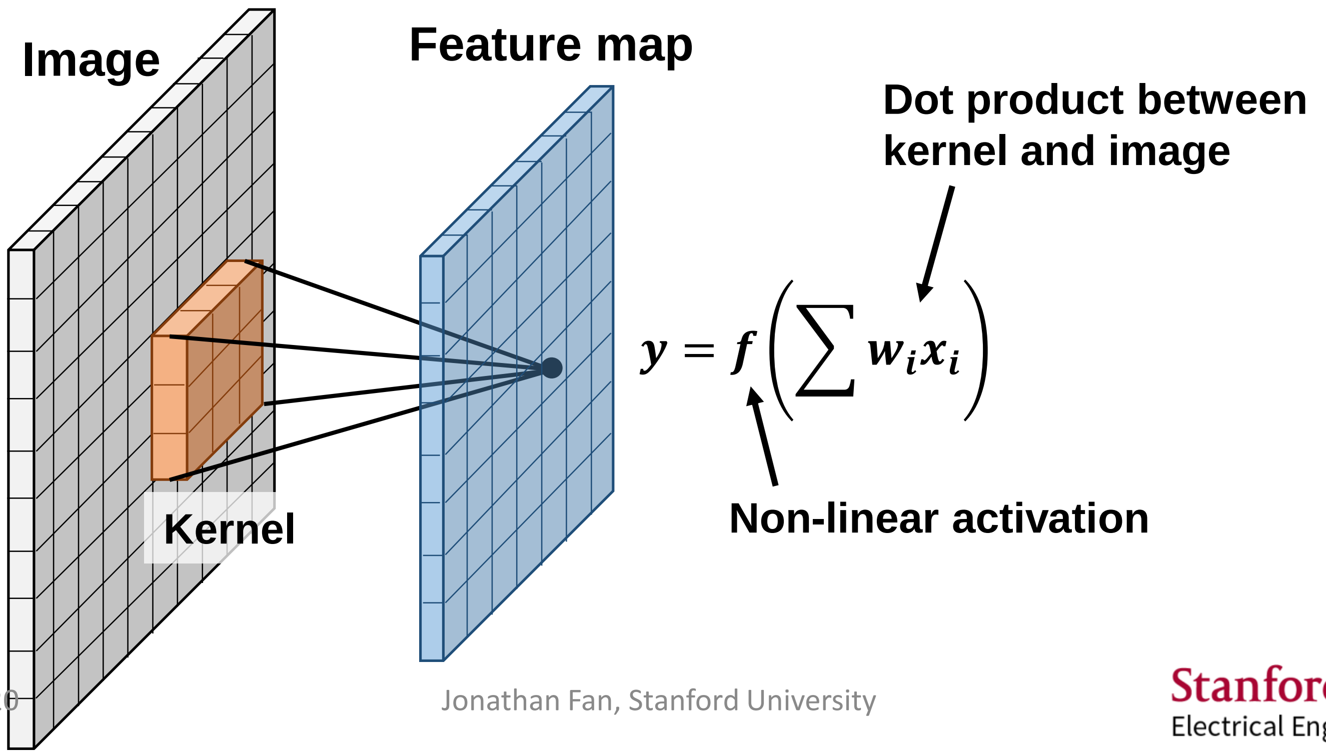
- Input values from prior layer are each multiplied by weight values w_i, added together, and then processed by a nonlinear activation function.
- The weights are determined from network training.



Building blocks: convolutional layers

In a convolutional layer, a kernel is convolved over all spatial locations of an image to produce a feature map.

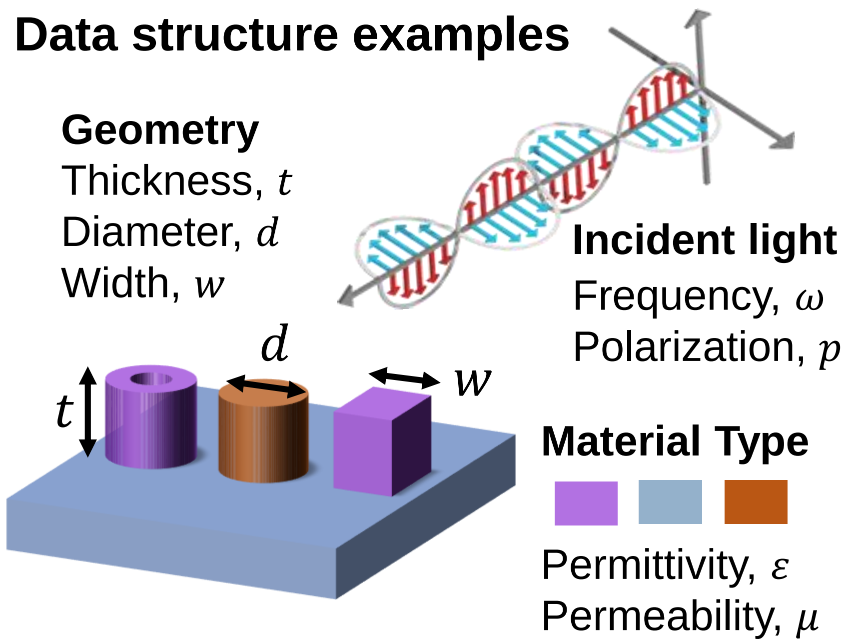
- The weights in the kernel are trainable.
- The use of the convolution operation leads to local, translationally invariant data processing.

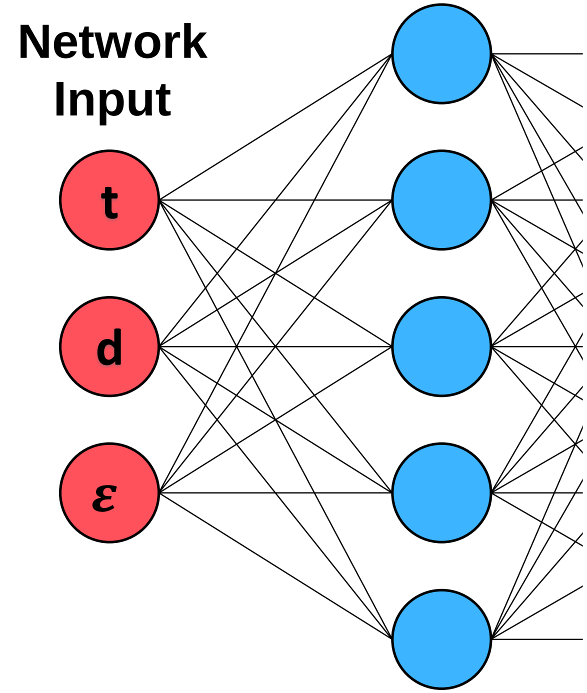


Data structure: discrete values

Any set of discrete physical variables can be inputted into the network as discrete values.

- Typically normalized and inputted into a layer of fully connected neurons.



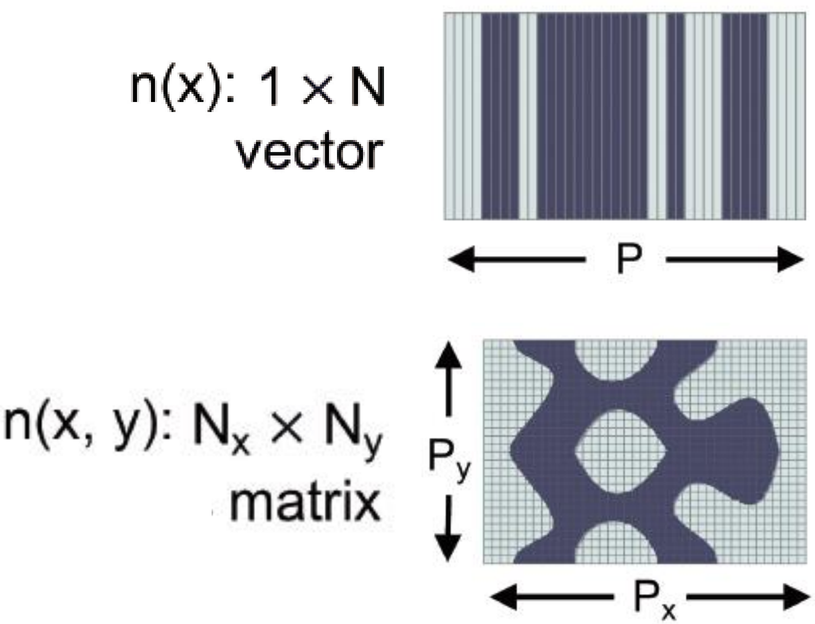


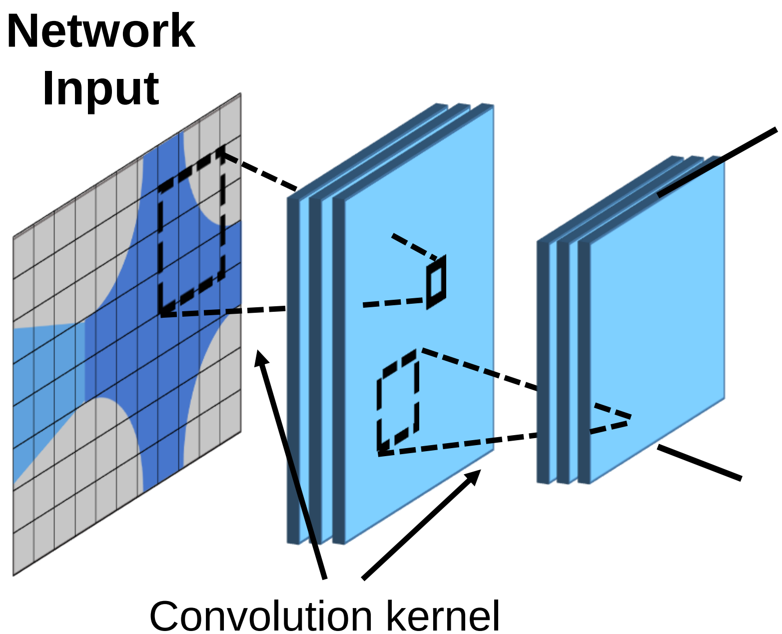
Data structure: images

Freeform photonic devices that cannot be described by a few discrete values can be processed as images.

- Images can be 1D (vectors), 2D (matrices), or 3D (tensors).
- Typically normalized and inputted into a convolutional layer.

Data structure examples

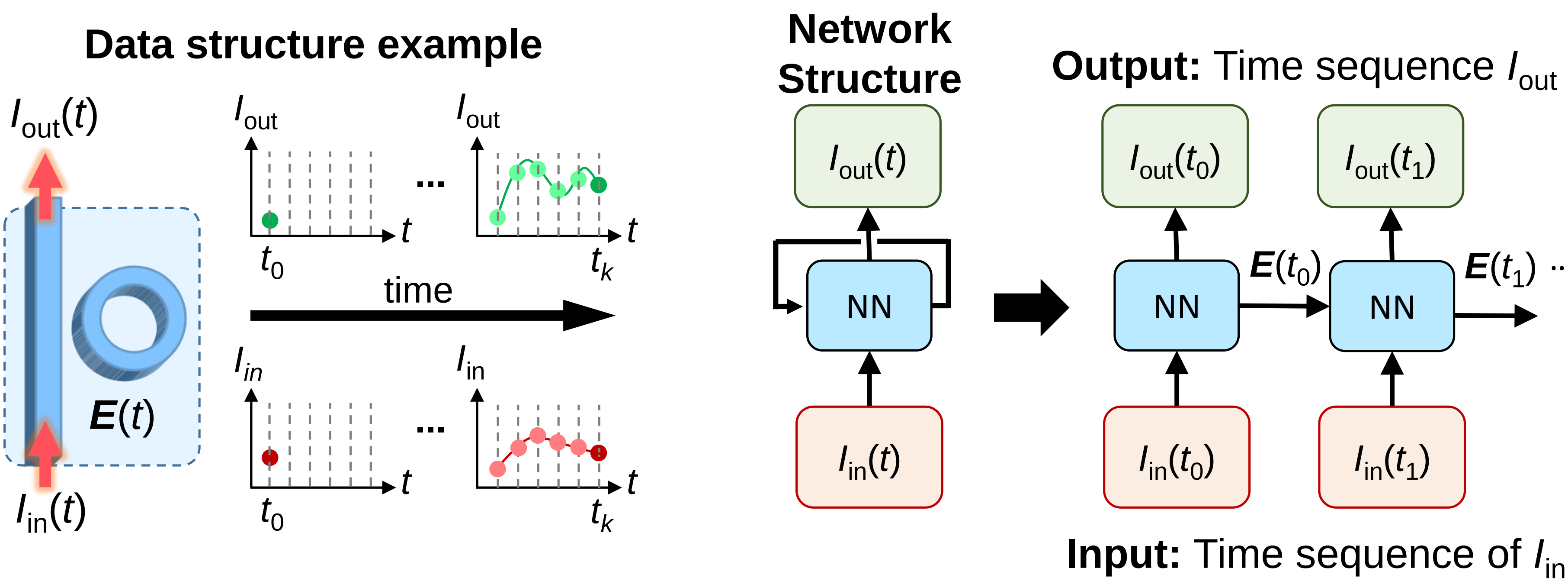




Data structure: time sequences

Time-dependent electromagnetic phenomena can be captured using recurrent neural networks (RNNs).

- The network uses feedback to capture system history.
- The network can be configured for discrete value, image, and graph data structures.



Training process

- The goal of training is to minimize the **loss function**, which represents error between the training set response and the network response.
- The network training process is as follows:
  - Create a training set and subdivide it into training, validation, and test datasets.
  - Initialize neural network with random weights, set hyperparameters, and perform network training until loss function asymptotically plateaus.
  - Use validation dataset to test the performance of the network, tweak hyperparameters, and repeat training.
  - Once the network is set, use the test dataset to determine final network performance.

Discriminative networks: loss functions

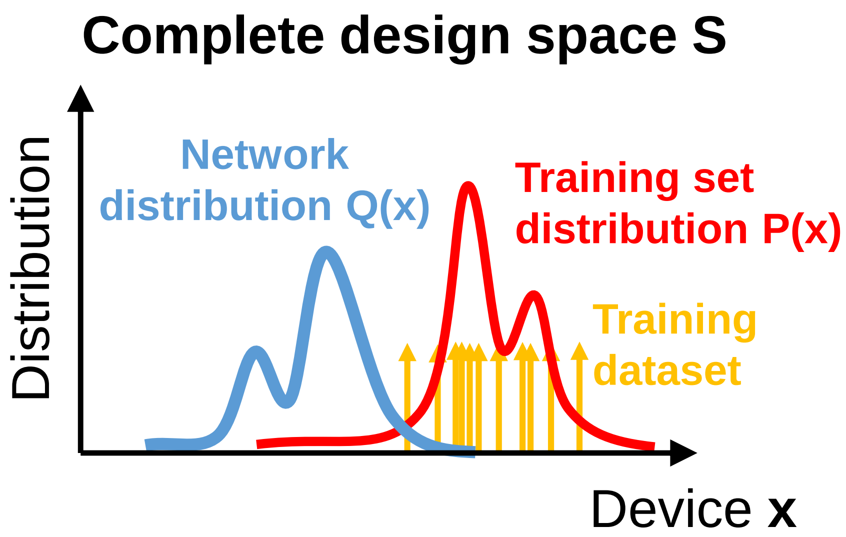
- Training data comprises input/output pairs $\left(\hat{x}^{(n)}, \hat{y}^{(n)}\right)$.
- For an input $\hat{x}^{(n)}$, the network output is $y^{(n)}$. We want $y^{(n)}$ to be as close to $\hat{y}^{(n)}$ as possible.
- The most popular loss function is least mean squares:

$$L(y, \hat{y}) = \frac{1}{2N} \sum_{n=1}^{N} \left(y^{(n)} - \hat{y}^{(n)}\right)^2$$

The loss function here is calculated for batch size N for a given training epoch.

Generative networks: loss functions

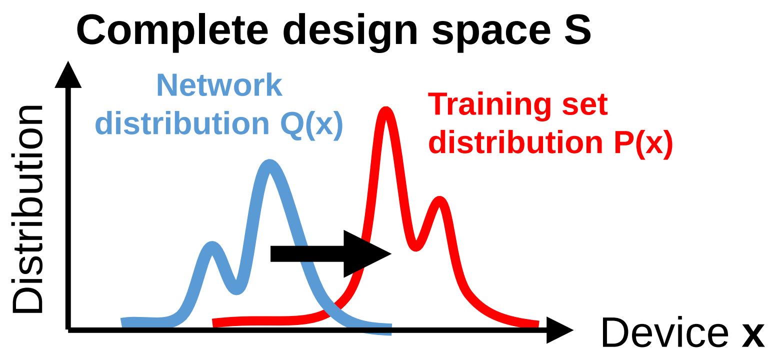
- The training set and generated device distributions should be treated as probability distributions.
  - The training set devices can be regarded as samples from the probability distribution P(x).



  - The distribution of devices produced by the generative network can be treated as the probability distribution Q(x).

Generative networks: loss functions 2

- We want the training process to get Q(x) to match with P(x). Two types of loss functions are typically used:



- **KL Divergence** quantifies how different one probability distribution is from another:

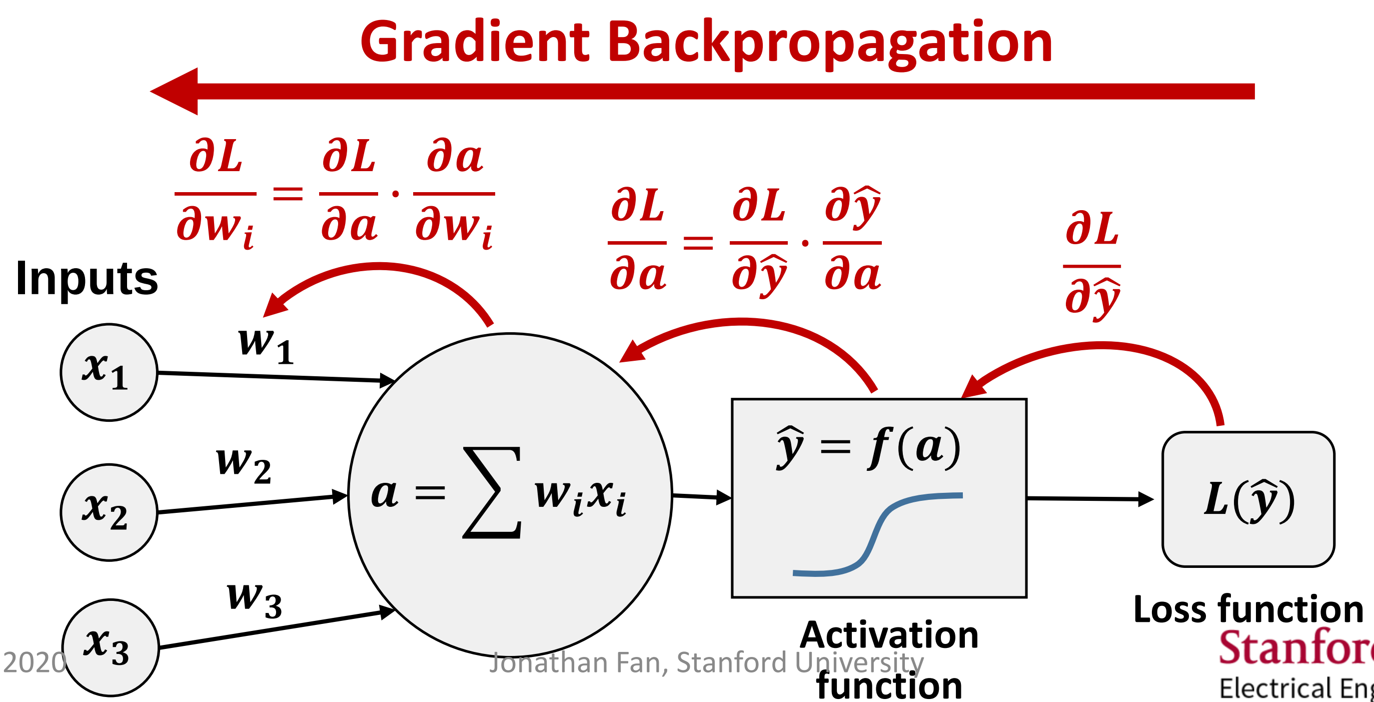
$$D_{KL}(P||Q) = \int P(\boldsymbol{x}) \log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} d\boldsymbol{x}$$

- **Cross entropy** describes how many bits of information are required when a coding scheme optimized for Q(x) is applied to P(x):

$$H(P,Q) = -\int P(\boldsymbol{x}) \log Q(\boldsymbol{x})\, d\boldsymbol{x}$$

Backpropagation

- To perturb the network weights in a manner that reduces the loss function, we perform backpropagation.
  - Backpropagation is based on the chain rule.
  - Backpropagation is used to calculate gradients.
- **Example:** a network consisting of just a single neuron:



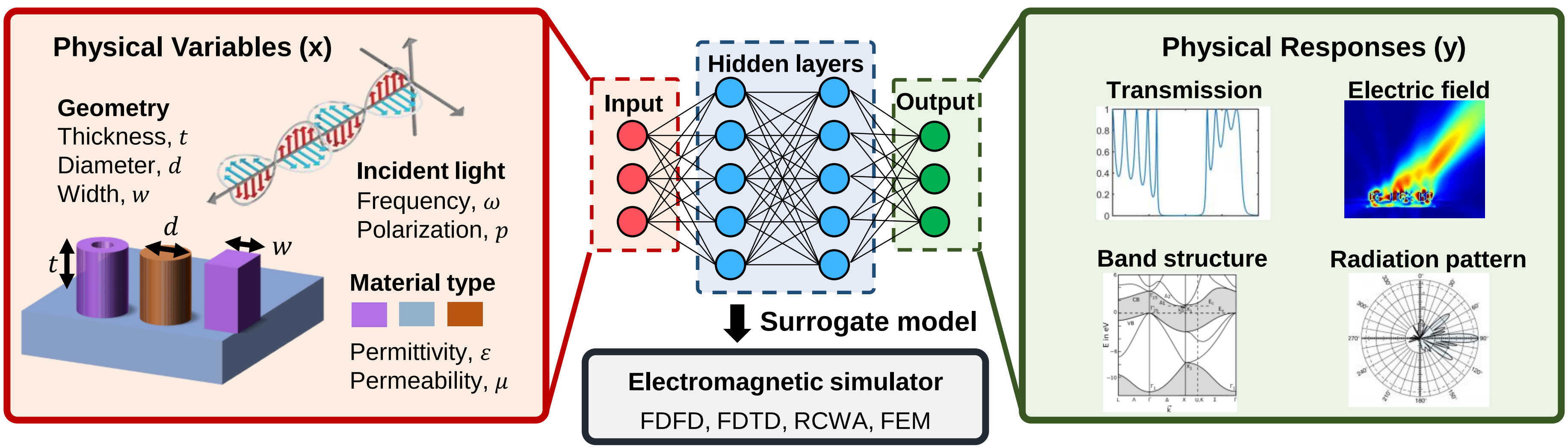
Outline

- Network classes and mathematical formulation
- Discriminative networks
    - Discriminative networks as surrogate solvers
    - Discriminative networks for inverse design
    - The curse of dimensionality
- Generative networks
- Dataless training of networks for optimization

Surrogate solvers

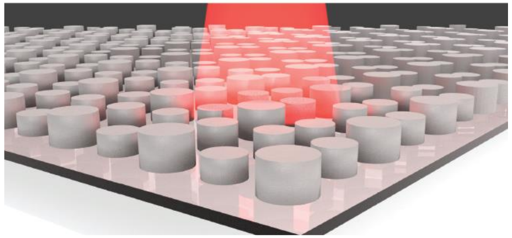
A popular application of discriminative networks is to create surrogate EM solvers that replace standard solvers.

- There is a substantial one time computation cost for creating training data and training the network.
- The trained network can perform inference with orders of magnitude faster times than a standard solver.

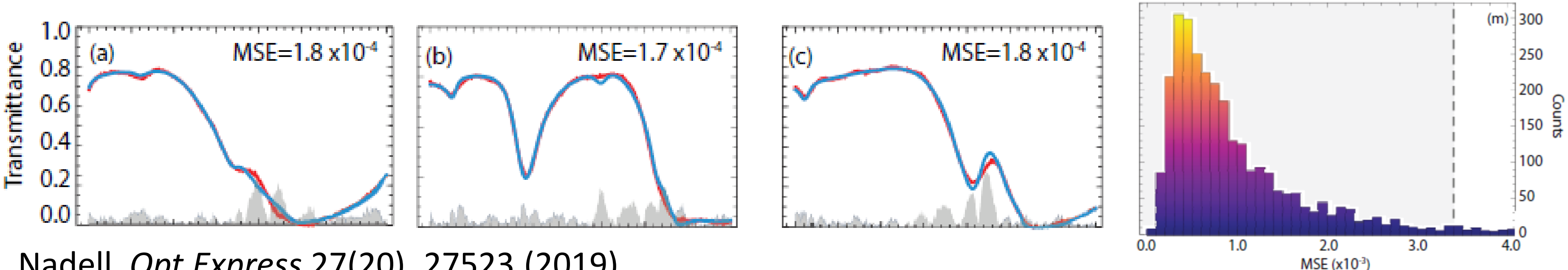


How fast?

- Network is trained (18,000 training data) to predict transmission spectra of a metasurface comprising silicon cylinders.



- Trained network, evaluated using a Tesla Quadro M6000 GPU, can compute 9,400 spectra/second.
  - Takes 23 hours to produce 815 million spectra.
  - 8.2×10^5 faster than simulations in CST Microwave Studio.
- 95% of devices have a mean squared error less than 3.4×10^{-3}.

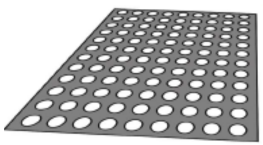


Nadell, *Opt Express* 27(20), 27523 (2019)

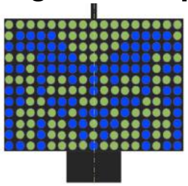
Guided wave devices

Guided photonic structures in the form of fibers, photonic crystals, and other photonic/plasmonic on-chip components have been modeled over the last decade.

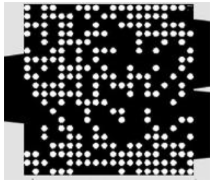
Photonic crystal



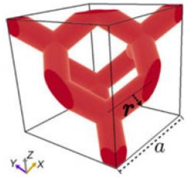
Waveguide coupler



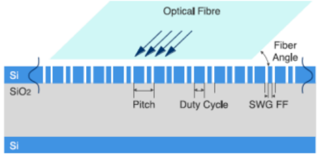
Waveguide splitters



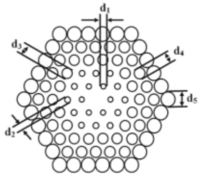
3D photonic crystal



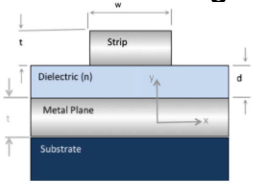
2D Grating coupler



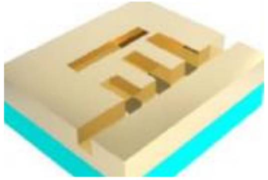
Photonic crystal fiber



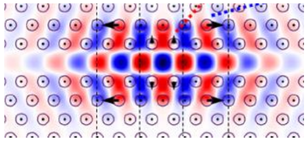
Plasmonic waveguide



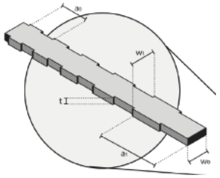
Plasmonic filter



Photonic crystal cavity



Si Bragg grating

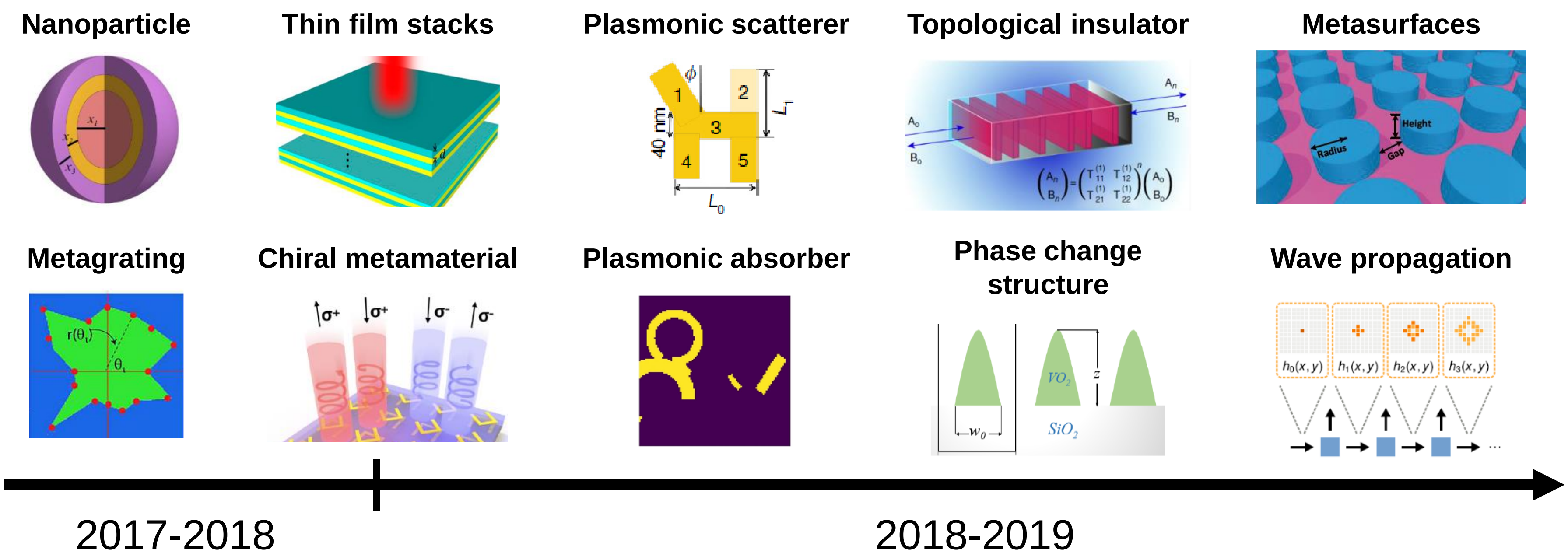


2012-2016

2018-present

Free space nanophotonic devices

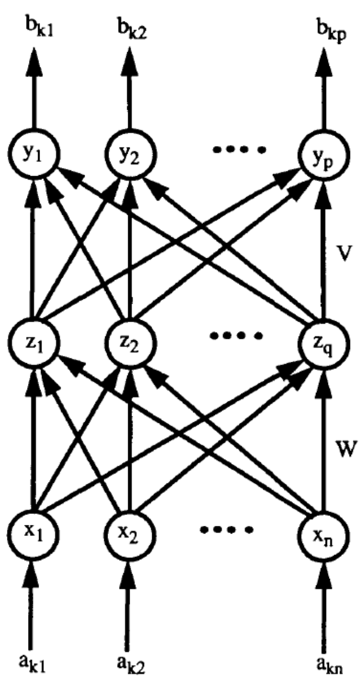
Neural network models of nanophotonic scatterers, metasurfaces, and absorbers have only recently been researched.



Early deep networks

A fully connected deep network was used to model nonlinear electronic components, such as MESFETS.

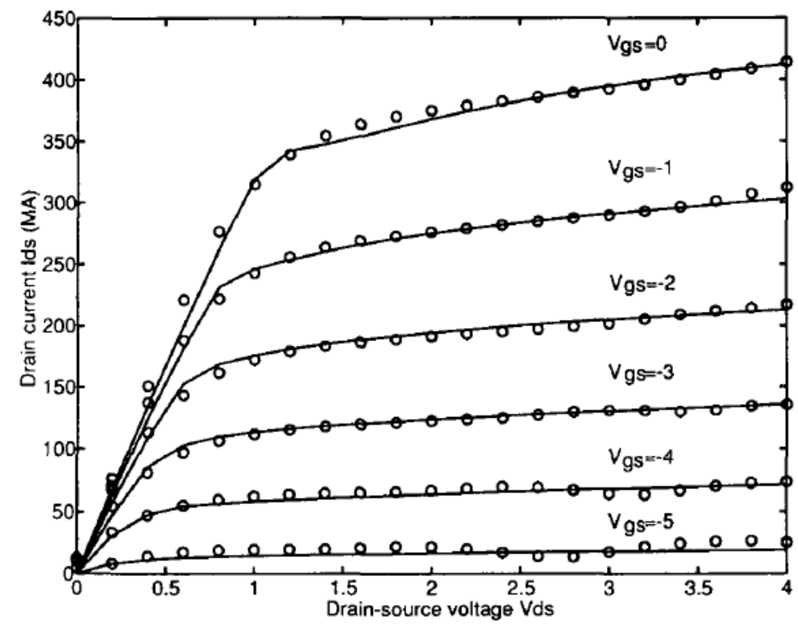
- Network is trained with 1000 training data.
- Is integrated with commercial CAD optimization software.



Output: currents and charges at the gate, drain, and source

Input: gate length/width, channel thickness, doping density, gate-source voltage, gate-drain voltage

Dots: network **Lines:** physical model



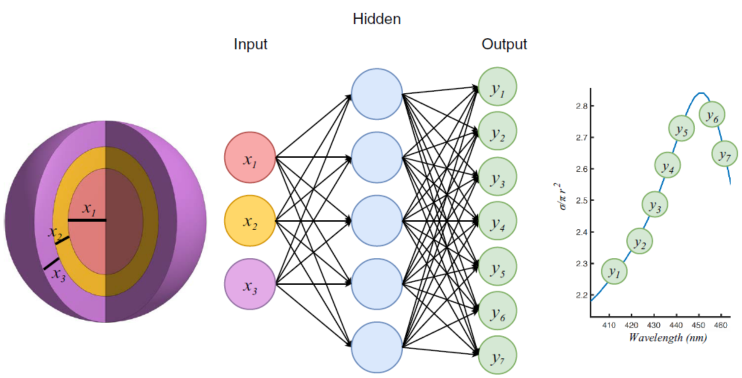
Zaabab, *IEEE Trans Microwave Theory and Techniques* 43(6), 1349 (1995)

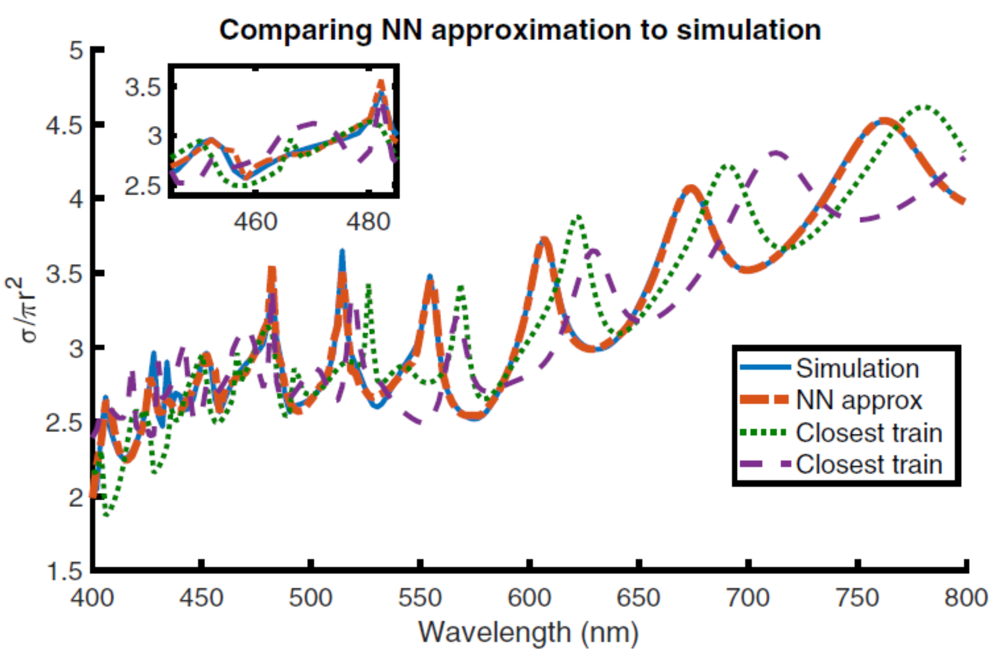
Nanoparticle scatterers

A fully connected neural network could predict the scattering spectra of concentric nanoshell scatterers.

- System comprises 8 shells of alternating dielectric material.
- 50,000 training data points were used.

Network architecture



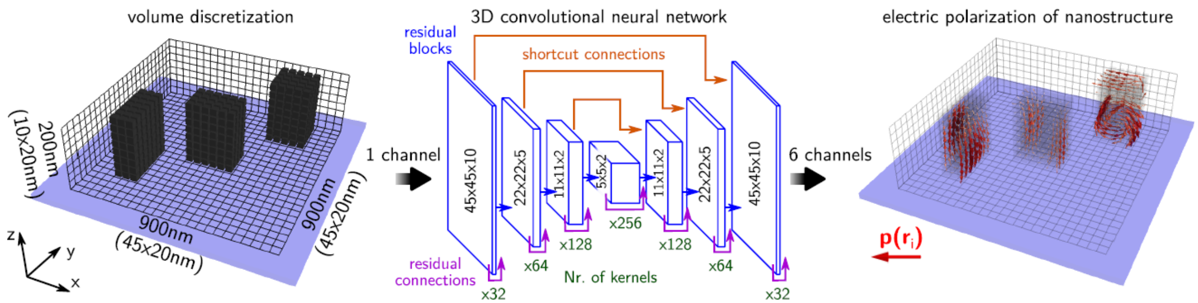


Peurifoy, *Sci Adv* **4**, eaar4206 (2018)

Stanford ENGINEERING
Electrical Engineering

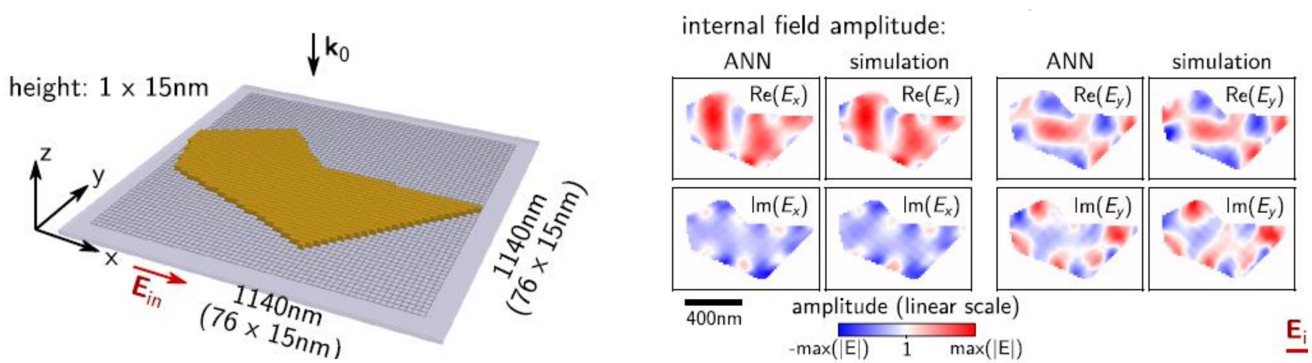
Electric polarization

A convolutional network can be trained to predict the electric polarization distribution within a set of illuminated nanostructures.



- The network architecture is a U-Net.
- 30,000 training data points were used.

Example: metallic scatterer



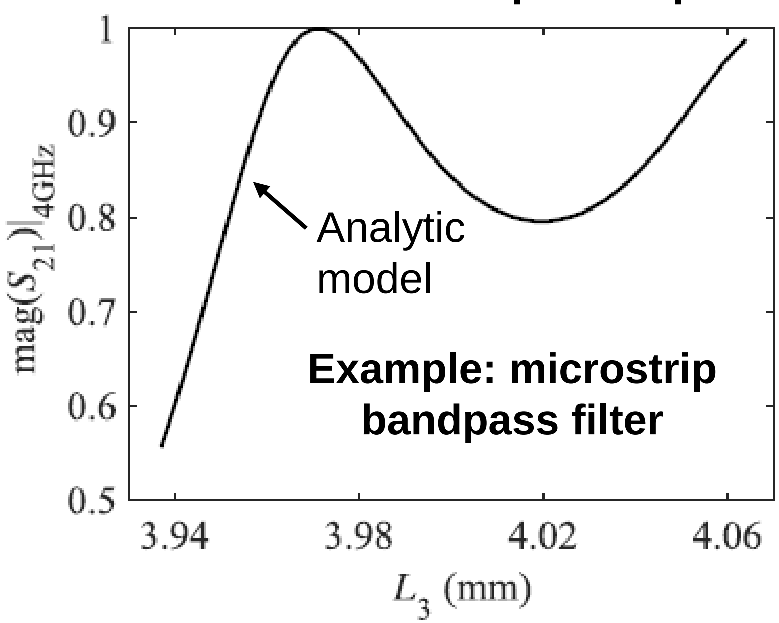
Wiecha, *Nano Lett* 20, 329 (2020)

Inverse design with discriminative models

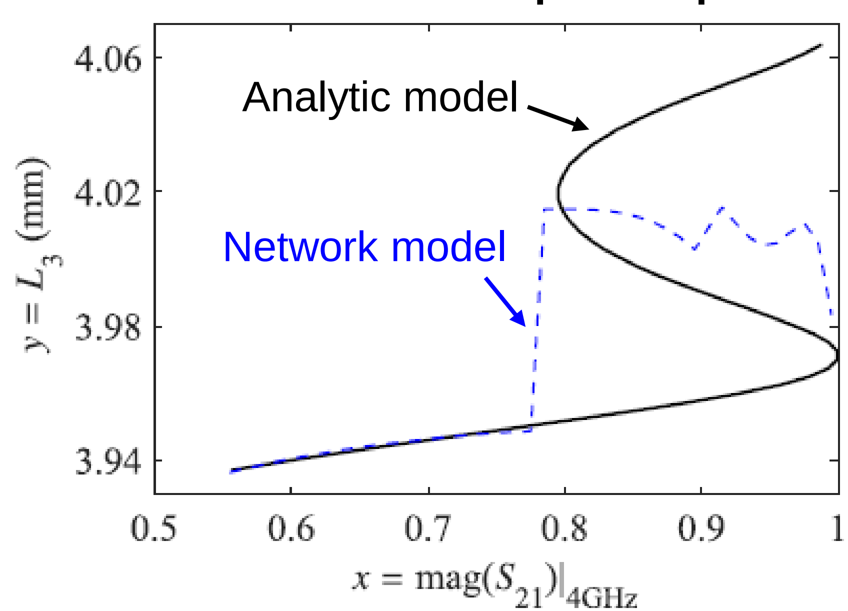
Discriminative models are suitable for solving forward problems but cannot be directly trained to solve inverse problems.

- Most inverse problems in electromagnetics involve one-to many mappings which destabilize network training.

Forward model response space



Inverse model response space

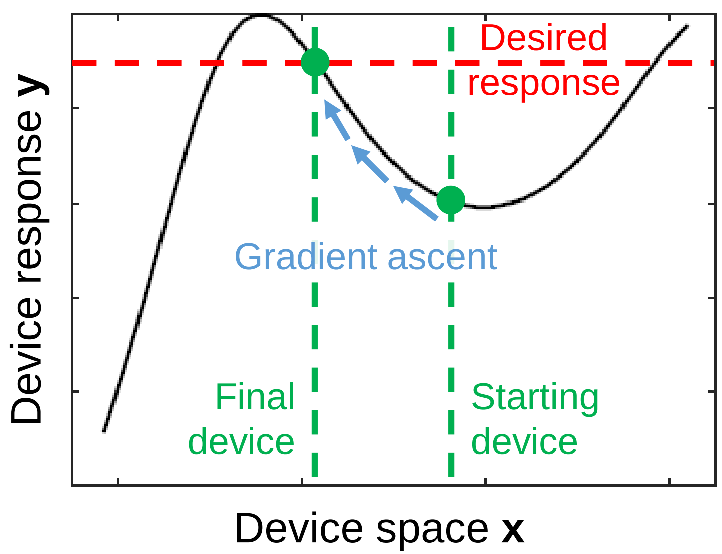


Zhang, *IEEE Trans Microwave Theory and Tech* **66(8)**, 3781 (2018)

Inverse design: backpropagation

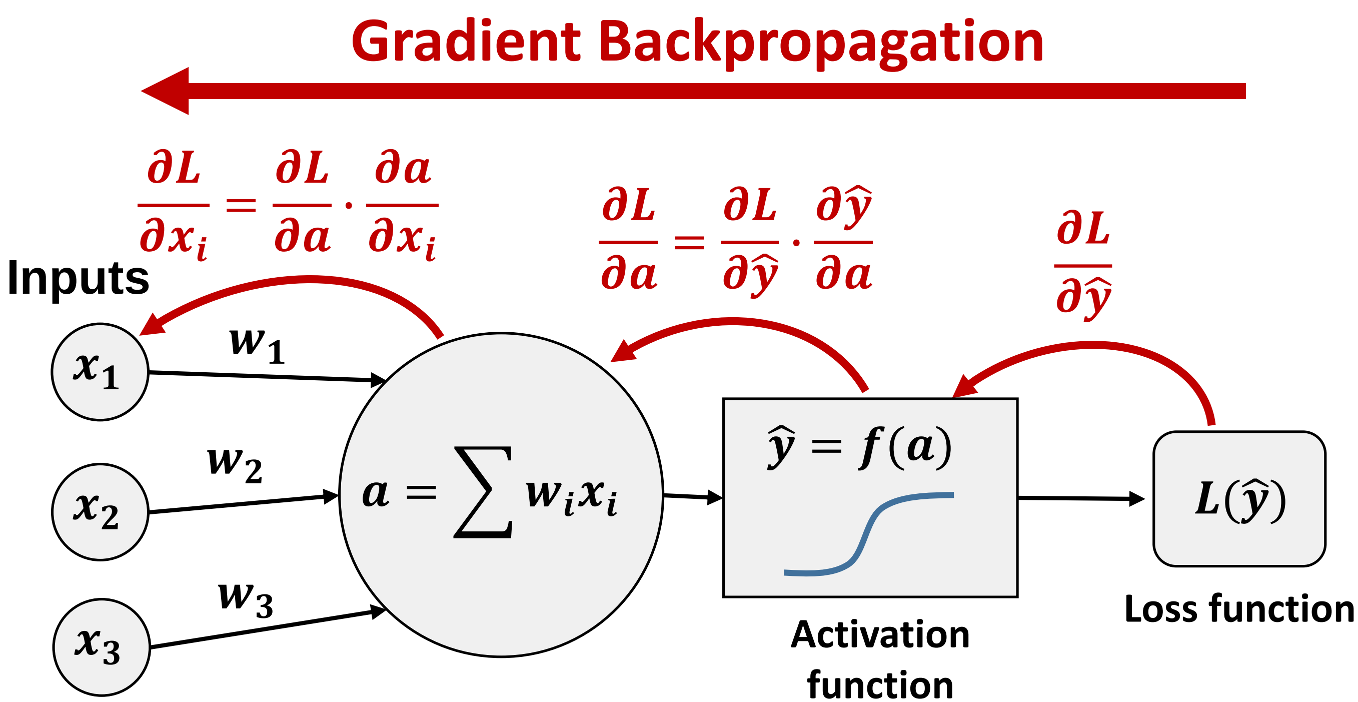
We can use backpropagation to perform gradient ascent within the design space.

- The loss function is defined as the difference between the outputted and desired optical response.
- The input device metrics are iteratively adjusted to reduce the loss function.



Inverse design: backpropagation 2

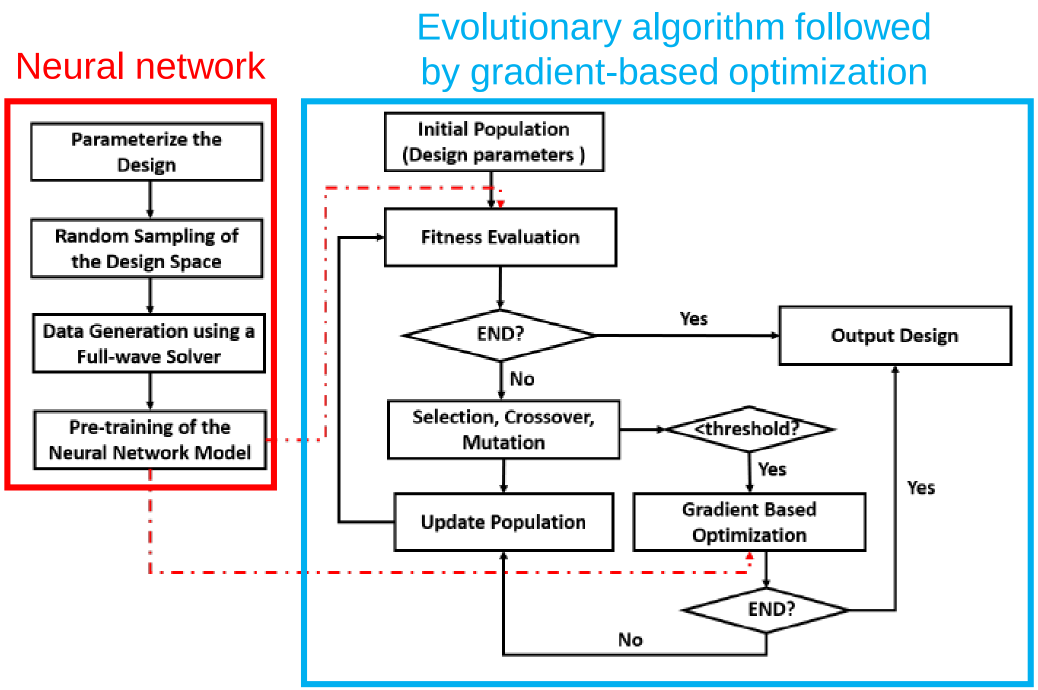
The backpropagation method is consistent with that for adjusting the weights of the network, except that the weights are kept constant and perturbations are applied to the input values.



Inverse design: conventional optimizer

Surrogate network solvers can be used together with conventional optimizers.

- Optimization methods include Newton's methods, interior-point algorithms, evolutionary algorithms, trust-region methods, and particle swarm optimization.
- **Example:** patch antenna design.

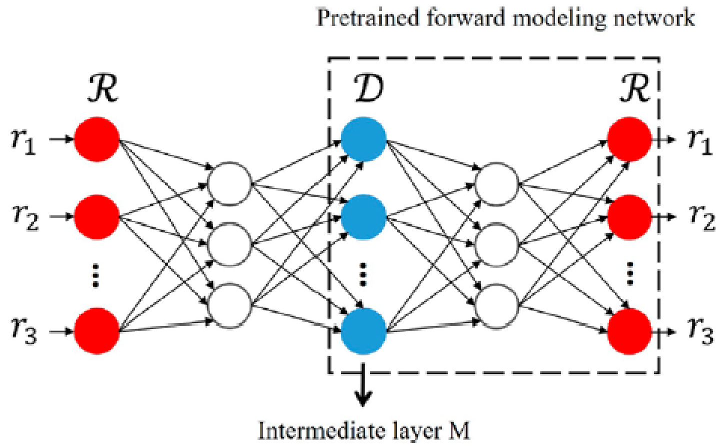


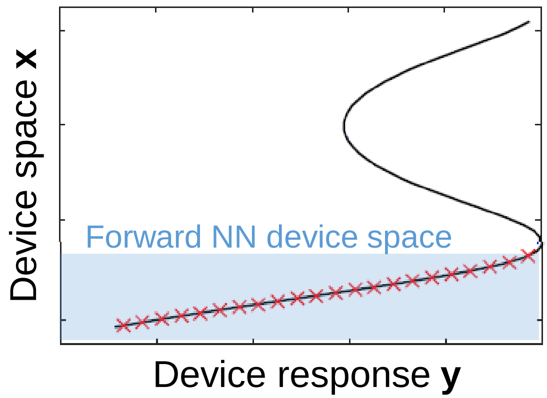
Liu, *IEEE EPEPS* 261 (2018)

Inverse design: tandem network

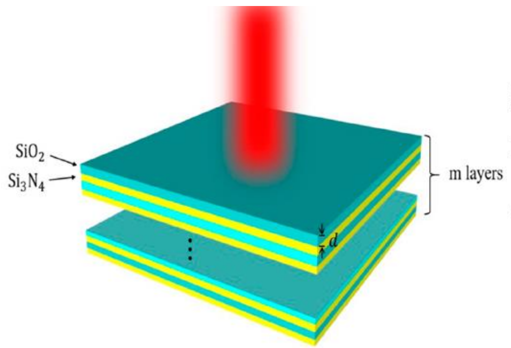
To reduce the one-to-many mappings problem, a forward network is first trained. It is then fixed and combined with an inverse network that is subsequently trained.

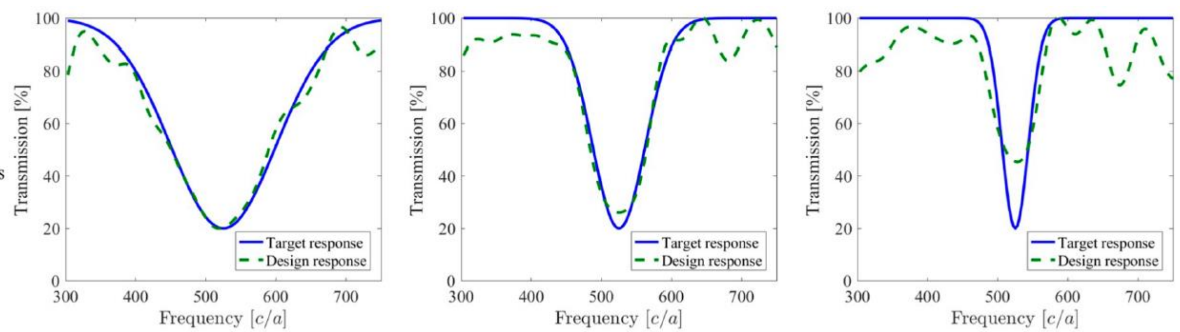
Neural network model





Example: thin film stacks

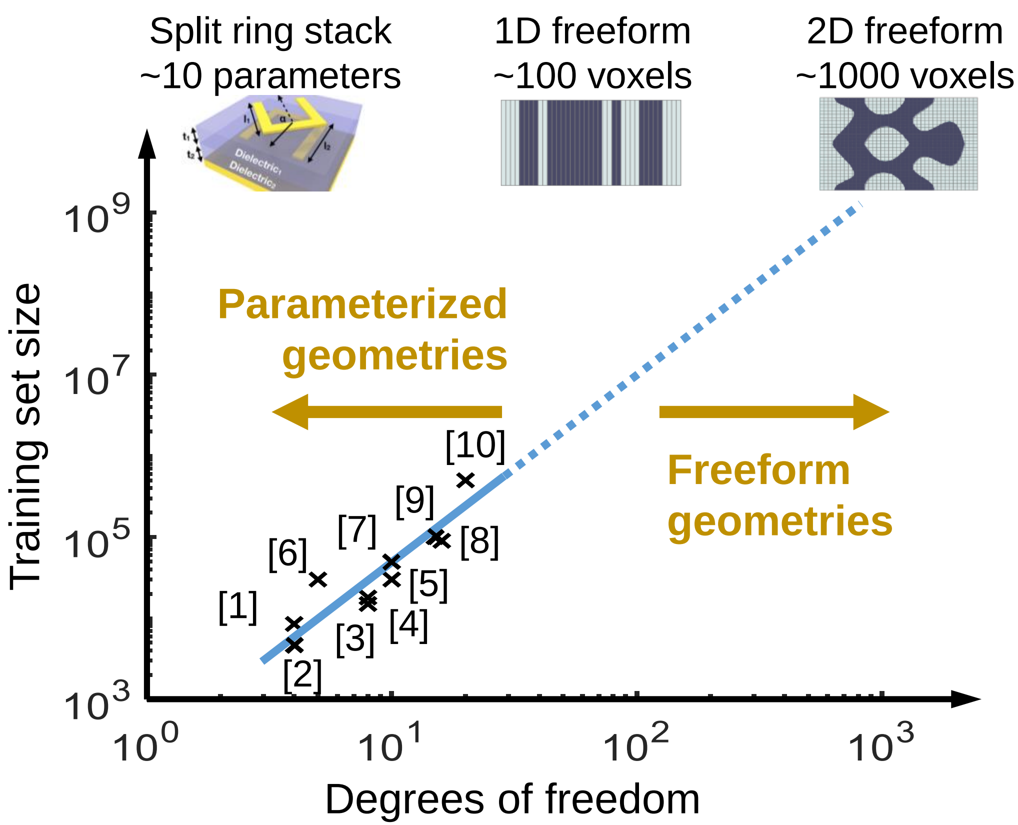




Z. Yu, *ACS Phot 5*, 1365 (2018)

Dimensionality reduction

A fundamental problem issue with scaling neural networks is termed the **curse of dimensionality**: the amount of required training data increases exponentially as the system degrees of freedom increase.



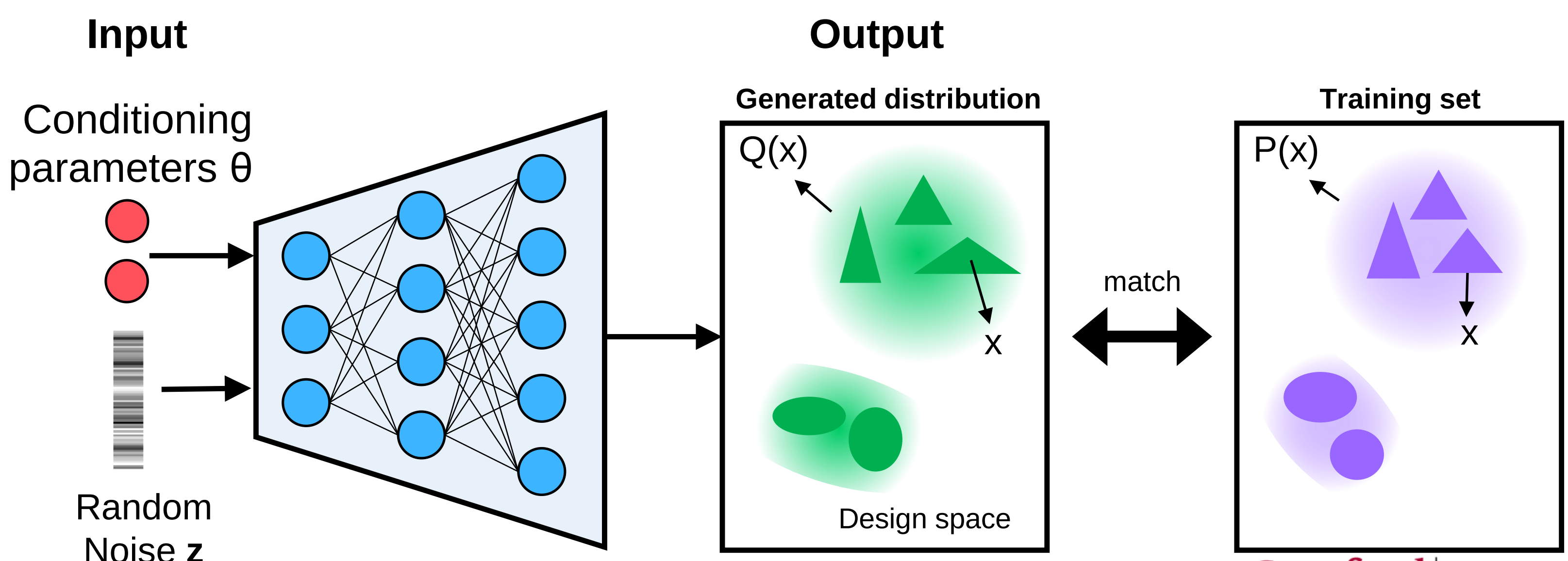
[1] Zhang, Adv. Theory Sim. 2, 1800132 (2019).
[2] Malkiel, Light. Sci. & Appl. 7, 60 (2018).
[3] Zhang, Photon. Res. 7, 368 (2019).
[4] Ma, ACS Nano 12, 6326 (2018).
[5] Zaabab, IEEE MTT-S 1, 393 (1994).
[6] Andrawis, Appl. Opt. 55, 2780 (2016).
[7] Ferreira, J. Light. Technol. 36, 4066 (2018).
[8] Peurifoy, Sci. Advances 4, eaar4206 (2018).
[9] Asano, Opt. Express 26, 32704 (2018).
[10] Pilozzi, Commun. Phys. 1, 57 (2018).

Outline

- Network classes and mathematical formulation
- Discriminative networks
- Generative networks
    - Variational autoencoders
    - Generative adversarial networks
- Dataless training of networks for optimization
- Demonstrations (http://metanet.stanford.edu/)

Generative networks

- Recall, generative networks have a latent space input, which enables one-to-many mappings.
- The training process for generative networks is different from that of discriminative networks because different assumptions can be made about the form of P(x) and Q(x).

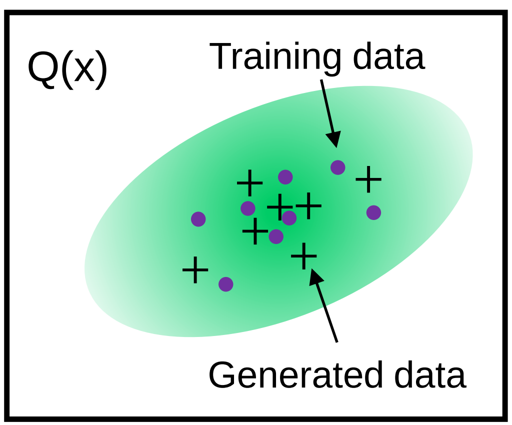


Ways to use generative networks

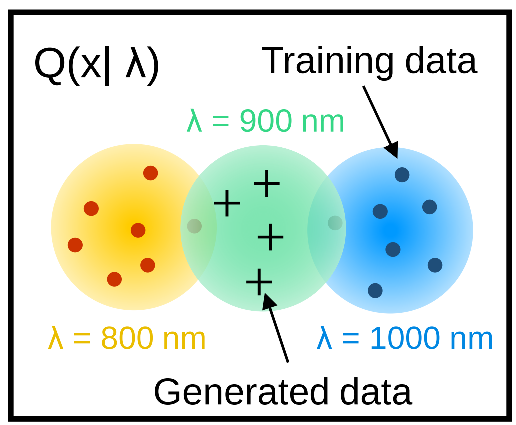
For inverse design, generative networks are typically used in one of three ways:

- An unconditional network generates variations of devices from a training set, some which are high performing.
- A conditional network generates distributions of devices operating at interpolated operating parameters.
- Classical optimization is performed in the latent space.

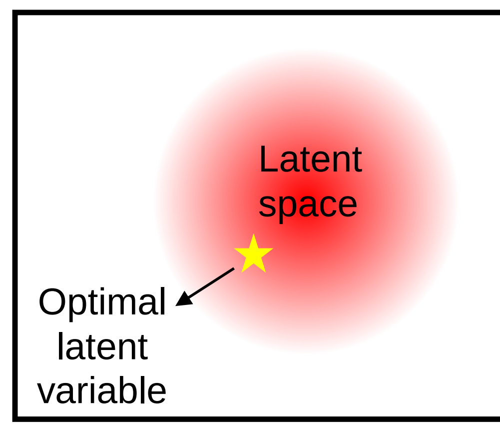
Unconditional distribution



Conditional distribution

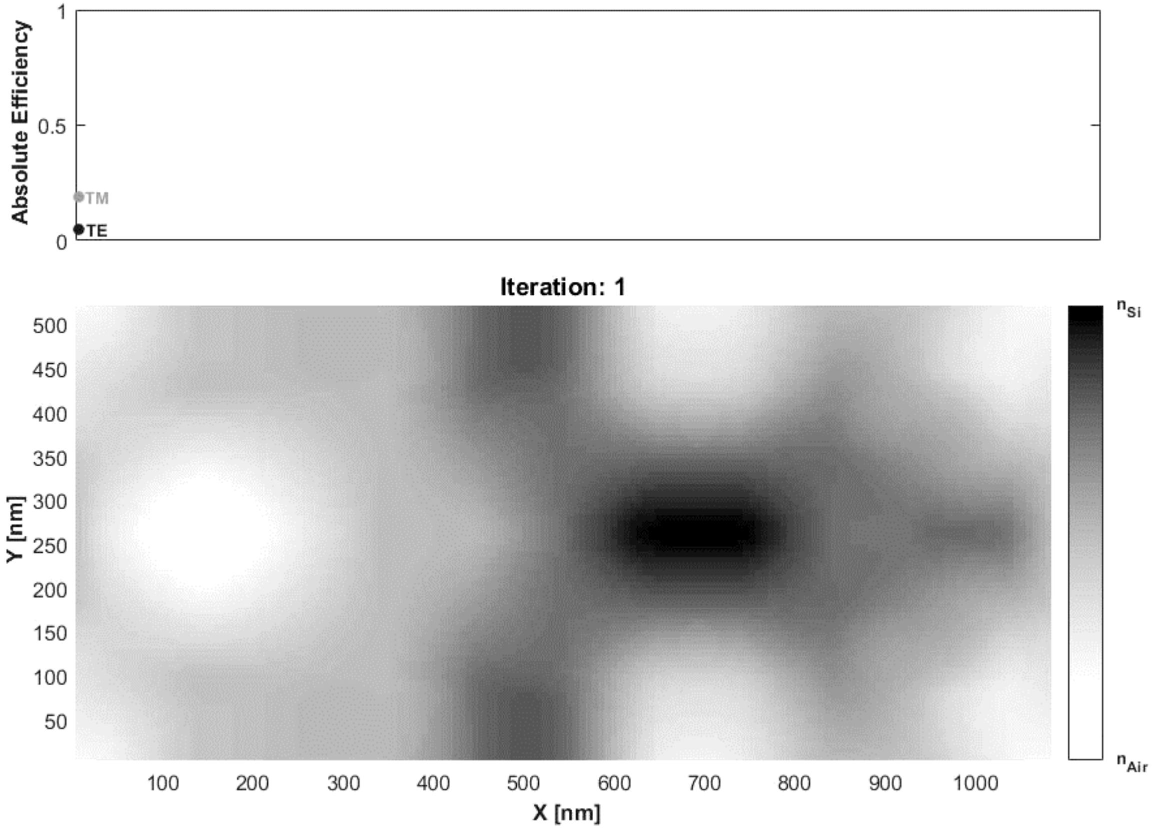


Optimization in the latent space



Images of freeform devices

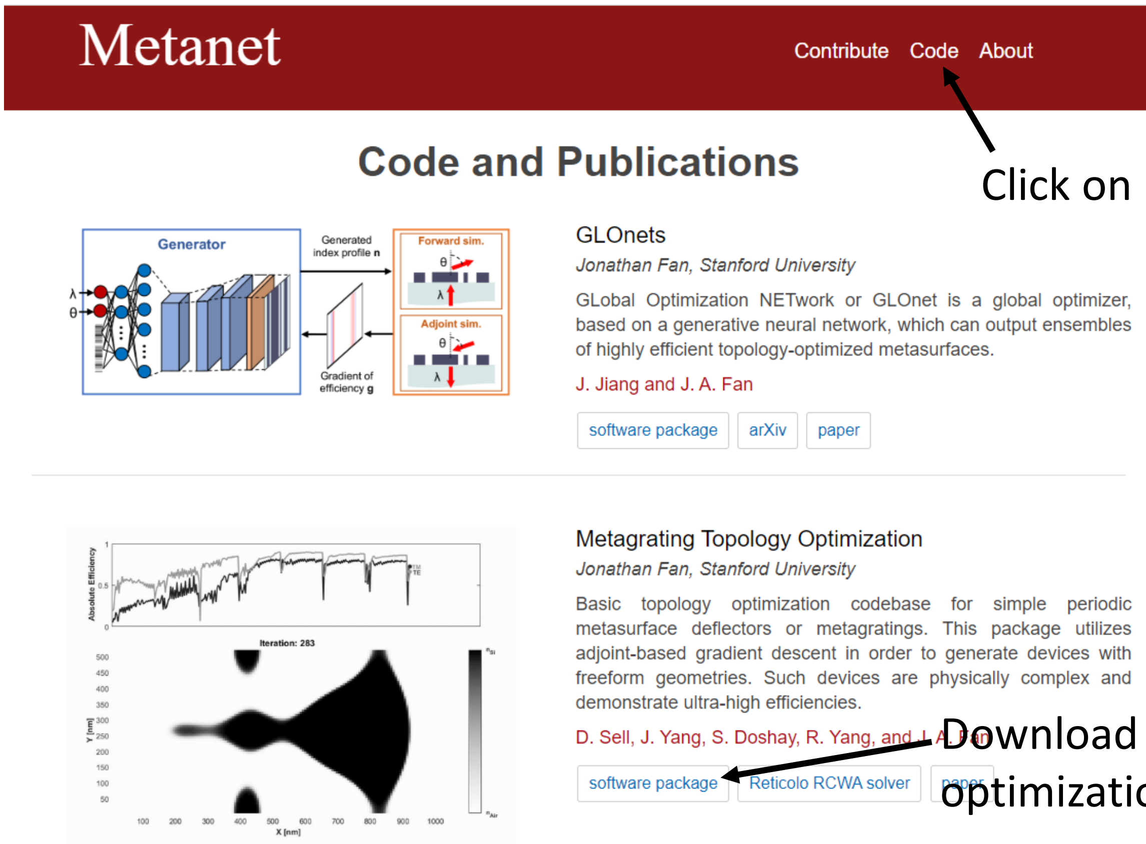
We can use generative networks to learn from images of complex devices, such as freeform nanophotonic structures.



Nano Lett., **17**(6), 3752-3757 (2017)

Download and install software

- If you would like to see topology optimization code, please go to http://metanet.stanford.edu/

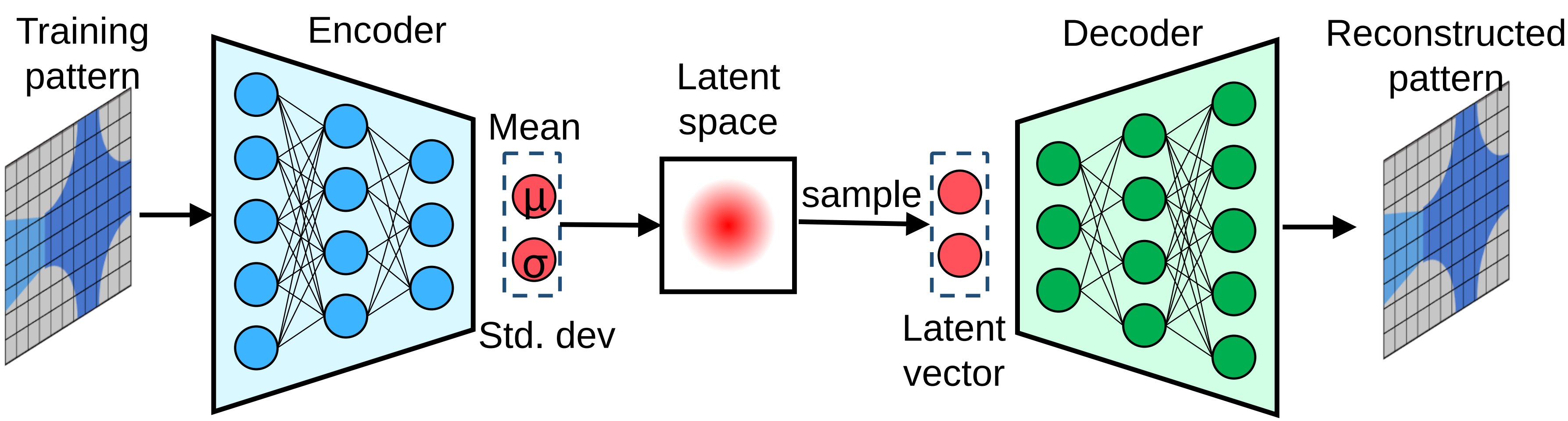


Click on “Code” tab

Download adjoint topology optimization code

Variational autoencoders (VAEs)

- VAEs are trained to reconstruct input data. As a generative network, we want to interpolate training data by sampling the latent space.
- The VAE encodes input data to latent space distributions.
  - The distributions are Gaussian defined by a mean and covariance matrix.
- The decoder is a generative network with a latent space input.

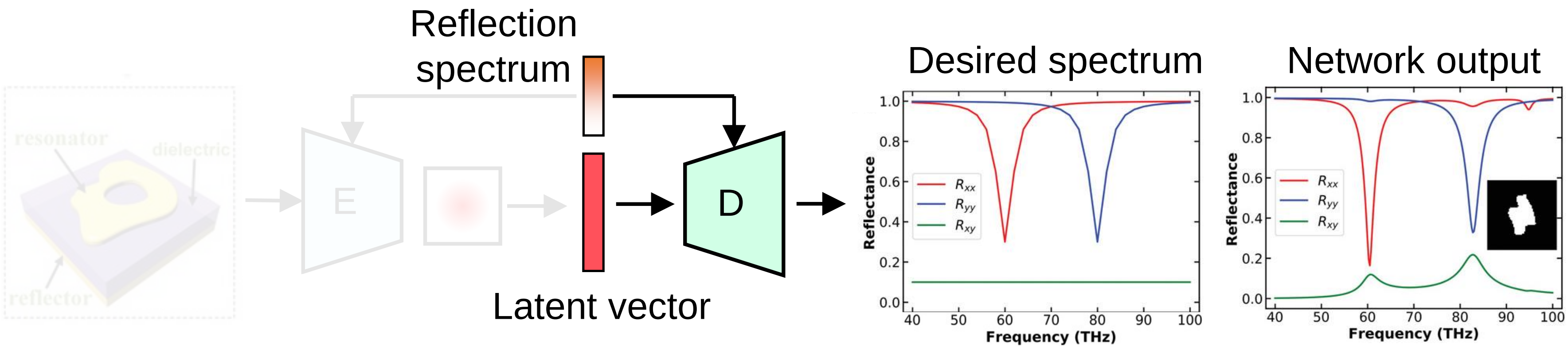


Conditional VAE for inverse design

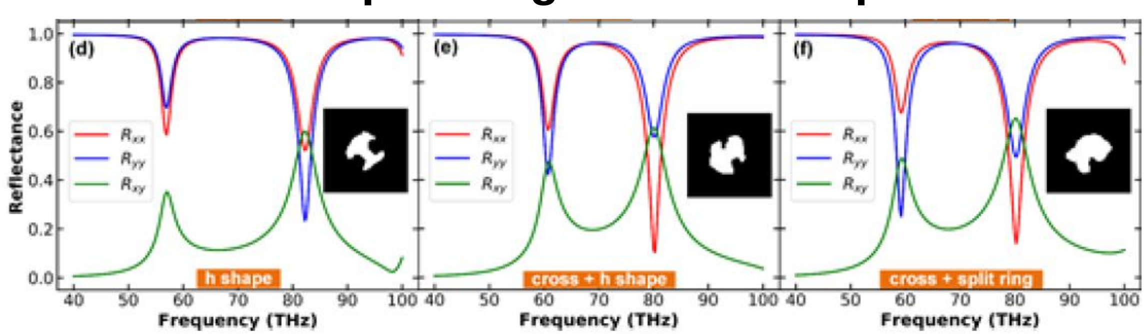
A VAE can be conditioned to output freeform metasurface structures as a function of desired spectral response.

- Training set includes H-shape, cross, and split ring shapes.

Network architecture



Examples of generated shapes

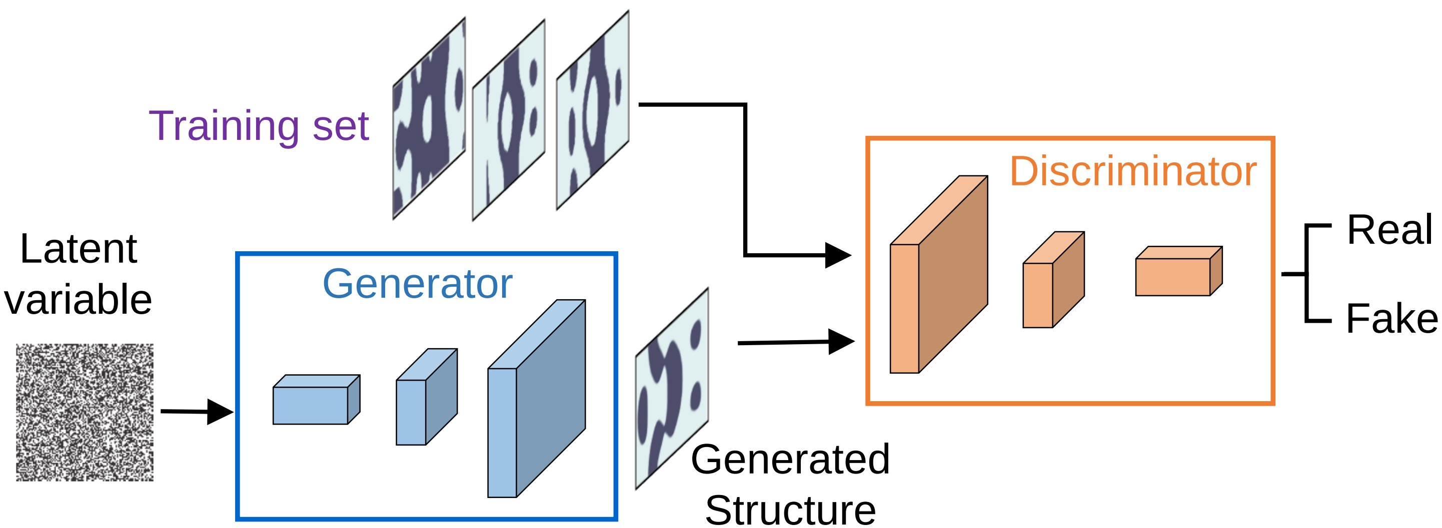


Ma, *Adv Mat* 31, 1901111 (2019)

Generative adversarial networks (GANs)

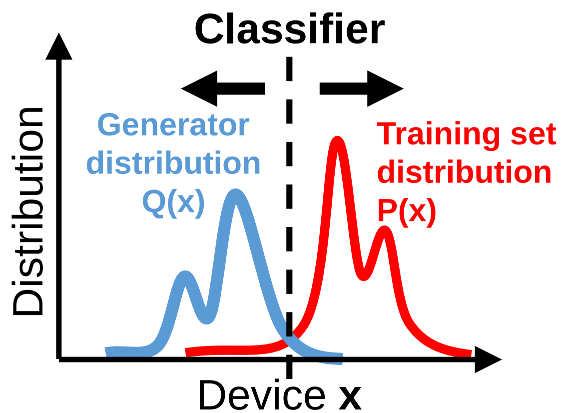
GANs are a method to train a generator to fit the implicit probability distribution of a training set.

- The discriminator is a classifier that attempts to determine whether the inputted data is from the training set or generator.
- The generator attempts to generate devices from latent variable inputs that match the training set.

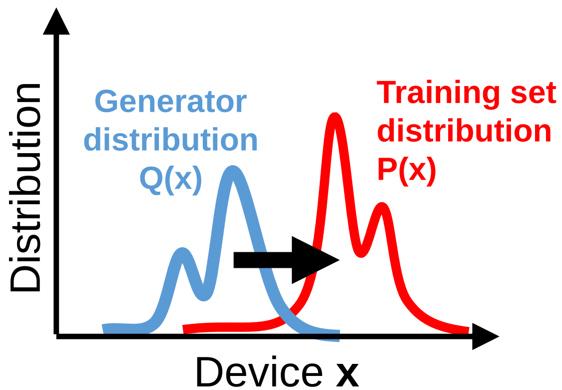


GAN loss function

- The training process can be considered as a two player game.
- The discriminator attempts to beat the generator by differentiating it from the training set.
    - Maximize cross entropy loss between the generated and training set distributions.

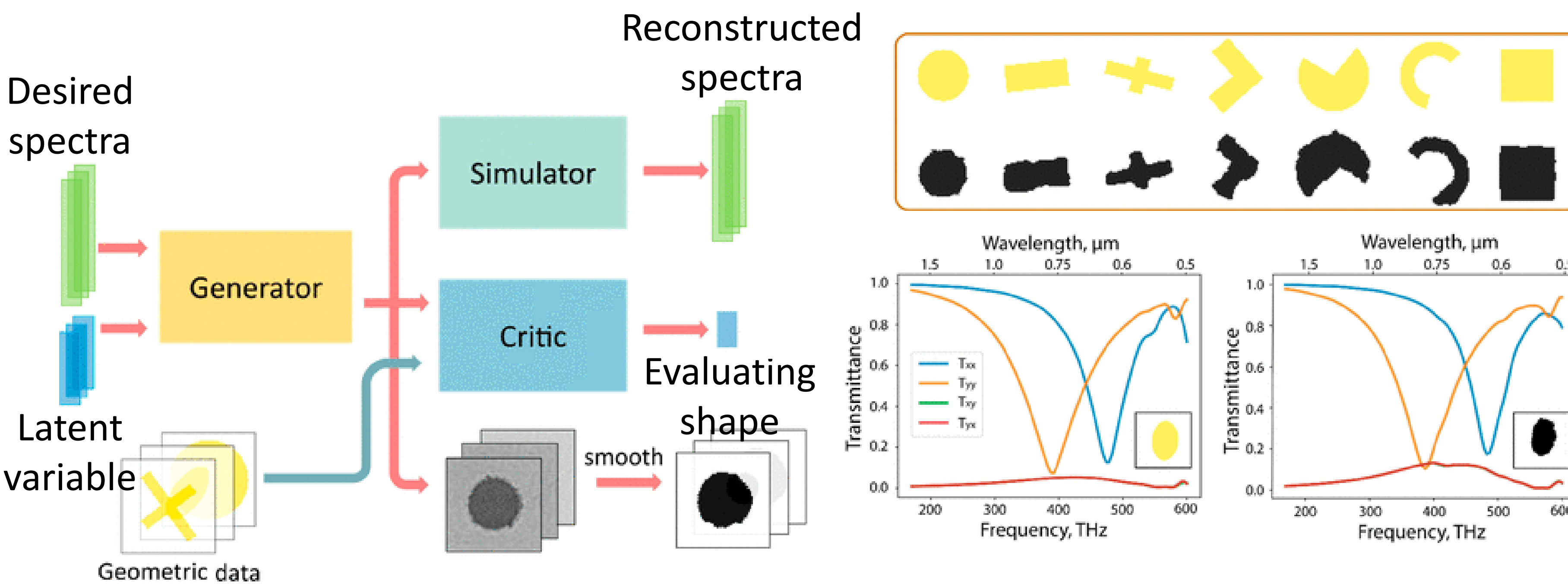


- The generator attempts to fool the discriminator by mimicking the training set.
    - Minimize cross entropy loss between the generated and training set distributions.



Conditional GAN as inverse model

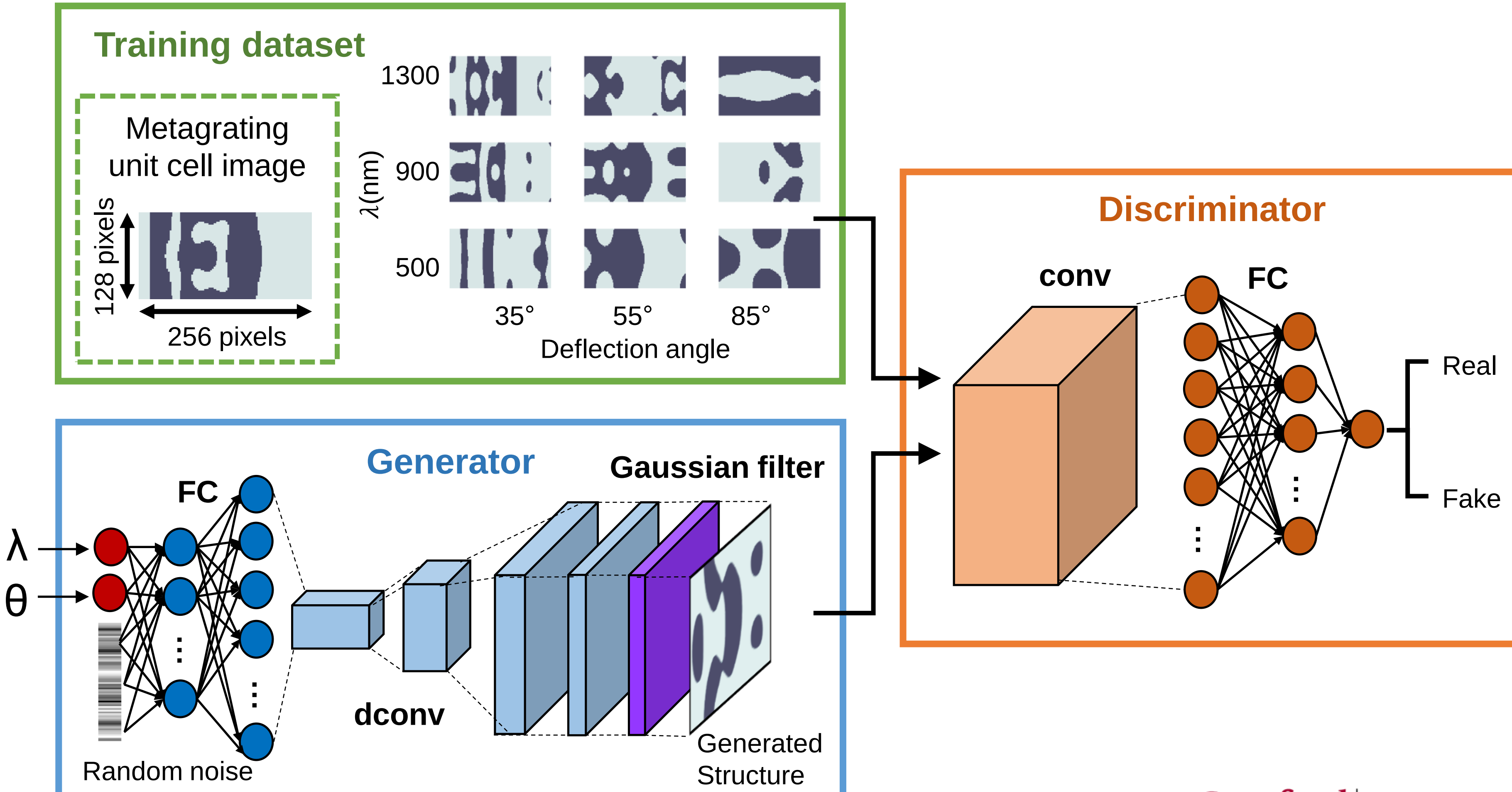
- The GAN is trained to learn basic shapes.
- Generator is conditioned on spectral response.
- There are two discriminators, a “simulator” that enforces proper spectral response and a “critic” that enforces shape from training set.



Liu, *Nano Lett*, 18, 6570 (2018)

Conditional GANs for interpolation

We use high efficiency, topology-optimized metagratings in our training data set.

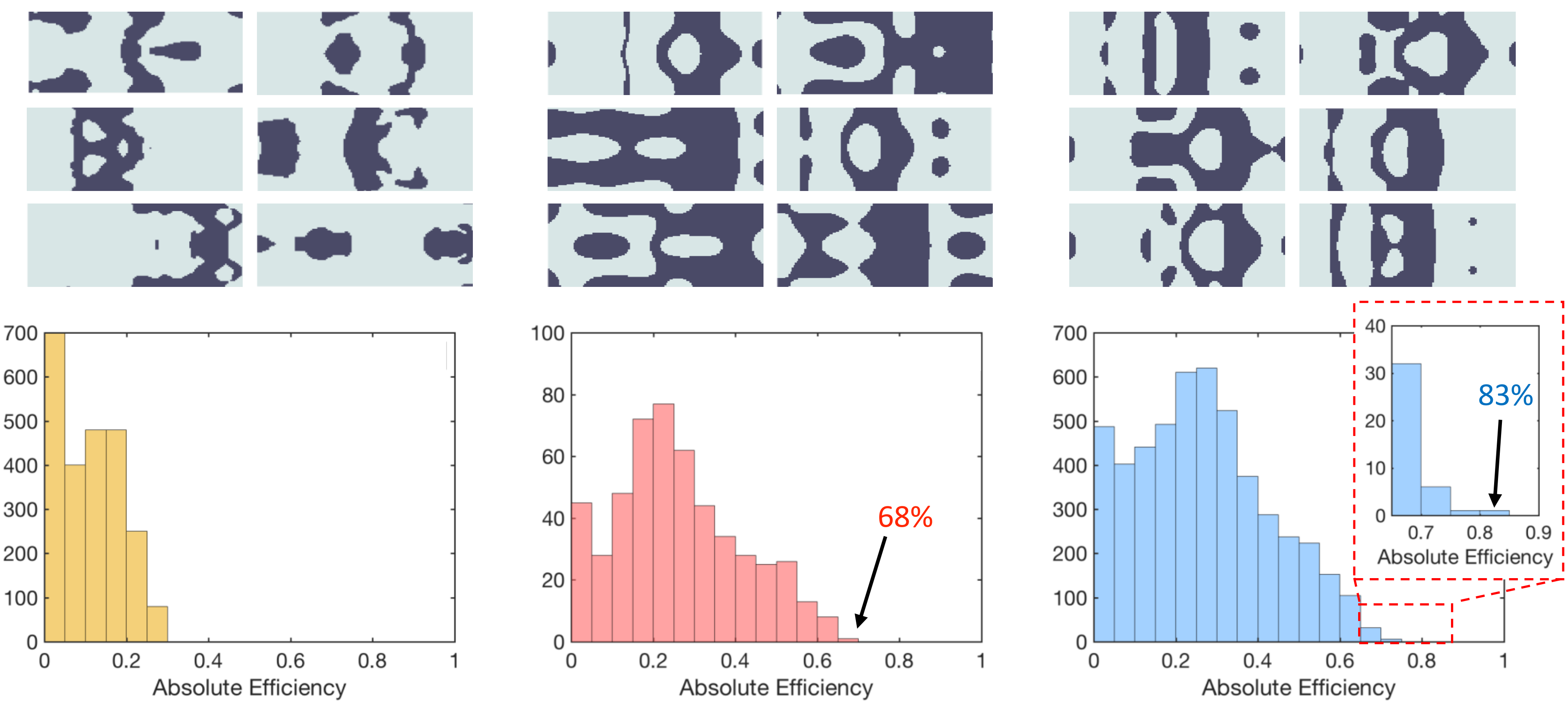


J. Jiang, *ACS Nano* 13, 8872 (2019)

Mimicking real devices

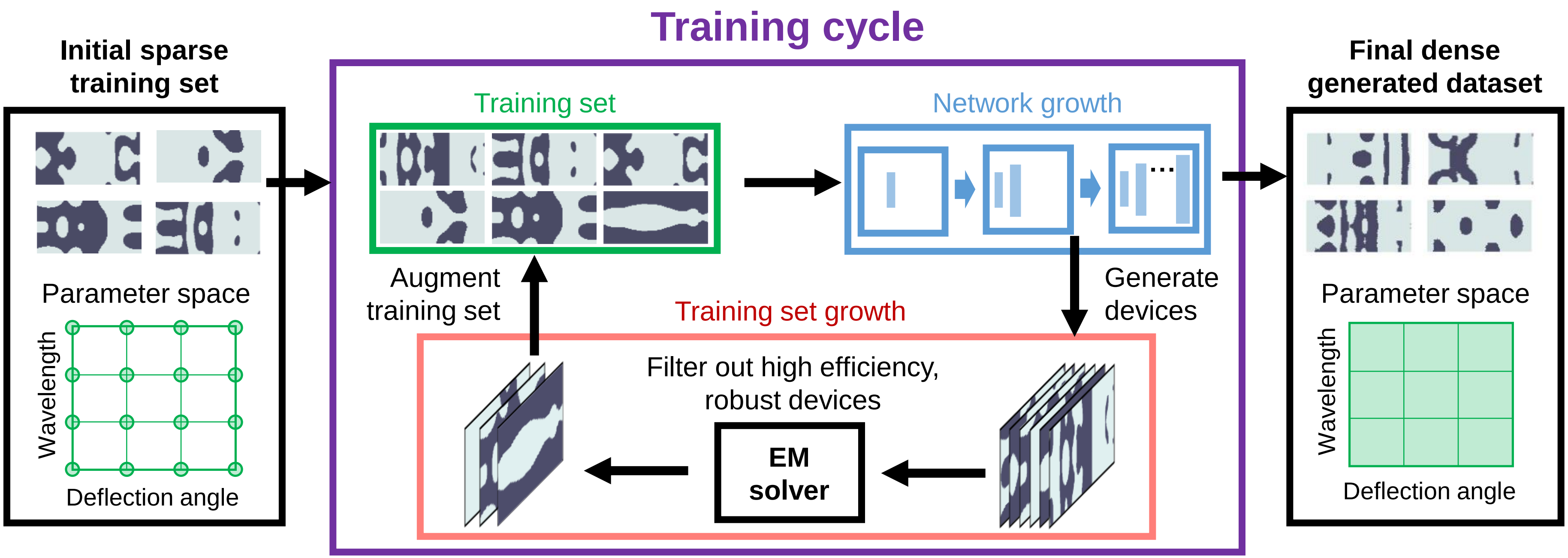
Do GAN-generated structures perform as working metagratings? We test them as 65° output, TE-pol devices operating at **λ = 1100nm**.

Random binary patterns | **Training set patterns** | **GAN-generated patterns**



Progressive growth of GANs

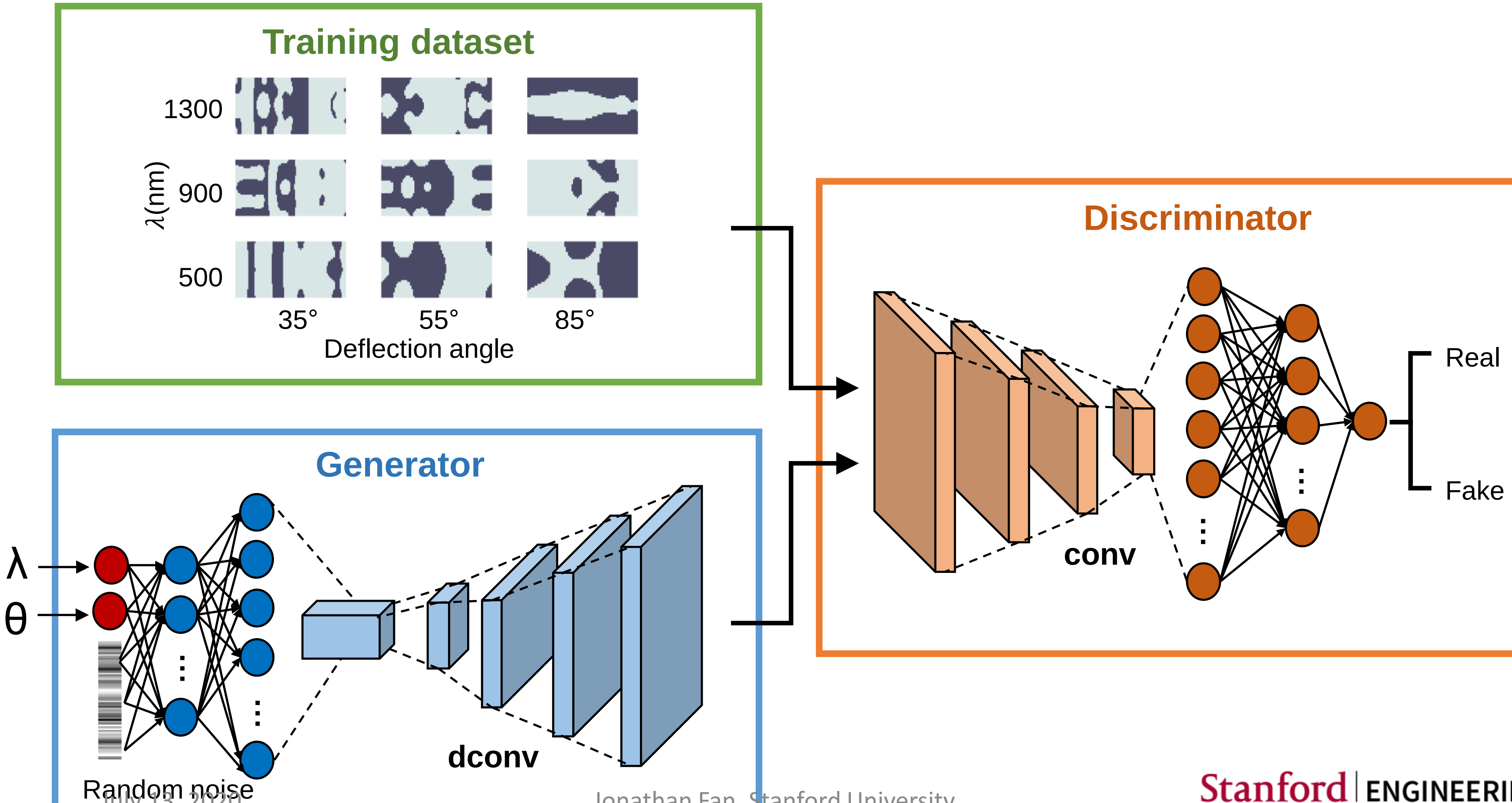
A new training scheme based on progressive growth of the network architecture and training set can lead to a substantially improved GAN.



J. Jiang, *ACS Photonics* ASAP

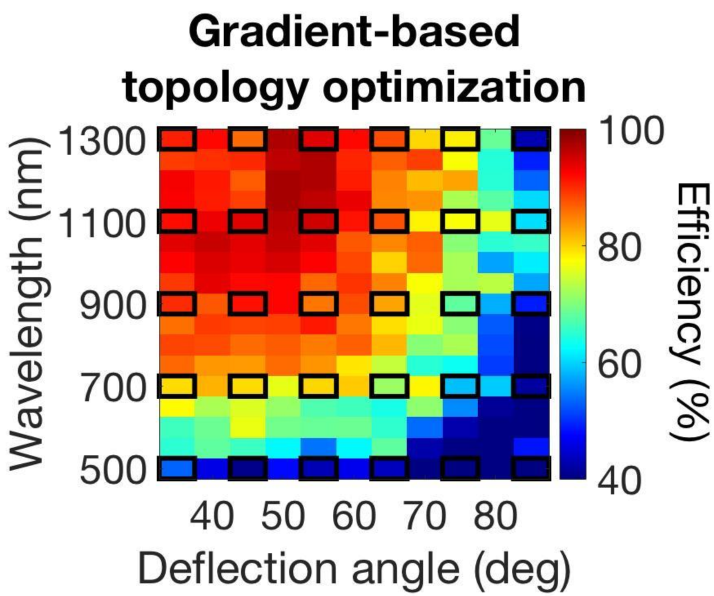
Progressively growing networks

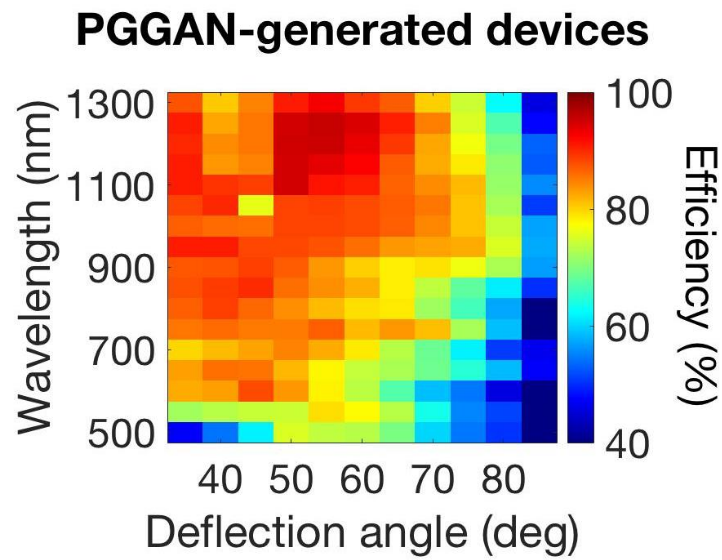
Progressively increasing the spatial resolution of the training set and network helps the GAN learn.



Benchmarking performance

Over 50% of the best PGGAN devices have higher efficiencies than the best topology-optimized devices and are robust.





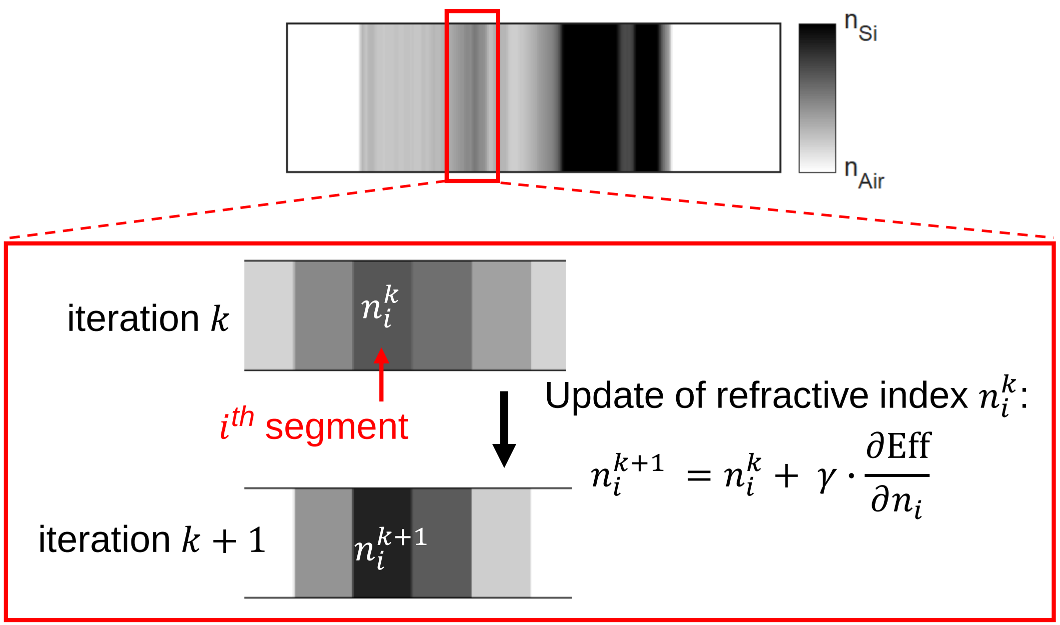
Stanford ENGINEERING
Electrical Engineering

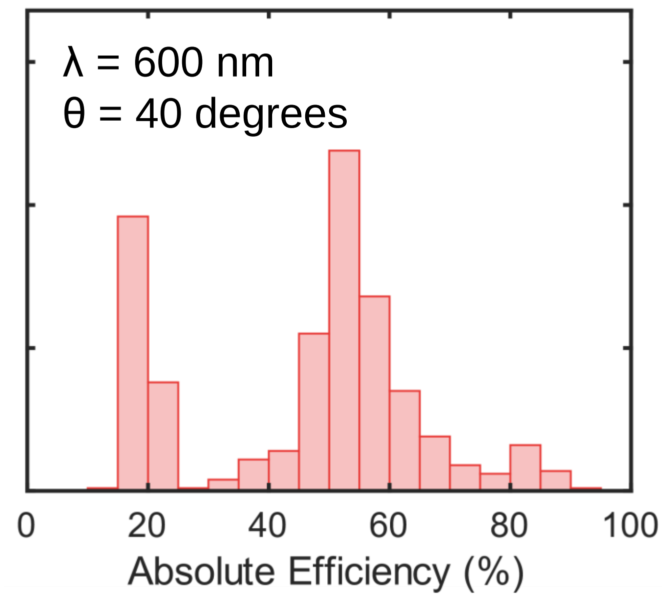
Outline

- Network classes and mathematical formulation
- Discriminative networks
- Generative networks
- Dataless training of networks for optimization
  - Global topology optimization networks (GLOnets)
  - Demonstration (http://metanet.stanford.edu/)

Topology optimization revisited

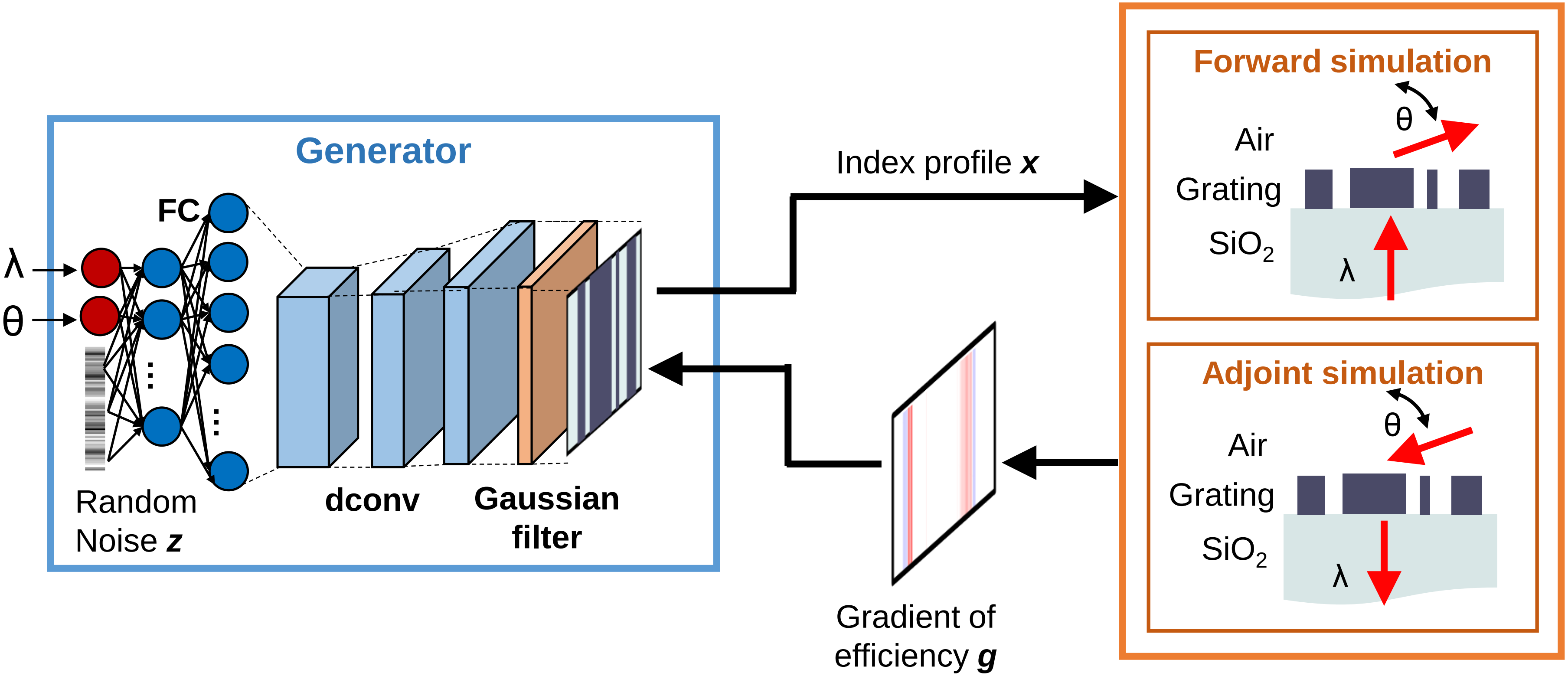
The adjoint variables method is a **local optimizer** that uses gradient descent to improve device performance.





Reframing topology optimization

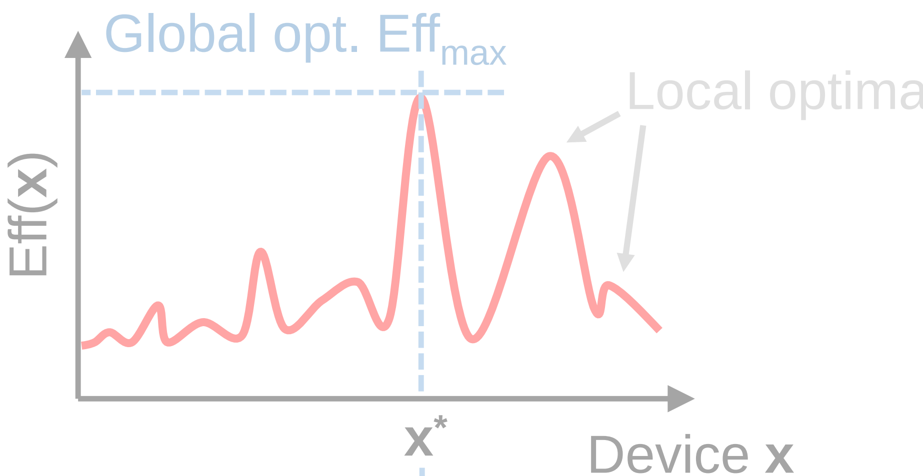
We introduce **global topology optimization networks** (GLOnets) as a new method for global optimization.

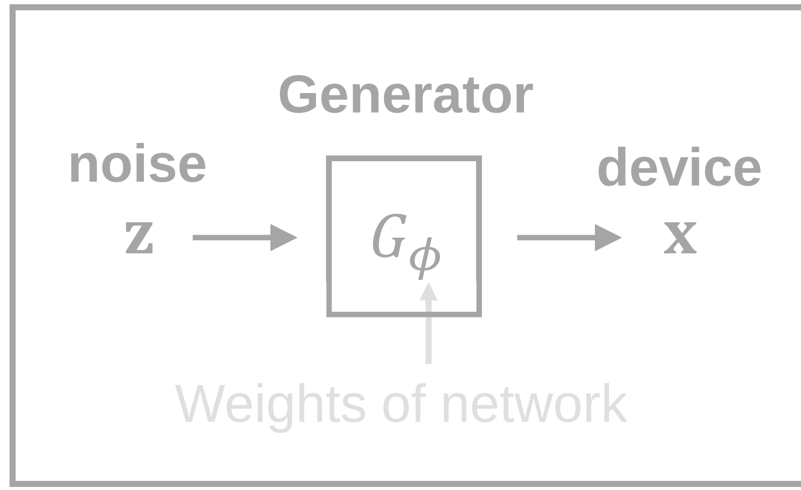


Nano Letters 19(8), 5366 (2019)

GLOnets: theoretical background

Complete design space



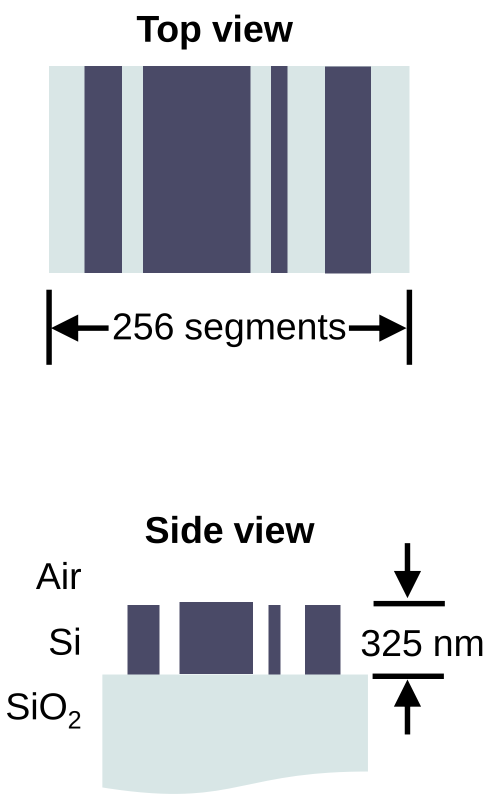


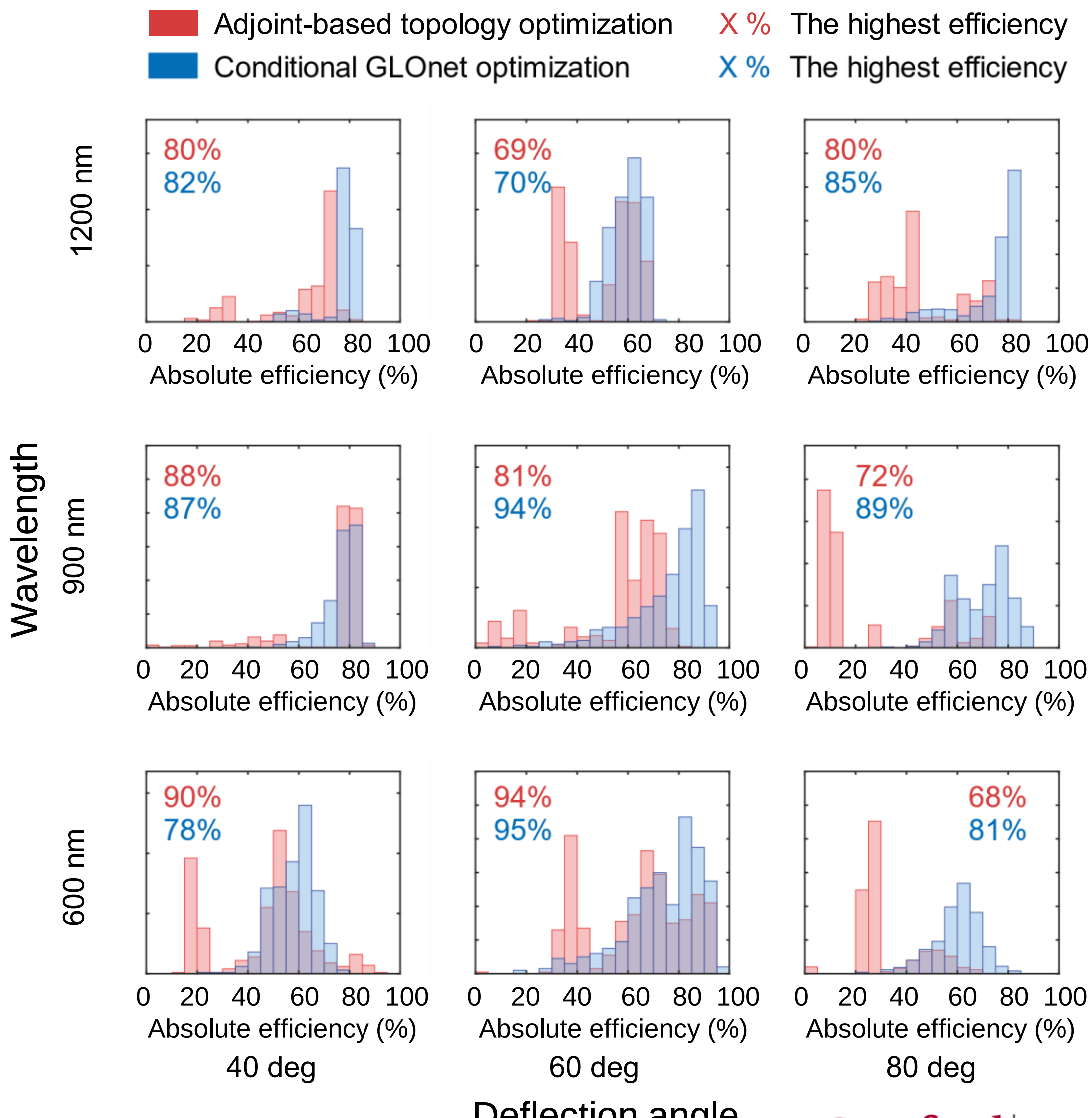
Optimization problem

$$\phi^* := \underset{\phi}{\operatorname{argmax}} \int_{\mathcal{S}} \delta\left(\mathrm{Eff}(\mathbf{x}) - \mathrm{Eff}_{max}\right) \cdot P_\phi(\mathbf{x}) d\mathbf{x}$$

Nanophotonics-2019-0330

Efficiency distributions

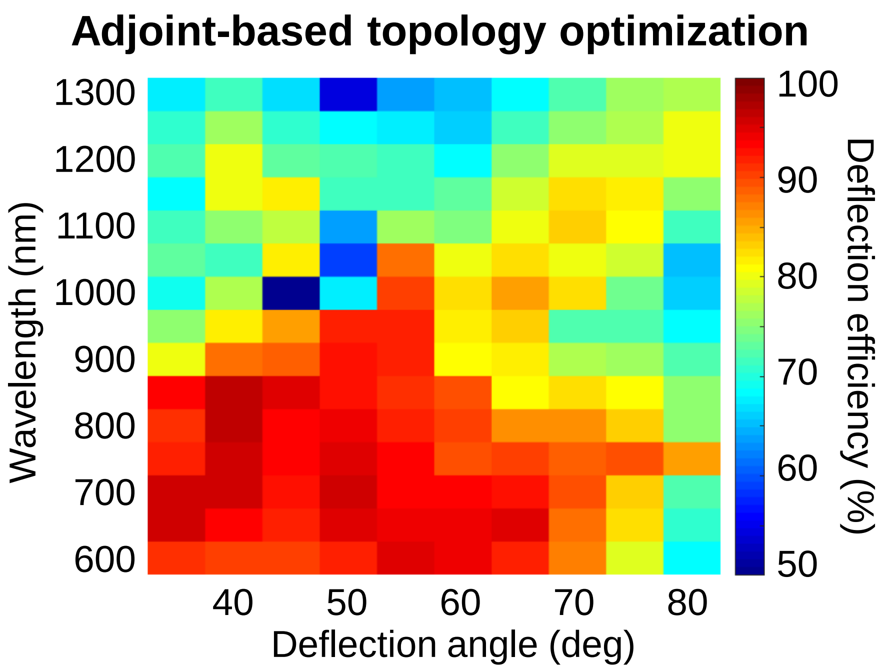


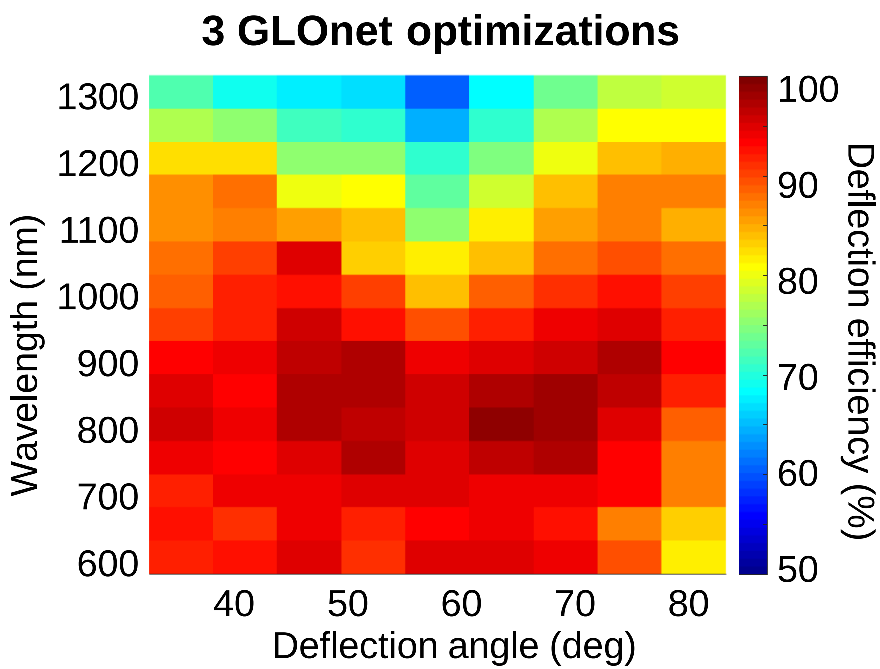


Nano Letters 19(8), 5366 (2019)

Comparative results

A comparison of the best overall devices shows that GLOnet can generate higher efficiency devices for the majority of operating parameters.



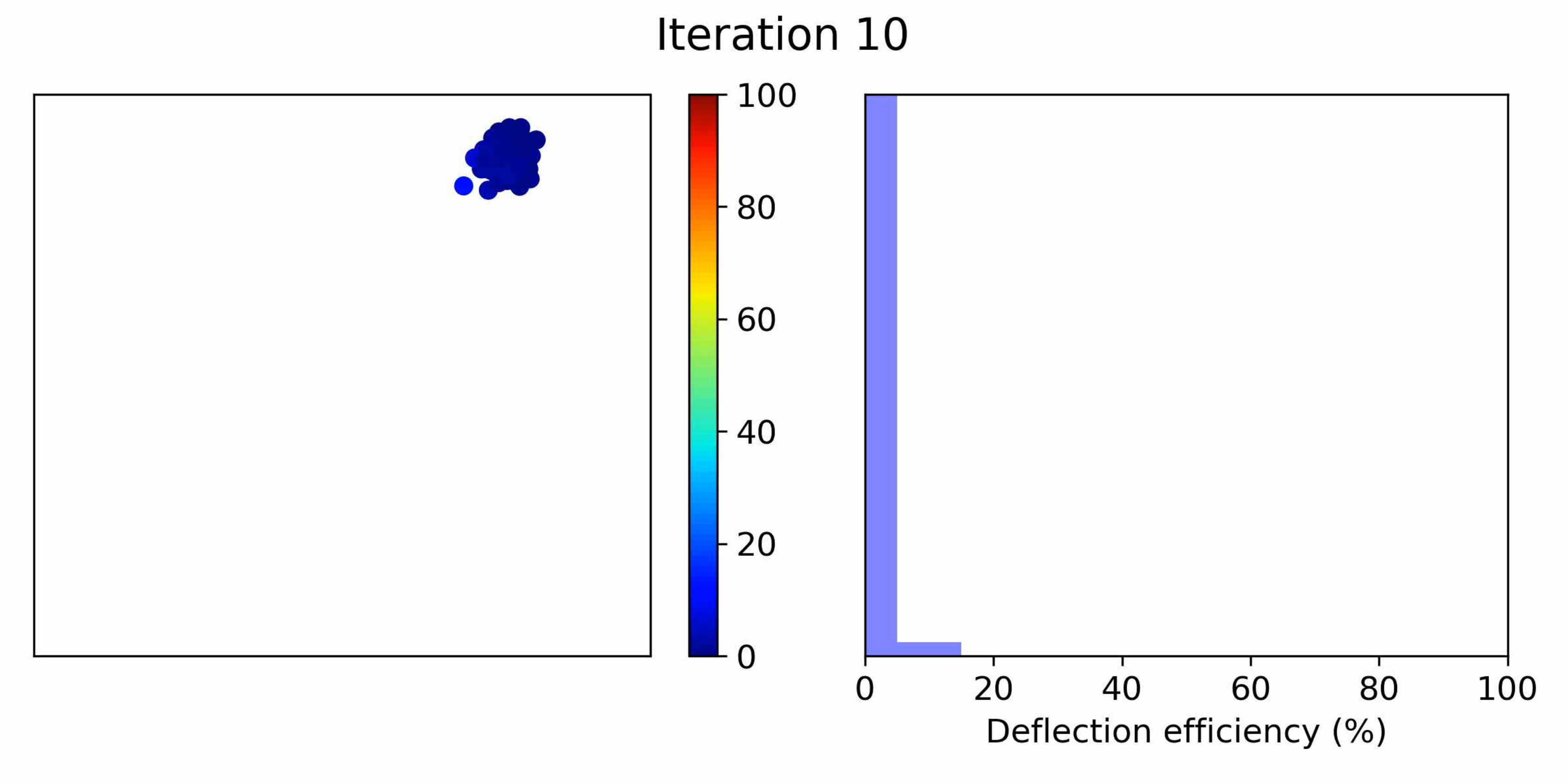


GLOnet required **10x** less computational cost compared to brute force topology optimization.

Visualizing GLOnet

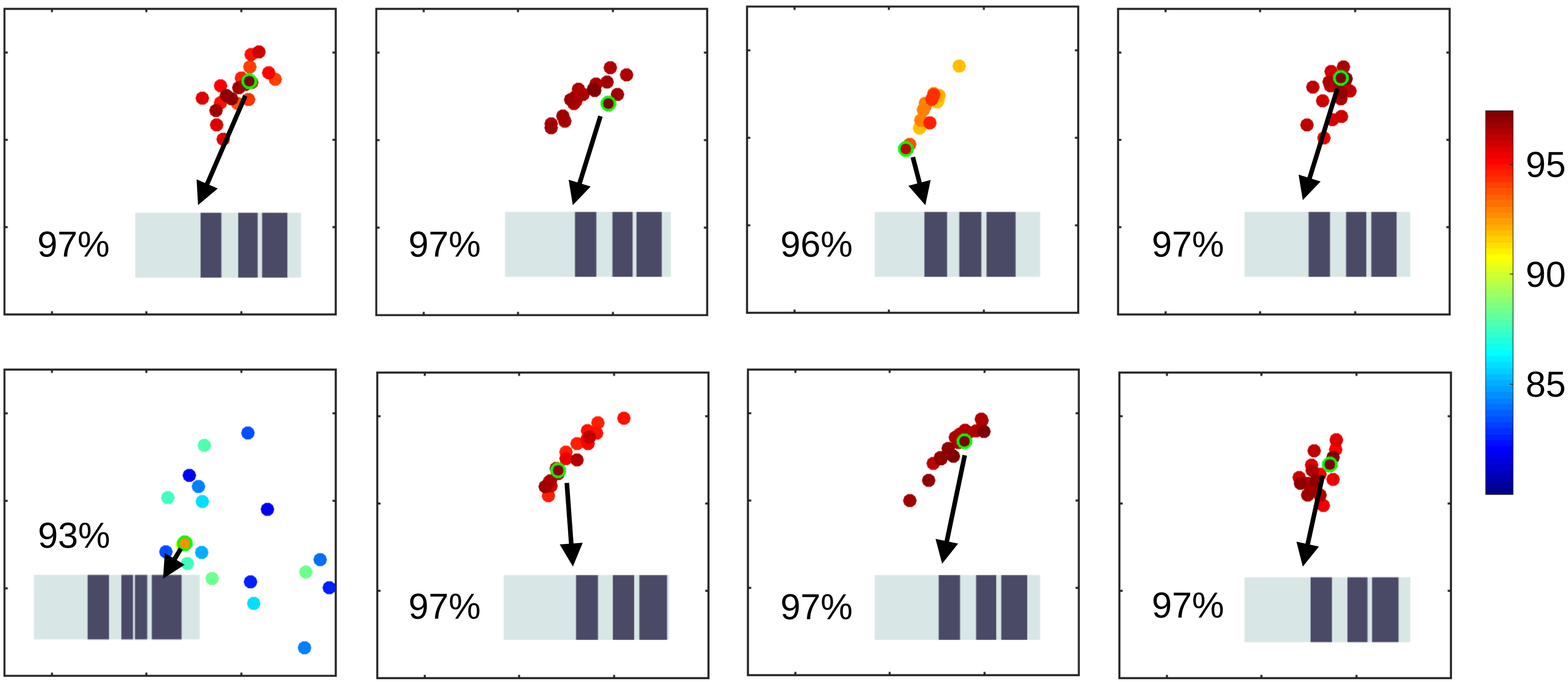
Plot of device efficiencies and geometries, depicted using principle components analysis.

- Device parameters: λ = 850 nm and θ = 65°



GLOnet stability

We train 8 unconditional GLOnets independently and the networks converge to the same optimal device 6 times.



Summary and Outlook

- Machine learning provides powerful methods to process data in highly non-linear and non-intuitive ways.
- There are still a lot of challenges and opportunities.
  - We require new concepts that intimately combine the physical structure of Maxwell with machine learning.
  - We require new electromagnetic simulators that can operate at significantly faster time scales.
  - We need to better streamline the training and refinement of neural networks for solving photonics problems, both from a data usage and user interface point of view.
  - We need to coordinate research efforts better to benchmark algorithms and devices.

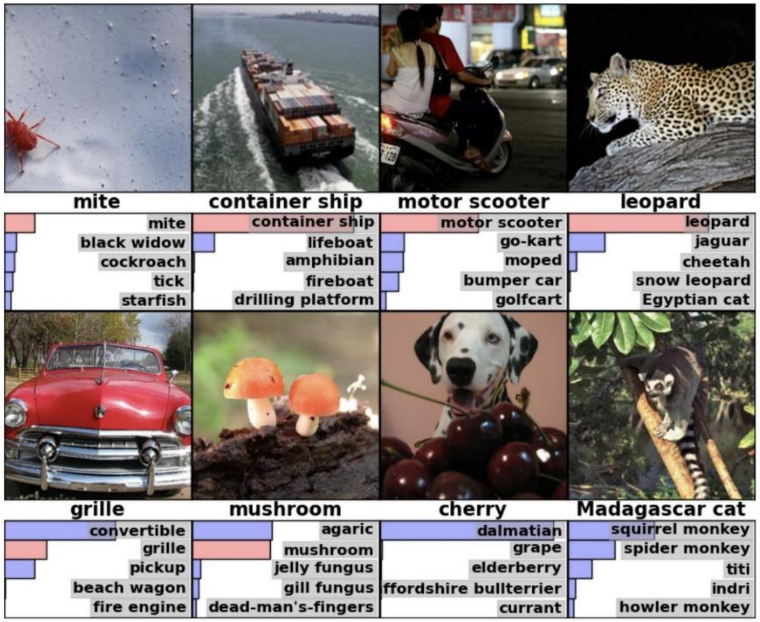
Coordinating research efforts

Machine learning research in the CS community is driven through open source coding and proper benchmarking of algorithms with common training data.

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



http://metanet.stanford.edu

http://metanet.stanford.edu

Metanet

Contribute Code About

Filter Metagratings ?

Geometry Type: ---------

Material Type: ? ---------

Angle Range:

Wavelength Range:

Thickness Range:

Efficiency Range:

Polarization: ---------

Optimization Method: ? ---------

Research Group: ---------

Filter | Download Filtered

Metagratings

Angle (°)	Wavelength (nm)	Thickness (nm)	Efficiency	Download Link	Optimization Method	Research Group
35.0	1000.0	325	0.9527	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9365	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9533	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9504	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9440	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9655	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9519	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9524	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9503	Download	Brute Force	Jonathan Fan Lab
35.0	1000.0	325	0.9528	Download	Brute Force	Jonathan Fan

1 of 5423 > >>

Metamaterials events and design contest

- I am the new chair of the OSA Photonic Metamaterials technical group, I am planning to have a design contest.
- If you want to stay in the loop, please sign up:

https://www.osa.org/en-us/get_involved/technical_groups/