Student Paper Finalist

Advanced Practice Paper Finalist

We2A-319-HC775

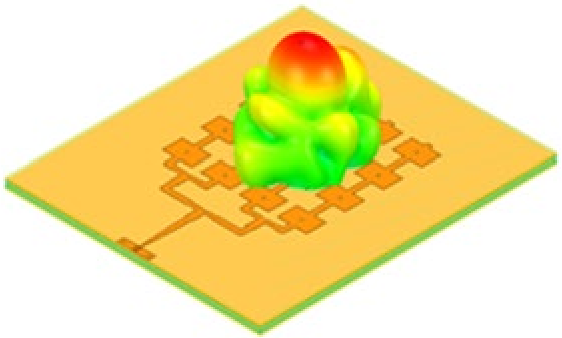
Surrogate Modeling with Complex-valued Neural Nets and its Application to Design of sub-THz Patch Antenna-in-Package

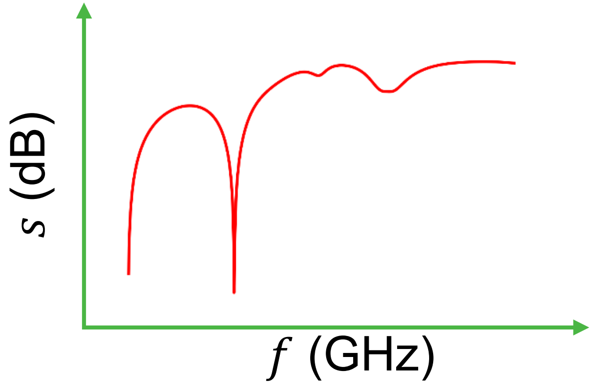
O. Akinwande[1], O. W. Bhatti[1], K.-Q. Huang[1], X. Li[1], and M. Swaminathan[1,2]

[1]Georgia Institute of Technology

[2]Pennsylvania State University

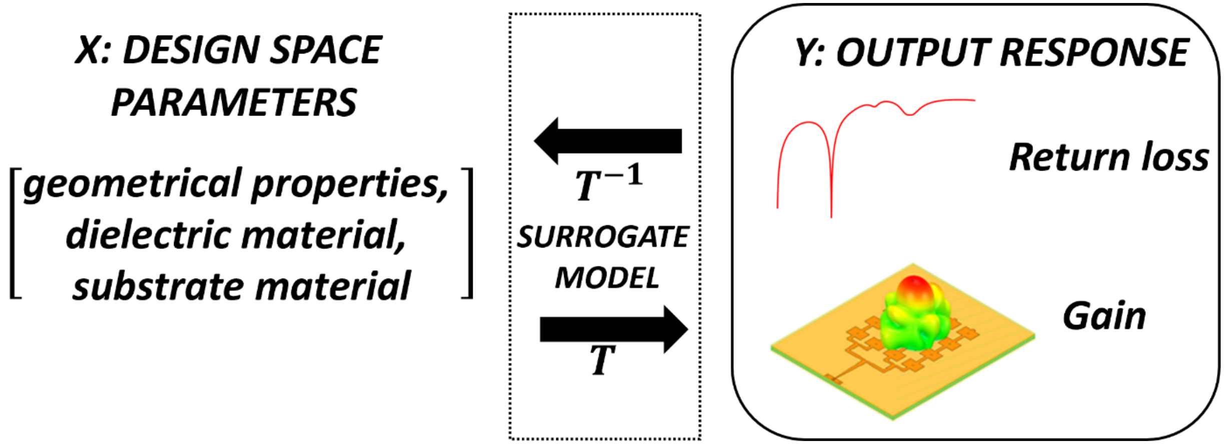
Motivation





- Simulating RF front-end modules like antennas is essential for design of wireless communications
- Simulating their behavior can be computationally and time–intensive
  - Size & complexity of the structure
  - Frequency range of interest
  - Operating environment
- Designing an antenna involves determining the suitable set of design parameters that generate the desired output response

Objectives



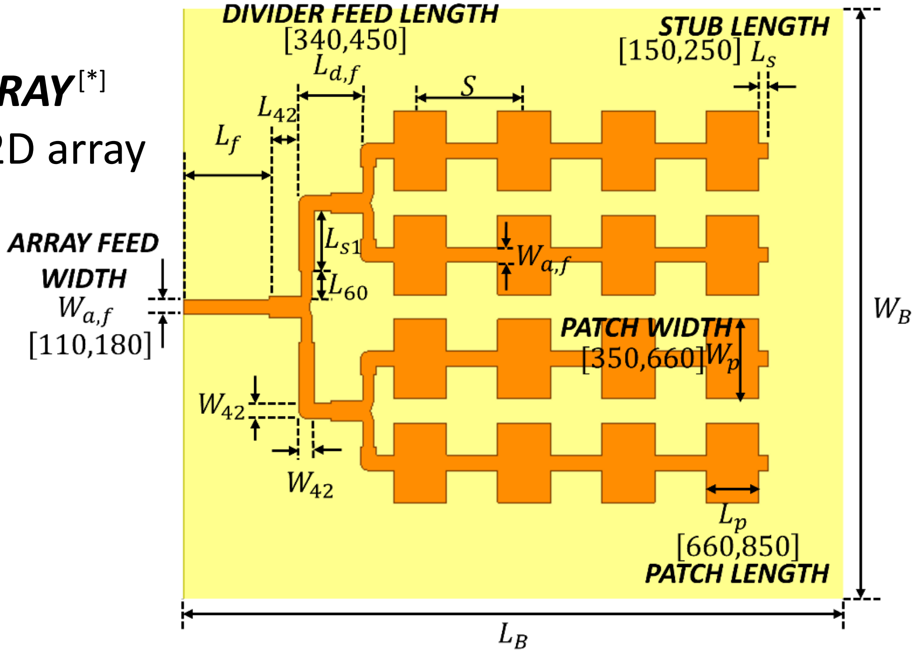
- ❑ Since Machine Learning (ML) techniques provide good representations of data
- ❑ Build a fast ML-based surrogate model that enables the designers to:

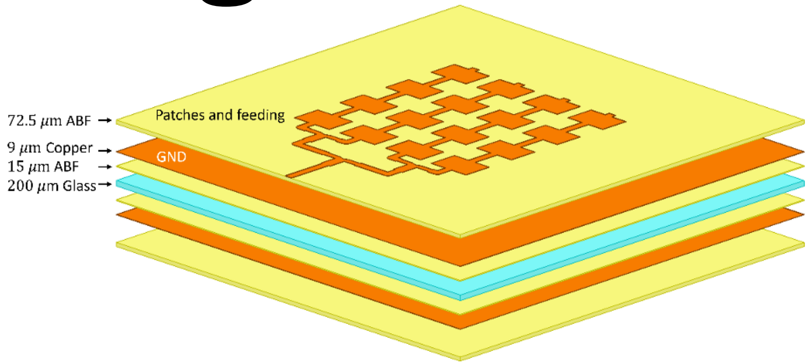
1. Simulate their designs to meet a target spec
2. Obtain the design parameters that correspond to a given spec

Example: Design of sub-THz Patch Array Antenna-in-package

PATCH ARRAY [*]

- ❑ 4 × 4 2D array





Objectives:

- Forward modeling → ***FREQUENCY RESPONSE*** S_{11} from [130.1, 150.05] GHz
- Inverse modeling → ***DESIGN PARAMETERS*** $W_p, L_p, W_{a,f}, L_s, L_{d,f}$

DESIGN SPACE PARAMETERS OF SUB-THZ PATCH ARRAY

Parameter		Unit	Min	Max
Patch width	W_p	μm	350	660
Patch length	L_p	μm	660	850
Array Feed width	$W_{a,f}$	μm	110	180
Stub length	L_s	μm	150	250
Divider feed length	$L_{d,f}$	μm	340	450

DATA:

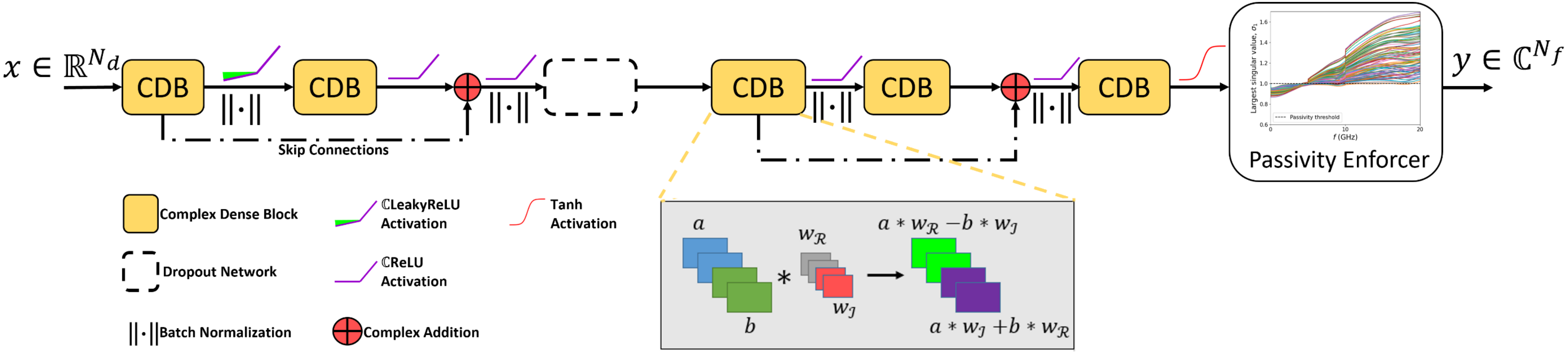
- ❑ Input design space $X \in R^5$
- ❑ Output specs $Y \in \mathbb{C}^{134}$

MODEL:

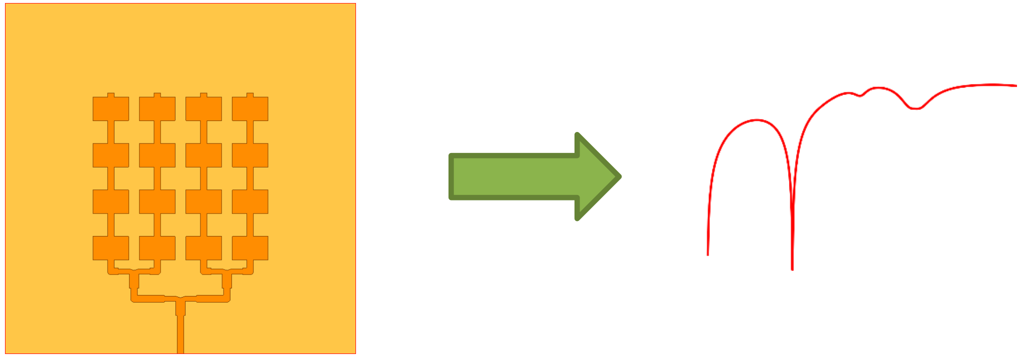
- ❑ ℂDNet
  - ➢ 6 complex dense blocks

[*] K. -Q. Huang and M. Swaminathan, "Antennas in Glass Interposer For sub-THz Applications," (ECTC), 2021

Surrogate modeling with deep complex dense net (ℂDNet)



Objective I: Forward modeling



- Obtain a frequency response based on given design parameters
- $\mathbb{C}$DNet learns the forward mapping between the patch array design space x and the frequency response y
- Train with an ℓ_2-supervised loss

$$\mathcal{L} = \mathbb{E}_{x,y}[\|\hat{y}_{\mathcal{R}} - y_{\mathcal{R}}\|_2^2 + \|\hat{y}_{\mathcal{J}} - y_{\mathcal{J}}\|_2^2]$$

where $\hat{y}$:=predicted S_{11}, y :=actual S_{11}

Modeling physically consistent response: Passivity of *S*-parameters

- A multiport network is passive if it cannot generate energy
- ⇔S-parameter matrix is unitary bounded, i.e.,

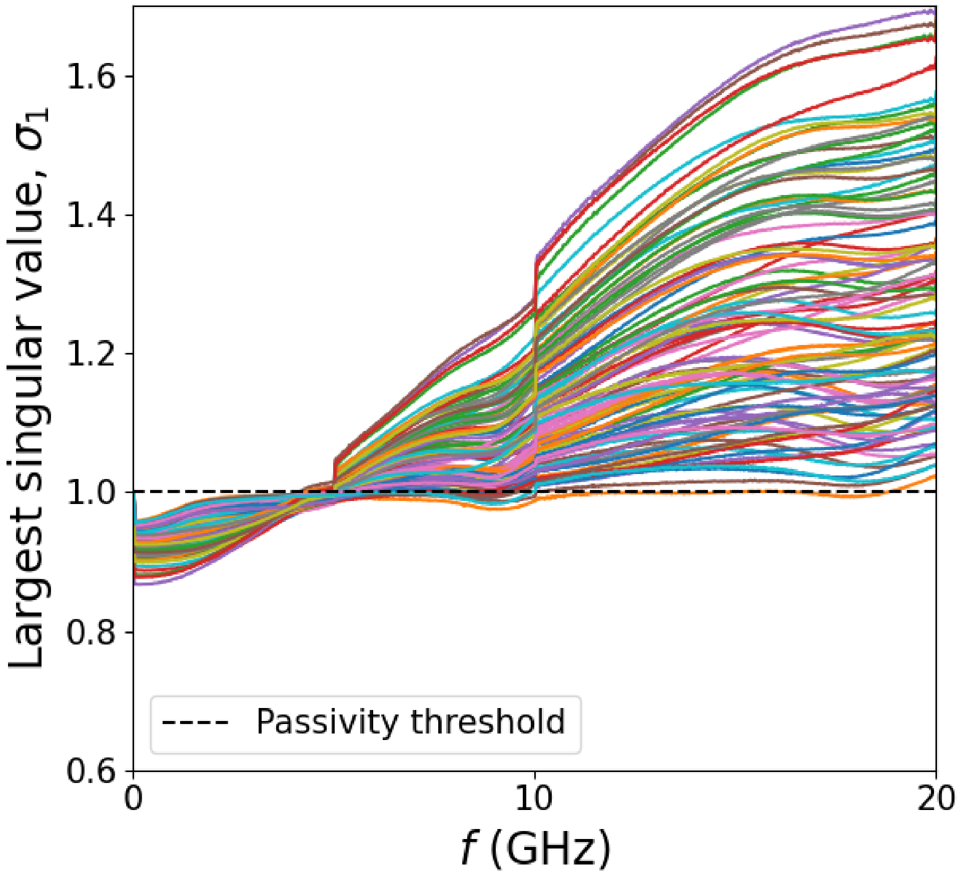
$$S^H(f)S(f) \le I \quad \forall f \in B$$

- $\Leftrightarrow \max_{i,f} \sigma_i(f) \le 1, \quad i: f_i \in B$
- All singular values must be bounded by one at all frequencies
- Passivity enforcer is added as the last layer of the NN model

Achieving physical consistency

$\sigma_1 > 1$

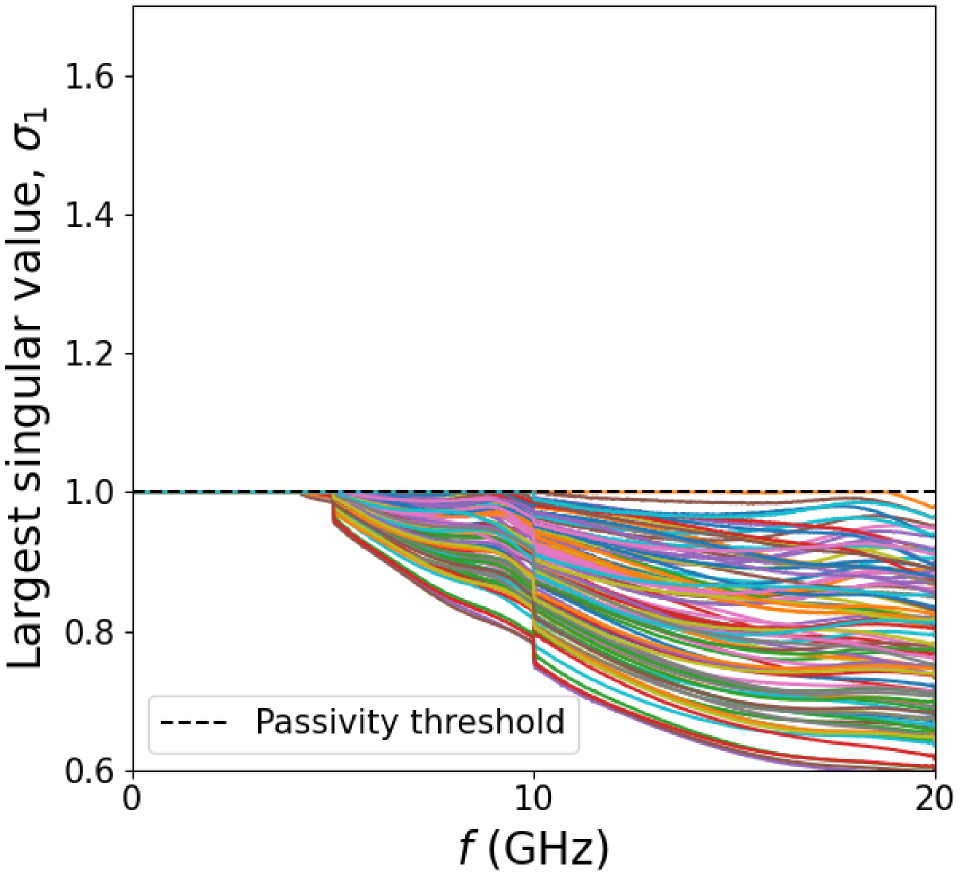
$\sigma_1 \in [0.867, 1.703]$



(a)

$\sigma_1 \leq 1$

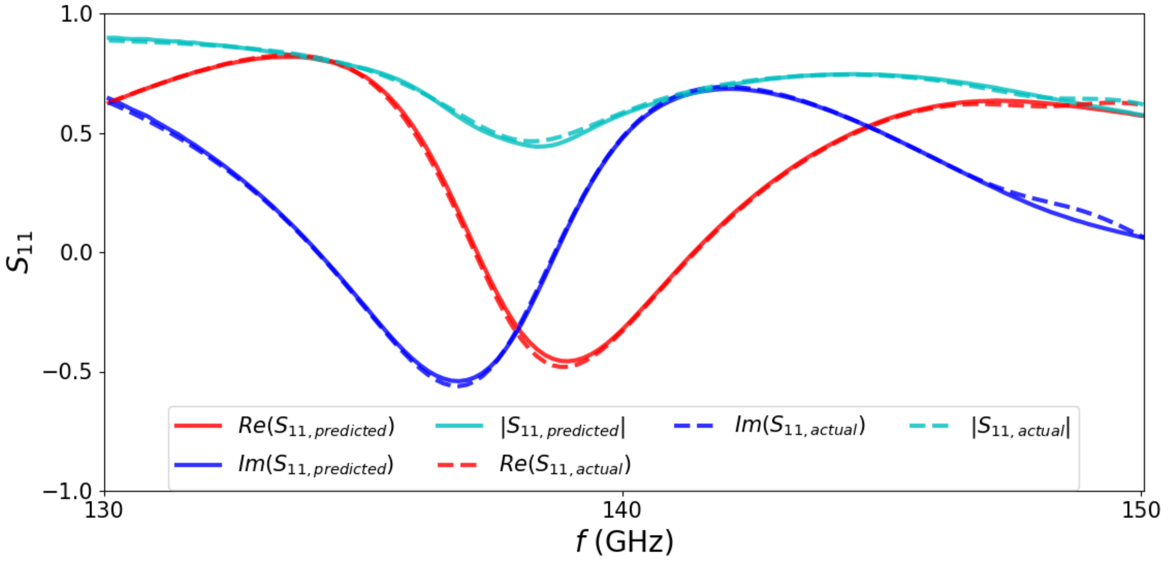
$\sigma_1 \in [0.587, 1]$



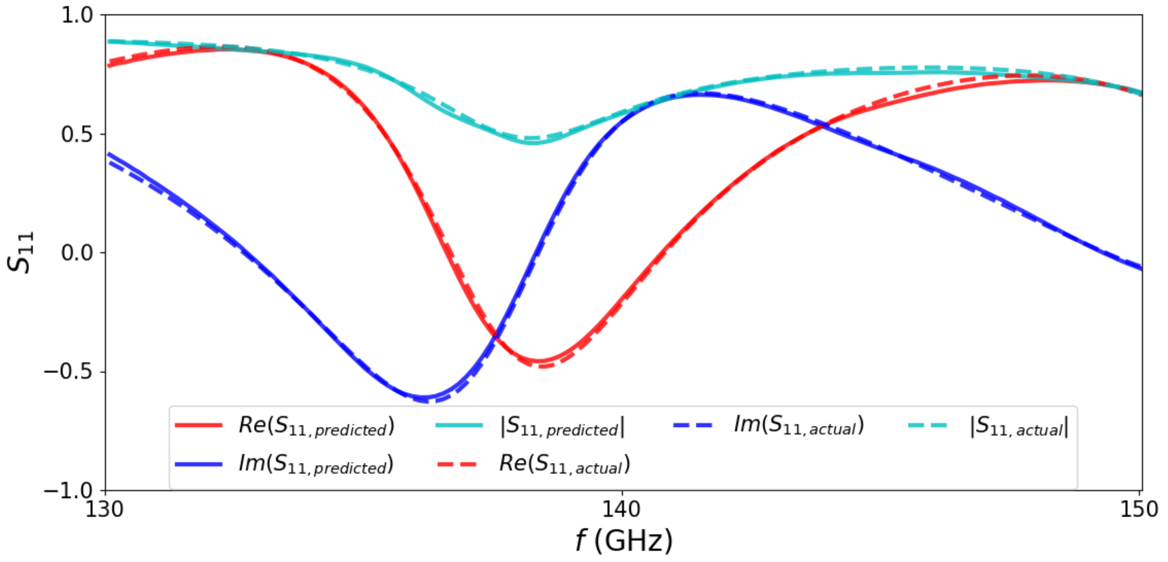
(b)

Forward modeling results

- Perform forward inference on random samples in test set

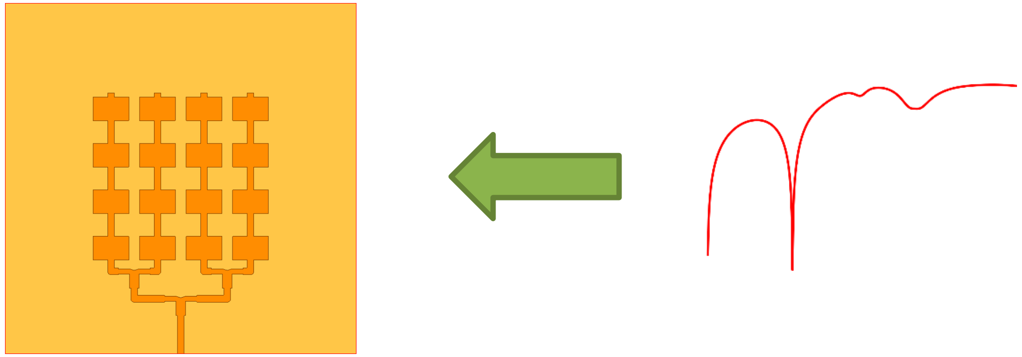


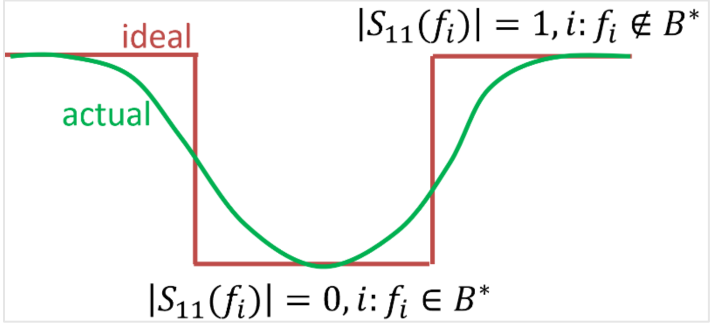
Sample A



Sample B

Objective II: Inverse optimization





- ❑Obtain design parameters that correspond to a given spec of $|S_{11}|$
- ❑Objective: ℓ_2-norm of the difference between the ideal $|S_{11}|$ (i.e., y^*) and that delivered by the forward model (i.e., $\hat{y}(x)$)

$$\hat{x} = \underset{x}{\operatorname{argmin}} \sum_{i:f_i \in B^*} |\hat{y}_i(x)|^2 + \sum_{i:f_i \notin B^*} (|\hat{y}_i(x)| - 1)^2$$

where $\hat{x}$:=inverse solution, B^* :=target band

Objective II: Inverse optimization

Algorithm 1: Inverse optimization

Input: Initialization $x^{(0)} \in dom(g)$, trained model g with the set of all network parameters θ, target band B^*, learning rate λ

Output: estimated x

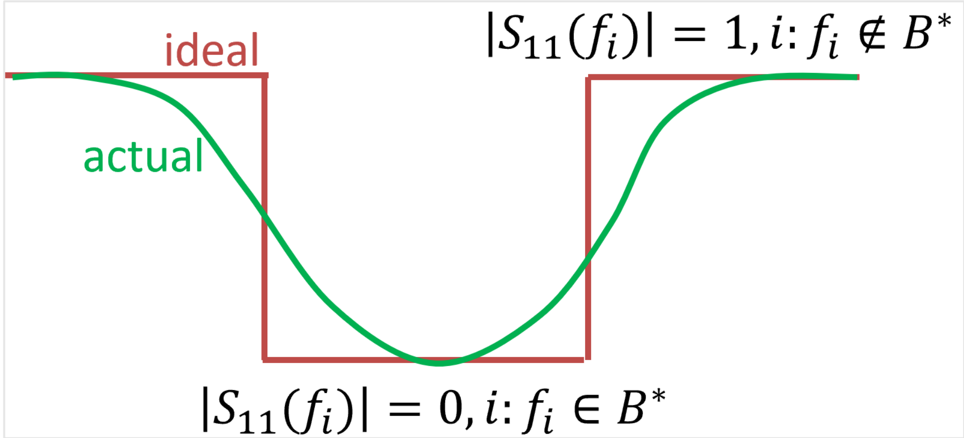
for $k = 0, 1, 2, \ldots,$ *until convergence,* **do**

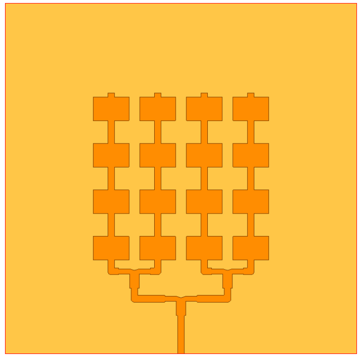
$$\hat{y}^{(k)} = g(x^{(k)}, \theta)$$

$$\mathcal{J}(\hat{y}^{(k)}) = \sum_{i: f_i \in B^*} |\hat{y}_i^{(k)}|^2 + \sum_{i: f_i \notin B^*} (|\hat{y}_i^{(k)}| - 1)^2$$

$$\Delta x^{(k)} = -\frac{\partial \mathcal{J}(\hat{y}^{(k)})}{\partial \hat{y}^{(k)}} \frac{\partial \hat{y}^{(k)}}{\partial \theta} \frac{\partial \theta}{\partial x^{(k)}}$$

Update: $x^{(k+1)} \leftarrow x^{(k)} + \lambda \Delta x^{(k)}$





Evaluation metrics

- Return loss passband
  - Region where the $|S_{11}|$ is lower than -10 dB in the resonant band

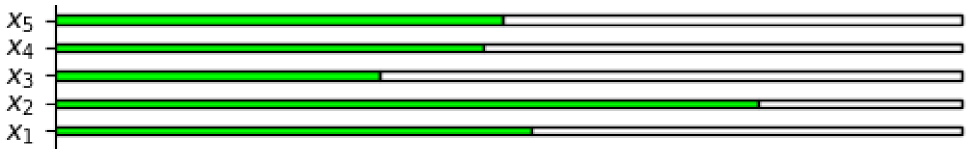
$$B = \{[f_L, f_H]:\ |S_{11}| < -10\ \mathrm{dB}\}$$

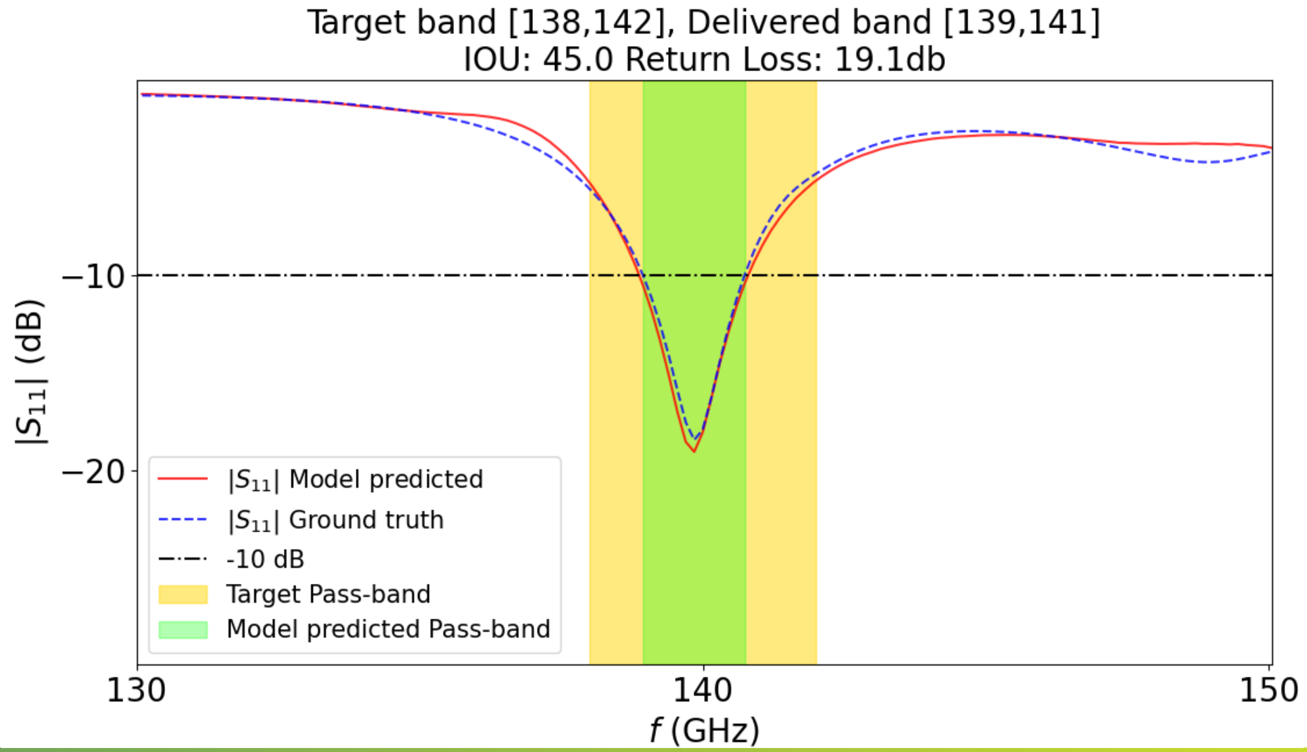
- Intersection-over-Union (IoU)
  - Percentage overlap between the target band and our prediction passband

$$\mathrm{IoU} = \frac{B^* \cap \hat{B}}{B^* \cup \hat{B}}$$

Inverse optimization results

- Given a target band $B^* = [138, 142]$ GHz
  - Optimize to find the design parameters
  - $\hat{x} = \{W_p, L_p, W_{a,f}, L_s, L_{d,f}\} = \{502.7, 789.8, 176.7, 210.3, 340.4\}\ \mu m$
  - Validate with forward design





Performance Summary

- ❑EM simulator: ~617 CPU hours to generate 2500 training data samples
- ❑CDNet: ~2.5 seconds to inference 2500 samples
  - ❑ ~1.8 minutes to train

Table 1. Performance Summary of the Proposed Surrogate Model

Design parameters	Frequency points	Train error (for 2400 samples)	Inference error (for 100 samples)
5	134	0.641 dB	0.357 dB

$$\mathrm{Error(dB)} = \frac{1}{N}\sum_{i=1}^{N}\left|20\log_{10}|y_i| - 20\log_{10}|\hat{y}_i|\right|$$

where y is the ground truth, and $\hat{y}$ is the prediction

Conclusion

- We present both forward and inverse modeling of RF systems using complex-valued neural networks
- Forward modeling gives us a fast prototype of the circuit
- Inverse modeling involves finding the best design parameters to generate a desired response
- Surrogate modeling helps in reducing the design cycle time

THANK YOU!

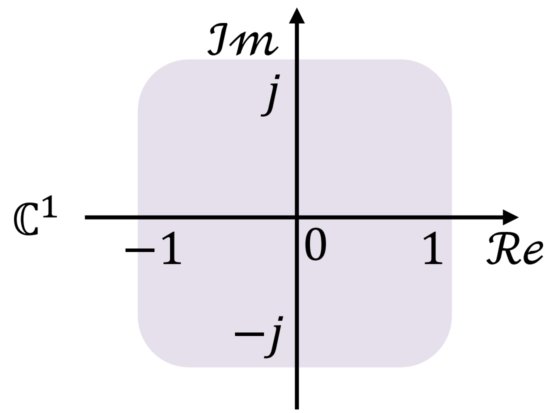
This research is supported in part by

NSF I/UCRC Center for Advanced Electronics

Through Machine Learning (CAEML)

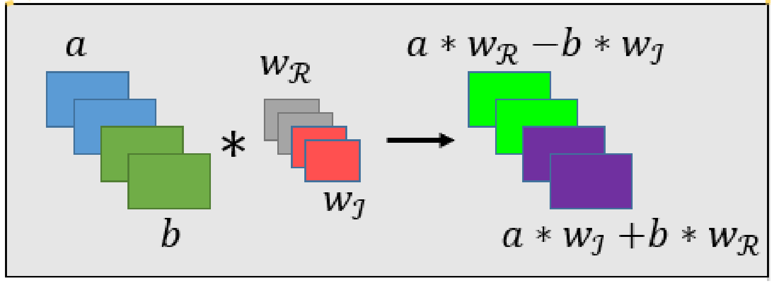
Surrogate Modeling with Complex-valued Neural Nets

- We predominantly train NNs in the real $\mathbb{R}$ domain
- But phase is important too!
- Complex domain $\mathbb{C}$ offers a richer set of numbers
  - Better data representation
    - Mapping for complex-valued NNs $g(z)$: $\mathbb{C}^N \leftrightarrow \mathbb{C}^M$
    - Mapping for real-valued NNs $h(z)$: $\mathbb{R}^{2N} \leftrightarrow \mathbb{R}^M$
  - Higher functionality
    - Weights do not just change amplitude
    - Can be rotated too!
  - Classification capability: A simple perceptron can only learn linearly separable functions
    - XOR: linearly non-separable
    - Single real-valued neuron fails
    - Single complex-valued neuron succeeds



Complex building blocks

- Complex convolution
  - $w *_{\mathbb{C}} z = (a + jb) *_{\mathbb{C}} (W_{\mathcal{J}} + jW_{\mathcal{R}})$
    $= (a * W_{\mathcal{R}} - b * W_{\mathcal{J}}) + j(a * W_{\mathcal{J}} + b * W_{\mathcal{R}})$



- Complex activation
  - $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$
  - $\mathbb{C}\mathrm{ReLU}(z) = \mathrm{ReLU}(a) + j\mathrm{ReLU}(b)$
- Complex residual block
  - Given mapping $T(z)$ from input to output
  - $R(z) = T(z) - z \Rightarrow T(z) = R(z) + z$

Modeling physically consistent response: Passivity of S-parameters

Algorithm 1: Passivity enforcement of S-parameters

Input: S: Predicted complex S-parameter matrix, n: Number of ports, B: Frequency band

Output: S_P: Passive S-parameter matrix

1 Reshape S into a batched matrix form for an n-port network.

2 Transform S into $\tilde{S}$ using isomorphism:

$$\tilde{S} = \begin{bmatrix} \Re(S) & \Im(S) \\ -\Im(S) & \Re(S) \end{bmatrix}$$

3 **for** $i : f_i \in B$ **do**

4 Calculate an upper bound for the largest singular value using:

$$\hat{\sigma}_1(f_i) = \sqrt{\frac{P(f_i)}{n} + \sqrt{\frac{n-1}{n}\left(Q(f_i) - \frac{P(f_i)^2}{n}\right)}}$$

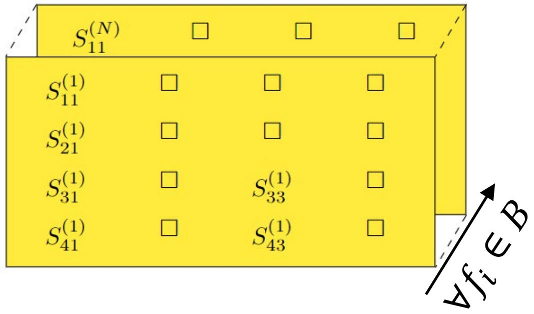
where

$$P(f_i) = \sum_{j=1}^{n} |\tilde{S}_{jj}(f_i)|^2 \text{ and}$$

$$Q(f_i) = \sum_{j,k=1}^{n} \left[\left(\tilde{S}^*(f_i)\tilde{S}(f_i)\right) \circ \left(\tilde{S}(f_i)\tilde{S}^*(f_i)\right)\right]_{jk}.$$

⋮

Step 1



⋮

5 Implement minimum-phase filter as:

$$\Sigma(f_i) = |\Sigma(f_i)| e^{j\phi(f_i)}$$

where

$$|\Sigma(f_i)| = \begin{cases} \frac{1}{\hat{\sigma}_1(f_i)}, & \text{for } \hat{\sigma}_1(f_i) > 1 \\ 1, & \text{for } \hat{\sigma}_1(f_i) \leq 1 \end{cases},$$

$$\phi(f_i) = \mathscr{H}\{\log|\Sigma(f_i)|\}.$$

/* $\mathscr{H}\{\cdot\}$ is the Hilbert transform, operated using a fast Fourier transform approach. */

6 Enforce passivity as:

$$S_P(f_i) = \tilde{S}(f_i) \odot \Sigma(f_i).$$