Supplementary Material: Deep-learning Enabled Generalized Inverse Design of Multi-Port Radio-frequency and Sub-Terahertz Passives and Integrated Circuits

E. A. Karahan, et.al.

Detailed Description of mm-Wave RFIC Datasets and Training Strategy

The table below summarizes the datasets consisting of higher accuracy simulations.

Supplementary Table 1. Training sets corresponding to distinct CNNs

	Frequency Range	Number of Simulations	After Data Augmentation	Training-Validation-Test Split	Number of Epochs
300×300 μm 16×16 2-port	30-100	75K	204K	170K-17K-17K	120
300×300 μm 12×12 2-port	30-100	36K	108K	90K-9K-9K	75
300×300 μm 10×10 2-port	30-100	28K	88K	72K-8K-8K	60
200×200 μm 16×16 2-port	30-100	15K	50K	40K-5K-5K	50
500×500 μm 16×16 2-port	30-100	80K	180K	150K-15K-15K	90
400×400 μm 25×25 3-port	24-80	180K	324K	270K-27K-27K	75
500×500 μm 20×20 3-port	24-80	180K	324K	270K-27K-27K	75
500×500 μm 20×20 4-port Quadrature Symmetric	24-80	60K	60K	55K-5K-5K	20

It is worth noting that performing data augmentation does not increase the dataset diversity as much as performing additional simulations[1]. Hence, for the high pixel counts we rely more on additional simulations rather than geometric

transformations. Datasets detailed above were used for the second stage of the transfer learning. For the following predictors, significantly larger number of lower accuracy (hence faster) simulations were conducted for the initial training:

- 300×300 μm 16×16 2-port: initial dataset consisting of 250K simulations was augmented to 600K.
- 300×300 μm 12×12 2-port: initial dataset consisting of 200K simulations was augmented to 500K.
- 300×300 μm 10×10 2-port: Initial dataset consisting of 80K simulations was augmented to 300K.
- 500×500 μm 16×16 2-port: Initial dataset consisting of 50K simulations was augmented to 125K.
- 400×400 μm 25×25 3-port: Initial dataset consisting of 250K simulations was augmented to 350K
- 500×500 μm 24×24 3-port: Initial dataset consisting of 250K simulations was augmented to 350K

For the simulations detailed above, a simplified stackup setup, consisting of air dielectric with 2 metal layers (pixelated structure and ground plane) was utilized. To obtain the same electrical length for the pixelated structure, target area (for example 300×300 μm) was upscaled by $\sqrt{\varepsilon_{Eff}}$. As the electrical length is similar in both cases, features extracted from the initial training would be useful[2]. The total one-time training time would take a few hours to a few days, depending on the design parameters on the computing system (400 CPU cores) that was available for this work.

To demonstrate the benefit of the proposed training loop, two different training strategy were compared for 300×300 μm 16×16 2-port dataset. In the proposed approach, a randomly initialized network was trained for 40 epochs with the low accuracy simulations. This intermediate network is then retrained with the high accuracy simulations for 120 epochs. In this case resulting test set rms error was found to be 0.50. On the contrary, if we train a randomly initialized network solely with the high accuracy simulations for 160 epochs, test set error was evaluated as 0.54. Note that in both cases CNN architecture is same. We can see that transfer learning helps with the performance improvement that would otherwise require expanding the dataset. Morevoer, with smaller datasets, clearer improvement could be obtained. This is in fact the case for 200×200 μm 16×16 EM structures as the corresponding predictor network was transfer learnt from the CNN model for 300×300 μm 16×16. Similarly, 500×500 μm quadrature symmetric 20×20 4-port predictor, was initialized from the 300×300 μm 12×12 2-port network. This was possible due to the quadrature symmetry imposed on the 4-port structures in the dataset.

We should also add that, simulations reported in these tables correspond to randomized pixelated grids. In order to improve representation capacity of the model, structures that resemble to transmission line networks (as exemplified at Fig. 2-b) were created and simulated for 300×300 μm 16×16 2-port EM data. This extra 38K simulation is then augmented to 130K for the re-training of the corresponding network.

Fig. S1 shows two filter synthesis examples for D-Band. We must emphasize that for the training of D-Band frequency range, we utilized transfer learning from an already existing forward model. As a result, relatively low number of simulations (25K in this case) was sufficient for training. In these examples, target is to pass the single frequency while rejecting other points as much as possible.

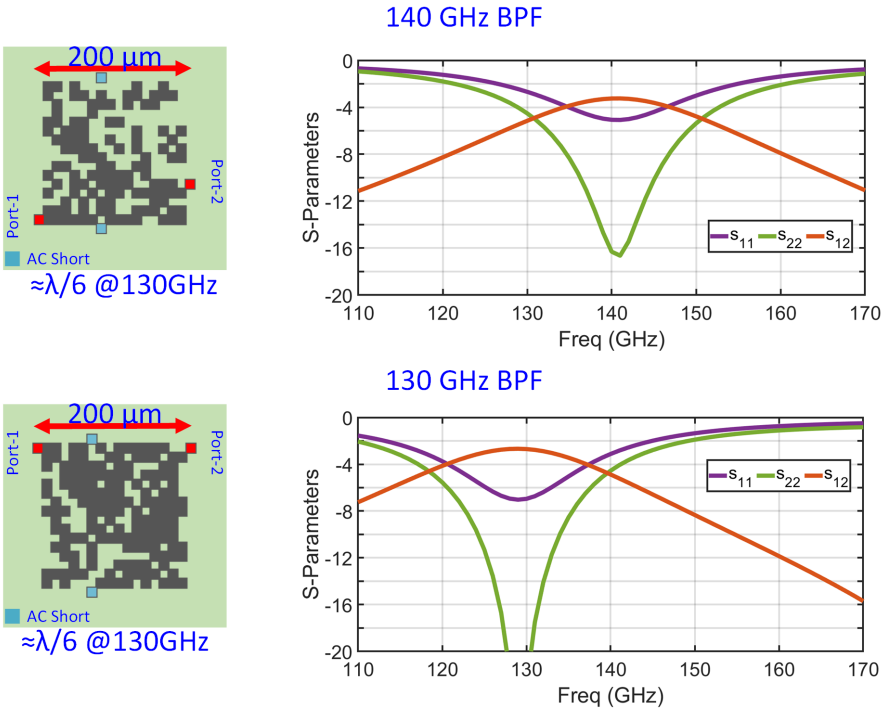

Figure S1. Band pass filter structures synthesized at sub-THz frequencies.

Detailed Description of Antenna Datasets and Training Strategy

The information of antenna performance in air dielectric can be utilized to create robust models for scaled antennas embedded in dielectric through transfer learning. Intuitive reasoning for the transfer learning flow was described in the previous section. This is critically important since, simulations with high accuracy can require 10 times more resources (due to required mesh density). Moreover, a one-time air simulation model can be utilized to create rapid antenna predictor models for different dielectrics through transfer learning.

Due to its rich representation, CNN trained with mm-Wave probe-fed antenna dataset was used as the initial point for transfer learning of other cases. In parallel to this, we refer to this CNN as 'framework network'. The initial training time for the proposed framework model is around 3 hours on a single A100 GPU. Model was trained for 60 epochs. Details for the training was provided at Table 2. Since other s_{11} models were initialized from an already converged CNN, their training requires less data.

As it can be seen, the input to the network is the 12×12 matrix representing the antenna structure and feed location. The output of the forward models is the return loss sampled at 81 frequency points for the mm-Wave antennas. Radiation patterns of mm-Wave antennas were sampled at 11 frequency points. Models were trained to predict $\phi = 0$ and $\phi = 90$ cuts. Each cut sampled from $\theta = 90$ to $\theta = -90$ over 72 points. This totals to 72×2 for each frequency point and 1584 points over the entire range of frequencies.

Frequency responses predicted by the emulator will depend on the statistical distribution of frequency bins. Contradicting this point, emulator is expected to perform well at predicting minority cases, which is impedance matching below -10 dB at multiple frequency bins. Due to randomized dataset generation, good matching and radiation pattern properties at lower frequencies (where the antenna is more compact) are a lot less common. This could result in CNN getting stuck at a local optimum during training. To be precise, CNN model can converge while ignoring the minority samples. This necessitates to construct a dataset that has good representation for minority cases. For this purpose, we use informed dataset generation to skew statistical properties of s_{11} values towards well-matched cases. A semi-optimization during the dataset generation can populate the dataset with larger number of optimum samples. Hereby, search rules similar to genetic algorithm were applied to search for more optimal antenna structures. Description of this search algorithm can be summarized as following.

1. Step I: Calculate the costs of the entire dataset for the frequency bins of interest, i.e. 20 to 25 GHz where antenna structures are compact.

2. Step II: Perform an exponentially distributed sampling by giving samples with lower cost values more probability of being chosen.

3. Step III: Explore the vicinity of selected samples (by introducing random mutations) or perform cross over between different samples to generate a new sample.

The framework network for probe feed antenna dataset consists of a total of 560 K random samples and 500 K informed samples. After geometric transformations, the dataset is split as 10 : 1 : 1 between the training, validation, and testing samples, respectively. To obtain the optimal network with the appropriate number of convolutional and fully connected layers, we sweep the hyper-parameter space. This finalized network is then utilized as the starting point for the remaining antenna predictors.

Dataset	Random Simulations	Informed Simulations	Data Augmentation	Training Size	Validation Size	Test Size	Number of Epochs	Batch Size	Transfer Learning
mm-Wave probe-fed (S_{11})	560K	500K	340K	1400K	150K	150K	60	1400	Framework network
mm-Wave probe-fed (Radiation Patt.)	383K	0	0	345K	25K	13K	75	2500	Initialized from framework network
mm-Wave edge-fed (S_{11})	670K	0	230K	750K	150K	150K	30	1000	Initialized from framework network
mm-Wave edge-fed (Radiation Patt.)	536K	0	0	345K	25K	166K	75	2500	Initialized from framework network

Supplementary Table 2. Summary of the datasets generated for antennas in air.

Dataset	Random Simulations	Informed Simulations	Data Augmentation	Training Size	Validation Size	Test Size	Number of Epochs	Batch Size	Transfer Learning
mm-Wave probe-fed (S_{11})	43K	0	100K	100K	20K	23K	30	1000	Initialized from framework network
mm-Wave edge-fed (S_{11})	40K	0	40K	70K	5K	5K	15	1000	Initialized from framework network

Supplementary Table 3. Summary of the datasets generated for antennas embedded to dielectric.

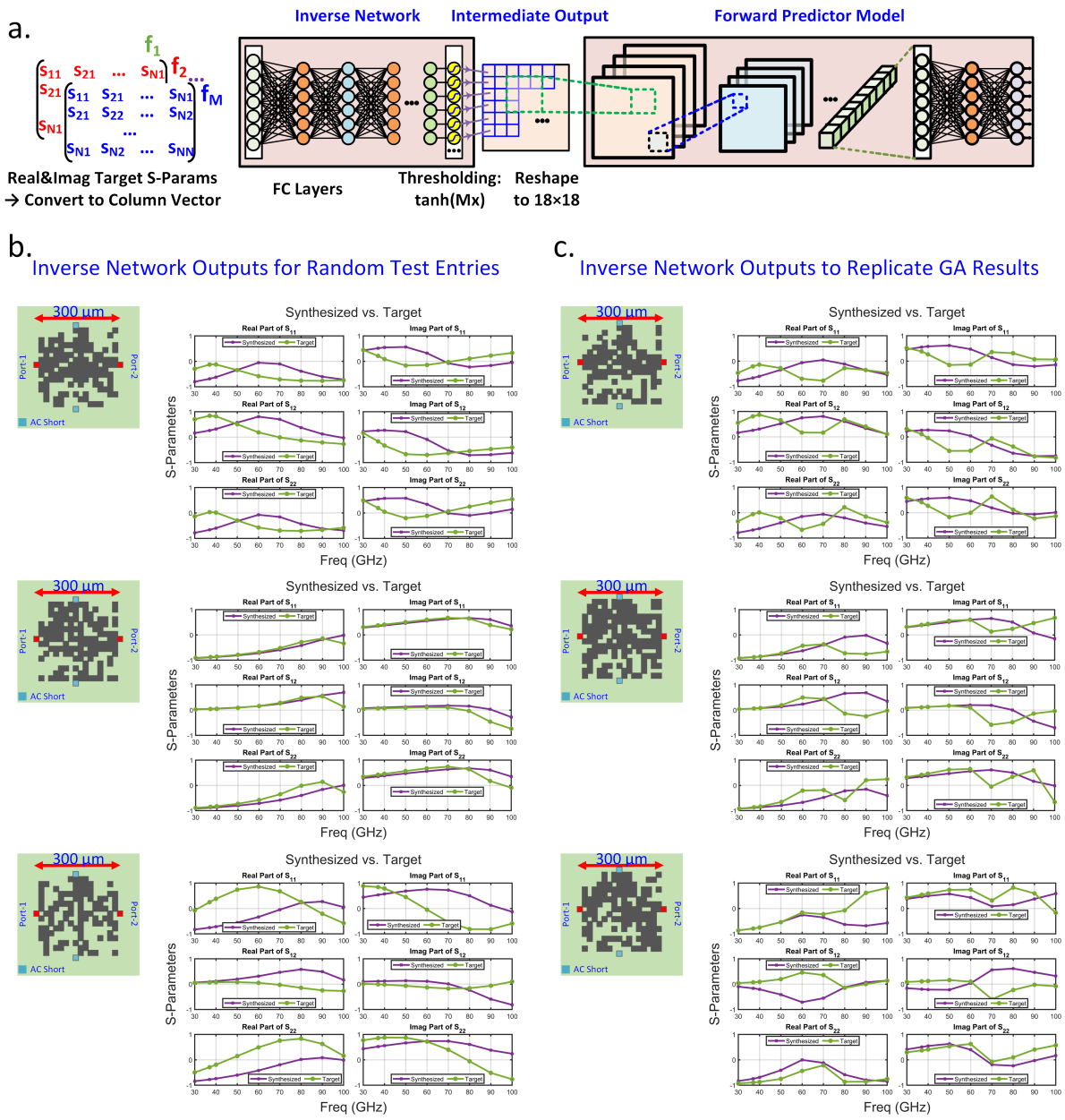

Figure S2. Inverse generative network training and synthesis overview. **a.** Input to the inverse neural network is real and imaginary parts of S-Matrix terms. Here, in order to train the inverse network, intermediate output, which is an 18×18 matrix denoting the pixelated structure, is fed to the forward model. Error and gradients were calculated using the difference between predicted and targeted S-Parameters. However, only the weights of inverse network is updated. **b.** Inverse network outputs for the S-Parameter entries selected from the test set. **c.** Inverse network outputs for the S-Parameters of GA synthesis results. While there is some success with the inverse network, poor generalization ability causes most of the results to fall short of desired specification.

We demonstrate a CNN based forward model for prediction of scattering parameters of arbitrary shaped passive

EM structures, and an approach to inverse synthesis utilizing this forward model. One can use this forward model with a generative AI framework or with RL approach for synthesis. Both of these methods relate to the design, and not to the modeling aspect of it, and therefore, can be used with the CNN-based forward model. In prior work[3], it was shown that a tandem neural network approach, inspired from a variational autoencoder approach, can perform synthesis of single-port antennas. Extending to much more complex multi-port structures that interact with active devices, however, requires far more extensive generalization capabilities. This might require considering physics-informed architecture. Here, we show some examples to demonstrate the challenges of the classical generative framework.

In order to provide a benchmark, we have created an inverse neural network that aims to instantly synthesize a structure that can approximate inputted S-Parameters. As the test case, 2-port structures within the frequency range of 30-100 GHz is considered. Fig. S2-a demonstrates the architecture and training scheme of the inverse generative network. Training of inverse network is done with the help of forward predictor model. Input to the inverse network is real and imaginary parts of S-Matrix elements. Input layer is followed by 6 fully connected layers. While activation function is leaky ReLU in other layers, in order to perform binary thresholding, last layer activation function is $tanh(Mx)+0.5$. Here, parameter M is increased after each 10 epochs. Output of this layer (a vector of length 324), is reshaped to 18×18 matrix, which is the proper input format to the forward model. Forward model prediction is carried out with respect to the 18×18 input. Difference between predicted S-Parameters and targeted S-Parameters is used in back-propagation. It should be noted that only the inverse network is updated while weights of the forward model is kept frozen.

An important challenge is to avoid forward model getting 'tricked' by the generative inverse network. For example, generative network can output an 18×18 matrix consisting of non-discrete values, which is not representative of a pixelated grid type of structure. While a hard thresholding function can ensure that output of the inverse network is solely 1s and 0s, performing back-propagation proved to be unstable due to challenges of defining derivatives in such cases. Moreover, during the initial training steps, it is plausible to relax the binary output requirement in order to boost inverse network training. We approached this by changing the value of M in the thresholding function at every 10 epochs, M was assigned as 1 for the first 10 epoch and adjusted to 4 and 8 in later steps. On the other hand, during the deployment of inverse network, hard thresholding was performed at the intermediate output.

Fig. S2-b illustrates deployment example outputs for the S-Parameters selected from the test set. While inverse network tries to generate a structure that attempts to approximate the desired S-Parameter input, the achieved results often fall short of the target values. Additionally, in Fig. S2-c, S-Parameters of an actual synthesized structure with the deep-learning enabled GA methods are used as input to the inverse network. The first example is a 70 GHz band stop filter, and latter two are 70 GHz band pass filters. In most of these synthesis methods, the achieved results fail to approximate the targeted values. These examples, while not being comprehensive, demonstrate that the inverse

network may not able to generalize well. In general, generative networks are a lot more data intensive to train and prone to mode collapses and training stagnation. We should note that generative AI models, such as transformers and diffusion models are notoriously more data and training intensive, but can generalize better and can be explored further.

Genetic Algorithm Heuristic Mechanisms

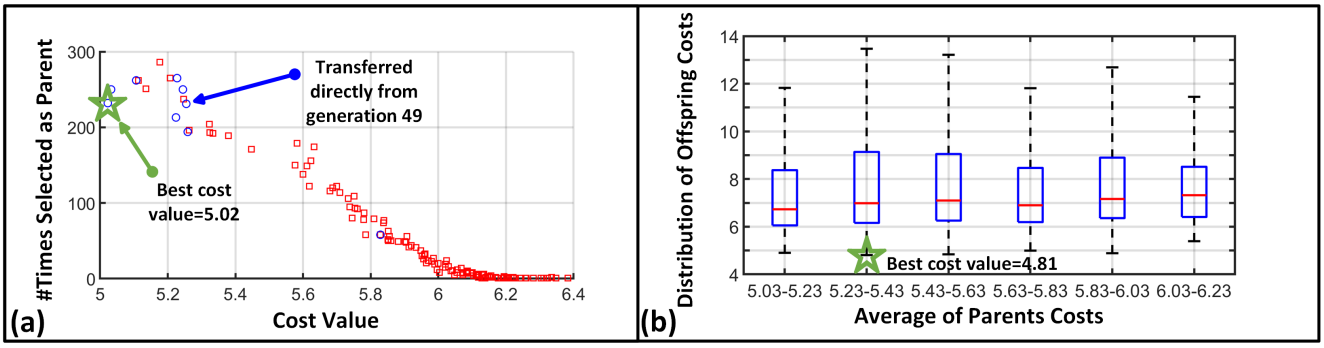

Figure S3. Snapshot from the 50th iteration from the optimization of a 2-port EM structure. **a.** Parent selection was performed separately for each of the next generation members. Due to the selection scheme, low cost samples have a good chance of being selected as parents frequently. Moreover, a diverse set of parents were selected. This contributes to the exploratory aspect of the optimization algorithm. **b.** Average cost value of parents versus cost values of offspring. The best cost value for successive generation is improved in comparison to the current generation. Interestingly offspring cost minimization does not exactly correlates with the parents'. This is in fact due to non-convexity of the design space. In the box plot, median, 25th and 75th percentiles are highlighted with red lines and blue boxes respectively. Whiskers indicate 1.5× interquartile range.

It may not be immediately evident that heuristic approach implemented with the GA can result in cost minimization. In fact, due to non-convex nature of the design space, we cannot guarantee that each and every offspring are going to have better cost value than the parents. However, crossover and mutation operations resemble the evolution process in nature. Thanks to this heuristic, the cost function gets better asymptotically.

To give better justification, we can show a snapshot from the 50th iteration of an optimization loop (for a 2-port network) in Fig. S3. Fig. S3-a plots cost value versus number of times a sample was chosen as parent. 8 inherited members from the 49th iteration were also annotated. We can observe that there is a diverse parent selection. In addition, we see that from generation 49 to generation 50, several population members that outperform best 8 members of generation 49 were obtained and generation 51 will still have 4 members from generation 49 thanks to their success.

Fig. S3-b plots the distribution of the cost function for the 51st iteration with respect to the cost values of parents. Compared the 5.02 value reported at (a), minimum cost is now 4.81, indicating the success of heuristic approach. Interestingly, minimum cost function is achieved at the 2nd bin even though average cost of parents are not minimum. This showcases the importance of exploratory strategy.

Example Cost Functions

A set of example cost functions are detailed as the following.

Filters

$$C = \sum_{i=1}^{N} a_1(i) \times |s_{21}(i)| + a_2(i) \times [1 - |s_{21}(i)|]^2 \tag{1}$$

For the filter design problem pass-band insertion loss and stop-band rejection are considered (Eq. 1). By adjusting the a_1(i) and a_2(i) across the frequency, band pass and band stop behaviors can be implemented. Also, to allow for a transition band, band edges can be ignored by assigning a_1(i) and a_2(i) to 0.

Power Divider

$$C = \sum_{i=1}^{M} a_1(i) \times |s_{32}^{\sim}(i) - |s_{32}^{*}(i)|| + a_2(i) \times |s_{31}^{\sim}(i) - |s_{31}^{*}(i)|| + a_3(i) \times \left| \mathrm{Arg}\left(\frac{s_{31}^{*}(i)}{s_{32}^{*}(i)}\right) - \varphi(i) \right| + a_4(i) \times \left| \frac{s_{32}^{\sim}(i)}{s_{31}^{\sim}(i)} \left| \frac{s_{31}^{*}(i)}{s_{32}^{*}(i)} \right| - \frac{s_{31}^{\sim}(i)}{s_{32}^{\sim}(i)} \left| \frac{s_{32}^{*}(i)}{s_{31}^{*}(i)} \right| \right| \tag{2}$$

In the Eq. 2, s_{31}^{*} and s_{32}^{*} denotes the predicted complex valued S-Parameters. To optimize a power divider (possibly with filtering characteristics) we assign frequency dependent targets for the absolute value of s_{32} and s_{31} (denoted as $s_{31}^{\sim}$ and $s_{32}^{\sim}$). Hence, the first 2 terms of the equation penalizes the deviation from target value. The second consideration is the phase difference between 2 output ports. The third term in the equation penalizes the deviation from the frequency dependent target phase value, which is shown as $\varphi(i)$. Lastly, we observed that adding the 4th term improves the amplitude balance since it is minimized by ensuring the ideal amplitude ratio. Any of the a_1(i)-a_4(i) could be assigned as 0 to ignore the corresponding specification or a range of frequencies. For example, in the case of filtering divider, amplitude and phase balance terms were ignored outside of the pass-band.

Example Comparison With the Traditional Design Method

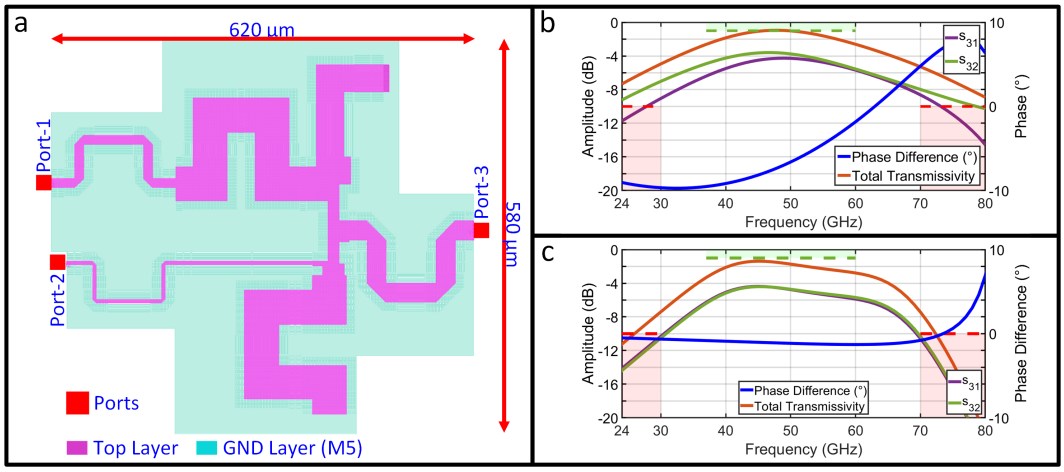

Figure S4. **a.** Layout of a transmission line based filtering divider. The classical method usually involves schematic/template optimization followed by layout level tuning and EM verification. **b.** EM simulation results for the classical filtering divider. **c.** Solution found by inverse design. We can see that the filtering response provided by the inverse design is sharper.

In Fig. 4-b a 3-port EM structure that simultaneously performs filtering and equal power division was synthesized using inverse design method. To compare this with traditional optimization, consisting of transmission line elements provided with the 90 nm BiCMOS PDK was designed on the schematic and layout level. While translating the schematic to layout, bends and meander lines were utilized to reduce the footprint of the circuit. This in turn required additional manual tuning. Finally resulting layout was simulated with EMX. Fig. S4-(a-b) demonstrates the final layout and EM simulation result. While this design approach takes 5-6 hours for a circuit designer, inverse design is able to find a solution within a few minutes, whose performance was shown on Fig. S4-c. It should also be emphasized that, neither inverse nor traditional design is assumed to be the global optimal. Rather, this comparison aims to illustrate time spent in both cases to optimize a circuit block. Furthermore, in terms of chip area, inverse design was able to find a more compact solution.

Measurement of Quadrature Coupler

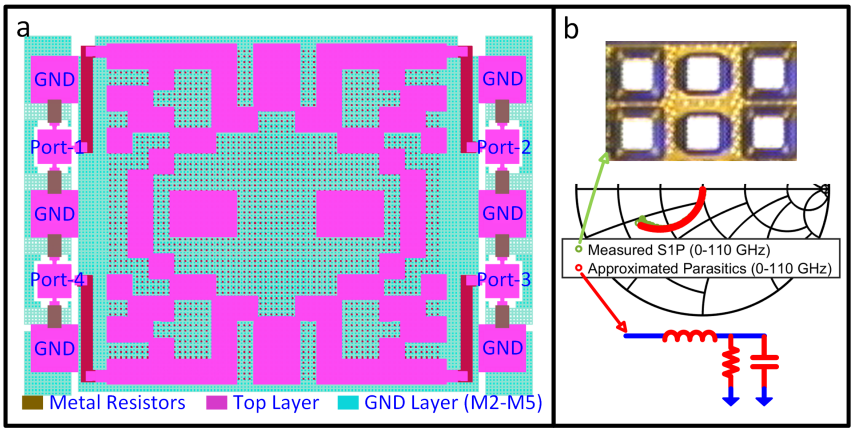

Figure S5. **a.** Layout of the quadrature coupler at Fig 6. As it can be seen, a 50Ω resistor was placed in parallel to each port. While this is helpful for performing 2-port measurements (since the other 2 ports are already terminated), parasitics of the resistor can create non-idealities in the measurement. **b.** A 50Ω resistor test structure was placed and measured to characterize associated parasitics. After performing the measurement, the parasitic LC network and termination R were de-embedded.

With our current measurement capabilities, full 4-port on-wafer measurement is challenging to perform. To work around this issue, 50Ω terminations were put in parallel to each of the ports. As a result, while performing 2-port measurement between input (port-1) and output (port-2 and 3) remaining ports are already terminated with 50Ω. Excess 50Ω resistors can then be de-embedded from this measured 2-port S-Parameters.

Fabrication of mm-Wave Antennas

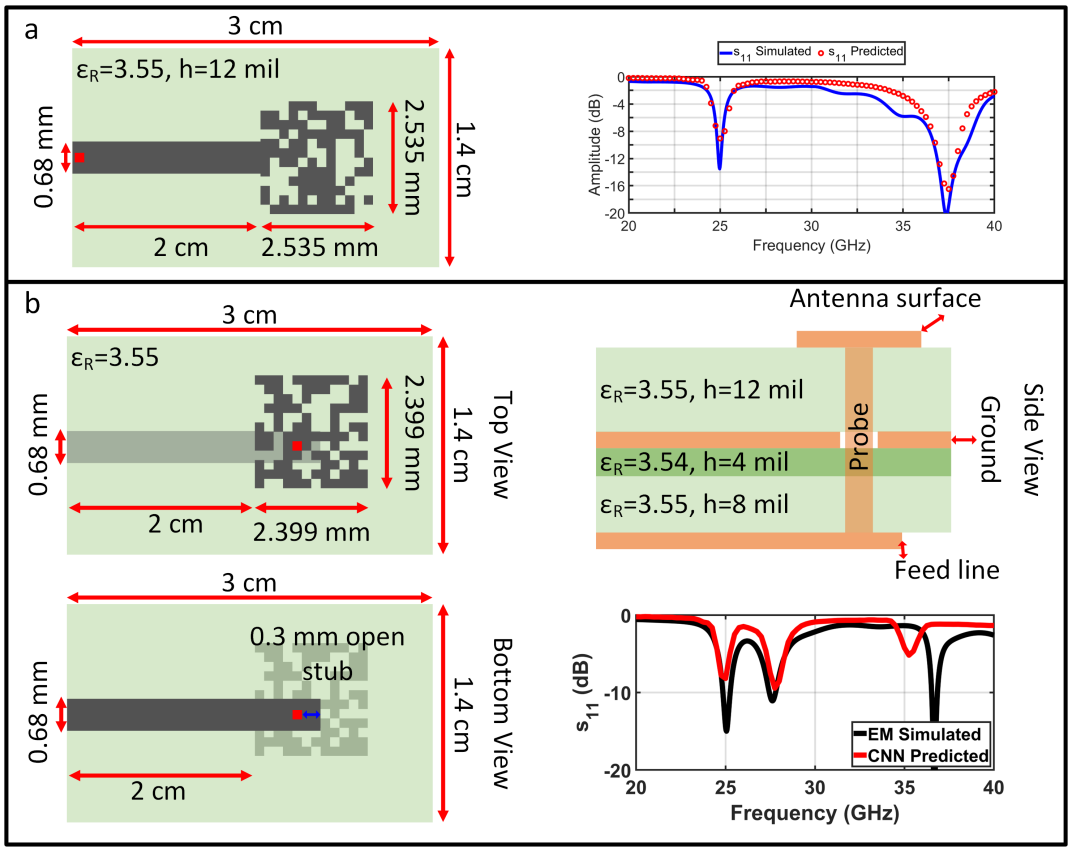

Figure S6. Practical aspects of mm-Wave antenna manufacturing. **a.** For edge feed mm-Wave antennas a 50Ω line and edge feed connector were used on the antenna layer. **b.** On the contrary, probe feed antennas requires a feeding network that is underneath the antenna layer. While a 50Ω line is utilized in the bottom copper, a probe needs to connect between the feed line and antenna excitation location. Addition of this probe can incur extra reactance, which was cancelled with a short open circuited (or shunted) stub.

Feed network for edge feed mm-Wave antennas consists of a 50Ω transmission line. To reduce proximity effects of PCB edge feed connector, a 2 cm line ($= 2\lambda_0$ at 30 GHz) is put in place as shown in Fig. S6-a. On the other hand, probe feed antennas need an extra layer to excite antenna from the desired location (Fig. S6-b). Addition of this feeding via can add extra reactance to antenna. This was cancelled by adding an open or shorted stub line in parallel to feeding point.

Antenna Measurement Setup

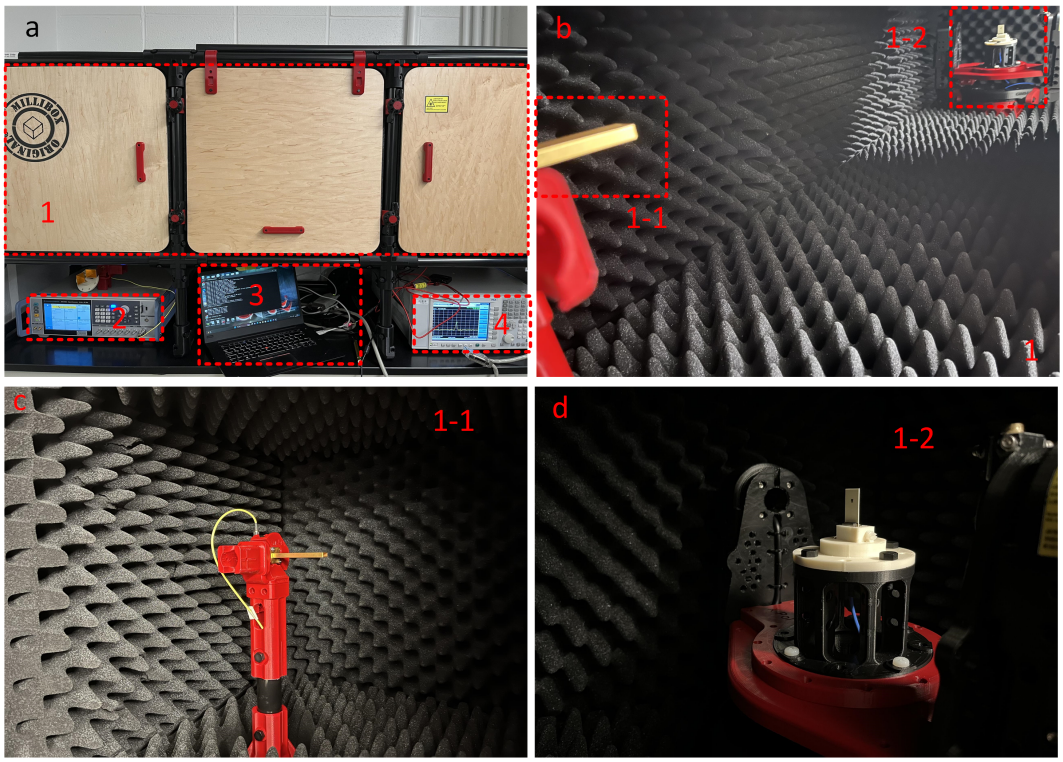

Figure S7. Radiation pattern measurement setup for mm-Wave antennas. **a.** Measurement setup components. We use compact anechoic chamber MBX03 Millibox (1) a 67 GHz signal generator (2) and 50 GHz spectrum analyzer (4) to measure DUT. Motor was controlled with Millibox software (3). **b.** Interior of MBX03. **c.** Horn antenna (WR42 open ended waveguide or WR28 horn antenna to cover mmWave range) and its mount. **d.** 2D rotational hardware (Gimball) where DUT is mounted. To fix DUT, a custom 3D printed holder is designed.

The S-Parameter measurements of the antennas were performed with Anritsu Vector Network Analyzer (MS4647B). Fig. S7 illustrates the radiation pattern measurement set up for mm-Wave antennas. 18-26.5 GHz were covered with a WR42 open ended waveguide while between 26.5-40 GHz, WR28 horn antenna was used. A customized holder that can bring DUT antennas to rotational axis was developed. Spectrum analyzer and signal generator were used concurrently to measure received power at different angles.

Detailed Analysis of Broadband mm-Wave PA Circuit

Power combining is often needed in power amplifiers to generate high power. A classical approach to designing such power combining architecture is shown in Fig. S8-a. The input signal gets split into two identical paths, gets amplified, and the output power gets combined symmetrically at the output port. The input splitter and output combiner are co-designed with the active devices to ensure optimal matching for maximum power transfer. One key observation is that the two paths being identical, all the voltage nodes at each of the two identical paths is the same. As a result, in spite of having a second order network in each path of the combiner (i.e. a fourth order network in total), the combiner acts as a second-order network and therefore, limits the bandwidth.

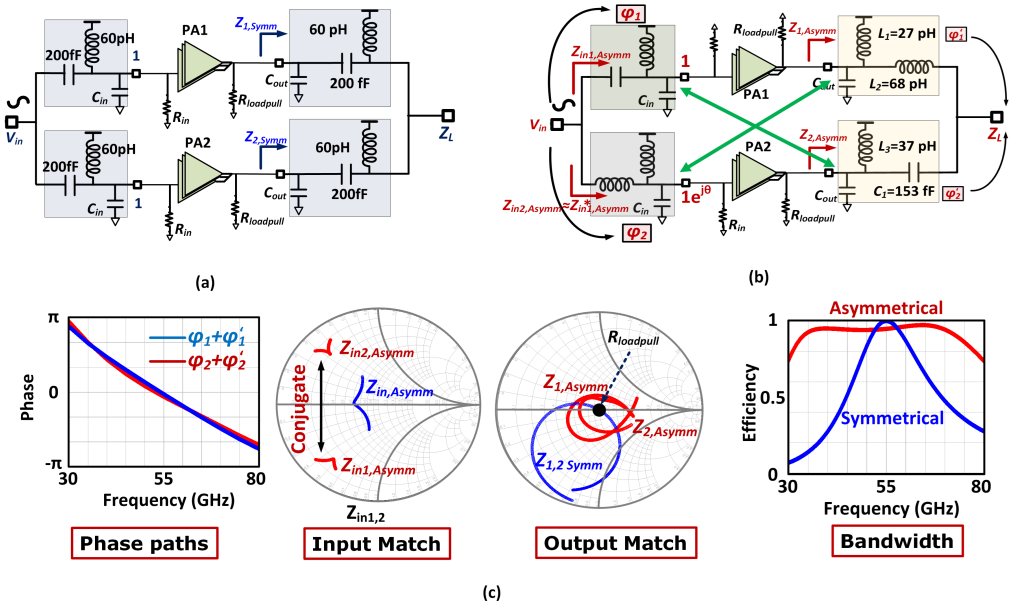

Figure S8. Asymmetrical two-way combining circuits allowing broader bandwidths compared to symmetrical circuits. **a.** Classical symmetrical two-way power combining power amplifier. **b.** Asymmetrical two-way power combining power amplifier with different input and output matching at each branch. **c.** Simulated results of the symmetrical and asymmetrical circuits showing bandwidth extension of the asymmetrical one. Even with asymmetry, the phase path $\phi_1 + \phi_2$ in the top path matches exactly the one in the bottom path $\phi_1' + \phi_2'$. This figure was adapted from "Chappidi, C. R., Sharma, T. and Sengupta, K. Multi-port Active Load Pulling for mm-Wave 5G Power Amplifiers: Bandwidth, Back-off Efficiency, and VSWR Tolerance,"[4].

In[4], we demonstrated an approach where we break the symmetry of the two paths as shown in Fig. S8-b. The input splitter and output combiner in each of the two paths are not identical. Therefore, the voltages, currents at each nodes in the two paths are not identical, and the two paths cannot collapse into an equivalent one path. This results in a quasi-fourth order behavior at the input and output, which results in a broader bandwidth compared to the symmetrical case. The two different paths, however, need to be designed in such a way that the input signals, while

splitting into two identical paths, experiences the same exact gain and collective phase shifts as they finally combine at the output. When such a phase condition is maintained (Fig. S8-c), the bandwidth can be extended significantly without additional losses. However, there is no systematic approach to realizing these multi-port networks with asymmetric frequency-dependent S-parameters.

In this work, the AI-model and the synthesis approach determines this by itself. As shown in Fig. S9 the synthesis generates the asymmetric input and output networks while matching the total collective phase shifts between the paths. Fig. S9 provides detailed schematic of the multi-port mm-Wave amplifier that was implemented in 90 nm BiCMOS process. For the optimization of output combiner, frequency dependent RC values corresponding to optimum termination impedances were calculated and stored in a lookup table. During the optimization, these parasitic capacitances (arising from the device and output pad) were lumped to the CNN predicted S-Parameters of the pixelated structure. Total transmissivity across the target frequencies were then calculated by assuming optimum phase and equal amplitude excitation. As a secondary objective, presence of a DC path between the input ports and AC shorting location was ensured.

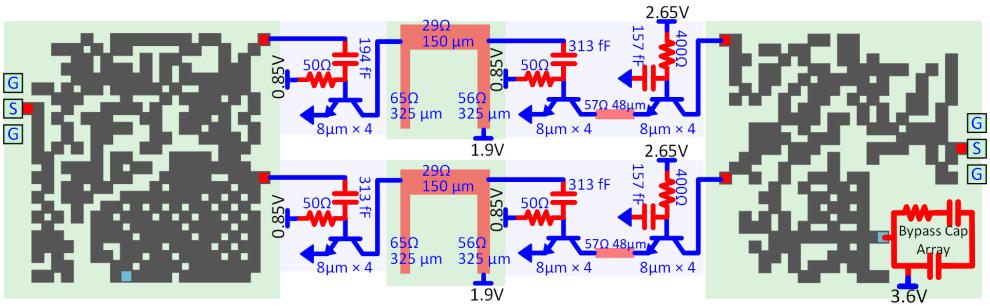

Figure S9. Detailed schematic of the circuit.

References

1. Yang, S. *et al.* Image data augmentation for deep learning: A survey (2023). 2204.08610.

2. Karahan, E. A., Liu, Z. & Sengupta, K. Deep-learning-based inverse-designed millimeter-wave passives and power amplifiers. *IEEE J. Solid-State Circuits* **58**, 3074–3088, DOI: 10.1109/JSSC.2023.3276315 (2023).

3. Gupta, A., Karahan, E. A., Bhat, C., Sengupta, K. & Khankhoje, U. K. Tandem neural network based design of multiband antennas. *IEEE Transactions on Antennas Propag.* **71**, 6308–6317, DOI: 10.1109/TAP.2023.3276524 (2023).

4. Chappidi, C. R., Sharma, T. & Sengupta, K. Multi-port active load pulling for mm-wave 5g power amplifiers: Bandwidth, back-off efficiency, and vswr tolerance. *IEEE Transactions on Microw. Theory Tech.* **68**, 2998–3016 (2020).