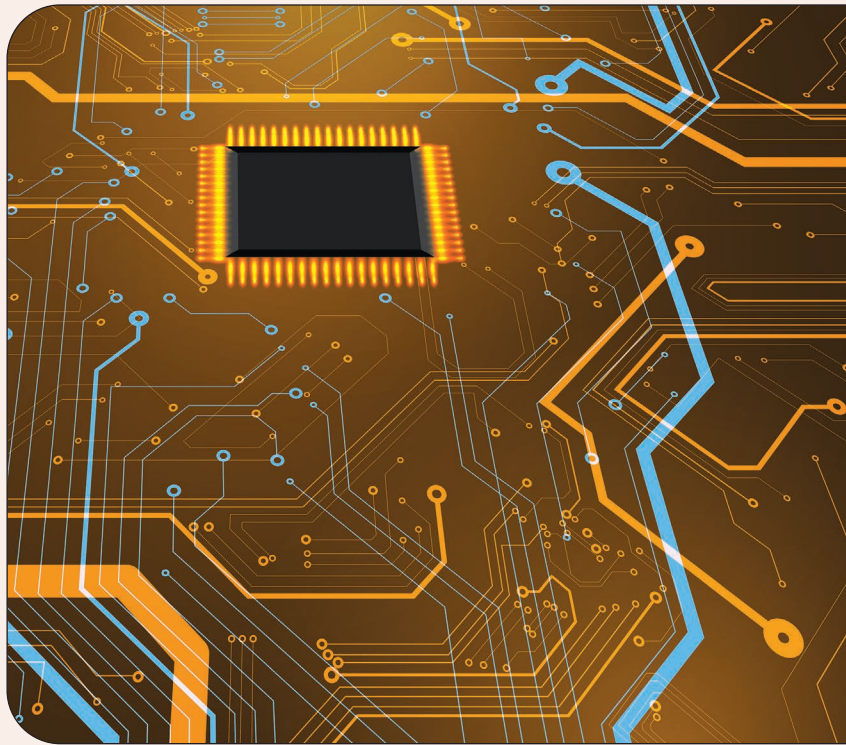


Accelerating RF Device Modeling and Circuit Design via Derivative-Free Optimization

*Rafael Perez Martinez¹,
Tae Hoon Choi,
Masaya Iwamoto,
Stephen Boyd², and
Srabanti Chowdhury³*

The design and development of RF circuits and systems span several levels of abstraction. At the device level, accurate transistor compact models serve as a vital bridge between physical semiconductor devices and electronic design automation (EDA) tools. These models mathematically represent the electrical and thermal behavior of active devices. Before they can be used for circuit



©SHUTTERSTOCK.COM/HNK

Rafael Perez Martinez (rafapm@alumni.stanford.edu) and Tae Hoon Choi (tae-hoon.choi@broadcom.com) are with the Wireless Semiconductor Division, Broadcom, Inc., San Jose, CA 95131 USA. Masaya Iwamoto (masaya_iwamoto@keysight.com) is with Keysight Labs, Keysight Technologies, Inc., Santa Rosa, CA 95403 USA. Stephen Boyd (boyd@stanford.edu) and Srabanti Chowdhury (srabanti@stanford.edu) are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA. Masaya Iwamoto is a Member of IEEE, Srabanti Chowdhury is a Fellow of IEEE, and Stephen Boyd is a Life Fellow of IEEE. Rafael Perez Martinez is the corresponding author.

Digital Object Identifier 10.1109/MMM.2026.3713927

design, their parameters must be extracted so that the compact model reproduces device characteristics obtained from experimental measurements or technology computer-aided design (TCAD) simulations with sufficient accuracy [1]. In recent years, however, parameter extraction has become increasingly complex. As new semiconductor technologies emerge, compact models have expanded to include more parameters to capture a broader range of physical effects not present in earlier technologies [2]. For instance, the Advanced SPICE Model for GaN HEMTs (ASM-HEMT) includes more than 200 model parameters [3].

Once the compact model has been extracted, the circuit design process begins. Compact models maintain a sufficient level of simplicity to be integrated into circuit simulators while still providing the accuracy needed to deliver reliable results across various geometries and a wide range of bias conditions [4], [5]. Using these models, a designer can synthesize a circuit comprising tens of transistors and passive components to achieve a desired RF response. This introduces another layer of complexity, as designers must simultaneously satisfy multiple objectives and stringent constraints. When optimizing for more than one objective, the goal is typically to identify Pareto-optimal designs: that is, designs that are not dominated by any other feasible solution and therefore represent the best achievable tradeoffs among competing objectives [6].

Traditionally, both parameter extraction and circuit design have relied heavily on expert intuition and manual tuning. In device modeling, achieving a good fit often involves a series of iterative steps that can extend over several days or weeks just to extract a single model card. This approach frequently leaves device modeling engineers uncertain whether the resulting model card represents a near-optimal parameter set or whether the model fit can be further improved. Similarly, circuit design often requires extensive simulations to understand underlying behaviors and meet target specifications, making it difficult to determine whether performance was left on the table when balancing competing objectives.

To reduce manual effort, engineers frequently turn to the built-in optimizers available in standard EDA tools [7]. Historically, these optimizers have been gradient-based, relying on algorithms, such as Levenberg-Marquardt or gradient descent [8]. However, most EDA tools do not readily provide gradient information, and it is not possible to obtain gradients directly from physical experiments. Consequently, gradients must often be approximated using computationally intensive numerical methods. While gradient-based optimization can be effective for fine-tuning a nearly complete model card or circuit design, it is often susceptible to becoming trapped in local minima unless the user

strictly constrains the parameter ranges, reduces the number of parameters, or provides an excellent initial guess.

In this article, we review how advanced derivative-free optimization (DFO) methods can be applied to RF/microwave design. DFO methods seek to minimize an objective function using only objective values, meaning that no gradient information is required. Beyond simply not using gradients, DFO methods provide a practical and reproducible framework for solving expensive black-box optimization problems. When the variables, constraints, and objective functions are carefully crafted, these methods can substantially reduce the cost of exploring high-dimensional RF design spaces that would otherwise be computationally prohibitive to search directly [9], [10], [11].

To make these ideas concrete, we focus on two closely related applications in the RF design flow. First, we consider compact model parameter extraction, where tens of model parameters must be extracted simultaneously to reproduce DC device behavior accurately across the relevant bias conditions [2]. Second, we consider robust transistor sizing, where competing objectives and robustness considerations must be incorporated directly into the optimization problem [6]. Together, these examples show how DFO methods can make black-box optimization in RF/microwave design both practical and computationally efficient.

Overview of DFO in RF Design

Many core problems in RF/microwave engineering can be written as optimization problems

$$\begin{aligned} &\text{minimize } F(x) \\ &\text{subject to } x \in X \end{aligned} \quad (1)$$

where x denotes a vector of design variables and the feasible set X captures physical bounds, discrete choices, and implicit feasibility constraints [12]. The form of $F(x)$ depends on the RF task of interest.

Device Modeling and Parameter Extraction

In compact model parameter extraction, we seek to determine the model parameters x that make the simulated curves align as closely as possible with the measured or TCAD-generated data. This can be formulated through the objective function

$$F(x) = \frac{1}{N} \sum_{i=1}^N L(\hat{y}_i(x), y_i) \quad (2)$$

where $L(\hat{y}_i(x), y_i)$ denotes the loss function to be minimized with respect to the model parameters x . Here, x is the optimization variable, $\hat{y}_i$ is the simulated value predicted by the compact model, and y_i is the corresponding measured or TCAD-generated value [2].

Multiobjective Circuit Design

In circuit design, the optimization problem often involves k competing objectives, denoted by $f_1(x), \dots, f_k(x)$. In such cases, the objective vector can be written as

$$F(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T. \quad (3)$$

In this multiobjective setting, a design is said to dominate another if it performs at least as well in every objective and strictly better in at least one. A design that is not dominated by any other feasible solution is said to be *Pareto optimal*. The set of such designs forms the Pareto frontier in the case of two objectives, or more generally, the Pareto surface in higher dimensions [6].

Practical Intuition for RF Optimization

At a fundamental level, derivative-free optimization follows an iterative process: 1) Propose a candidate design vector x based on historical data, and 2) evaluate the objective or objective vector $F(x)$ through simulation or measurement. This loop is then repeated for a fixed number of evaluations, which is often the most practical stopping criterion when simulations are expensive, or until satisfactory performance is reached [10].

This approach aligns seamlessly with RF/microwave engineering, as many practical design problems are already black-box in nature. For instance, in circuit design, a designer selects a set of parameters, launches an expensive circuit or electromagnetic (EM) simulation, and then analyzes the results to decide on the next iteration. From this perspective, DFO is not a departure from traditional RF workflows. Instead, it automates the manual processes that engineers already perform, making them more efficient, systematic, and reproducible.

Unlike gradient-based optimization methods, DFO does not require the objective function to be differentiable, nor does it assume that derivative information is analytically available. This is particularly important in RF workflows because design spaces often include discrete choices, feasibility constraints, nonconvergent simulations, and strongly coupled parameters [1]. Even when gradients can in principle be computed, their cost is not justified, especially when each function evaluation corresponds to an expensive simulation, such as a modulated large-signal simulation or a high-fidelity 3D EM extraction.

In practice, DFO is best viewed in RF design as an intelligent way to allocate a limited simulation or measurement budget, rather than as optimization without gradients. In addition, when each evaluation is expensive, the key question in RF design is not whether the global optimum can be guaranteed, since in most practical nonconvex problems it cannot [12], but whether the available simulation budget is being used effectively. DFO addresses this problem by using prior evaluations to decide where to search next, making each

new simulation more informed and therefore making better use of the available budget.

Many modern DFO methods implement this idea through some balance of global exploration and local exploitation. Early in the search, the algorithm samples broadly to identify promising regions of the design space. As more evaluations are collected, the search gradually becomes more focused on high-performing regions. In probabilistic and population-based methods, this is often done by identifying the best-performing subset of evaluated designs and using their distribution to guide subsequent proposals. In practical RF terms, the optimizer gradually shifts from learning where good designs might be to spending more of the budget near designs that are most likely to yield further improvement [13], [14].

To illustrate why this behavior is valuable in RF optimization, Figure 1 shows a representative objective-function landscape from RF device modeling. Inspired by loss-landscape visualizations from the machine learning (ML) community [15], [16], the high-dimensional parameter space is projected onto a 2D subspace defined by two principal components (PC1 and PC2). The resulting contour and 3D surface plots reveal a highly nonconvex projected landscape with multiple local minima. In such settings, gradient-based optimization methods can be sensitive to initialization and may converge to nearby local solutions. In comparison, the superimposed projected trajectory in the contour plot illustrates the broader exploratory behavior of a DFO algorithm, which can help the search traverse multiple basins and identify higher-quality solutions.

A Taxonomy of DFO Algorithms

DFO encompasses a broad range of algorithms, which can be grouped roughly into random-search methods, direct-search methods, population-based methods, and surrogate-based methods. This classification is not exhaustive, but it provides a useful framework for RF/microwave engineers. More important than the category itself, however, are the practical tradeoffs associated with each method, since engineers must weigh robustness, sample efficiency, and implementation complexity against the cost of each evaluation.

At the simplest end of the spectrum, a random search evaluates candidate designs by sampling points directly from the feasible design space [17]. Although conceptually simple, it remains an important baseline because it is easy to implement, straightforward to parallelize, and sometimes competitive when only a subset of variables strongly influences the objective. In most RF problems, however, its main value is as a reference point rather than a final design tool.

Direct-search methods, such as coordinate search or pattern search, improve on this idea by evaluating nearby points and accepting improving moves [10].

These methods are appealing because of their simplicity and can work well for local improvement, particularly in lower-dimensional problems or when a good initial design is available. Their main limitation is that they are less well suited to many RF compact model extraction and circuit-design problems, where the design space is high-dimensional and the variables are strongly coupled.

Population-based methods work with a set of candidate designs at each iteration rather than only one current design. A well-known example is the covariance matrix adaptation evolution strategy (CMA-ES), which adapts a search distribution over time to favor promising regions of the design space [18]. These methods can be robust in rugged or multimodal landscapes and are often less sensitive to initialization, but this robustness typically comes at the cost of more function evaluations.

Surrogate-based methods use previously collected data to construct an auxiliary approximation of the

objective function and then use that model to guide the next expensive evaluation [19]. These methods are particularly attractive when each evaluation is costly and only a limited number of evaluations can be afforded. Bayesian optimization is the best-known example. Related methods, such as the Tree-Structured Parzen Estimator (TPE), follow the same sequential model-based philosophy, but are often convenient when the design space includes a mix of continuous, integer, categorical, or log-scaled variables [20].

The main tradeoffs and RF/microwave applications of the aforementioned DFO methods are summarized in Table 1. Ultimately, in RF/microwave design, no single DFO algorithm is uniformly best. The right choice depends on the cost of each evaluation, the structure of the design space, and whether the main need is local refinement, global exploration, or efficient use of a limited evaluation budget.

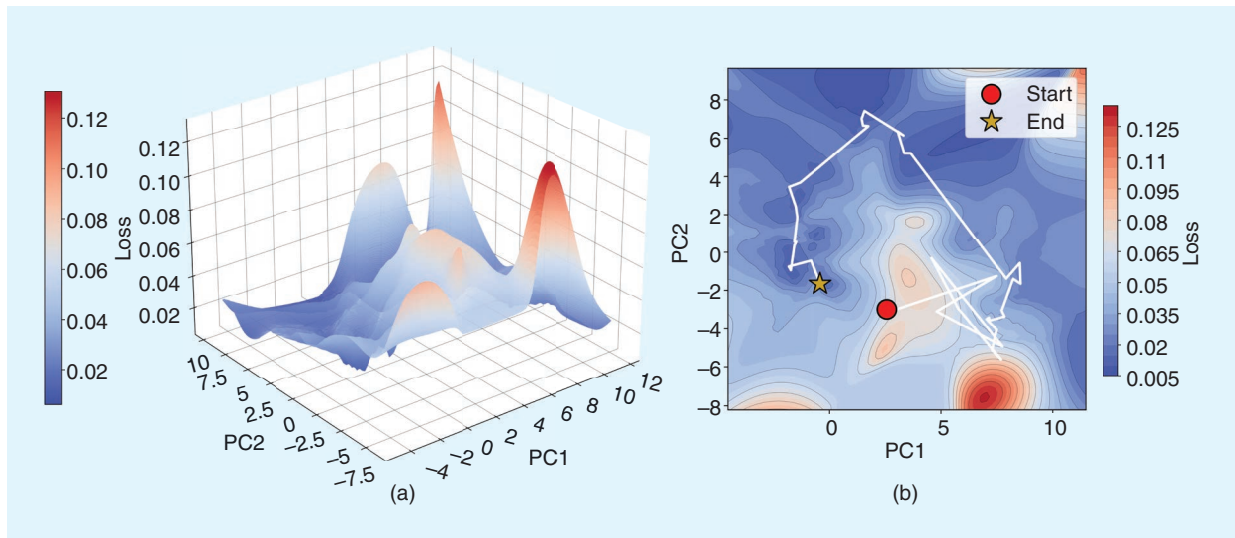


Figure 1. Representative objective-function landscape from RF device modeling visualized in a 2D principal-component subspace. (a) Three-dimensional surface plot of the loss as a function of the first two principal components, PC1 and PC2. (b) Corresponding contour plot with an illustrative projected search trajectory overlaid. The projected landscape is highly nonconvex, with multiple basins and local minima. Such visualizations help explain why local optimization methods can be sensitive to initialization, while derivative-free methods with stronger exploratory behavior may traverse multiple basins and discover better solutions.

TABLE 1. Representative DFO methods, key tradeoffs, and example applications in RF/microwave design.

DFO Method	Well Suited For	Key Tradeoffs	RF Example
Random search	Early exploration or baseline studies	Simple to implement and parallelize, but often requires many trials	Initial parameter sweeps for device or circuit optimization
Pattern or coordinate search	Local refinement near a good design	Straightforward to apply, but sensitive to the starting point	Fine-tuning compact model parameters or matching networks
CMA-ES and evolutionary methods	Rugged or strongly coupled design spaces	Broad exploration of complex landscapes, but usually requires many trials	Compact model parameter extraction with low-cost simulations
Bayesian optimization and TPE	Expensive evaluations or mixed continuous–discrete spaces	Uses past trials efficiently, but can be less effective in high dimensions	Load-pull, EM, or mixed-variable circuit design

Compact Model Parameter Extraction

Compact model parameter extraction is a clear example of a black-box optimization problem in RF/microwave design. A candidate parameter vector is proposed, the compact model is evaluated in a simulator, and the simulated characteristics are compared against measured or TCAD-generated data through a loss function. This loop is repeated until the mismatch between simulation and data is reduced to an acceptable level [2].

In the examples that follow, the DFO workflow was implemented using Optuna [14] with the TPE algorithm, which models favorable and unfavorable regions of the parameter space and then samples new candidates from regions predicted to perform well [20].

A practical distinction between the conventional manual extraction flow and the proposed DFO-based workflow can be seen in Figure 2. In a conventional manual extraction process, parameters are often tuned sequentially by physical effect or operating region, with repeated fine-tuning required as parameter coupling

causes changes in one region of operation to degrade the fit in another. On the other hand, the proposed approach formulates parameter extraction as a single optimization problem in which parameter ranges, loss functions, and validation strategy are defined up front, and the optimizer iteratively searches for a parameter set that provides a good fit across the full training dataset.

Although the workflow may appear conceptually straightforward, the main difficulty lies in the interaction among model complexity, parameter coupling, and objective-function design. Modern compact models can contain tens or even hundreds of parameters, many of which affect multiple device characteristics simultaneously. As a result, the optimization landscape is often highly nonconvex, and sequential manual tuning becomes slow, labor-intensive, and difficult to standardize across different devices.

More importantly, parameter extraction is not solved simply by embedding a good optimizer into a simulator. A low numerical error alone is not convincing evidence of

a reliable compact model extraction. The outcome still depends critically on which parameters are exposed to optimization, what bounds are imposed, how fit quality is measured, and how the extracted model is validated beyond the data used directly during fitting. In this sense, DFO does not replace compact model expertise, but instead provides a systematic way to bring that expertise into the extraction workflow.

The main takeaway from this case study is not that DFO can automate compact model parameter extraction. Rather, it becomes useful only when the optimization workflow is built around carefully chosen model parameters, physically meaningful parameter bounds, a loss function tailored to compact modeling rather than borrowed naively from ML, and a validation strategy that checks whether the model remains reliable beyond the fitting set [2].

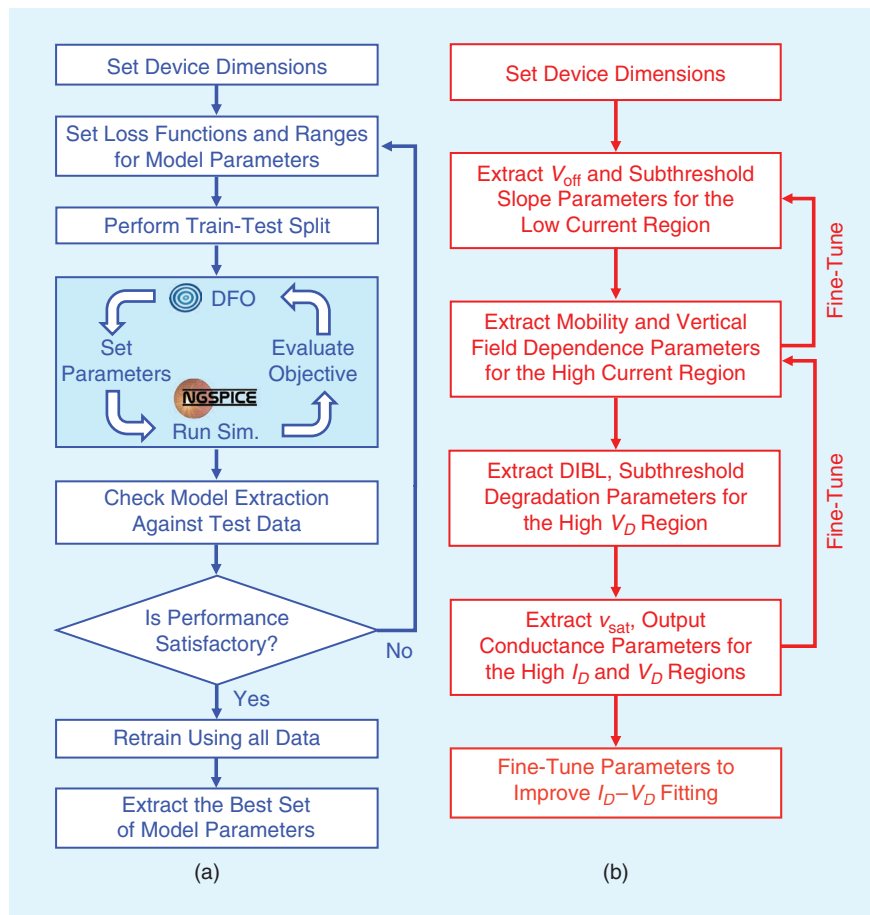


Figure 2. Comparison of two workflows for DC model parameter extraction in ASM-HEMT: (a) The proposed DFO-based workflow, in which parameter ranges, loss functions, and a validation strategy are defined in advance and the search is performed iteratively (adapted from [2]); and (b) the conventional manual workflow, in which parameters are extracted and repeatedly fine-tuned by operating region (adapted from [3]).

Loss Function Design

In compact model parameter extraction, the optimizer is only one part of the story. A fundamental issue is the

careful construction of a loss function that reflects the main priorities in device modeling. For instance, naively using the absolute error (or L1 loss) to fit DC I-V characteristics tends to place more weight on larger current values, often leading to poor accuracy at lower currents

$$\mathcal{L}_1(\hat{y}_i, y_i) = |\hat{y}_i - y_i|. \quad (4)$$

To address this issue, we apply a log transformation, which shifts the emphasis of the loss function from absolute errors to relative errors

$$\mathcal{L}_{\log}(\hat{y}_i, y_i) = |\log(\hat{y}_i) - \log(y_i)|. \quad (5)$$

We also introduce a threshold parameter ϵ into the logarithmic transformation

$$u(\hat{y}, y) = \log\left(1 + \frac{\hat{y}}{\epsilon}\right) - \log\left(1 + \frac{y}{\epsilon}\right). \quad (6)$$

This transformation has a simple interpretation in the context of DC I-V characteristics. For current levels

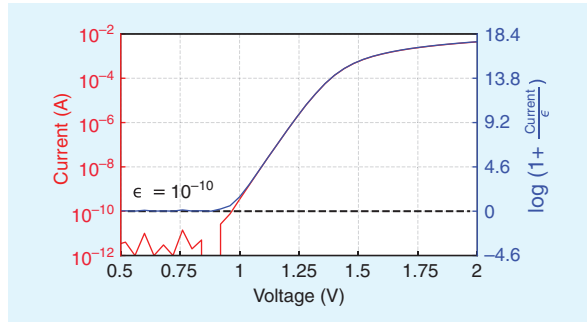


Figure 3. Transformation of the loss function used to reduce the influence of current values below a threshold ϵ during fitting. The left y-axis (red) shows the original current values, while the right y-axis (blue) shows the transformed current values used in the loss function (after [2]).

below the threshold ϵ , the contribution to the loss is strongly reduced. In practice, this discourages the optimizer from spending excessive effort in regions that are physically unimportant, such as those dominated by measurement noise.

The effect of this transformation on the fitting objective is illustrated in Figure 3. The red curve shows the original current on a logarithmic scale, while the blue curve shows the transformed quantity used in the loss function. Once the current falls below ϵ , the transformed response is effectively flattened, so differences in that region contribute much less to the optimization objective. As a result, the fitting process places greater emphasis on current levels and bias regions that are more relevant for model extraction.

To further improve robustness to outliers or measurement error, we clip the transformed loss using a threshold δ

$$\mathcal{L}_{\text{clip}}(\hat{y}, y) = \begin{cases} u^2, & |u| \leq \delta \\ \delta^2, & |u| > \delta \end{cases}. \quad (7)$$

This clipped loss imposes a fixed penalty once the magnitude of the transformed error exceeds δ , which reduces the sensitivity of the fitting process to outliers and corrupted data [12].

Figure 4 conceptually illustrates how clipping improves robustness to large residuals. Under a conventional quadratic loss, a small number of large residuals can dominate the objective and pull the fitted model away from the main data trend. In the clipped-quadratic loss, however, the influence of such residuals is limited once they exceed the threshold δ , so the resulting fit is determined more by the majority of the data. This behavior is evident in Figure 4(c), where the penalty saturates beyond the clipping threshold. In the extraction process, this improves robustness to outliers, corrupted samples, and occasional measurement artifacts.

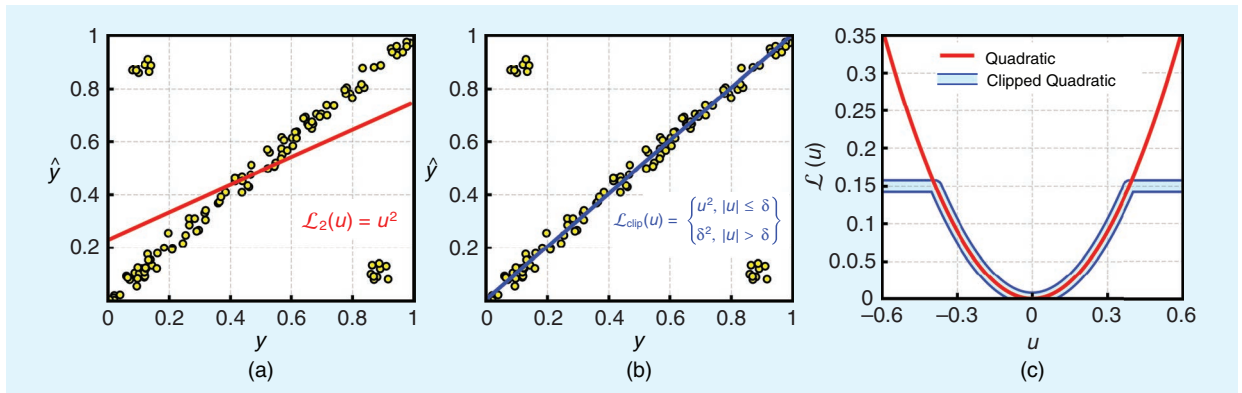


Figure 4. Conceptual illustration of robust regression for data fitting and parameter extraction, adapted from [21]. (a) and (b) A regression analogy for the effect of outliers under quadratic and clipped-quadratic losses. (c) Comparison of the corresponding quadratic and clipped-quadratic penalties as functions of the residual variable u . In the compact model extraction workflow considered here, clipping is applied to the transformed residual u defined in (6).

Model Assessment

When performing parameter extraction with tens of parameters, model assessment becomes essential. The goal is not simply to obtain a low fitting error on the data used during extraction, but to ensure that the compact model remains physically valid across different voltages, currents, and operating conditions that were not used for fitting. This issue is closely related to generalization in statistics and ML, where good agreement with training data alone does not guarantee good performance on unseen data [22].

To address this issue, the available data in each experiment are divided into training and test sets (e.g., 80% for training and 20% for testing). The optimizer uses only the training set during extraction, while the test set provides an independent check on whether the model generalizes beyond the fitting data. This distinction is important because, as the number of fitted parameters increases, it becomes easier to improve

agreement on the training set without achieving comparable improvement on unseen data. In the following examples, both training and test errors are tracked throughout the optimization process to guard against overfitting and assess generalization.

Pedagogical Example: Diamond Schottky Diode

The DFO workflow is first demonstrated through a simple three-parameter example using the SPICE diode model [23] to fit the DC I–V characteristics of a diamond Schottky diode [24], [25]. The fitted parameters are the ideality factor (n), saturation current (I_s), and series resistance (R_s).

Although this example is relatively simple and does not, by itself, require a derivative-free optimizer, it provides a clear setting in which to illustrate the main ideas introduced earlier. In particular, it highlights the role of the loss function, the importance of model assessment during parameter extraction, and the effect of robustness in the presence of corrupted data.

Because many DFO algorithms are stochastic, optimization progress is often reported in terms of the best error found to date, denoted here as Error^* , as a function of the number of trials. This is also the style of the convergence plot commonly used in the optimization and ML communities [20]. The progression of Error^* for the training and test sets, together with the retrained result on the full dataset, is shown in Figure 5. The plot illustrates how the train-test split can be used to monitor generalization during parameter extraction, while the retraining step produces the final parameter set using all available data.

Finally, the diode example provides a convenient way to assess robustness to outliers by intentionally introducing corrupted data points. Under these conditions, the clipped-loss formulation remains much less sensitive to the corrupted samples, whereas the non-robust loss exhibits visibly degraded fitting performance. The effect of corrupted samples on the fitted diode characteristics can be seen in Figure 6. Even when 50% of the measured samples are randomly corrupted, the fit obtained with the robust loss remains closer to the main I–V trend, while the nonrobust fit is more strongly affected by the outliers. This simple example highlights why robust loss design can be important in compact model parameter extraction, especially when the data contain measurement artifacts or corrupted samples.

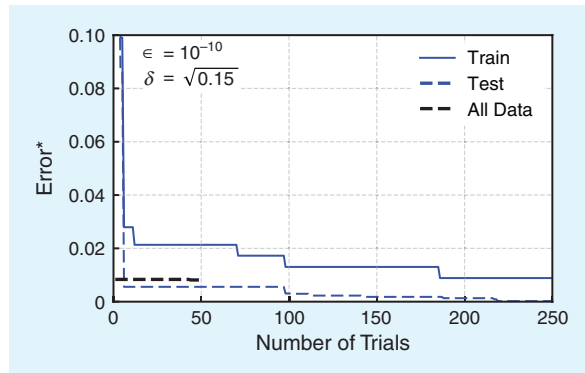


Figure 5. Progression of Error^* across 250 trials for the diamond Schottky diode example, showing the training and test curves, together with the error obtained after retraining on the full dataset. The loss function used hyperparameter values of $\epsilon = 10^{-10}$ and $\delta = \sqrt{0.15}$ (after [2]).

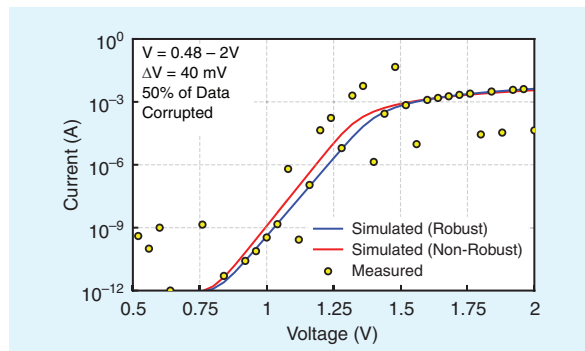


Figure 6. Measured and simulated I–V characteristics of a diamond Schottky diode using robust and non-robust loss functions. The voltage sweep ranges from 0.48 to 2 V in 40-mV steps, for a total of 39 measurements. In this example, 50% of the measurements were randomly corrupted to evaluate the robustness of the two loss functions (after [2]).

GaN HEMT Compact Model Extraction

After building intuition from the diode example, the same workflow is applied to the DC portion of the ASM-HEMT model for a 150-nm GaN-on-SiC HEMT. In this case, 35 parameters are extracted simultaneously

to capture the device's DC characteristics, including transport parameters, access resistances, self-heating terms, and gate-current parameters [3].

The resulting agreement between measured and simulated DC characteristics is shown in Figure 7. The extracted parameter set provides a good overall fit to both the output characteristics in Figure 7(a) and the transconductance behavior in Figure 7(b), indicating that the optimization captures the main bias-dependent trends of the device across a wide range of bias conditions. This result is especially important because the parameters are extracted simultaneously rather than being tuned one operating region at a time, allowing the final solution to balance coupled effects within a single optimization workflow.

This case study more closely reflects the difficulty of practical compact model extraction, where the parameter space is high-dimensional and many of the fitted quantities are strongly coupled [2]. Under such conditions, sequential hand-tuning becomes slow, labor-intensive, and difficult to reproduce. In contrast, the DFO-based workflow is able to obtain a good fit in fewer than 6,000 trials. Although physical intuition remains essential for defining meaningful parameter bounds and selecting the relevant model parameters, DFO substantially reduces manual tuning effort and improves reproducibility.

It is important to emphasize, however, that optimization alone does not resolve all of the challenges in compact modeling. If the selected model is inadequate and fails to capture the relevant device physics, a good fit may be impossible regardless of the optimizer [4]. The same is true when critical physical effects are omitted, when important model parameters are excluded from the optimization, or when the imposed constraints are not physically meaningful. In such cases, the optimizer may reduce the error for one set of characteristics, yet the extracted model may still fail to remain physical or to generalize to others. DFO should therefore be viewed as a powerful tool for parameter extraction, but not as a substitute for model adequacy, careful parameter selection, physically meaningful constraints, or validation under the operating conditions relevant to subsequent circuit design.

Robust Pareto Transistor Sizing

Once an accurate compact model with geometry scaling has been extracted and validated across a wide range of bias conditions and temperatures, the next step is to use that model for circuit design. A typical challenge for circuit designers is to find the best transistor size for the intended RF application. In practice, however, this is rarely a single-objective optimization problem. Instead, there are usually many candidate solutions with competing tradeoffs.

For millimeter-wave power amplifiers, output power and efficiency must be balanced against thermal constraints, stringent linearity requirements, and reliability [26]. For low-noise amplifiers (LNAs), gain and $f_{\max}$ must be balanced against minimum noise figure. More importantly, the best device size depends on the operating conditions, since a design that appears optimal at one nominal bias point may shift or degrade significantly as the bias changes during typical circuit operation.

As a result, transistor sizing can be formulated naturally as a robust multiobjective problem rather than as a nominal search for a single best device size. The goal is not simply to identify a device geometry that looks attractive under one nominal condition, but to find designs that remain competitive across realistic operating variation. From this viewpoint, robustness is not something to be considered only after optimizing for a particular transistor size. In many cases, it changes which designs are worth considering at all. This is what makes transistor sizing a natural candidate for DFO [6].

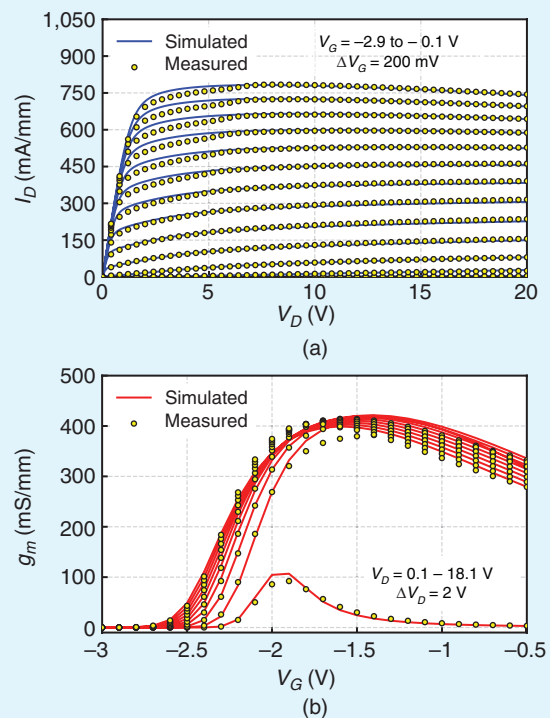


Figure 7. Measured and simulated (a) I_D and (b) g_m characteristics of the GaN HEMT after DC parameter extraction. For (a), the drain voltage V_D was swept from 0 to 20 V with a step size of 100 mV, and the gate voltage V_G was swept from -2.9 to -0.1 V with $\Delta V_G = 200$ mV. For (b), the gate voltage V_G was swept from -3 to -0.5 V with $\Delta V_G = 100$ mV, and the drain voltage V_D was swept from 0.1 to 18.1 V with $\Delta V_D = 2$ V (after [2]).

Robust Pareto Formulation

To account for bias sensitivity, transistor sizing can be formulated as a robust multiobjective optimization problem. Let x denote the design variables and let $p \in \mathcal{P}$ denote an operating parameter. In this setting, the operating parameter is V_{GS} , while the design variables correspond to transistor bias and geometry choices under the designer's control. Because some performance metrics are to be minimized and others maximized, objectives to be maximized are multiplied by -1 . For each objective $f_i(x, p)$, we define the corresponding worst-case objective as

$$F_i(x) = \max_{p \in \mathcal{P}} f_i(x, p). \quad (8)$$

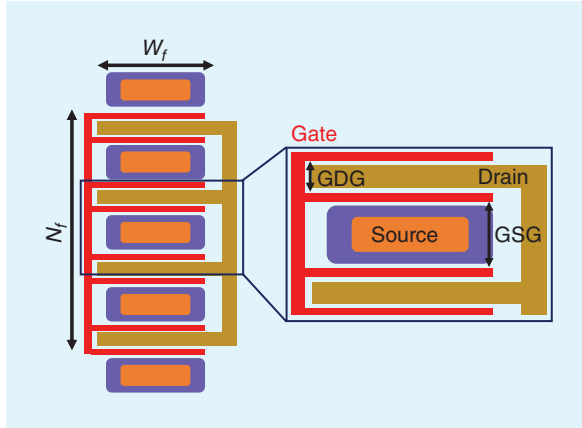


Figure 8. Qualitative illustration of the HEMT layout and its corresponding design variables, including the finger width W_f , number of fingers N_f , and the GDG and GSG layout spacings (after [6]).

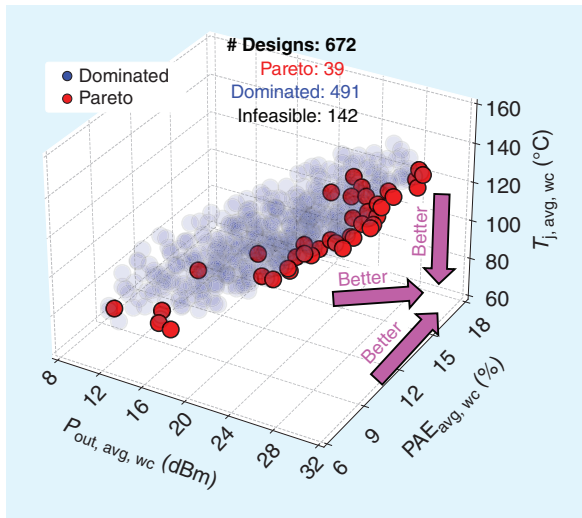


Figure 9. Robust Pareto tradeoff surface illustrating the relationship among $T_{j,avg,wc}$, $PAE_{avg,wc}$, and $P_{out,avg,wc}$. The Pareto-optimal designs are highlighted in red, while the dominated designs are shown in blue (after [6]).

A design is considered robust Pareto optimal if there is no other feasible design whose worst-case performance is at least as good in every objective and strictly better in at least one [27]. In this way, the goal is not just to optimize performance at a single operating point, but rather to identify transistor designs that remain competitive across realistic operating variation.

Design Variables and Simulation Flow

To demonstrate the methodology, a 150-nm GaN-on-SiC HEMT technology is used together with a foundry compact model in Keysight ADS [28]. In this problem, the design variables are

$$x = (V_{DS}, W_f, N_f, GDG, GSG) \quad (9)$$

corresponding to the drain bias, finger width, number of fingers, and the gate-drain-gate (GDG) and gate-source-gate (GSG) layout spacings. A qualitative HEMT layout helps visualize the meaning of W_f , N_f , GDG, and GSG, as shown in Figure 8. For each candidate design, performance is evaluated across multiple V_{GS} values, and the worst-case value for each objective is retained. This produces a robust assessment of device quality under bias variation rather than a nominal ranking based on a single operating condition.

Power Amplifier Case Study

We consider two case studies. In the power amplifier setting, the device is evaluated at 30 GHz under a modulated 5G NR 64-QAM stimulus. Here, the objectives are chosen to reflect realistic transmitter operation. Specifically, we seek to maximize worst-case average output power and power-added efficiency while minimizing worst-case average junction temperature. In this study, self-heating effects are included, since junction temperature is geometry-dependent and directly affects the overall tradeoff among power, efficiency, linearity, and reliability. Constraints are also incorporated, and designs that violate adjacent channel power ratio or gain requirements are treated as infeasible.

Once the worst-case objectives across the gate-bias range are obtained, a robust Pareto surface can be constructed to capture the tradeoffs among power, efficiency, and thermal behavior. This tradeoff surface shows that there is no single universally best transistor size. Rather, there is a family of robust tradeoff solutions whose value depends on the designer's priorities and system constraints [6]. This behavior is illustrated in Figure 9, where the Pareto-optimal designs form the boundary of the achievable tradeoff surface. Designs on this front cannot improve one objective without degrading at least one of the others, whereas dominated designs are inferior in all objectives relative to some other feasible solution.

Low-Noise Amplifier Case Study

In the LNA setting, the same device-geometry space is evaluated using small-signal simulations. Here, the objectives are to maximize worst-case gain and $f_{\max}$ while minimizing the worst-case minimum noise figure at the operating frequency of 30 GHz. Unlike in the power amplifier case, self-heating is not included in the LNA simulations, which reduces computational

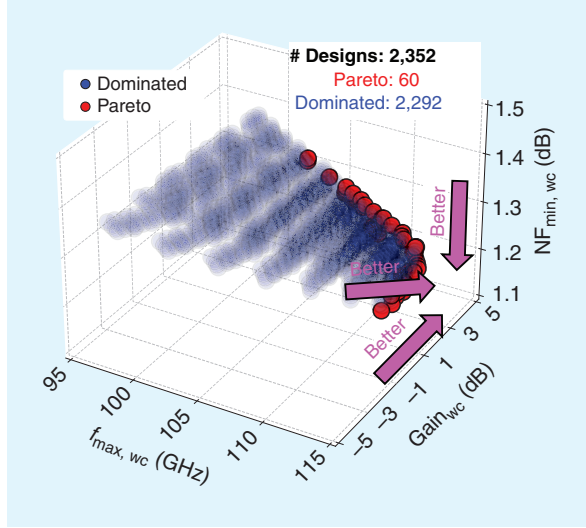


Figure 10. Robust Pareto tradeoff surface illustrating the relationship among $NF_{\min,wc}$, $Gain_{wc}$, and $f_{\max,wc}$. The Pareto-optimal designs are highlighted in red, while the dominated designs are shown in blue (after [6]).

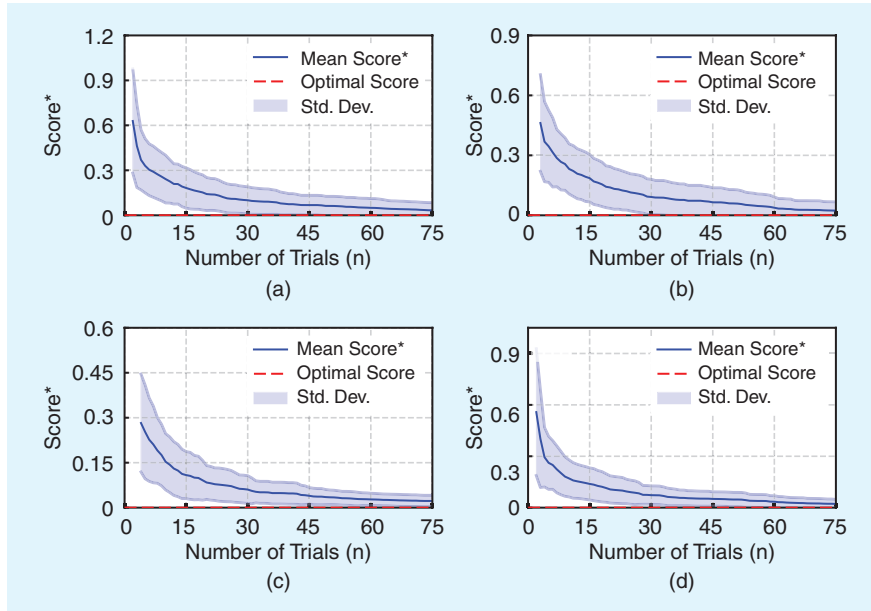


Figure 11. (a)–(d) Progression of $Score^*$ as a function of the number of trials for four experiments with different targets, limits, and priorities for the objectives $P_{out,avg,wc}$, $PAE_{avg,wc}$, and $T_{j,avg,wc}$. Each experiment consists of 100 independent optimization runs, with 75 trials per run. The plots show the mean and standard deviation of $Score^*$ across runs, together with the optimal score for the corresponding experiment (after [6]).

cost and enables a larger design sweep. Even so, only a small fraction of the explored geometries lie on the robust Pareto frontier. This again makes the tradeoff space among the competing objectives explicit and shows that, once robustness is imposed, many designs are no longer as competitive as they may appear under nominal conditions. This behavior can be seen in Figure 10, where the Pareto-optimal designs define the boundary of achievable tradeoffs among noise figure, gain, and $f_{\max}$.

Efficient Design Discovery With DFO

After characterizing the full design space, we use HOLA [13] to recover strong device designs without performing an exhaustive search. Rather than optimizing the raw multiobjective metrics directly, HOLA converts the robust design problem into a scalar preference function based on designer-defined targets, acceptable limits, and priorities [13]. This formulation is well suited to RF design, where some metrics correspond to hard requirements, while others define tradeoffs among acceptable designs. The resulting optimization behavior is illustrated in Figure 11, which shows the progression of $Score^*$ as a function of the number of trials for four different target, limit, and priority settings. In each case, the mean score improves rapidly during the early trials and then gradually approaches the corresponding optimal score, showing that strong robust designs can be identified after exploring only a small fraction of the full design space.

The main result of this case study is that HOLA identifies near-optimal robust designs after exploring only a small fraction of the full design space. In transistor sizing, the main practical bottleneck is often not whether a good design exists, but whether it can be found without an impractically large number of simulations.

This is where DFO becomes useful in practical RF workflows. It reduces the number of expensive simulations needed to reach designs that a skilled engineer would consider competitive, while still accounting for robustness, tradeoffs, and design priorities.

Summary and Outlook

The two case studies presented in this review article

showcase adjacent stages in a typical RF design flow. In the first case, DFO is used to extract the parameters of a compact model that accurately reproduces measured device characteristics. In the second case, compact models are used to explore robust transistor sizing tradeoffs under realistic operating variation. This progression also highlights an important point: Meaningful circuit optimization depends on accurate compact modeling, since any circuit-level search is limited by the fidelity of the device model used in the simulator [5].

Across both case studies, the underlying problems share commonalities. In each case, we define a feasible design space using physics-based bounds and design rules, evaluate performance through a black-box simulator, and use a derivative-free optimizer to decide where to spend the next expensive evaluation. From this perspective, DFO is less about any single algorithm and more about a general workflow that can be applied to RF design.

A key takeaway from both case studies is that the optimizer itself is only one part of the solution. In compact model parameter extraction, the critical design choice is the loss function used to measure the error between measurement and simulation. In transistor sizing, the analogous design choice is the objective formulation used to capture robustness and system-level tradeoffs. In both settings, the quality of the result depends on whether the objective carefully reflects the engineering goals of interest, such as a better fit in certain regions, higher performance across frequency, or satisfying key design specifications.

These examples also show that DFO can change the way RF design is done today. Rather than relying on manual tuning or exhaustive sweeps, the search for high-quality RF designs can be automated while preserving physical intuition. This is particularly useful in RF problems where gradients are unavailable, unreliable, or prohibitively expensive to compute.

At the same time, DFO does not replace engineering intuition. Domain knowledge remains essential for defining parameter bounds, selecting operating regions, formulating constraints, and interpreting the resulting designs. DFO instead offers a systematic, reproducible framework for applying that intuition more efficiently.

Looking ahead, the same principles extend beyond compact model extraction and transistor sizing to many other RF workflows, including passive-component synthesis, pixelated RF/microwave layout design, EM structure optimization, and circuit-level codesign [29]. These problems are well-suited to DFO because the design variables are often binary or discrete, and performance must be evaluated through expensive EM or circuit simulations. More recently, surrogate-assisted optimization has emerged as an increasingly active

direction, in which neural networks are trained to approximate these computationally costly evaluations and then coupled with DFO methods to accelerate circuit design-space exploration [30]. In that setting, surrogate models can provide faster and cheaper evaluations to guide the search toward optimized designs, while expensive full-wave or circuit simulations are reserved for physical validation and selective refinement.

As RF systems continue to grow in complexity, workflows that combine compact models, surrogate models, and modern DFO methods will become increasingly important for navigating large design spaces with limited simulation budgets. The broader opportunity is not simply faster optimization, but a more scalable and reproducible way to explore RF design tradeoffs. To support reproducibility and encourage further exploration, code for the optimization workflows discussed in this article is publicly available on GitHub [31], [32].

References

- [1] R. P. Martinez, "Advancing mm-Wave gallium nitride technology through innovative modeling approaches," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2024.
- [2] R. P. Martinez et al., "Compact model parameter extraction via derivative-free optimization," *IEEE Access*, vol. 12, pp. 123,224–123,235, 2024, doi: 10.1109/ACCESS.2024.3453198.
- [3] S. Khandelwal, S. Ghosh, S. A. Ahsan, A. Dasgupta, and Y. S. Chauhan, "Advanced SPICE model for HEMTs," Tech. Manual, Tech. Rep. ASM-HEMT 101.4.0, 2023.
- [4] R. P. Martinez et al., "Hybrid physical ASM-HEMT model using a neural network-based methodology," in *Proc. IEEE BiCMOS Compound Semicond. Integr. Circuits Technol. Symp. (BCICTS)*, 2024, pp. 38–41, doi: 10.1109/BCICTS59662.2024.10745660.
- [5] R. P. Martinez et al., "Assessment and comparison of measurement-based large-signal FET models for GaN HEMTs," *IEEE Trans. Microw. Theory Techn.*, vol. 72, no. 5, pp. 2692–2703, May 2024, doi: 10.1109/TMTT.2023.3349172.
- [6] R. P. Martinez, S. Boyd, and S. Chowdhury, "Robust pareto design of GaN HEMTs for millimeter-wave applications," *IEEE Access*, vol. 13, pp. 34,105–34,114, 2025, doi: 10.1109/ACCESS.2025.3543787.
- [7] A. C. Goad et al., "Bring your own code: A simulation framework to harness optimization, artificial intelligence, and automation techniques for microwave design," *IEEE Microw. Mag.*, vol. 27, no. 8, pp. 60–70, Aug. 2026, doi: 10.1109/MMM.2025.3617448.
- [8] L. Duc-Hung, P. Cong-Kha, N. T. T. Trang, and B. T. Tu, "Parameter extraction and optimization using Levenberg–Marquardt algorithm," in *Proc. 4th Int. Conf. Commun. Electron. (ICCE)*, 2012, pp. 434–437, doi: 10.1109/CCE.2012.6315945.
- [9] O. Kramer, D. E. Ciaurri, and S. Koziel, "Derivative-free optimization," in *Computational Optimization, Methods and Algorithms*, S. Koziel and X. S. Yang, Eds., Berlin, Germany: Springer-Verlag, 2011, pp. 61–83.
- [10] J. Larson and M. Menickelly, *Derivative-Free Optimization Methods*. Philadelphia, PA, USA: SIAM, 2019.
- [11] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. Philadelphia, PA, USA: SIAM, 2009.
- [12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [13] G. Maher et al., "A light-weight multi-objective asynchronous hyper-parameter optimizer," 2022, *arXiv:2202.07735*.
- [14] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2623–2631.

- [15] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 6373–6384.
- [16] Z. Liu, "Loss landscape visualization 1 – Seeing sticky plateau," *Ziming Liu*, Mar. 3, 2026. Accessed: Mar. 23, 2026. [Online]. Available: <https://kindxiaoming.github.io/blog/2026/loss-visualization-1/>
- [17] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, 2012.
- [18] N. Hansen, "The CMA evolution strategy: A tutorial," 2016, *arXiv:1604.00772*.
- [19] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016, doi: 10.1109/JPROC.2015.2494218.
- [20] S. Watanabe, "Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance," 2023, *arXiv:2304.11127*.
- [21] S. Lall and S. Boyd, "EE104/CME107: Introduction to machine learning," Stanford Univ. Course Website, 2025. Accessed: Mar. 22, 2026. [Online]. Available: <https://ee104.stanford.edu/>.
- [22] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.
- [23] P. Antognetti and G. Massobrio, *Semiconductor Device Modeling with SPICE*. New York, NY, USA: McGraw-Hill, 1988.
- [24] K. Woo et al., "Control of Schottky barrier height in diamond using UV-generated ozone and its effect on barrier inhomogeneity and temperature dependent properties," *Diamond Relat. Mater.*, vol. 145, May 2024, Art. no. 111059, doi: 10.1016/j.diamond.2024.111059.
- [25] K. Woo et al., "From wide to ultrawide-bandgap semiconductors for high power and high frequency electronic devices," *J. Phys., Mater.*, vol. 7, no. 2, 2024, Art. no. 022003, doi: 10.1088/2515-7639/ad218b.
- [26] R. P. Martinez, D. J. Munzer, B. Shankar, B. Murmann, and S. Chowdhury, "Linearity performance of derivative superposition in GaN HEMTs: A device-to-circuit perspective," *IEEE Trans. Electron Devices*, vol. 70, no. 5, pp. 2247–2254, May 2023, doi: 10.1109/TED.2023.3259383.
- [27] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization*, vol. 28. Princeton, NJ, USA: Princeton Univ. Press, 2009.
- [28] "Foundry & design services," MACOM, Lowell, MA, USA, 2023. Accessed: Jul. 22, 2026. [Online]. Available: <https://www.macom.com/foundry-services>
- [29] V. Chenna and H. Hashemi, "Algorithmic design and optimization of on-chip multilayered pixelated passive networks," *IEEE Trans. Microw. Theory Techn.*, vol. 74, no. 1, pp. 356–366, Jan. 2026, doi: 10.1109/TMTT.2025.3630084.
- [30] H. Chae, H. Yu, S. Li, and D. Z. Pan, "PulseRF: Physics-augmented ML modeling and synthesis for high-frequency RFIC design," in *Proc. 43rd IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2024, pp. 1–9, doi: 10.1145/3676536.3676768.
- [31] R. P. Martinez. "dfo_parameter_extraction." GitHub. Accessed: Mar. 23, 2026. [Online]. Available: https://github.com/rafapm/dfo_parameter_extraction
- [32] R. P. Martinez. "robust_pareto_design." GitHub. Accessed: Mar. 23, 2026. [Online]. Available: https://github.com/rafapm/robust_pareto_design

