End-to-End Neural Modeling of EM Response and Design Performance for Free-Form RFIC Passives

Yuhao Mao[1]* **Chenhao Chu**[2,*] **Martin Vechev**[1] **Hua Wang**[2]
[1]Department of Computer Science
[2]Department of Information Technology and Electrical Engineering
ETH Zurich, Switzerland
{yuhao.mao,martin.vechev}@inf.ethz.ch
{chenhao.chu,hua.wang}@iis.ee.ethz.ch

Abstract

Passive matching networks are critical components in radio-frequency integrated circuits (RFICs), but their design is bottlenecked by expensive finite-difference-based electromagnetic (EM) simulation. We propose an end-to-end neural framework for free-form RFIC passives that predicts both EM response and downstream design performance under varying frequency and circuit contexts, combining a frequency-conditioned neural simulator with a circuit-conditioned neural solver. To improve design performance prediction, the simulator outputs a neural EM response consisting of both the explicit EM response and a complementary response, together with a dependence regularizer that encourages the two responses to capture orthogonal information. To evaluate the proposed framework, we construct a large-scale dataset of free-form passive layouts, synthesized from both expert-designed contours and random paths. Our 21M-parameter model achieves high accuracy across a broad frequency range, circuit contexts and passive matching settings, with $R^2 > 0.98$ for almost all targets, while providing orders-of-magnitude faster evaluation than traditional EM simulation. Code available at `https://github.com/ETH-IDEAS/RFIC-NeuralSim`.

1 Introduction

Radio-frequency integrated circuits (RFICs) are core building blocks of modern wireless systems. Within an RFIC, the power amplifier (PA) is a central component of the transmit chain and has therefore received sustained design attention. A key challenge in PA design is the passive matching network[2]. Its layout geometry determines the electromagnetic (EM) response, which in turn shapes downstream design performance. Traditional finite-difference-based EM simulation is extremely expensive, preventing fast design optimization for RFIC passives. This cost has motivated fast surrogate models for RFIC passives, including recent neural approaches.

Existing neural approaches have made progress on RFIC passive modeling, but their coverage remains limited, as summarized in Table 1. First, many prior methods operate on templated or otherwise restricted passive layouts, which simplifies modeling but significantly narrows the design space. Second, most prior work focuses on EM response prediction without modeling downstream design performance. This is a serious but overlooked limitation because small EM-response errors can cause large design-performance degradations when passed to a ground-truth solver, as shown in Figure 1. Third, prior methods often omit explicit auxiliary circuit conditioning, operate on narrow or sparse frequency regimes, or require retraining when the frequency grid changes. These gaps leave open the central problem we study here: end-to-end neural modeling of free-form passive

*Equal contribution.

[2]In this paper, "network" refers to a passive matching network rather than a neural network unless stated otherwise.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Comparison between our method and prior work on neural modeling of RFIC passives.

	Free-form Passive Layout	Modeling Outcome		Modeling Conditions		Open Source
		EM Response	Design Performance	Frequency	Circuit Context	
Yan et al. (2018)	✗	✓	✗	100 pts, 1 GHz range	not applicable	✗
Guan et al. (2021)	✗	✓	✗	20 pts, 20 GHz range	not applicable	✗
Li et al. (2022)	✗	✗	✓	✗	✗	✗
Lee et al. (2022)	✗	✓	✓	10 pts, 100 GHz range	✗	✗
Karahan et al. (2024)	✓	✓	✗	pts not reported, 76 GHz range†	not applicable	✗
Sahu et al. (2025)	✗	✓	✗	pts not reported, 2 GHz range†	not applicable	✗
Zhang and Zhou (2026)	✗	✓	✗	pts not reported, 20 GHz range	not applicable	✗
Ours	✓	✓	✓	501 pts, 50 GHz range	✓	✓

† Instead of explicit conditioning, they use separate output dimensions for different frequencies, which requires retraining the model to operate on unseen frequencies.

matching networks, from layout to both EM response and design performance, under varying circuit and frequency settings.

Here, *design performance* refers to the downstream circuit quantities induced by the passive matching network within its surrounding circuit context, such as auxiliary capacitors.

In this work, we address these limitations with an end-to-end model for free-form passive matching networks under varying circuit and frequency settings. Our model combines a frequency-conditioned *neural simulator* with a circuit-conditioned *neural solver*, as shown in Figure 4. The neural simulator maps passive layout and frequency to a *neural EM response*, which consists of the explicit EM response and a complementary response. The complementary response preserves information that is useful for design-performance prediction when the explicit EM response is imperfect. The neural solver then combines the neural EM response with circuit context to predict design performance.

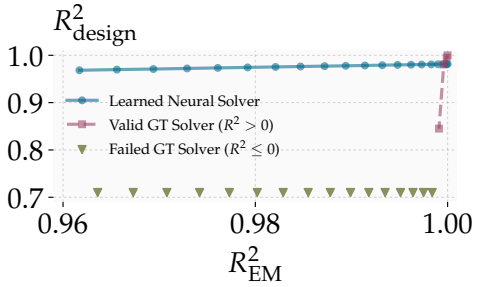


Figure 1: Solver stability under imperfect EM response. The ground-truth (GT) solver achieves $R^2 = 1$ with noiseless input, but its design-performance R^2 becomes negative even at $R^2_{\text{EM}} \approx 0.9984$. In contrast, the learned neural solver remains stable given imperfect EM responses. App. E details the evaluation protocol and illustrates the magnitude of the perturbations.

Our main contributions are as follows:

- We introduce an end-to-end neural modeling framework for free-form RFIC passives. Two components are proposed to further improve this framework: a complementary response that avoids the information bottleneck of the imperfect EM response prediction, and a dependence regularizer that encourages independence between the EM response and complementary response.
- We construct a large-scale dataset of free-form RFIC passives, with layouts synthesized from both expert-designed contours and random paths, and use it for a comprehensive empirical study.
- Our 21M-parameter model achieves high accuracy ($R^2 > 0.98$ for almost all targets) across a broad frequency range. On an RTX 2080 Ti GPU, the neural simulator runs at 121 full-frequency simulations per second, and the neural solver runs at 25,069 full-frequency predictions per second. Compared with traditional EM simulation, which can take up to 20 minutes for one full-frequency evaluation, our method provides orders-of-magnitude acceleration while maintaining high accuracy, enabling fast design optimization for RFIC passives.

Aside from the closest prior works in Table 1, a more detailed related work can be found in App. A.

2 Background

A typical RFIC power amplifier (PA) uses passive matching networks at its input, between stages, and at its output, as shown in Figure 2. Informally, these networks are small on-chip passive circuits that reshape how RF signals pass between blocks. Since this behavior is set by the metal geometry, the layout affects both the local electromagnetic (EM) response and the end-to-end PA performance.

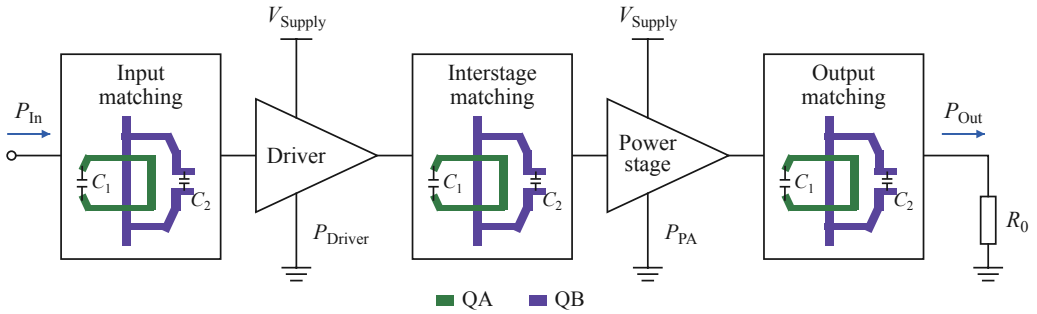


Figure 2: Typical RFIC PA with three passive matching networks: input matching, interstage matching, and output matching.

We study a realistic passive layout setting. Each passive layout is symmetric, its ports are mutually connected, and its geometry is represented on a fixed 18×18 pixel grid. The layout contains two metal layers, *QA* and *QB*, shown as the green and purple layers in Figure 2. The chosen passive structures are reciprocal because their EM model uses linear, time-invariant conductors and dielectrics with reciprocal material responses. The surrounding circuit context is parameterized by two scalar capacitor values, C_1 and C_2. The input variables for the prediction task are therefore the layout mask, the frequency, and the circuit context. Dual-metal-layer pixelated matching networks have been used in fabricated CMOS PAs in GlobalFoundries 22nm FDX+ technology (Chu et al., 2025a,b). In particular, Chu et al. (2025b) use these networks for input, interstage, and output matching in a measured 12–16 GHz PA.

The prediction targets include both EM response and design performance. The EM response is represented by frequency-dependent complex S-parameters, which describe how signals scatter between ports. At each frequency, the full S-matrix has shape 8×8. Reciprocity gives $S_{ij} = S_{ji}$; together with layout symmetry, it reduces the matrix to 20 distinct complex entries. The design performance consists of the complex input impedance Z_{in}, together with the real-valued passive efficiency and max gain.

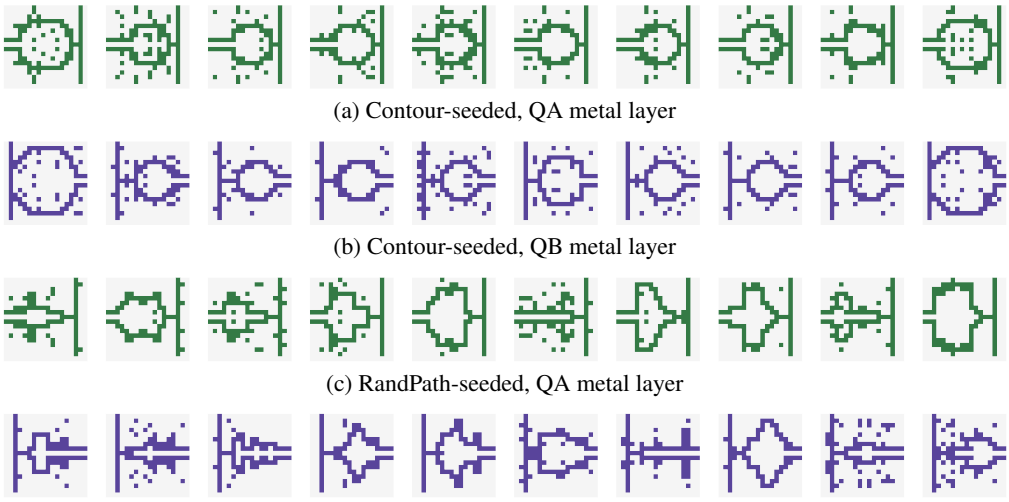
The traditional workflow evaluates a passive matching network in two stages. First, a finite-difference-based EM simulator is run on the layout over frequency to obtain its EM response. Second, this EM response is embedded into the surrounding circuit, and a downstream solver evaluates the resulting design performance under the circuit context. This workflow is physically grounded and widely used in practice, but the EM simulator is computationally expensive and can require up to 20 minutes for a single full-frequency evaluation. As a result, iterative design and large-scale exploration remain slow. Neural modeling of RFIC passives replaces this expensive pipeline by predicting EM response and design performance directly with neural networks, enabling faster evaluation and design iteration.

3 Method

Problem Formulation. We study one passive matching network at a time and train separate models for input, interstage, and output matching. Each passive layout is represented by a binary mask $\boldsymbol{M} \in \{0,1\}^{2\times 18\times 18}$. The masks are drawn from two data distributions, illustrated in Figure 3. Contour-seeded masks start from high-performing expert-designed contours and then add random pixels. RandPath-seeded masks start from random paths between the two ports, mirror the paths to satisfy symmetry, and then add random pixels. These two families cover both structured layouts near established designs and more random connected layouts.

Here, *free-form* denotes pixel-based geometry within the fixed grid and stated structural constraints, allowing layouts beyond predefined geometric templates. This describes the flexibility of the geometric representation, while prediction accuracy on unseen layout distributions is evaluated separately.

For each layout, we sample 200 circuit contexts $\boldsymbol{c} = (C_1, C_2)$ by drawing C_1 and C_2 uniformly at random from $[0, 400]$ fF and $[10, 400]$ fF, respectively, and then quantizing them to 0.01 fF. Each context is evaluated on the frequency grid $f \in \mathcal{F} = \{10 + 0.1k \text{ GHz} : k = 0, \ldots, 500\}$. The prediction task is to learn a mapping from $(\boldsymbol{M}, f, \boldsymbol{c})$ to the EM response $S(\boldsymbol{M}, f) \in \mathbb{R}^{40}$ and the design performance $Y(\boldsymbol{M}, f, \boldsymbol{c}) \in \mathbb{R}^4$, as defined in Section 2.



(a) Contour-seeded, QA metal layer

(b) Contour-seeded, QB metal layer

(c) RandPath-seeded, QA metal layer

(d) RandPath-seeded, QB metal layer

Figure 3: Example layouts from the two passive layout distributions.

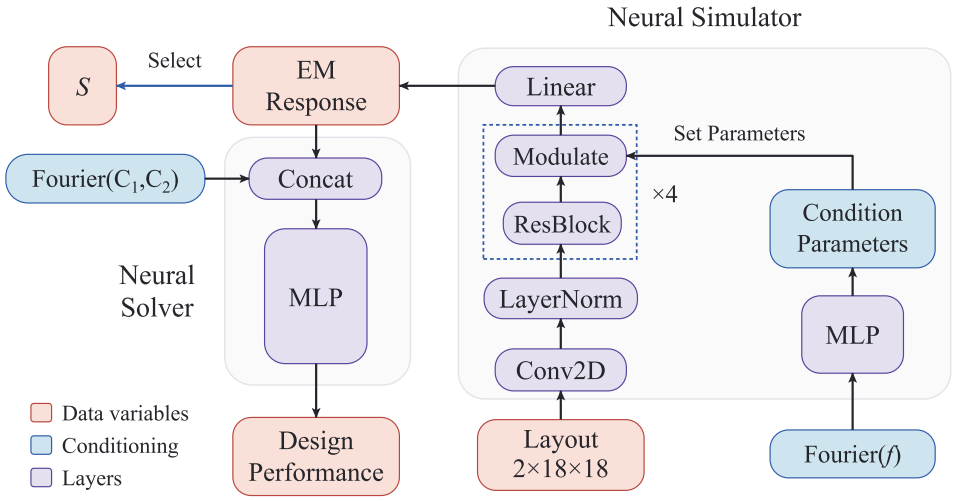


Figure 4: Overview of the proposed model. The frequency-conditioned neural simulator maps the layout $\boldsymbol{M}$ and frequency to a neural EM response, consisting of an explicit EM response and a complementary response. The circuit-conditioned neural solver then combines this neural EM response with the circuit context to predict design performance. Frequency information is injected through modulation across four residual blocks.

Overview. Our framework combines a frequency-conditioned *neural simulator* with a circuit-conditioned *neural solver*, as shown in Figure 4. The neural simulator maps the passive layout $\boldsymbol{M}$ and frequency f to a *neural EM response*. This response has two parts: an explicit EM response $\hat{S}$, supervised against the target S-parameters, and a complementary response Z_{comp}, which preserves additional information useful for downstream design prediction. The neural solver then combines $(\hat{S}, Z_{\text{comp}})$ with the circuit context $\boldsymbol{c} = (C_1, C_2)$ to predict design performance.

Frequency-Conditioned Neural Simulator. The neural simulator processes $\boldsymbol{M}$ with a convolutional backbone comprising an initial 2D convolution, layer normalization, and four residual blocks (He et al., 2016). To condition this shared spatial backbone on frequency, we first compute the Fourier encoding

$$\phi(f) = [f, \sin(2^0\pi f), \cos(2^0\pi f), \ldots, \sin(2^{K-1}\pi f), \cos(2^{K-1}\pi f)], \tag{1}$$

where $K = 10$ is the number of Fourier scales. This fixed-size encoding captures both low- and high-frequency variation. Inspired by the affine conditioning in DiT (Peebles and Xie, 2023), we inject

frequency information through a modulation layer that follows the feature-wise linear modulation (FiLM) principle (Perez et al., 2018). Conventional CNN FiLM shares each channel's scale and shift across spatial positions. Our layer instead predicts a separate scale and shift for every channel and spatial position from the Fourier-encoded frequency. This design keeps the layout-processing backbone shared across frequencies while allowing intermediate activations to vary smoothly with f. Specifically, for each modulation layer, a conditioning MLP maps $\phi(f)$ to modulation parameters $\boldsymbol{\gamma}(f)$ and $\boldsymbol{\beta}(f)$. Let $\boldsymbol{h}(\boldsymbol{M}) \in \mathbb{R}^d$ denote the vectorized input feature tensor to one such layer. Then $\boldsymbol{\gamma}(f), \boldsymbol{\beta}(f) \in \mathbb{R}^d$, and the modulated feature is

$$\widetilde{\boldsymbol{h}}(\boldsymbol{M}, f) = \boldsymbol{\gamma}(f) \odot \boldsymbol{h}(\boldsymbol{M}) + \boldsymbol{\beta}(f), \tag{2}$$

where $\odot$ denotes elementwise multiplication. Continuous frequency conditioning allows queries between the discrete training frequencies, as evaluated within the training band in Section 4. Since the conditioning parameters depend only on frequency, they can be cached when evaluating multiple layouts at the same frequency.

Circuit-Conditioned Neural Solver. The neural solver predicts design performance from the neural EM response and circuit context. We encode C_1 and C_2 with the same Fourier feature construction, using $K = 5$ for each scalar, concatenate the resulting embedding with the neural EM response at each frequency, and process the concatenated representation with an MLP to obtain $\hat{Y}$. This design mirrors the traditional workflow in Section 2: the simulator provides a learned surrogate for EM analysis, and the solver maps the resulting neural EM response into downstream design quantities under the specified circuit context.

The learned solver also improves robustness to imperfect EM predictions. As shown in Figure 1, a ground-truth downstream solver is highly sensitive to perturbations in its EM-response input. Therefore, high EM-response accuracy alone is insufficient for accurate design-performance prediction when the traditional solver is applied. Similar observations have also been reported in different settings by Carlucci et al. (2023, 2025). The learned neural solver can adapt to errors in the neural EM response, enabling design performance prediction even with imperfect EM response.

Joint Training Objective. We train the simulator and solver end-to-end with normalized mean squared error losses. Let $\hat{S}$ and S denote the predicted and ground-truth EM responses, and let $\hat{Y}$ and Y denote the predicted and ground-truth design performances. We define

$$\mathcal{L}_{\mathrm{EM}} = \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} \|\hat{S}(f) - S(f)\|_2^2, \tag{3}$$

$$\mathcal{L}_{\mathrm{design}} = \frac{1}{|\mathcal{C}|\,|\mathcal{F}|} \sum_{\boldsymbol{c} \in \mathcal{C}} \sum_{f \in \mathcal{F}} \|\hat{Y}(\boldsymbol{c}, f) - Y(\boldsymbol{c}, f)\|_2^2, \tag{4}$$

where $\mathcal{F}$ is the frequency grid and $\mathcal{C}$ is the sampled set of circuit contexts. The prediction loss is

$$\mathcal{L}_{\mathrm{pred}} = \mathcal{L}_{\mathrm{EM}} + \mathcal{L}_{\mathrm{design}}. \tag{5}$$

This joint supervision keeps the simulator faithful to the EM target while encouraging representations that support accurate downstream design-performance prediction.

Dependence Regularization. The complementary response Z_{comp} is intended to capture information beyond the explicit EM-response, thus reducing the information bottleneck due to the imperfect EM response. To encourage independence between Z_{comp} and $\hat{S}$, we regularize Z_{comp} additionally to reduce its correlation with $\hat{S}$. Let $\boldsymbol{u}$ denote the simulator feature before the final linear projections, and let

$$\hat{S} = \boldsymbol{W}_S \boldsymbol{u} + \boldsymbol{b}_S, \qquad Z_{\mathrm{comp}} = \boldsymbol{W}_Z \boldsymbol{u} + \boldsymbol{b}_Z. \tag{6}$$

Since both are affine transformations of the same feature, their covariance has the closed form

$$\mathrm{Cov}(\hat{S}, Z_{\mathrm{comp}}) = \boldsymbol{W}_S \,\mathrm{Cov}(\boldsymbol{u})\, \boldsymbol{W}_Z^\top. \tag{7}$$

For a minibatch with feature vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$, we estimate $\mathrm{Cov}(\boldsymbol{u})$ by the empirical covariance

$$\widehat{\mathrm{Cov}}(\boldsymbol{u}) = \frac{1}{n-1} \sum_{i=1}^{n} (\boldsymbol{u}_i - \bar{\boldsymbol{u}})(\boldsymbol{u}_i - \bar{\boldsymbol{u}})^\top, \qquad \bar{\boldsymbol{u}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{u}_i. \tag{8}$$

We then define

$$\mathcal{L}_{\text{corr}} = \frac{1}{d_S d_Z} \left\| \text{sg}(\boldsymbol{W}_S)\ \text{sg}(\widehat{\text{Cov}}(\boldsymbol{u}))\, \boldsymbol{W}_Z^\top \right\|_F^2, \tag{9}$$

where d_S and d_Z are the dimensions of $\hat{S}$ and Z_{comp}, respectively, $\|\cdot\|_F$ denotes the Frobenius norm, and $\text{sg}(\cdot)$ denotes stop gradient. This regularization thus only affects the complementary response and put no constraint on the prediction of the explicit EM response $\hat{S}$. The full training objective is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{corr}}\, \mathcal{L}_{\text{corr}}. \tag{10}$$

As shown in Section 4, this regularization substantially reduces interdependence with little loss in accuracy.

4 Experiments

Experimental Details. We split the full dataset of 61,515 passive layouts into training, validation, and test partitions in a 7:1:2 ratio. During training, all target variables are normalized by subtracting the dataset mean and dividing by the dataset standard deviation. Conditioning variables are normalized to $[-1, 1]$ according to their ranges. We also apply layout augmentation with random shifts, flips, and right-angle rotations. We use R^2 as the prediction accuracy metric, defined as $R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$, where y_i and $\hat{y}_i$ are the ground-truth and predicted values, respectively, and $\bar{y}$ is the mean of the ground-truth values. We report R^2 for both EM response and design performance targets, denoted as R^2_{EM} and R^2_{design}, respectively. When the target variable has multiple dimensions, e.g., $S \in \mathbb{R}^{40}$, we compute R^2 for each dimension and report the average.

We study two model scales: a base model with 5.4M parameters and a large model with 21M parameters. Both are optimized with AdamW, using learning rate 10^{-4}, weight decay 10^{-2}, and a cosine annealing schedule that decays the learning rate to 10^{-6} over training. For efficient storage and loading, we store the dataset in Zarr format. The stored dataset occupies about 200 GB after compressing 32-bit floating-point values to 16 bits, followed by Bitshuffle (Masui, 2017) and Blosc compression. Training uses distributed data parallelism over 8 NVIDIA GeForce RTX 2080 Ti GPUs with batch size 4 per GPU, and we report test results from the checkpoint with the lowest validation loss. Unless stated otherwise, all models are trained for 500 epochs, with validation after every epoch. This takes roughly 45 hours for the 5.4M model and roughly 120 hours for the 21M model to train end-to-end. After initial tuning on the 5.4M model with few epochs, we keep the hyperparameters fixed across all model scales and matching settings, without further tuning. We did not perform cherry-picking of random seeds, and all results are from a single end-to-end run per model, due to the high computational cost. Nevertheless, the results are stable across model scales and matching settings in independent runs.

Main Results Across Model Scales and Matching Settings. Table 2 compares the 5.4M and 21M models across input, interstage, and output matching. Results are reported on the full distribution and on the two constituent distributions. Scaling from 5.4M to 21M consistently improves both EM response and design performance prediction across all three matching settings. These gains suggest that model capacity is an important factor for free-form RFIC passive modeling. Further scaling may yield additional improvements; we leave it for future work due to the high cost. App. C further compares parameter-matched plain CNN and residual backbones trained for 100 epochs on input matching. Their comparable aggregate accuracy indicates that residual blocks are not essential to high performance in this setting.

Interpolation Between Training Frequencies. Figure 5 evaluates the 21M input matching model on a frequency grid that is $100\times$ denser than the training grid, using a query step of 0.001 GHz, for five randomly sampled test layouts. Corresponding interstage and output matching results are provided in App. B.3. This evaluation queries unseen frequencies within the 10–60 GHz training band and assesses interpolation. We do not evaluate extrapolation because RFIC usually has an operating range and does not need to extrapolate to other bands. The model remains well behaved under this much finer frequency query, indicating that the frequency-conditioned simulator interpolates smoothly between training frequencies. The denser evaluation also reveals visibly less smooth responses in high frequencies, possibly due to the increased sensitivity of the targets. This observation suggests

Table 2: Main results for the 5.4M and 21M models across input, interstage, and output matching. We report EM response R^2_{EM} and design performance R^2_{design} on the full distribution and on its two constituent distributions. Scaling to 21M improves both metrics across all three matching settings.

#param	Matching	Full		Contour-seeded		RandPath-seeded	
		R^2_{EM}	R^2_{design}	R^2_{EM}	R^2_{design}	R^2_{EM}	R^2_{design}
5.4M	input	0.9879	0.9439	0.9885	0.9327	0.9940	0.9442
21M		**0.9944**	**0.9666**	**0.9934**	**0.9782**	**0.9968**	**0.9641**
5.4M	interstage	0.9844	0.9282	0.9864	0.8759	0.9920	0.9191
21M		**0.9932**	**0.9394**	**0.9921**	**0.9647**	**0.9962**	**0.9381**
5.4M	output	0.9878	0.9503	0.9881	0.9350	0.9937	0.9492
21M		**0.9939**	**0.9714**	**0.9936**	**0.9810**	**0.9968**	**0.9723**

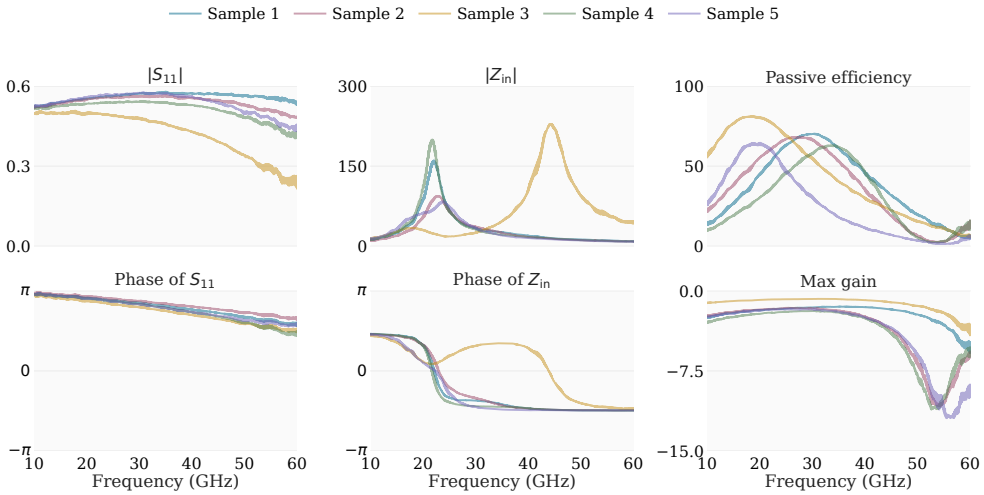


Figure 5: Predictions of the 21M input matching model on a frequency grid with 0.001 GHz spacing. Wider curve means the prediction changes less smoothly.

that future work could explore non-uniform frequency sampling during training, with denser coverage in high-frequency regions.

Per-Target Accuracy Across Frequency. Figure 6 compares the 21M and 5.4M input matching models on the full distribution, broken down by target and frequency. Corresponding interstage and output matching results are provided in App. B.1. Scaling to 21M improves all targets across frequency band, with the clearest gains in more difficult prediction regimes where the 5.4M model struggles, including harder targets and frequency ranges. Among the six targets, max gain remains the most difficult for the 21M model, with accuracy about 10% lower than the other targets.

Effect of Training Compute Scaling. Figure 7 tracks the 5.4M input matching model across training epochs. Corresponding interstage and output matching results are provided in App. B.2. Longer training improves accuracy across all targets, with the largest gains on challenging targets and frequency ranges. These results demonstrate the benefit of increasing training compute.

Effect of Dataset Scaling. We train both model sizes on nested subsets containing 25%, 50%, 75%, and 100% of the RandPath-seeded training layouts for input matching, using a fixed number of steps equivalent to 50 full-data epochs. Dataset scaling consistently improves R^2 for almost all targets (Figure 8). As with training compute scaling, the gains are most pronounced for challenging targets with lower R^2.

Effect of the Complementary Response. Figure 9 compares the 21M input matching model with and without the complementary response. Adding Z_{comp} improves the average R^2 for every target.

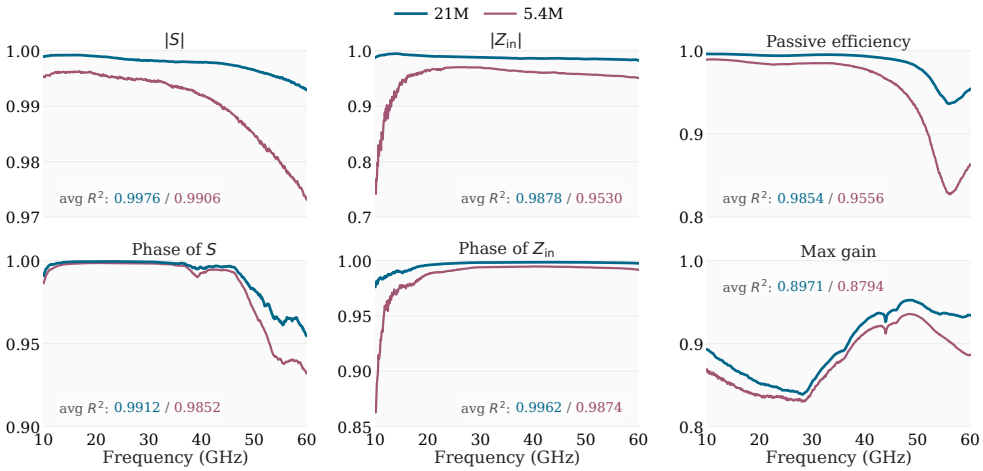


Figure 6: Per-target R^2 over frequency for the 21M and 5.4M input matching models.

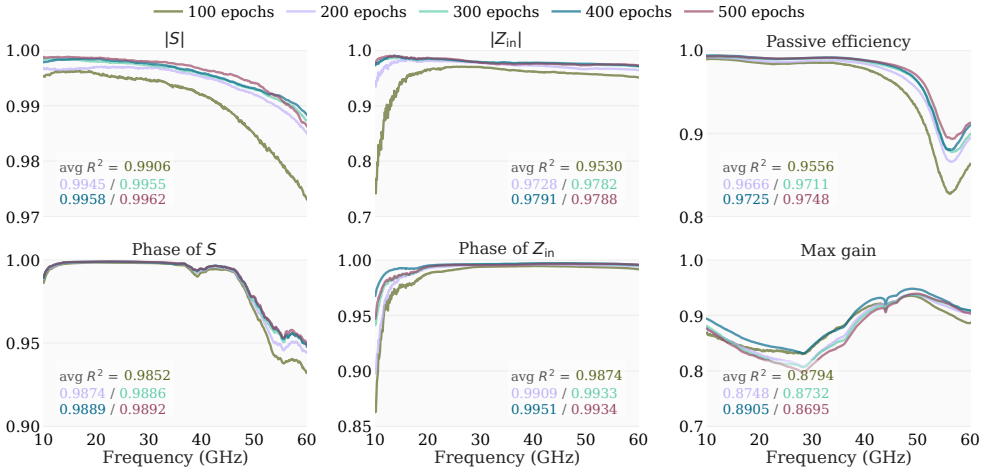


Figure 7: Per-target R^2 over frequency for the 5.4M input matching model across training epochs.

The largest gain appears on $|S|$, where removing Z_{comp} causes a pronounced high-frequency drop. This is possibly because that without Z_{comp}, the model has to adapt to the sensitive dependence between the design performance and S, thus preventing it from learning better $\hat{S}$. For $|Z_{\text{in}}|$ and phase of Z_{in}, Z_{comp} prevents sharp degradation in the low-frequency region. The remaining targets also receive consistent gains from Z_{comp} across the frequency band.

Generalization to Unseen Layout Distributions. We test cross-distribution generalization between the two constituent distributions, Contour-seeded and RandPath-seeded, which have different layout generation procedures and layout characteristics. As shown in Figure 10, the direction of transfer matters substantially. Models trained on Contour-seeded masks perform well on the same family ($R^2_{\text{design}} = 0.955$ and 0.967 for 5.4M and 21M), but generalize poorly to the more random RandPath-seeded family ($R^2_{\text{design}} = -0.054$ and 0.036). By contrast, models trained on RandPath-seeded masks transfer much better to the unseen Contour-seeded family ($R^2_{\text{design}} = 0.614$ and 0.646). Thus, training on the more random mask distribution provides stronger out-of-distribution generalization, although it still does not match same-family performance. This suggests that broader and more randomized layout distributions may improve generalization to unseen layouts compared with training only on established layout families.

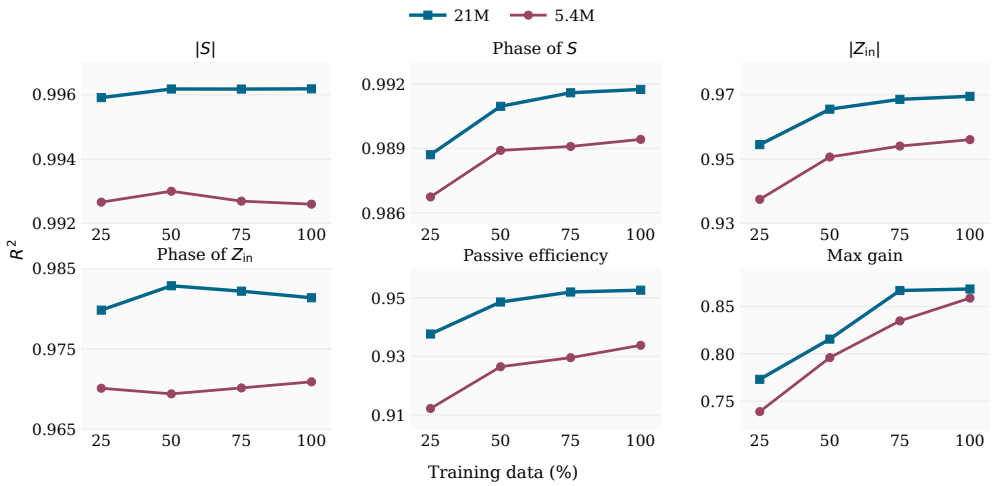


Figure 8: In-distribution data scaling for input matching. Both models are trained and tested on RandPath-seeded layouts with a fixed training budget of 52,450 optimizer steps. Each panel reports test R^2 for one target. Panel-specific y limits differ.

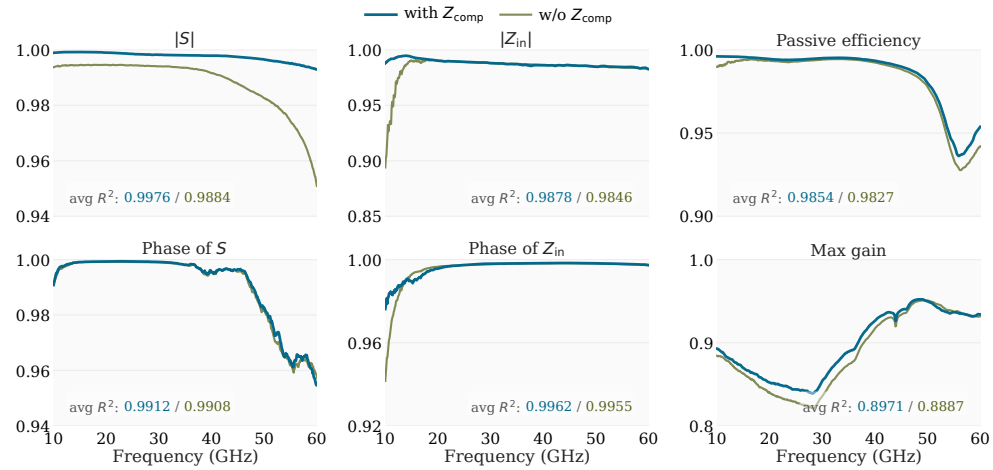


Figure 9: Effect of the complementary response for the 21M input matching model.

Effect of Dependence Regularization. Figure 11 evaluates the effect of the dependence regularization. With regularization ($\lambda_{\text{corr}} = 0.1$), $\text{Corr}(\hat{S}, Z_{\text{comp}})$ decreases from 0.149 to 0.006, corresponding to a 96% reduction. This reduction comes with only a small change in design performance, with R^2_{design} moving from 0.967 to 0.964. These results show that the regularizer yields a cleaner decomposition of the neural EM response while preserving design-performance accuracy. A per-target accuracy breakdown across regularization strengths is provided in App. B.4.

5 Discussion

Limitations. Our study focuses on one RFIC passive setting with a fixed 18×18 layout grid, two metal layers, and three PA matching placements. The models are trained separately for input, interstage, and output matching, and the data cover only two layout-generation families. Cross-family accuracy remains below in-distribution accuracy, and the current evidence does not establish generalization to broader layout distributions. The current checkpoints are not expected to transfer to different layout resolutions or process technologies without adaptation or retraining. The dense-frequency evaluation demonstrates interpolation within the training band rather than extrapolation beyond it. Training is expensive, particularly for the 21M model, and we do not quantify variability

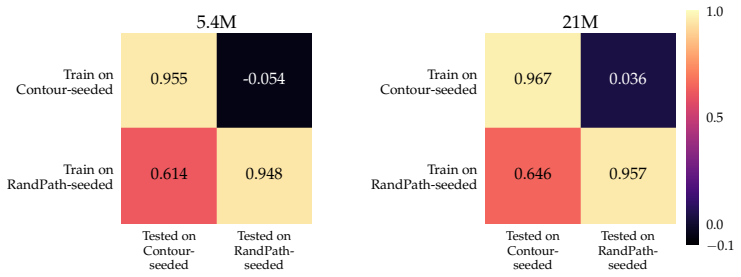


Figure 10: Generalization to unseen mask distributions between the Contour-seeded and RandPath-seeded families. Each cell reports R^2_{design}. Models are trained for 100 epochs for this analysis.

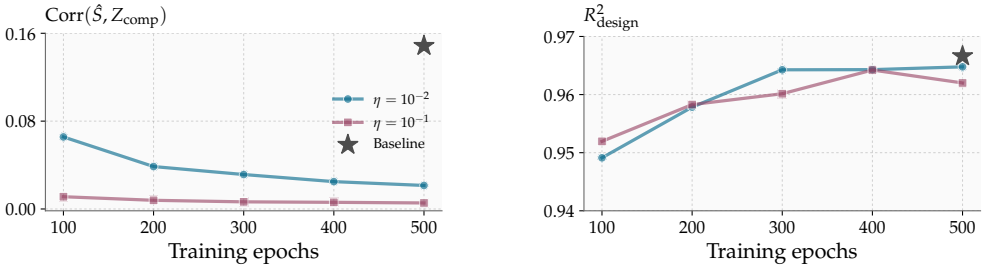


Figure 11: Effect of dependence regularization. $\text{Corr}(\hat{S}, Z_{\text{comp}})$ is the average correlation.

across random seeds. The predicted EM responses do not guarantee physical properties such as passivity; App. D quantifies violations of passivity across all three matching settings and shows lower violation rates with better models.

Future Work. *(a) Flexible layout sizes.* Even though this work removes the restriction from a templated layout, the fixed layout size still restricts the design space. Encoding layouts as variable-length sequences of patch tokens (Dehghani et al., 2023) could further relax this constraint and support an even broader design space. *(b) Further scaling.* We demonstrate that larger models (Table 2), longer training (Figure 7), and more training data (Figure 8) improve accuracy consistently. These gains motivate further scaling neural simulation models in these directions. *(c) Broader conditioning.* This work provides a general way to condition neural simulation models; extending the current conditioning to other properties, *e.g.*, material effects, could potentially allow one simulator to cover more diverse settings. *(d) Randomized training data.* Stronger transfer from RandPath-seeded than Contour-seeded training (Figure 10) suggests that broader training-distribution coverage should accompany increases in dataset size. For broad free-form generalization, this motivates prioritizing randomized simulation data over exclusive reliance on template-seeded layouts (Chu et al., 2025a,b) or predefined transformer families (Lee et al., 2022; He et al., 2026).

6 Conclusion

We present an end-to-end neural modeling framework for free-form RFIC passive matching networks. The framework combines a frequency-conditioned neural simulator with a circuit-conditioned neural solver, enabling direct prediction of both EM response and design performance under varying frequency and circuit contexts. On a large-scale dataset of free-form RFIC passives, the proposed 21M model achieves high accuracy across input, interstage, and output matching settings, while providing orders-of-magnitude faster evaluation than traditional EM simulation. These results suggest that neural EM and design-performance modeling can substantially accelerate RFIC passive design-space exploration, with future extensions toward more diverse layouts, flexible layout dimensions, and broader circuit settings.

LLM Disclosure

This paper was assisted by PIRA (PIRA Project, 2026), a research-assistant system powered by GPT-6-Astra and GPT-6.1-Sol. The assistance included implementation aids and writing polishes. The authors are fully responsible for the presented final content.

References

S. Yan, Y. Zhang, X. Jin, W. Zhang, and W. Shi, "Multi-physics parametric modeling of microwave passive components using artificial neural networks," *Progress In Electromagnetics Research M*, vol. 72, pp. 79–88, 2018.

Z. Guan, P. Zhao, X. Wang, and G. Wang, "Modeling radio-frequency devices based on deep learning technique," *Electronics*, vol. 10, no. 14, p. 1710, 2021.

X. Li, Y. Tang, P. Zhao, S. Chen, K. Xu, and G. Wang, "A deep learning approach for efficient electromagnetic analysis of on-chip inductor with dummy metal fillings," *Electronics*, vol. 11, no. 24, p. 4214, 2022.

D. Lee, G. Shin, S. Lee, K. Kim, T.-H. Oh, and H.-J. Song, "Neural-network-based automated synthesis of transformer matching circuits for rf amplifier design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 11, pp. 4726–4739, 2022.

E. A. Karahan, Z. Liu, A. Gupta, Z. Shao, J. Zhou, U. Khankhoje, and K. Sengupta, "Deep-learning enabled generalized inverse design of multi-port radio-frequency and sub-terahertz passives and integrated circuits," *Nature Communications*, vol. 15, no. 1, p. 10734, 2024.

K. C. Sahu, S. Koziel, and A. Pietrenko-Dabrowska, "Surrogate modeling of passive microwave circuits using recurrent neural networks and domain confinement," *Scientific Reports*, vol. 15, no. 1, p. 13322, 2025.

H. Zhang and P. Zhou, "Efficient rf passive components modeling with bayesian online learning and uncertainty aware sampling," in *2026 31st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2026, pp. 1145–1151.

C. Chu, E. Liu, Y. Liu, B. Lin, M. Eleraky, B. Abdelaziz, M. Ghorbanpoor, T.-Y. Huang, A. Wang, and H. Wang, "Deep learning-assisted RFIC design with dual-metal-layer passive matching networks: A 15–22 GHz CMOS PA for 6G in 22nm FDX+," in *2025 16th German Microwave Conference (GeMiC)*. IEEE, 2025, pp. 76–79.

C. Chu, J. Xu, Y. Liu, J. Zeng, A. Wang, T. Torii, S. Shinjo, K. Yamanaka, and H. Wang, "AI-assisted template-seeded pixelated design for multi-metal-layer high-coupling EM structures: A Ku-band 6G FR3 PA in 22nm FDX+," in *2025 IEEE/MTT-S International Microwave Symposium (IMS)*. IEEE, 2025, pp. 922–925.

K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018, pp. 3942–3951.

A. Carlucci, T. Bradde, and S. Grivet-Talocia, "Addressing load sensitivity of rational macromodels," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 13, no. 10, pp. 1591–1602, 2023.

A. Carlucci, T. Bradde, and S. Grivet-Talocia, "Improving robustness to termination conditions in passivity enforcement of rational macromodels," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 15, no. 1, pp. 64–74, 2025.

K. Masui, "Bitshuffle: Filter for improving compression of typed binary data," Astrophysics Source Code Library, ascl:1712.004, 2017. [Online]. Available: https://www.ascl.net/1712.004

M. Dehghani, B. Mustafa, J. Djolonga, J. Heek, M. Minderer, M. Caron, A. Steiner, J. Puigcerver, R. Geirhos, I. Alabdulmohsin, A. Oliver, P. Padlewski, A. Gritsenko, M. Lučić, and N. Houlsby, "Patch n' pack: NaViT, a vision transformer for any aspect ratio and resolution," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 2252–2274.

H. He, Y. Xu, L. Xia, Y. Hu, F. Cai, and T. Chi, "MOTIF-RF: Multi-template on-chip transformer synthesis incorporating frequency-domain self-transfer learning for RFIC design automation," in *2026 31st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2026, pp. 1138–1144.

PIRA Project, "PIRA: PI Research Assistant," GitHub repository. https://github.com/AlgebraLoveme/PIRA, 2026.

D. Munzer, S. Er, M. Chen, Y. Li, N. S. Mannem, T. Zhao, and H. Wang, "Residual network based direct synthesis of EM structures: A study on one-to-one transformers," in *2020 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. IEEE, 2020, pp. 143–146.

S. Er, E. Liu, M. Chen, Y. Li, Y. Liu, T. Zhao, and H. Wang, "Deep learning assisted end-to-end synthesis of mm-wave passive networks with 3d EM structures: A study on a transformer-based matching network," in *2021 IEEE MTT-S International Microwave Symposium (IMS)*. IEEE, 2021, pp. 66–69.

F. Passos, N. Lourenço, E. Roca, R. Martins, R. Castro-López, N. Horta, and F. V. Fernández, "PACOSYT: A passive component synthesis tool based on machine learning and tailored modeling strategies towards optimal RF and mm-wave circuit designs," *IEEE Journal of Microwaves*, vol. 3, no. 2, pp. 599–613, 2023.

C. Chu, Y. Mao, and H. Wang, "Transfer learning assisted fast design migration over technology nodes: A study on transformer matching network," in *2024 IEEE/MTT-S International Microwave Symposium (IMS)*. IEEE, 2024, pp. 188–191.

Z. Liu, E. A. Karahan, and K. Sengupta, "Deep learning-enabled inverse design of 30–94 ghz Psat,3dB SiGe PA supporting concurrent multiband operation at multi-gb/s," *IEEE Microwave and Wireless Components Letters*, vol. 32, no. 6, pp. 724–727, 2022.

H. Chae, H. Yu, S. Li, and D. Z. Pan, "PulseRF: Physics augmented ML modeling and synthesis for high-frequency RFIC design," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. ACM, 2024, pp. 1–9.

S. H. Chai, H. Chae, H. Yu, D. Z. Pan, and S. Li, "A d-band InP power amplifier featuring fully AI-generated passive networks," *IEEE Microwave and Wireless Technology Letters*, vol. 35, no. 6, pp. 824–827, 2025.

Y. Guo, E. A. Karahan, Z. Li, Z. Shao, Z. Zhang, M. Wang, and K. Sengupta, "Dall-EM: Generative AI with diffusion models for new design space discovery and target-to-electromagnetic structure synthesis," in *2025 IEEE/MTT-S International Microwave Symposium (IMS)*. IEEE, 2025, pp. 926–929.

S. Ghosh, S. S. Peri, S. Ramprasath, S. A. Berhan, E. Y. Gebru, R. Harjani, and S. S. Sapatnekar, "COmPOSER: Circuit optimization of mm-wave/RF circuits with performance-oriented synthesis for efficient realizations," in *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, 2026.

A Related Work

Neural Modeling of RFIC Passives and Downstream Applications. The high cost of electromagnetic (EM) simulation has motivated neural surrogates for RF and microwave passive components. Existing methods learn geometry- and frequency-conditioned mappings to EM responses, including multiphysics passive models (Yan et al., 2018), RF-device S-parameter models (Guan et al., 2021), on-chip inductor models with dummy metal fillings (Li et al., 2022), recurrent surrogates with domain confinement for passive microwave circuits (Sahu et al., 2025), and uncertainty-aware Bayesian online learning for RF passive components (Zhang and Zhou, 2026). These models primarily accelerate EM-response evaluation, thereby supporting design-space exploration, optimization, uncertainty analysis, and inverse synthesis. For free-form and multi-port structures, Karahan et al. (2024) use a deep forward model of scattering parameters to support inverse design of RF/sub-THz passives and integrated circuits. Our framework differs from this line of work by jointly predicting the explicit EM response and the downstream design performance of free-form PA matching networks under varying frequency and circuit contexts.

RFIC Design with Neural Methods. Neural methods have also been used directly for RFIC design and synthesis. Early direct-synthesis methods map target specifications to transformer geometries for RF/mm-wave matching networks (Munzer et al., 2020; Er et al., 2021), and Lee et al. (2022) propose an automated transformer matching-circuit synthesizer for RF amplifier design. Subsequent tool-oriented and data-efficient methods include PACOSYT for inductor and transformer synthesis (Passos et al., 2023), transfer learning for fast migration of transformer matching networks across technology nodes (Chu et al., 2024), and multi-template transformer synthesis with frequency-domain self-transfer learning (He et al., 2026). For broader free-form and circuit-co-designed synthesis, learned forward models have enabled inverse design of mm-wave passives and PAs (Liu et al., 2022; Karahan et al., 2024), physics-augmented modeling and synthesis in PulseRF (Chae et al., 2024), fully AI-generated passive networks in a D-band PA (Chai et al., 2025), diffusion-based target-to-EM-structure generation (Guo et al., 2025), and end-to-end RF/mm-wave automation flows such as COmPOSER (Ghosh et al., 2026). Closest to our PA matching-network setting, recent 22nm FDX+ demonstrations combine template-guided or template-seeded pixelated dual- and multi-metal-layer passive networks with CNN-based EM emulators to realize measured 6G FR3 CMOS PAs (Chu et al., 2025a,b). These approaches show that neural models can move beyond fast evaluation toward automated RFIC design, but they mainly optimize or synthesize structures through EM-response targets, inverse-design objectives, or full design-flow validation.

B Additional Frequency-Dependent Analyses

This appendix complements the input-matching analyses in Figures 5–7, 9 and 11 with corresponding frequency-dependent analyses for the interstage and output matching settings. We group the figures by objective: scaling behavior across frequency (Appendix B.1), training evolution across frequency (Appendix B.2), dense-frequency-query visualizations for the 21M model (Appendix B.3), and a frequency-level breakdown of dependence regularization (Appendix B.4).

B.1 Frequency-Level Scaling Analysis

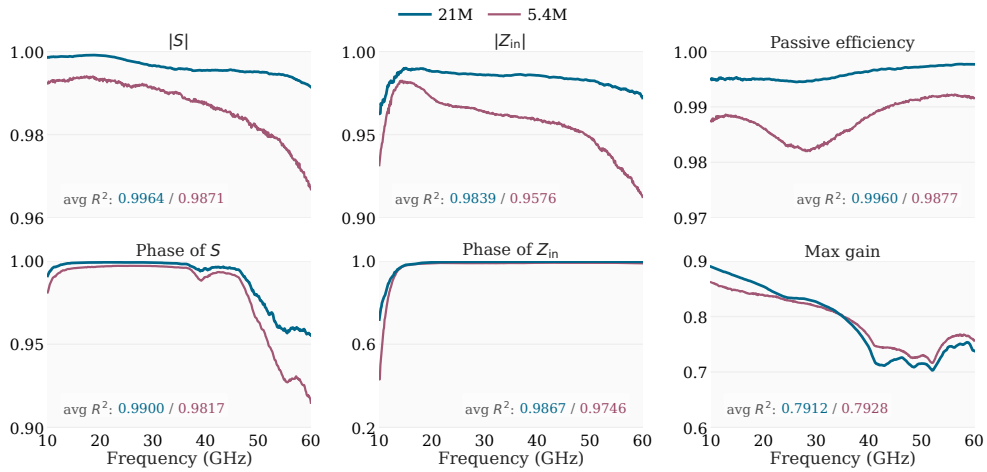


Figure 12: Per-target R^2 over frequency for the 21M and 5.4M interstage matching models on the full distribution.

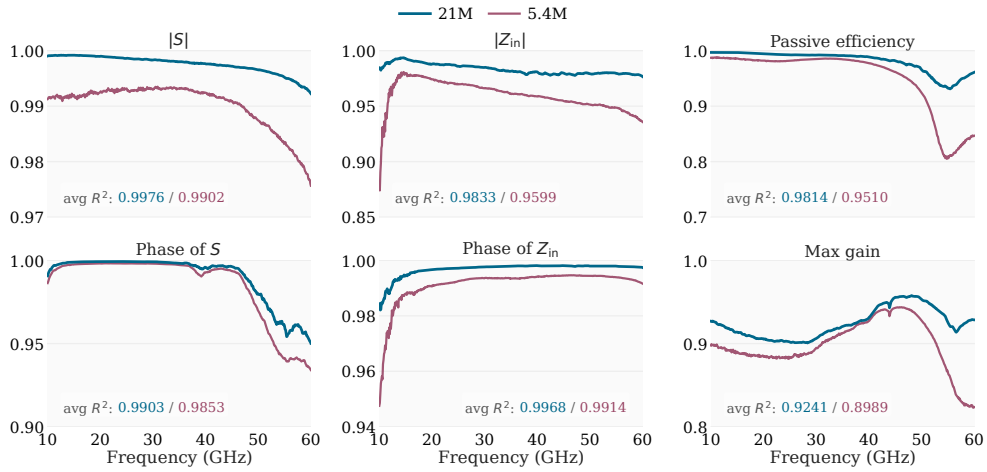


Figure 13: Per-target R^2 over frequency for the 21M and 5.4M output matching models on the full distribution.

B.2 Frequency-Level Training Analysis

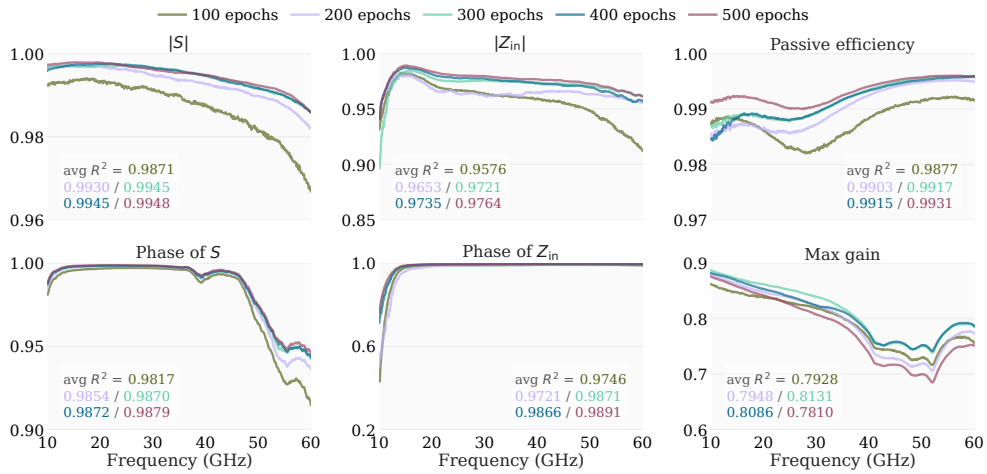


Figure 14: Per-target R^2 over frequency for the 5.4M interstage matching model across training epochs on the full distribution.

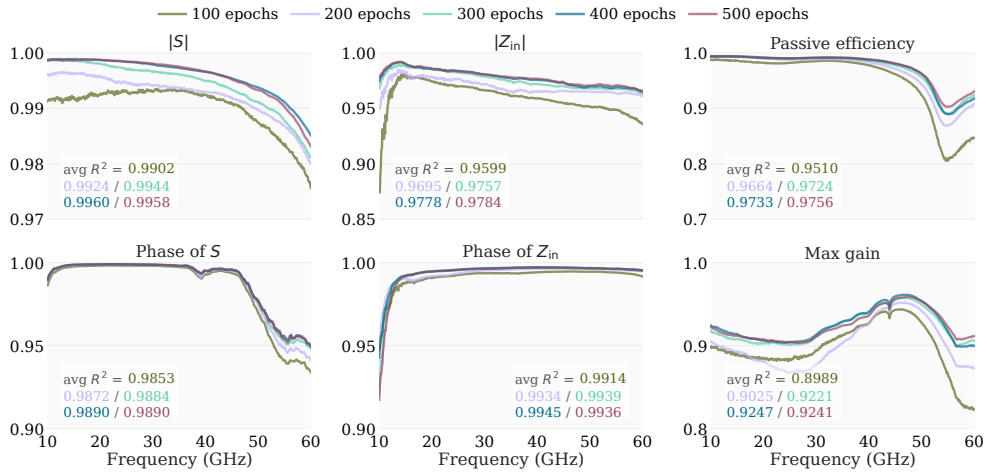


Figure 15: Per-target R^2 over frequency for the 5.4M output matching model across training epochs on the full distribution.

B.3 Dense-Frequency Query Analysis

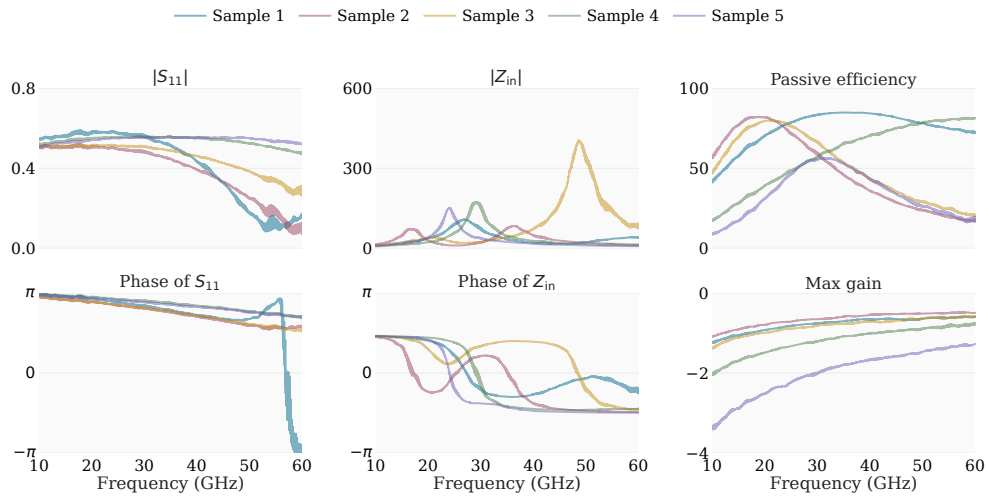


Figure 16: Predictions of the 21M interstage matching model on a frequency grid with 0.001 GHz spacing, which is $100\times$ denser than the training grid.

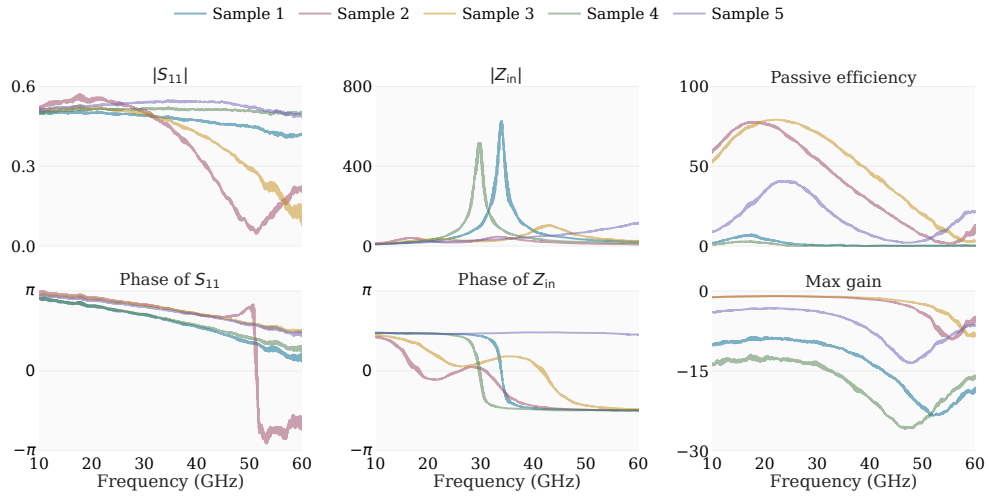


Figure 17: Predictions of the 21M output matching model on a frequency grid with 0.001 GHz spacing, which is $100\times$ denser than the training grid.

B.4 Dependence-Regularization Frequency Analysis

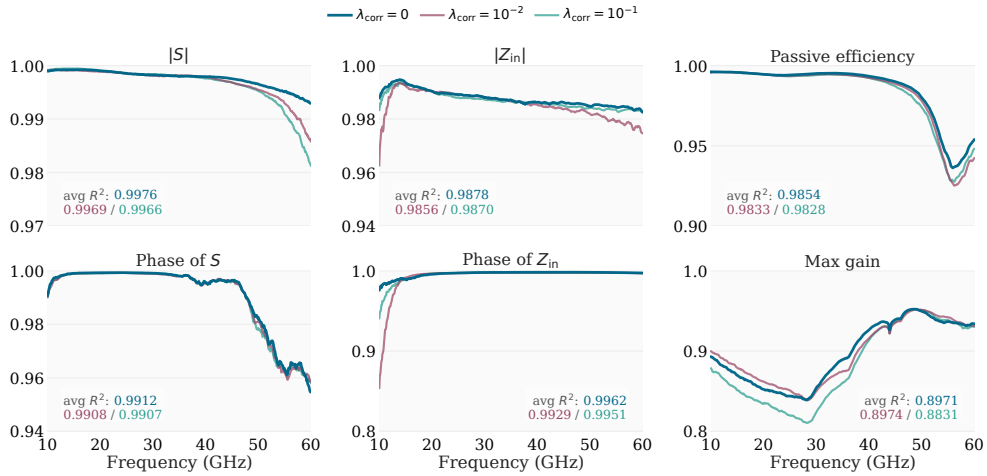


Figure 18: Per-target R^2 over frequency for the 21M input matching model under different dependence-regularization strengths on the full distribution.

C Sensitivity to the Simulator Backbone

We evaluate whether the residual backbone is essential to the performance of the conditional neural simulator–solver framework. We replace the neural simulator's residual backbone with a shallow convolutional neural network (CNN) of comparable parameter count, retaining frequency conditioning and the neural solver. Both models are trained for 100 epochs on the full mixed-distribution training split for input matching, with seed 1, batch size 4 per device, learning rate 10^{-4}, and weight decay 10^{-2}. We evaluate the checkpoint with the lowest validation loss on the full test distribution.

Table 3: Simulator-backbone comparison for input matching after 100 training epochs. Both models retain the conditional simulator–solver framework.

Backbone	Parameters	R^2_{EM}	R^2_{design}
Plain CNN	5,347,884	0.98839	0.94128
Residual	5,383,468	0.98792	0.94386

Table 3 shows comparable aggregate accuracy for the two backbones. The CNN achieves slightly higher R^2_{EM}, while its R^2_{design} is lower by 0.00258. The design-performance difference is concentrated in maximum gain, where the CNN and residual model achieve R^2 values of 0.86045 and 0.87938, respectively. These results indicate that residual blocks are not essential to achieving high aggregate accuracy in this setting.

D Physical Consistency of Predicted EM Responses

Prediction accuracy alone does not guarantee that a learned EM response satisfies physical constraints. Our reduced S-parameter representation enforces reciprocity through reconstruction of the full matrix, but the model does not explicitly enforce passivity. We therefore measure passivity violations in the predicted responses of both model sizes.

We sample 1,000 layouts without replacement from the test set and evaluate all 501 frequencies for each layout. The same layouts are used for both model sizes and all three matching settings. After denormalizing the predictions and reconstructing the full complex S-parameter matrix S, we count a violation when its largest singular value satisfies $\sigma_{\max}(S) > 1 + 10^{-6}$. The reported rate is the fraction of the resulting $N = 501{,}000$ layout–frequency pairs that violate this criterion.

The 21M model exhibits fewer violations in every matching setting, with rates below 7% across the evaluated layout–frequency pairs. Neither model guarantees passivity. For reference, the ground-truth

Table 4: Passivity-violation rates in percent. Error bars are one IID Bernoulli standard error over layout–frequency pairs, computed as $100\sqrt{\hat{p}(1-\hat{p})/N}$ for the observed violation fraction $\hat{p}$. This calculation treats pairs as independent and does not account for dependence among frequencies of the same layout.

Matching	5.4M	21M
Input	7.118 ± 0.036	3.925 ± 0.027
Interstage	8.170 ± 0.039	4.134 ± 0.028
Output	13.874 ± 0.049	6.890 ± 0.036

responses on the same sample have a violation rate of 0.138% under the same numerical criterion. Thus, improved prediction accuracy accompanies fewer passivity violations, while explicit physical consistency remains a separate requirement.

E Solver-Stability Evaluation Protocol

Figure 1 compares the robustness of the learned neural solver and the ground-truth (GT) solver when their input responses are imperfect. The experiment tests how small response errors propagate to design-performance predictions in the input-matching setting.

E.1 Noise Injection and Solver Evaluation

For the GT solver, we add independent zero-mean Gaussian noise to every dimension of the ground-truth EM response. We vary the noise standard deviation from 1% to 20% of the empirical standard deviation of the corresponding response dimension, in increments of one percentage point, and include a noiseless reference. Noise is applied to the normalized magnitude and phase dimensions before denormalization and reconstruction of the complex S-parameter matrix. We then feed this matrix to the GT solver, keeping frequency and circuit context unchanged.

For the learned solver, we perturb the predicted neural EM response using the same dimension-wise noise-scaling rule. This input includes both the explicit EM response and the complementary response. The trained solver remains fixed throughout the experiment, and frequency and circuit context remain unchanged. We compare the resulting design-performance predictions from both solvers against the ground-truth design targets using R^2_{design}. For this study, we compute each design-target score separately at each frequency over layouts and circuit contexts, then average over frequencies and targets. The study implementation uses $1 - \mathrm{MSE}/s_y^2$, where s_y^2 is the sample variance with denominator $n - 1$ for n observations. The EM-response score uses the same variance convention, pooling layouts and frequencies within each response dimension before averaging over dimensions.

E.2 Magnitude and Effect of EM Simulation Response Noise

The GT solver's design-performance R^2 already becomes negative at 4% noise. To illustrate how much this noise level changes the simulated EM response, Table 5 shows the first complex S-parameter entry of three randomly sampled examples before and after perturbation.

Table 5: Example S-parameter values before and after adding 4% noise. Here j denotes the imaginary unit.

Example	Before	After
1	$-0.4457 + 0.2717j$	$-0.4396 + 0.2774j$
2	$-0.1701 + 0.3195j$	$-0.1654 + 0.3267j$
3	$-0.3637 + 0.3805j$	$-0.3585 + 0.3726j$

These examples show that the added noise changes the simulated values only slightly, preserving the dominant EM signal. The perturbed EM responses also retain an aggregate R^2_{EM} of 0.9984, confirming that the response remains close to the ground truth despite the GT solver's degraded performance. In contrast, the learned solver retains $R^2_{\mathrm{design}} = 0.9808$ at the same relative noise level, compared with 0.9814 without added noise. Its accuracy decreases gradually as the noise magnitude

increases. We do not inject artificial response noise during training to enforce this stability. The behavior emerges through joint training, possibly because the learned solver is exposed to imperfect EM-response predictions from the neural simulator. The GT solver's sensitivity to these small EM-response errors motivates direct joint prediction of EM response and design performance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The introduction states the motivation, scope, contributions, model components, dataset, accuracy, and speed claims. These claims are supported by the method and experiment sections, with limitations summarized in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 5, including the fixed-grid and process-specific setting, separate models for each matching placement, limited cross-distribution generalization, interpolation-only frequency evaluation, training cost, unquantified random-seed variability, and lack of guaranteed passivity.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results requiring assumptions and proofs. The mathematical expressions in Section 3 define the model objective and regularizer.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 2–4 disclose the task, dataset construction, model architecture, objective, data split, optimization setup, training duration, and evaluation protocol. App. C–E provide the additional backbone, physical-consistency, and solver-perturbation evaluation protocols. The dataset cannot be released due to commercial licensing constraints, but the paper describes the dataset construction details needed for reproduction, and the provided code supports reproducing the main experimental results supporting the paper's claims.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is provided, but the dataset cannot be publicly released due to commercial licensing constraints. To support reproducibility despite this restriction, the paper documents the dataset construction, model, training setup, and evaluation protocol in Sections 3 and 4, providing the details needed to recreate the dataset and reproduce the reported results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 specifies the data split, target and conditioning normalization, augmentation, model scales, optimizer, learning-rate schedule, batch size, checkpoint selection, training duration, and hyperparameter-selection protocol. Additional evaluation settings are described in App. C–E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The main predictive-performance results are from single training runs, and variability across random seeds is not quantified, as stated in Sections 4 and 5. App. D reports one IID Bernoulli standard error for passivity-violation rates and explicitly states the independence assumption. These error bars do not measure training-seed variability or account for dependence among frequencies of the same layout.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 reports the main hardware setup, including 8 NVIDIA GeForce RTX 2080 Ti GPUs, per-GPU batch size, storage footprint, and approximate training time for the 5.4M and 21M models. These details specify the resources used for the main experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses simulated RFIC passive-layout data and neural surrogate modeling. We are not aware of any deviation from the NeurIPS Code of Ethics.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper focuses on technical modeling of RFIC passives and does not include a dedicated broader-impact discussion. The main expected impact is faster RFIC passive design, with limited direct societal-risk pathways.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released asset is code, not a high-risk model, scraped dataset, human-subject dataset, or public release of the RFIC mask dataset requiring special misuse safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The work does not use existing external datasets or pretrained models as research assets. Relevant prior methods and software components are cited where they are discussed.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper provides code to support reproducibility and future research. The dataset construction, model setup, and experimental details are documented in Sections 3 and 4.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject experiments, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLM assistance is disclosed in the unnumbered LLM Disclosure section immediately before the references. The assistance included brainstorming, partial implementation assistance, and writing polish. The authors remain fully responsible for the final content.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.