

Extrapolation of Load-Pull Data: A Novel Use of GAN Artificial Intelligence Image Completion

Austin Egbert¹, *Member, IEEE*, Charles Baylis², *Senior Member, IEEE*,
and Robert J. Marks, II³, *Life Fellow, IEEE*

Abstract—Amplifier design, both in traditional approaches and real-time circuit optimization, greatly benefits from fast and thorough extraction of information from measurement data. Using only a few performance samples at varying impedances, deep learning image completion techniques can be utilized to extrapolate an entire set of Smith chart load-pull contours. In addition to speeding nonlinear device characterizations, this extrapolation can be performed in an iterative fashion for use as a circuit optimization algorithm with a very low number of measurements. The techniques of this work have been tested in the measurement of a nonlinear, large-signal amplifier. The load impedance can be estimated with a typical error of <0.1 linear units using as few as seven impedances and yields even better accuracy with larger sample sizes.

Index Terms—Artificial neural networks, circuit optimization, machine learning algorithms, measurement techniques, power amplifiers.

I. INTRODUCTION

CRUCIAL transistor performance information for amplifier design is often gathered through an extensive process that includes nonlinear measurement and characterization techniques, such as load-pull measurements. Performing load-pull measurements to characterize the maximum power, efficiency, or linearity performance of a device over changing load impedance often requires the measurement of many pre-selected impedance points. To reduce the number of measurements required in the design process, designers often rely on software nonlinear device models created from measurements over a subset of possible device configurations. Meanwhile, in reconfigurable power amplifiers, the optimum load impedance must be found through real-time measurements, which must be minimized to allow fast adaptivity. Both traditional and real-time design scenarios can benefit from techniques that are able to extract more information from fewer data points, either by reducing the size of load-pull measurement datasets, accelerating model development, or improving the rate of real-time optimization.

The ability of neural networks to learn and extrapolate performance of an individual large-signal microwave device using

a measured dataset has been previously demonstrated [1], [2], [3], [4], [5], [6], [7]. Such approaches often seek to train an individual network to model a single device for simulation and design purposes, accelerating system development. However, given the typical computational cost associated with training neural networks, these methods may struggle in application to real-time optimization problems.

Instead of using neural networks, many other approaches for impedance optimization have been explored in the literature, including locating constant power contours and proceeding toward increased performance [8], fitting measured data to an equation and solving for the optimal value [9], genetic algorithms controlling a hybrid varactor/MEMS switch tuner [10], and alternatively searching for optimal reflection phase and gradually increasing reflection magnitude [11]. However, Barkate et al. [12] demonstrated that gradient-based methods perform well for real-time circuit optimization when compared with other typical algorithms.

Rather than train neural networks to model a device (i.e., predict the output of a device given some input), it is possible to instead train a network to directly predict measured performance characteristics, such as output power or gain, from sample measurements at a small number of configurations. As an example, consider the proposed problem shown in Fig. 1. Here, a subset of a device's load impedance gain contours has already been evaluated, and it is desired to obtain the complete set of load-pull contours without additional measurement.

Saini et al. [13] previously demonstrated a form of load-pull extrapolation by generating polyharmonic distortion behavioral models [14] for the device in question, using as few as 15 measurements. It is possible to reduce the required information even further by instead applying image completion techniques from deep learning to extrapolate the unknown portions of the load-pull contours, as shown in Fig. 1. In general, image completion techniques have been successfully applied to complicated datasets, including full-color photographs [15] and partial electron microscopy [16].

In a preceding conference paper, we demonstrated the application of image completion for the simple case of simulated linear amplifier *S*-parameter data [17], with good predicative capability shown using as few as nine measurements near the center of the Smith chart. This present article demonstrates, unlike the conference paper [17], success at extrapolating nonlinear load-pull data using the image completion

Manuscript received 28 July 2022; accepted 8 September 2022. Date of publication 17 October 2022; date of current version 4 November 2022. (Corresponding author: Charles Baylis.)

The authors are with the Department of Electrical and Computer Engineering, Baylor University, Waco, TX 76798 USA (e-mail: charles_baylis@baylor.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TMTT.2022.3209700>.

Digital Object Identifier 10.1109/TMTT.2022.3209700

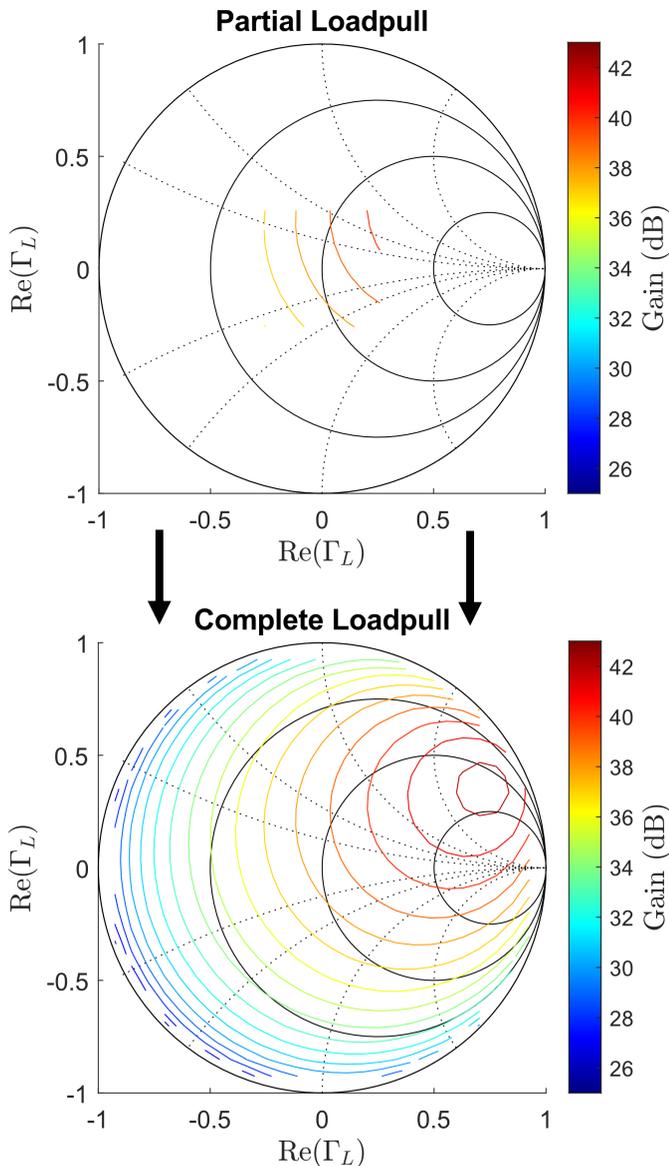


Fig. 1. Illustration of load-pull extrapolation objective: given the partial set of load-pull measurements (top), it is desired to produce the complete set of contours (bottom) without additional measurement. Reprinted from [17].

method trained on similar generated linear S -parameter data. The previous conference paper [17] used only generated S -parameter data to simulate contours and attempt to predict the linear, simulated results with image completion. This present article demonstrates the accurate prediction of measured, nonlinear device contours. The successful application of this method to nonlinear devices demonstrated using multiple nonlinear load-pull measurements is, as such, a much broader and applicable accomplishment than the scope of [17]. In addition, this work provides a process in which a small number of points are algorithmically and iteratively selected during measurement to provide quality extrapolation with a minimum number of measurements, improving on the use of fixed, square sets of measurements from [17] and establishing how these techniques could be employed for live circuit optimization in an adaptive system. Finally, this present article demonstrates the successful use of the

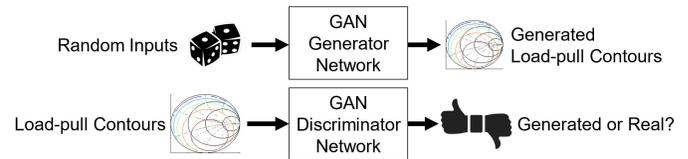


Fig. 2. Illustration of the function of the generator and discriminator networks utilized within the GAN architecture used in this present article.

extrapolation approach in measurement results for a nonlinear amplifier, while work [17] is limited to simulation demonstration of extrapolation for generated linear amplifier S -parameter models.

Section II provides an overview of the load-pull extrapolation method providing the basis for this work and introduces a novel family of circuit optimization algorithms built upon the iterative application of load-pull extrapolation. Section III then shows the performance of this optimization algorithm for live measurement data and compares the results to an existing, gradient-based method. Section IV provides a more detailed look at the quality of the individual extrapolations completed in the experiments of Section III. Finally, Section V presents conclusions from this work.

II. THEORY

The load-pull extrapolation and circuit optimization techniques here are built upon the deep image completion method demonstrated for simulated, linear devices in [17]. For purposes of clarity, here is a brief overview of the approach.

To extrapolate full load-pull contours from a partial dataset, performance measurements from the known impedances are treated as an image for the purposes of image completion. The term “image completion” is derivative of the use of this approach to images. The “images” here correspond to performance data plotted on the Smith chart, while the “image completion” corresponds to filling in unknown portions of these data.

The image completion process applies a gradient search to a generative adversarial network (GAN) trained on known load-pull contours. GANs utilize two networks pitted against each other with opposing goals [18]. In this present article, the discriminator network is trained to determine whether an image is a member of the training set of load-pull contours or a false, generated image. Meanwhile, the generator network is trained to produce load-pull images that meet the criteria learned by the discriminator, resulting in misclassification. By iteratively training these networks (and taking care to avoid overfitting), the discriminator learns how to recognize valid load-pull contours and the generator learns how to produce valid load-pull contours. The operation of these networks is shown in Fig. 2, and the relationship of these networks to one another is described in Fig. 3.

Given a well-trained GAN, load-pull extrapolation can be achieved by performing a gradient search on the inputs to the generator network, searching for an input \hat{z} that produces a generated image $G(\hat{z})$ that closely agrees with the performance at the measured impedances. This is shown in Fig. 3.

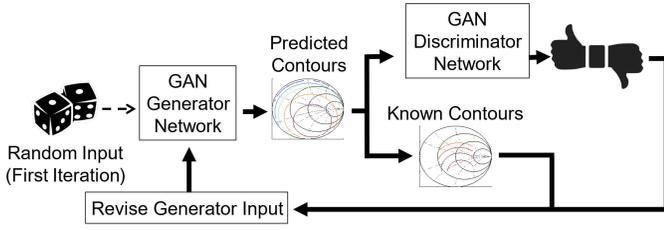


Fig. 3. Illustration of the image completion search process used in this present article. The input to the generator network is revised over time to produce load-pull contours that have good agreement with the known performance contours and present the appearance of a real load-pull dataset.

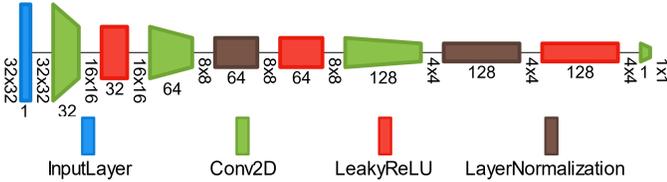


Fig. 4. WGAN-GP network topology for load-pull critic network. This network operates on a 32×32 pixel grayscale load-pull image and produces a single value estimating how far the input image is from the learned set of valid load-pull images. Illustration synthesized using [21].

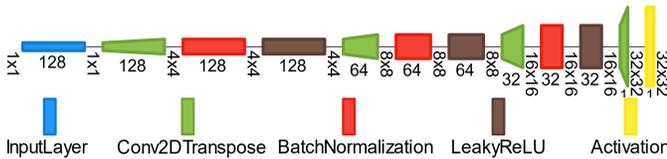


Fig. 5. WGAN-GP network topology for load-pull generator network. This network operates on a 128-element initialization vector and produces a synthesized 32×32 pixel grayscale load-pull image. A hyperbolic tangent activation layer provides the final scaling of the network output. Illustration synthesized using [21].

The method avoids the need to train the neural networks for the specific device under test, unlike the method used by Avolio et al. [1], which uses measurement data to train a network for modeling a specific device.

We use a Wasserstein GAN with gradient penalty (WGAN-GP). The Wasserstein GAN architecture reduces loss metric saturation, provides a more stable and robust training process, and is less susceptible to vanishing gradients and mode collapse [15], [19]. Because the WGAN-GP architecture uses the Wasserstein distance for quantifying how far the generated image is from the expected set (as opposed to categorizing the generated image as belonging to the set), the discriminator network is typically referred to as the “critic” network instead. The network implementation and layer topology for this present article are the same as in [17], originally adapted from [20]. Figs. 4 and 5 show the illustrations of the resulting network structure for the WGAN-GP generator and critic networks used in this present article.

These networks have been trained using 100 000 sets of simulated output power contours for linear amplifier devices [17], represented as 32×32 pixel images. These data have been obtained by randomly generating amplifier S -parameters and solving for the output power with various load impedances

and a fixed $50\text{-}\Omega$ source impedance using the transducer gain relationship from [22]. Each output power value is also subject to simulated measurement noise of up to $+720\ \mu\text{W}$, which corresponds to about 0.1 dB of error at 15 dBm. The resulting images are standardized and preprocessed with a hyperbolic tangent function as recommended by Li et al. [23], matching the output form of the generator network.

For image completion, this present article uses a slightly modified version of the algorithm in [17], which is originally derived from [24]. This approach provides known data, specified as a masked image \hat{x} , and produces a completed image containing valid load-pull contours that accurately reflect the known data. The mask of points providing measured inputs to the generator network is described [17] by

$$M(n) = \begin{cases} 1, & \hat{x}(n) \text{ is valid} \\ 0, & \hat{x}(n) \text{ is invalid} \end{cases} \quad (1)$$

where n is an index giving the image pixel number.

In comparing $G(\hat{z})$ the image that is generated by the generator network, to the measurement points, loss metrics are used. Contextual loss is the similarity between the generated image and the measured points \hat{x} , using the mask M , and is given by [17]

$$L_{\text{context}}(\hat{z}) = \|M * G(\hat{z}) - M * \hat{x}\|_1. \quad (2)$$

Perceptual loss, on the other hand, compares the image created by the generator network $G(\hat{z})$ to the dataset trained by the critic network $C()$ and is defined as follows [17]:

$$L_{\text{percept}}(\hat{z}) = \log(1 - C(G(\hat{z}))). \quad (3)$$

The overall loss is a linear combination of the contextual loss and perceptual loss, with the relative weight of each determined by a hyperparameter λ [17]

$$L(\hat{z}) = L_{\text{context}}(\hat{z}) + \lambda L_{\text{percept}}(\hat{z}). \quad (4)$$

In this work, we use $\lambda = 1$. The optimization attempts to obtain \hat{z} such that $L(\hat{z})$ is minimized.

For loss minimization, we use the Adam optimization [25] with learning rate 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. While work [17] uses a fixed number of optimization iterations when searching for an appropriate generator network input, this present article instead monitors the loss metrics used during optimization, greatly reducing the computation required for each extrapolation. When the combined loss metric reaches a local minimum, the search concludes, and the corresponding generator output provides the load-pull extrapolation result.

The general process flow for a circuit optimization algorithm using only load-pull extrapolation to drive exploration of the search space is shown in Fig. 6. For each of the given steps in the algorithm, different variations are possible.

The algorithm implementation used in this present article is built on a simple “maximum addition” technique, where the best predicted impedance from the previous load-pull extrapolation is chosen for measurement, and its performance is added to the dataset used for the next extrapolation. The initial point is chosen at random from one of the four pixels closest to $50\ \Omega$. Using this measurement, an (essentially random)

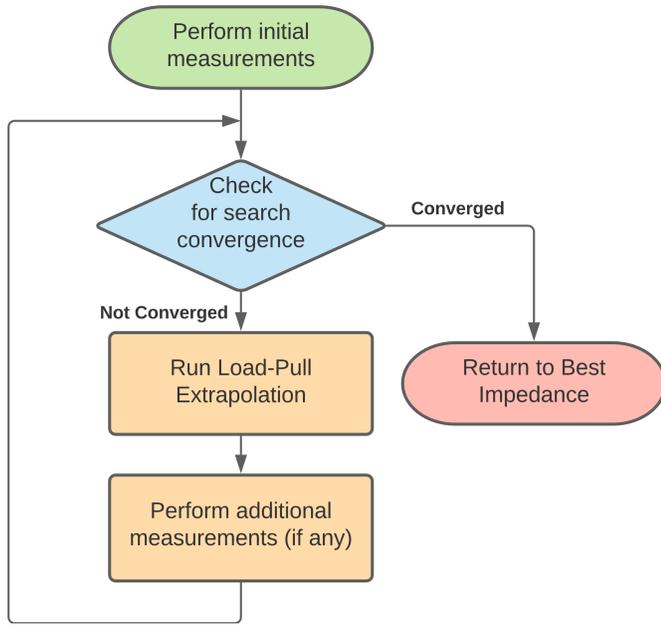


Fig. 6. Generic process for circuit optimization using load-pull extrapolation. Variations on the algorithm are possible by changing how the initial and subsequent measurement points are selected based on extrapolation results and by changing the convergence criteria.

optimum is generated by extrapolation, and this optimum is then measured and used with the measurement at the initial $50\text{-}\Omega$ point to predict the next optimum. The predicted optima continue to be added to the dataset of measured points, and the search converges when the most recently predicted best impedance has already been evaluated during a previous iteration, at which point the best measured impedance is reported as the optimum. If applied to a frequency hopping context or live system, the current impedance could be used as the initial point of the optimization process instead of first tuning near $50\text{ }\Omega$.

This maximum addition technique is well suited for systems where the time cost of performing a load-pull extrapolation is low relative to the time required to tune to and evaluate a new impedance value. However, if this situation is reversed (i.e., it is faster to tune and measure the performance extrapolations), it may be beneficial to evaluate multiple impedances per extrapolation. For example, the eight pixels surrounding a predicted optimal impedance could also be evaluated in addition to the optimal impedance. This also permits a more sophisticated convergence check that looks for the presence of some local maximum performance. Given the typical convex nature of output power contours on the Smith chart, any local maximum would be assumed to be the global maximum without requiring an additional search.

III. DEMONSTRATION OF CIRCUIT OPTIMIZATION VIA LOAD-PULL EXTRAPOLATION

A. Test Configuration

The circuit optimization via load-pull extrapolation technique of Section II has been tested and measurement demonstrated using the Skyworks 65017-70LF InGaP packaged

amplifier with a bias voltage of 7 V. For each frequency under test, the Skyworks amplifier was power swept with a $50\text{-}\Omega$ load impedance to find the 3-dB compression input power; these 3-dB compression power values are used for all the results in this article. For this device, it is necessary to offset the measured powers by 20 dB to place them in the expected range for the image completion networks. In the event, this offset is unknown for a device, and it can be learned on the fly as part of the measurement process, offsetting the measured powers to fall near the network's expected average power. This avoids improper extrapolation caused by power values that lie in the highly nonlinear range of the hyperbolic tangent activation function.

In this simple load-pull setup, the harmonic terminations of the device under test were not controlled or known. Stancliff and Poulin [27] and Colantonio et al. [28] demonstrated the use of harmonic load-pull measurements in device design and optimum termination. The harmonic terminations of the device impact performance and contour shape. In this situation, since the harmonic terminations, in general, vary as the fundamental load reflection coefficient is varied, it is expected that the contours will be different than for fixed, known harmonic terminations. This, perhaps, may make it even more difficult to accurately predict the device performance in most modeling cases. However, in this case, an image processing approach is used to generate the contours, and it is limited to extrapolating from what it can “see” based on the limited measurement dataset, irrespective of device harmonic termination impacts.

All image completions in this section are run using a computer with an NVIDIA GeForce RTX 2070 Super GPU, AMD Ryzen 3700X CPU, and 32 GB of DDR4-3200 RAM, with an average image completion time of 1.1 s. Measurements are acquired using custom MATLAB software with a traditional load-pull system (Maury Microwave passive automated load-pull tuner and Keysight Technologies signal generator and power meter).

B. Measurement Results

The circuit optimization technique was tested at 21 frequencies between 2 and 4 GHz, spaced at 100-MHz intervals. The 50 searches were performed at each frequency (a total of 1050 searches). The results from several example searches at various frequencies are shown in Figs. 7–9. Each figure shows the similarity between the real and final predicted contours, as well as the true optimal impedance and the optimized impedance. “Original Data” represents the real contours, obtained for comparison through traditional load-pull measurement. “Final Extrapolated Data” represents the extrapolated load-pull contours based on the complete collection of “Measured Points.” The “Final Predicted Optimum Impedance” is the optimum point extrapolated from all measured data. The “Search Optimum Impedance” is the impedance with the best performance over the course of the search. These points are often different, as the final predicted optimum impedance point is extrapolated from all data, whereas the best measured point (labeled the “Search Optimum Impedance”) is the best performing point of the measured data, rather than

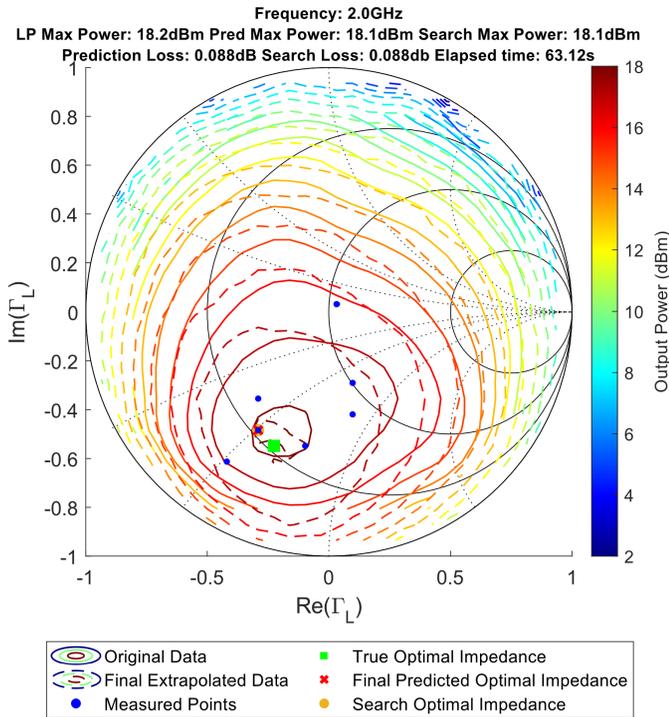


Fig. 7. Results for a single run of the load-pull extrapolation circuit optimization algorithm at 2 GHz. This search evaluated seven impedances in 63 s, converging to an impedance 1 pixel up and to the left of the true optimal impedance, with an associated performance loss of 0.09 dB.

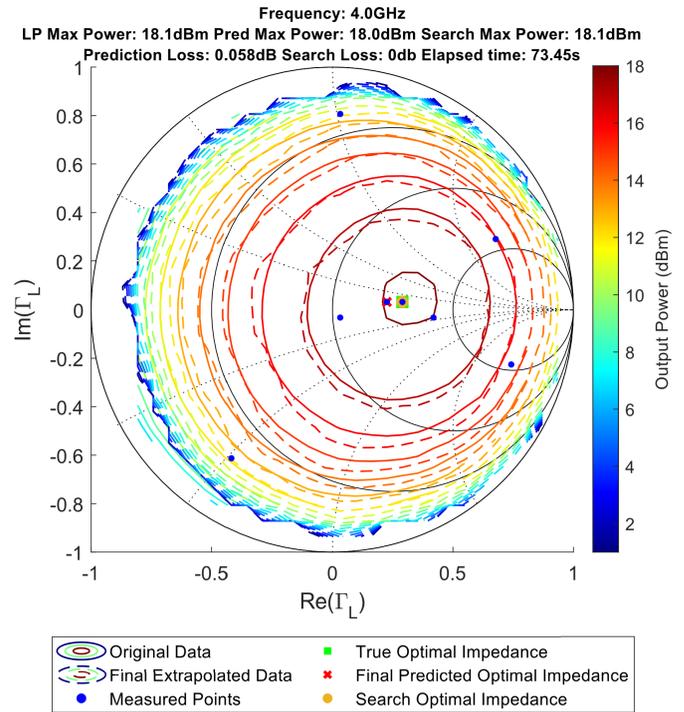


Fig. 9. Results for a single run of the load-pull extrapolation circuit optimization algorithm at 4 GHz. This search evaluated eight impedances in 73 s, converging to the true optimal impedance.

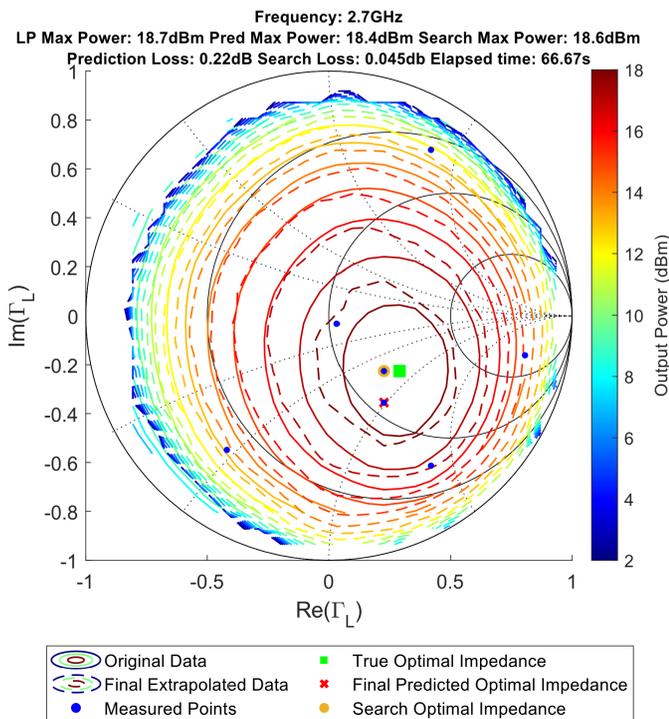


Fig. 8. Results for a single run of the load-pull extrapolation circuit optimization algorithm at 2.7 GHz. This search evaluated seven impedances in 67 s, converging to an impedance 1 pixel to the left of the true optimal impedance, with an associated performance loss of 0.05 dB.

the extrapolated optimum. In all three of these examples, the selected maximum power varies from the optimum power by less than 0.1 dB, with eight or fewer impedances evaluated.

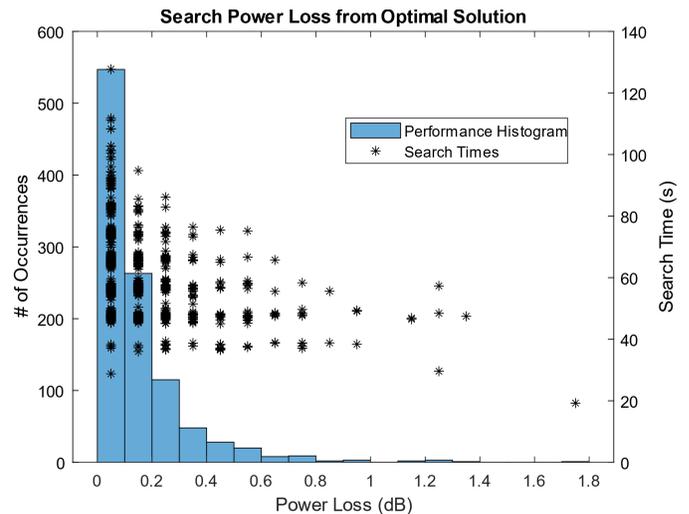


Fig. 10. Overview of search results for circuit optimization with load-pull extrapolation. Results are shown for 1050 searches run across 21 frequencies from 2 to 4 GHz. Power loss is the difference in performance for the optimal load-pull impedance and the final search impedance.

This is a significant improvement on traditional load-pull measurements and on impedance optimization algorithms, given both the accuracy and the small number of required measurements.

A complete overview of the achieved performance (compared to load-pull maximum) and elapsed time for all 1050 trials is shown in Fig. 10. To accurately compare the search performance against the maximum power reported by the load-pull for each frequency, the load-pull performance

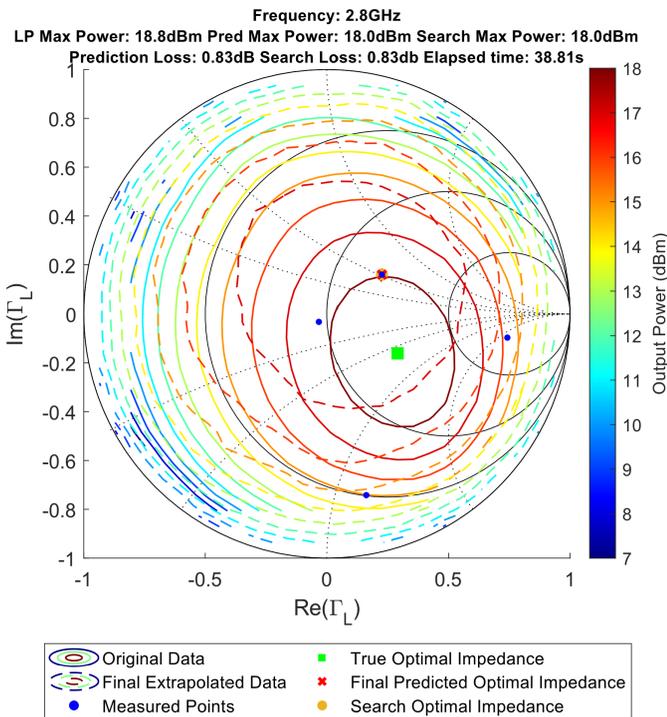


Fig. 11. Results for a suboptimal run of the load-pull extrapolation circuit optimization algorithm at 2.8 GHz. This search converged early after evaluating four impedances in 39 s, missing the true optimum by a significant margin.

(obtained from the traditional load-pull measurements) for the final search impedance is used in place of the power reported at the end of the search. This eliminates the impact of small variations in measured power that occur over the significant time needed to run all 1050 trials.

Fig. 10 shows that the search algorithm is generally successful, converging to within 0.5 dB of the optimal performance 95% of the time (1001 out of 1050 trials). Of these trials, the search converges within 60.5 s on average, with 95% of these searches converging within 84 s. In addition, more than half the searches obtain performance within 0.1 dB of the true optimum (547 out of 1050 trials). Instances that miss the optimal impedance by a significant margin generally obtained an early repeated maximum predicted impedance, leading to early search convergence, such as the result shown in Fig. 11. This can be mitigated by requiring a minimum number of measured points prior to convergence and selecting additional impedances to evaluate by some other means when an early repeat does occur. However, imposing a minimum number of measurements also imposes a floor on how quickly the search is allowed to converge.

Compared to a load-pull gradient search such as the one implemented in [26], this algorithm is able to converge much faster with significantly fewer measurements. The load-pull gradient search mentioned in [26] proceeds by beginning at a candidate point, measuring two neighboring points around the candidate, and then proceeding a search distance in the direction of steepest ascent. The gradient search mentioned in [26] stops when the search distance, decreased when the

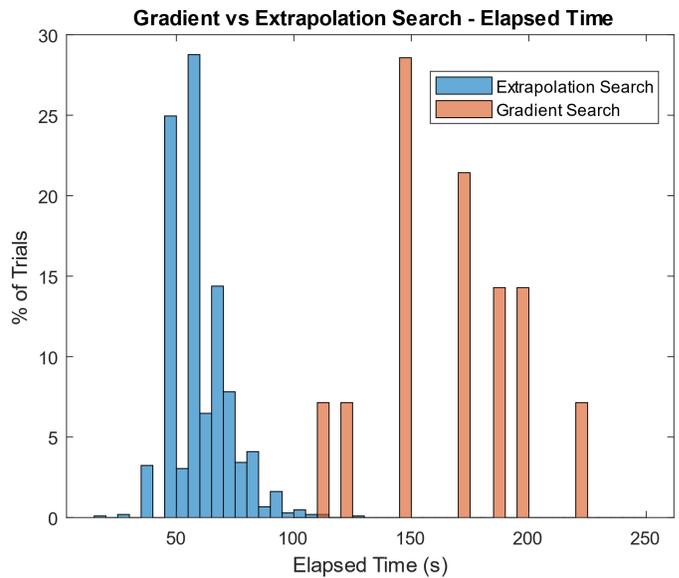


Fig. 12. Histograms of the elapsed times for the 14 gradient search trials of [26] and the 1050 extrapolation search trials of this work. For each search type, the percentage of the total searches performed for each algorithm finishing in a given time is plotted in the histogram.

next candidate performs more poorly than its predecessor, is decremented below a defined resolution distance. Histograms of the elapsed times for the 14 gradient search trials mentioned in [26] and the 1050 extrapolation search trials of this work are shown in Fig. 12. For each search type, the percentage of the total searches performed for each algorithm finishing in a given time is plotted in the histogram. While work [26] does not report elapsed times for each search, we have estimated the timing by applying the average tuning and measurement time for the traditional load-pull system to the number of performance queries reported in [26] (assuming that the computation time required for the gradient search is negligible). By this standard, the searches performed in [26] converge in an average time of 166.7 s—nearly double the 95th percentile time for the extrapolation searches that achieve within 0.5 dB of the optimal performance. This faster convergence comes at a cost of performance consistency: the total variation in final impedance for the gradient search presented is reported to be on the order of half a single pixel in the load-pull extrapolation search (each pixel is roughly 0.06 units wide on the Smith chart). Note that it may be possible to improve the accuracy of the load-pull extrapolation algorithm by training on a collection of nonlinear data, as opposed to the simulated, linear data used in this article, or by increasing the pixel density of the load-pull images.

IV. ANALYSIS OF NONLINEAR LOAD-PULL EXTRAPOLATION QUALITY

Following up on the results mentioned in [17] for linear devices, this section considers the quality of the load-pull extrapolations performed as part of the searches in Section III. Over the course of those searches, extrapolations were performed using anywhere from 1 to 14 measurements. However, only one extrapolation each used 13 and 14

TABLE I
NONLINEAR LOAD-PULL EXTRAPOLATION PERFORMANCE:
PREDICTED PEAK STATISTICS

Number of Impedances	Sample Size	Mean Power Error (dB)	Median Power Error (dB)	Mean Γ_L Error Magnitude	Median Γ_L Error Magnitude
1	1050	2.35	2.03	0.612	0.595
2	1050	3.65	3.51	0.655	0.609
3	1049	3.38	2.99	0.564	0.551
4	1047	1.35	0.477	0.275	0.204
5	1013	0.813	0.330	0.203	0.144
6	747	0.378	0.238	0.146	0.129
7	416	0.517	0.193	0.135	0.091
8	197	0.488	0.199	0.129	0.091
9	81	0.331	0.172	0.132	0.091
10	34	0.900	0.156	0.128	0.091
11	12	0.181	0.171	0.107	0.078

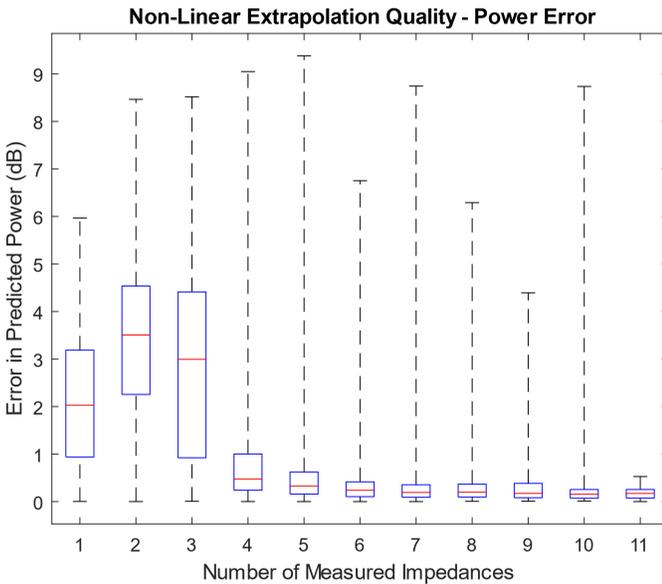


Fig. 13. Power error box plot (median, quartiles, and extremes) for the extrapolations performed in Section III. Here, an error is the difference between the measured load-pull maximum performance and the maximum power reported by the extrapolated load-pull.

measurements, and only five extrapolations used 12 measurements. Figs. 13 and 14 show statistics for the error in the predicted power and optimal impedance for all extrapolations with at least ten extrapolations per input dataset size. Table I provides a numeric summary of these results for comparison with the simulated, linear results mentioned in [17].

Comparing the results mentioned in Table I and [17], the extrapolations based on at least seven measurements perform comparably to the linear results with 81 measurements grouped around the center of the Smith chart, based on the median Γ_L error magnitude. (For comparison purposes, we prefer the median statistic as it more closely represents the typical extrapolation result without as much influence from the rare strong outliers visible in Figs. 13 and 14.) This finding suggests that the load-pull extrapolation benefits greatly from

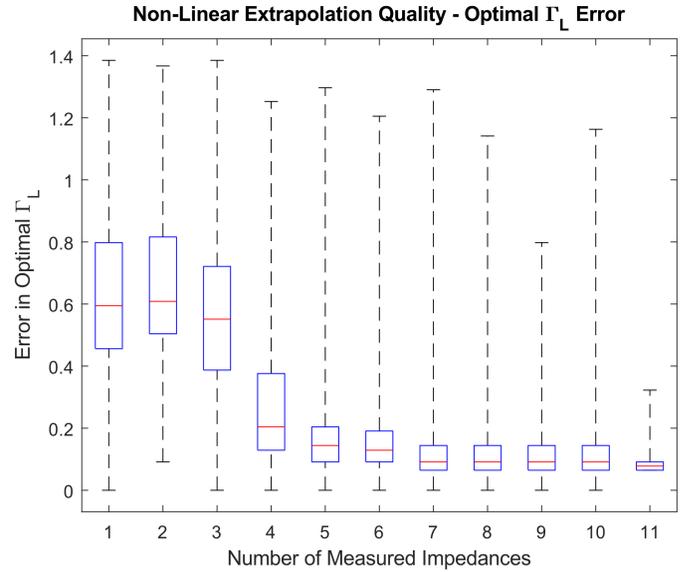


Fig. 14. Load impedance error box plot (median, quartiles, and extremes) for the extrapolations performed in Section III. Here, an error is the distance between the measured load-pull optimal reflection coefficient and the optimal reflection coefficient reported by the extrapolated load-pull.

having a collection of points scattered throughout the Smith chart, as opposed to a group of points near a single area. This finding may be useful for other microwave device modeling applications, such as the results mentioned in [1], which used measurements from a uniform grid in a single area of the Smith chart. In addition, the extrapolation method continues to perform quite well on nonlinear measurements, despite having only been trained on simulated, linear devices.

V. CONCLUSION

The ability to extrapolate nonlinear characteristics of amplifier output power versus reflection coefficient (load-pull) from partial datasets utilizing deep learning image completion techniques has been demonstrated. This approach has produced good predictive ability (median optimal Γ_L error vector magnitude below 0.1) with as few as seven measured impedances, even when the underlying neural networks are only trained on randomly generated linear S -parameters. Load-pull extrapolation accelerates amplifier optimization processes, such as traditional device characterization load-pull measurements or real-time impedance tuning techniques, by quickly identifying areas of interest. Accelerated impedance tuning has been demonstrated using an iterative extrapolation method, providing significant time savings over a gradient search method using the same hardware measurement platform.

REFERENCES

- [1] G. Avolio, A. Raffo, M. Marchetti, G. Bosi, V. Vadala, and G. Vannini, "GaN FET load-pull data in circuit simulators: A comparative study," in *Proc. 14th Eur. Microw. Integr. Circuits Conf. (EuMIC)*, Sep. 2019, pp. 80–83, doi: [10.23919/EuMIC.2019.8909451](https://doi.org/10.23919/EuMIC.2019.8909451).
- [2] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microw. Mag.*, vol. 11, no. 3, pp. 105–118, May 2010, doi: [10.1109/MMM.2010.936079](https://doi.org/10.1109/MMM.2010.936079).

- [3] C. Zhang, S. Yan, Q.-J. Zhang, and J.-G. Ma, "Behavioral modeling of power amplifier with long term memory effects using recurrent neural networks," in *Proc. IEEE Int. Wireless Symp. (IWS)*, Apr. 2013, pp. 1–4, doi: [10.1109/IEEE-IWS.2013.6616831](https://doi.org/10.1109/IEEE-IWS.2013.6616831).
- [4] W. Na and Q.-J. Zhang, "Automated knowledge-based neural network modeling for microwave applications," *IEEE Microw. Wireless Compon. Lett.*, vol. 24, no. 7, pp. 499–501, Jul. 2014, doi: [10.1109/LMWC.2014.2316251](https://doi.org/10.1109/LMWC.2014.2316251).
- [5] W. Na and Q. Zhang, "Automated parametric modeling of microwave components using combined neural network and interpolation techniques," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2013, pp. 1–3, doi: [10.1109/MWSYM.2013.6697547](https://doi.org/10.1109/MWSYM.2013.6697547).
- [6] M. Isaksson, D. Wisell, and W. Rönnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 11, pp. 3422–3428, Nov. 2005, doi: [10.1109/TMTT.2005.855742](https://doi.org/10.1109/TMTT.2005.855742).
- [7] Q.-J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design—from theory to practice," *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 4, pp. 1339–1350, Apr. 2003, doi: [10.1109/TMTT.2003.809179](https://doi.org/10.1109/TMTT.2003.809179).
- [8] S. Perlow, "New algorithms for the automated microwave tuner test system," *RCA Rev.*, vol. 46, pp. 341–355, Sep. 1985.
- [9] A. P. de Hek, "A novel fast search algorithm for an active load-pull measurement system," in *Proc. GaAs Symp.*, 1998, pp. 1–6.
- [10] D. Qiao, R. Molfino, S. M. Lardizabal, B. Pillans, P. M. Asbeck, and G. Jerinic, "An intelligently controlled RF power amplifier with a reconfigurable MEMS-varactor tuner," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 3, pp. 1089–1095, Mar. 2005, doi: [10.1109/TMTT.2005.843495](https://doi.org/10.1109/TMTT.2005.843495).
- [11] G. Berghoff, E. Bergeault, B. Huyart, and L. Jallet, "Automated characterization of HF power transistors by source-pull and multiharmonic load-pull measurements based on six-port techniques," *IEEE Trans. Microw. Theory Techn.*, vol. 46, no. 12, pp. 2068–2073, Dec. 1998, doi: [10.1109/22.739284](https://doi.org/10.1109/22.739284).
- [12] J. Barkate, A. Tsatsoulas, C. Baylis, L. Cohen, and R. J. Marks, "Comparison of multidimensional circuit optimization techniques for real-time transmitter use," in *Proc. Texas Symp. Wireless Microw. Circuits Syst. (WMCS)*, Mar. 2016, pp. 1–7, doi: [10.1109/WMCaS.2016.7577485](https://doi.org/10.1109/WMCaS.2016.7577485).
- [13] R. S. Saini, J. J. Bell, T. Williams, J. Lees, J. Benedikt, and P. J. Tasker, "Interpolation and extrapolation capabilities of non-linear behavioural models," in *Proc. 78th ARFTG Microw. Meas. Conf.*, Dec. 2011, pp. 1–4, doi: [10.1109/ARFTG78.2011.6183865](https://doi.org/10.1109/ARFTG78.2011.6183865).
- [14] J. Verspecht and D. E. Root, "Polyharmonic distortion modeling," *IEEE Microw. Mag.*, vol. 7, no. 3, pp. 44–57, Jun. 2006, doi: [10.1109/MMW.2006.1638289](https://doi.org/10.1109/MMW.2006.1638289).
- [15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5769–5779.
- [16] J. M. Ede and R. Beanland, "Partial scanning transmission electron microscopy with deep learning," *Sci. Rep.*, vol. 10, no. 1, pp. 1–10, May 2020, doi: [10.1038/s41598-020.65261.0](https://doi.org/10.1038/s41598-020.65261.0).
- [17] A. Egbert, A. Martone, C. Baylis, and R. J. Marks, "Partial load-pull extrapolation using deep image completion," in *Proc. IEEE Texas Symp. Wireless Microw. Circuits Syst. (WMCS)*, May 2020, pp. 1–5, doi: [10.1109/WMCs49442.2020.9172302](https://doi.org/10.1109/WMCs49442.2020.9172302).
- [18] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, 2014, pp. 1–47.
- [19] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 214–223.
- [20] D. Szurko. (2019). *TensorFlow 2.0 WGAN-GP*. [Online]. Available: <https://github.com/drewszurko/tensorflow-WGAN-GP>
- [21] A. Bauerle, C. van Onzenoort, and T. Ropinski, "Net2 vis—A visual grammar for automatically generating publication-tailored CNN architecture visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 6, pp. 2980–2991, Jun. 2021, doi: [10.1109/TVCG.2021.3057483](https://doi.org/10.1109/TVCG.2021.3057483).
- [22] G. Gonzalez, *Microwave Transistor Amplifiers*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1997.
- [23] H. Li et al., "Data preprocessing," in *Fuzzy Neural Intelligent Systems: Mathematical Foundation and the Applications in Engineering*. Boca Raton, FL, USA: CRC Press, 2000, ch. 15.
- [24] B. Amos. (2016). *Image Completion With Deep Learning in TensorFlow*. [Online]. Available: <http://bamos.github.io/2016/08/09/deep-completion/>
- [25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.* San Diego, CA, USA, 2015, pp. 1–15.
- [26] C. Baylis et al., "Efficient optimization using experimental queries: A peak-search algorithm for efficient load-pull measurements," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 15, no. 1, pp. 13–20, Jan. 2011.
- [27] R. B. Stancliff and D. B. Poulin, "Harmonic load-pull," in *IEEE MTT-S Int. Microw. Symp. Dig.*, May 1974, pp. 185–187.
- [28] P. Colantonio, F. Giannini, E. Limiti, and V. Teppati, "An approach to harmonic load- and source-pull measurements for high-efficiency PA design," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 191–198, Jan. 2004.



Austin Egbert (Member, IEEE) received the B.S. degree in electrical and computer engineering from Baylor University, Waco, TX, USA, in 2017, where he is currently pursuing the M.S. and Ph.D. degrees in electrical and computer engineering.

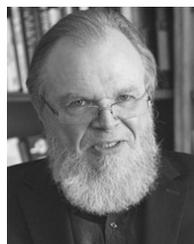
His research focuses on accelerating techniques for RF performance evaluation and real-time circuit optimization, especially for application to radar transmit amplifiers.



Charles Baylis (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of South Florida, Tampa, FL, USA, in 2002, 2004, and 2007, respectively.

He is currently a Professor of electrical and computer engineering with Baylor University, Waco, TX, USA, where he directs the Wireless and Microwave Circuits and Systems (WMCS) Program. His research focuses on spectrum issues in radar and communication systems and has been sponsored by the National Science Foundation, the Army, and the Navy. He has focused his work on the application of microwave circuit technology and measurements, combined with intelligent optimization algorithms, to creating reconfigurable transmitters.

Dr. Baylis founded the annual Texas Symposium on Wireless and Microwave Circuits and Systems, where he serves as the Chair for the Executive Committee.



Robert J. Marks, II (Life Fellow, IEEE) received the Ph.D. degree from Texas Tech University, Lubbock, TX, USA, in 1977.

He is currently a Distinguished Professor of Engineering with the Department of Engineering and Computer Science, Baylor University, Waco, TX, USA, and the Director and a Senior Fellow of the Walter Bradley Center for Natural and Artificial Intelligence, Seattle, WA, USA. He has consulted for Microsoft, DARPA, and Boeing. He is the author/coauthor of *Neural Smthing: Supervised Learning in Feedforward Artificial Neural Networks* (MIT Press), *Handbook of Fourier Analysis and Its Applications* (Oxford University Press), *Introduction to Shannon Sampling and Interpolation Theory* (Springer-Verlag), and *Introduction to Evolutionary Informatics* (World Scientific).

Dr. Marks, II, is a Fellow of the Optical Society of America. He is the Editor-in-Chief of *BIO-Complexity* and the former Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. His professional awards include the NASA Tech Brief Award and the IJCNN Pioneer in Neural Network Award. He is the Faculty Advisor for the Ratio Christi and American Scientific Affiliation Student Chapters at Baylor University.