Residual Network Based Direct Synthesis of EM Structures: A Study on One-to-One Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose using machine learning models for the direct synthesis of on-chip electromagnetic (EM) passive structures to enable rapid or even automated designs and optimizations of RF/mm-Wave circuits. As a proof of concept, we demonstrate the direct synthesis of a 1:1 transformer on a 45nm silicon on insulator (SOI) process using our proposed neural network model. Using pre-existing transformer s-parameter files and their geometric design training samples, the model efficiently predicts target geometric designs based on desired circuit specification.

1 Introduction

RF/mm-Wave circuits are often governed by the design/performance form factor of the passive components/networks used. Passives are extensively used for impedance matching, scaling, tuning, filtering, power combining/splitting, and signal generation (Figure 1). Therefore, maximizing passive structures' performance while minimizing their form factor is critical for RF/microwave designs. This process is very iterative and requires extensive EM design background to arrive at a faster, optimal solution. The fundamental reason of this existing iterative and computationally inefficient design flow is that most existing EM simulation software suites only act as "analysis tools" (Figure 2). These software suites (such as ANSYS's HFSS) take long simulation time, yet only analyze EM passive structures with given geometries and then yield their circuit performance parameters. The designers are required to do multiple remodeling before optimal geometry is found, with time spent and number of geometrical parameters increase exponentially as the complexity of the passive structure increases. Instead, designers need "synthesis tools" which could directly generate the passive geometries based on the required circuit specification (i.e., inverse engineering of the circuit specification for circuit geometries).

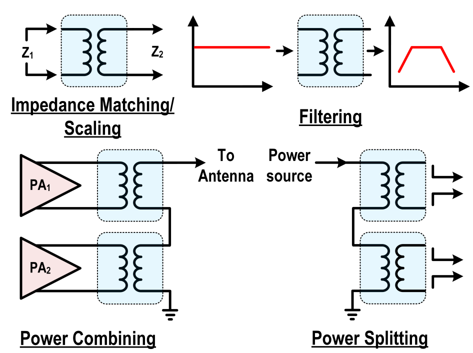


Figure 1: General uses of transformers in RF/mm-Wave Design

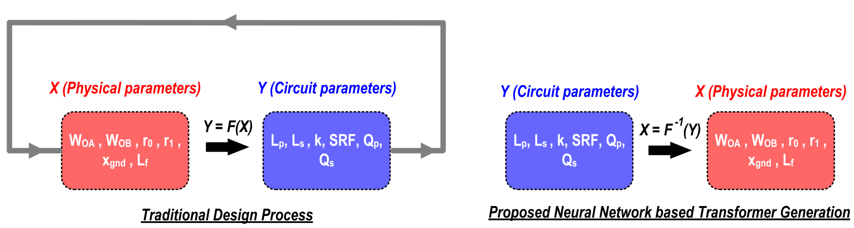


Figure 2: Comparison of the existing iterative cycle for designers in which they iteratively tune their passive structures' geometry based on EM simulations and our neural network predictive model which gives the optimal geometry based on the desired circuit parameters.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

2 EM Passive Structure Design Flow

On-chip transformers are used extensively in RF/mm-Wave designs, particularly for the upcoming 5Gcommunication. Specifically, 1:1 transformers are often used at mm-Wave due to their compact form factor, large achievable coupling, and broadband impedance transformation properties [7]. Therefore, we propose a machine learning based predictive model (Figure 2) for the direct EM synthesis of 1:1 transformers, which will generate the desired transformer design parameters, including the coil radiuses (r_0 and r_1), widths (W_{OA} and W_{OB}), ground spacing (x_{gnd}), and input/output feed length (ℓ_{f}), based on the targeted circuit parameters including the self-resonance frequency (SRF), primary and secondary inductance (L_{p} and L_{s}), coupling coefficient (k), and primary/secondary quality factor (Q_{p} and Q_{s}). See Figure 4 for the synthesized transformer structure, input parameters, and output geometrical parameters.

Our EM predictive model is built upon residual network architectures. Neural networks are known for their predictive power: they can provide a highly close fit to new data after training. Empirical results also suggest that overparameterized neural networks (the number of free parameters exceeds that of training data points) are easy to train, and surprisingly retain appealing predictive performance. More recently, residual networks [3] further ease the training, and enhance the prediction by allowing direct interactions between inputs and outputs. We train the model using a limited number of transformers s-parameter files and their geometric designs. *We use s-parameter files instead of measuring fabricated transformers since doing so would only verify the accuracy of the EM simulator instead of our algorithm.* When given targeted electrical parameters, the neural network outputs geometric designs which act as a starting point to close in on an optimal solution, hence acting as the "EM synthesis tool". Thus, the designers only need very few additional EM simulations to verify and fine tune the design parameters, which circumvents the tedious and resource intensive, iterative process.

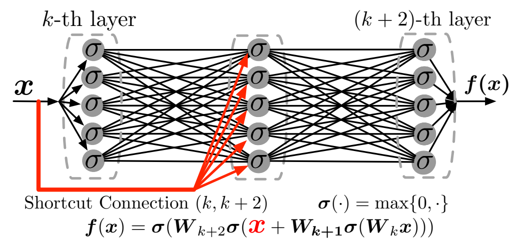
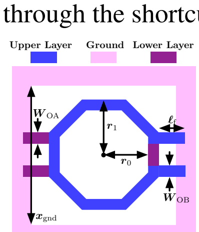


Figure 3: Illustration of residual block with shortcut connection $(k, k+2)$. The input of the k-th layer is directly added to the input of the $(k+2)$-th layer.

3 Residual Network Architecture and Training

The residual network architecture consists of a series of residual blocks. Each residual block is built upon a feedforward neural network by adding shortcut connections across layers. Figure 3 illustrates a residual block with a shortcut connection bypassing two hidden layers. Residual neural networks can be viewed as an ensemble of feedforward neural networks containing varying and adaptive numbers of layers. This largely improves the modeling ability of residual networks. More importantly, for general feedforwad neural networks, simply stacking layers does not promise a performance boost. One of the major reasons is that the vanishing/exploding gradient issue arises which makes the network very difficult to train [1]. However, the residual network architecture mitigates this error through the shortcut connections across layers with additional performance boosts.



Electrical Parameters (EM)	Definition
k	Coupling Coefficient
L_{p}	Inductance of Lower Coil
L_{s}	Inductance of Upper Coil
SRF	Self Resonant Frequency
Q_{p}	Quality Factor of Lower Coil
Q_{s}	Quality Factor of Upper Coil

Electrical Parameters (EM)	Definition
r_0	Radius of Lower Coil
r_1	Radius of Upper Coil
W_{OA}	Width for Lower Layer
W_{OB}	Width for Upper Layer
x_{gnd}	Ground Spacing
ℓ_{f}	Feed Length

Figure 4: Transformer with corresponding input circuit parameters and output geometrical parameters.

The training of neural networks can be written as minimizing the following penalized empirical loss:

$$\min_\theta \; L(\theta) = \Phi_n\left(\{f_\theta(x_i), y_i\}_{i=1}^n\right) + \frac{w}{2}\|\theta\|_2^2, \tag{1}$$

where $f_\theta(x_i)$ is the predicted output of neural network models, i.e., 1:1 transformer's geometric parameters, in this case. Here θ denotes the weight parameters, Φ_n is a properly chosen loss function and the subscript n emphasizes the dependence on n samples, and $R = \frac{w}{2}\|\theta\|_2^2$ is a penalty to avoid overfitting with the tuning parameter $w > 0$. Here (x_i, y_i)'s are samples with x_i denoting input (circuit parameters) and y_i denoting targeted response (geometric parameters), and n is the sample size. In practice, Φ_n are often chosen as an average of empirical errors, e.g., $\Phi_n = \frac{1}{n}\sum_{i=1}^n (f_\theta(x_i) - y_i)^2$ corresponds to the mean squared error. We apply the SGD-type algorithm to solve (1).

4 1:1 On-chip Transformer Direct Synthesis Demonstration

• **Experiment Setup**. We evaluate our predictive models on a 1:1 transformer design task using three residual network architectures. We also compare the predictive models with three baseline methods: linear regression (LR), gradient boosting (GB) [2], and feedforward neural network (FN). The network configurations are listed in Table 1. Note that feedforward neural network models FN_i and residual network models $\mathcal{N}_i$ for $i = 5, 6, 7$ have the same total number of parameters.

Table 1: Feedforward and Residual Neural Network Architecture

Model	Width of Hidden Layers	Shortcut Connections
FN_i	$\{2048 \times i\}, i = 2, \ldots, 7$	NA
$\mathcal{N}_5$	$\{2048 \times 5\}$	(1,5)
$\mathcal{N}_6$	$\{2048 \times 6\}$	(1,6)
$\mathcal{N}_7$	$\{2048 \times 7\}$	(1,3), (3,5), (5,7)

During the training, geometric and circuit parameters of randomly selected pre-solved on-chip 1:1 transformer designs are used as the input data, which we standardize before feeding to the neural networks. We use Adam [4] as our optimizer, one of the most widely used SGD type algorithms for training neural networks. Adam enjoys faster convergence in practice by using adaptive learning rate and momentum acceleration. In our experiments, we randomly select 16 samples from the training set at each iteration to form our mini-batch.

• **Model Comparisons**. We perform extensive comparisons on the prediction accuracy of different models. We use EM simulators to obtain 6400 pairs of 1:1 physical transformer parameters and their corresponding circuit parameters. We randomly select a testing set consisting of 1200 samples, and vary the size of training set in $\{600, 1200, 2400, 4800\}$.

We use two different training loss metrics: 1) Scaled Mean Squared Error (SMSE)

$$\text{SMSE}(\theta) = \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} \left(\frac{y_{i,j} - \hat{y}_{i,j}(\theta)}{y_{i,j}} \right)^2. \tag{2}$$

2) Scaled Dimensional Mean Squared Error (SDMSE) [6]

$$\text{SDMSE}(\theta) = \frac{1}{k} \sum_{j=1}^{k} \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_{i,j} - \hat{y}_{i,j}(\theta)}{y_{i,j}} \right)^2}. \tag{3}$$

In the above, $\hat{y}(\theta) = f_\theta(x_i)$ denotes the predicted geometrical parameters (totally k parameters). Note that SMSE minimizes the relative error of each prediction, which accounts for the different scales of physical parameters and stabilizes the training. Moreover, SDMSE puts additional emphasis on balancing the prediction error across testing samples. Our training objective $L(\theta)$ is (1), where Φ_n takes SMSE or SDMSE defined before.

We observe that residual networks consistently outperform other models on both evaluation criteria, when varying the size of training set. We summarize the experimental results corresponding to using 2400 training samples in Table 2.

Table 2: Performance of Predicting Geometrical Parameters using Circuit Parameters

Model	SMSE Loss		SDMSE Loss	
	SMSE	R^2	SMSE	R^2
LR	0.0358	0.5620	0.0358	0.5620
GB	0.0199	0.7468	0.0199	0.7468
FN_2	0.0114	0.7553	0.0054	0.9070
FN_3	0.0070	0.8457	0.0040	0.9360
FN_4	0.0065	0.8410	0.0038	0.9400
FN_5	0.0061	0.8183	0.0037	0.9431
FN_6	0.0079	0.8040	0.0036	0.9432
FN_7	0.0080	0.8061	0.0037	0.9051
$\mathcal{N}_5$	0.0051	0.8846	0.0030	0.9535
$\mathcal{N}_6$	0.0049	0.8868	0.0031	0.9553
$\mathcal{N}_7$	0.0043	0.9243	0.0030	0.9586

For all neural network models, using SDMSE as training loss improves the testing accuracy compared to SMSE. It is shown that residual neural networks yield superior performance compared to feedforward neural networks consisting of the same number of weight parameters: consistently better prediction accuracy and less sensitivity to the training loss we choose. In addition, adding more layers to the feedforward network does not improve the prediction accuracy. The feedforward network FN_7 shows substantial performance degradation compared to its shallow counterparts FN_5 and FN_6. These observations indicate that shortcut connections play a crucial role in the superior performance of residual networks.

We also observe that $\mathcal{N}_7$ consistently achieves the highest prediction accuracy. It is worth mentioning that the performance gap between residual networks $\mathcal{N}_7$ and $\mathcal{N}_5$ or $\mathcal{N}_6$ are more significant when we use SMSE as the training loss. Note that $\mathcal{N}_5$ and $\mathcal{N}_6$ both contain only one shortcut connection linking the input layer directly to the output layer, while $\mathcal{N}_7$ is equipped with more sophisticated shortcut connections. This observation indicates that a careful design of shortcut connections in $\mathcal{N}_8$ can achieve a significant performance boost.

• **Further Experiments on Residual Networks**. We further present more comprehensive experimental results for the residual network $\mathcal{N}_7$. As we have observed in Table 2, using SDMSE as the training loss improves the prediction accuracy compared to SMSE. Thus, we focus on SDMSE loss with weight decay.

Table 3: Performance of $\mathcal{N}_7$ using Different Training Sizes

Training Size	With Feed Length		Without Feed Length	
	SMSE	R^2	SMSE	R^2
600	0.0090	0.8940	0.0028	0.9217
1200	0.0052	0.9337	0.0024	0.9433
2400	0.0033	0.9586	0.0016	0.9593
4800	0.0022	0.9666	0.0018	0.9670

The results of using different sizes of training set are summarized in Table 3. We see that as the size of the training set increases, the prediction accuracy of $\mathcal{N}_7$ also improves.

Moreover, we evaluate the prediction power of the residual network $\mathcal{N}_7$ on each geometrical parameter. Table 4 reports the SMSE for predicting each geometrical parameter in , when using a training size of 2400 and SDMSE as the training loss.

It can be seen that the SMSE for predicting the feed length (ℓ_f) well exceeds those for predicting other geometrical parameters. This observation is consistent across different training sizes and both SMSE and SDMSE training loss. The low correlation between the circuit parameters and the feed length well matches theory, since feed length only influences the inductance of the primary/secondary coils. In addition, its choice is largely independent of the transformer geometric design, whereas highly relies on the physical layout of the RF/mm-Wave circuit.

Table 4: SMSE for Predicting Each Geometrical Parameter

Geometrical Parameter	W_OA	W_OB	r_0	r_1	x_gnd	ℓ_f
SMSE	0.0017	0.0038	0.0003	0.0007	0.0012	0.0123

Therefore, we further test using $\mathcal{N}_7$ to predict all the geometrical parameters except the feed length (ℓ_f). The results are summarized in the rightmost two columns of Table 3. By removing the feed length parameter, $\mathcal{N}_7$ enjoys a performance boost especially using a small number of training samples. This result is inspiring and suggests that $\mathcal{N}_7$ is indeed efficient in capturing the informative correspondence between circuit and geometrical parameters.

• **Validation Examples of the Direct Synthesis**. We demonstrate an example of using the trained predictive model $\mathcal{N}_7$ to directly synthesize geometrical parameters given a randomly selected set of desired circuit parameters. The obtained geometrical parameters are shown in Table 5. Then, we run one EM simulation using the predicted geometry to verify the prediction. We observe that the synthesized circuit parameters closely match the desired parameters.

Table 5: Predicting Geometrical Parameters using Circuit Parameters

Circuit Prameters		L_p(pH)	L_s(pH)	k	SRF(GHz)	Q_p	Q_s
I	Targeted	142.25	163.60	0.55	97.00	22.20	20.52
	Synthesized	142.91	164.41	0.56	96.50	22.39	20.42
II	Targeted	173.30	188.44	0.48	99.00	21.81	23.59
	Synthesized	168.48	184.43	0.47	99.00	21.89	24.36
III	Targeted	226.89	242.25	0.70	66.30	22.38	21.44
	Synthesized	236.02	252.69	0.69	65.00	22.79	21.80
IV	Targeted	111.26	128.72	0.59	95.80	23.25	19.97
	Synthesized	111.93	129.38	0.59	96.20	24.03	20.00
V	Targeted	245.09	294.89	0.62	78.70	21.79	16.73
	Synthesized	243.21	293.00	0.62	79.00	21.78	16.84

5 Conclusion

We propose a neural network based model for the direct synthesis of RF/mm-Wave EM passive structures. A proof of concept is demonstrated on a 1:1 transformer. Our trained residual network model generates near perfect predictions on transformer's geometrical parameters for given target circuit performance, and outperforms widely used machine learning baseline methods. Our proposed model can be further extended to more complex EM passive structures and revolutionize the design procedure and automation of RF/mm-Wave circuits.

References

[1] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.

[2] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[6] Han Liu, Lie Wang, and Tuo Zhao. Calibrated multivariate regression with application to neural semantic basis discovery. *Journal of Machine Learning Research*, 16:1579–1606, 2015.

[7] F Wang, K Xu, J Romberg, and H Wang. An artificial-intelligence (ai) assisted mm-wave doherty power amplifier with rapid mixed-mode in filed performance optimization. In *Proc. IEEE IMC-5G, August*, 2019.