

Partial Load-Pull Extrapolation Using Deep Image Completion

Austin Egbert¹, Anthony Martone², Charles Baylis¹, Robert J. Marks II¹

¹Baylor University, Waco, TX, USA

²Army Research Laboratory, Adelphi, MD, USA

Abstract—Many search and optimization techniques are influenced by the choice of initial starting location, including power amplifier circuit optimization. Intelligent choice of an initial starting location relies upon some understanding of the underlying search space. Given a small sample of the search space, deep learning image completion techniques can be utilized to extrapolate an understanding of the entire search space. This extrapolation can be used in lieu of a traditional search algorithm or can inform the selection of a starting location for a complete optimization. Using the techniques of this work applied to as few as nine sampled measurements, the optimum amplifier gain can be estimated with a typical error of < 0.6 dB and the corresponding load reflection coefficient can be estimated to a typical distance of < 0.2 linear units, with improved accuracy with larger measurement sample sizes.

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

I. INTRODUCTION

During amplifier design and optimization, as well any search process, it is necessary to choose some initial starting location. In cases where the optimization search locations are known in advance (dubbed “simultaneous search” by [1]), the choice of starting point has no impact on the final result. Examples of such search approaches include exhaustive searches and full amplifier load-pulls. However, significant time savings can be achieved by selecting a specific region of the search space, assuming the chosen region contains the desired optimum.

For techniques that select additional search locations based on previous measurements (dubbed “sequential search” by [1]) significant speed improvements can be gained based on the start location. Here, choosing a starting location near the final result typically allows the search to converge to the optimum value faster than from a more distant location.

In order to choose a good starting location, it is helpful to have at least a sparse or estimated understanding of the underlying search space. The common load-pull technique of performing consecutive load-pulls with increasing density utilizes the sparse understanding generated by previous iterations to select a region of focus for the next iteration [2]. For real time circuit optimization techniques, the conjugate of the device under test’s (DUT) output reflection coefficient is considered a good starting location [3]. Absent a priori knowledge of the DUT, all possible locations are equally likely to be the optimum, so beginning

at the center of the Smith Chart minimizes the average expected distance to the device optimum. For search spaces where convexity, and by extension convergence, is not guaranteed, “Sarvin’s method” of [4] has demonstrated the ability to select a good starting location by testing a few locations spread throughout the load tuner search space and choosing the best performing location, with the intention that the best performing point will be close enough to the optimum to ensure the search is able to converge.

This work presents an alternative method to generating an estimate of the underlying amplifier performance across the complete search space by employing a deep learning image completion on a generative neural network. The objective is to extrapolate a complete evaluation using only measured samples from a small region, as illustrated in Fig. 1. The optimum of this estimated search space can be used in lieu of some other optimization technique or refined by applying a more traditional search technique starting from this optimum, as opposed to the method of [4] that only provides feedback on directly measured points.

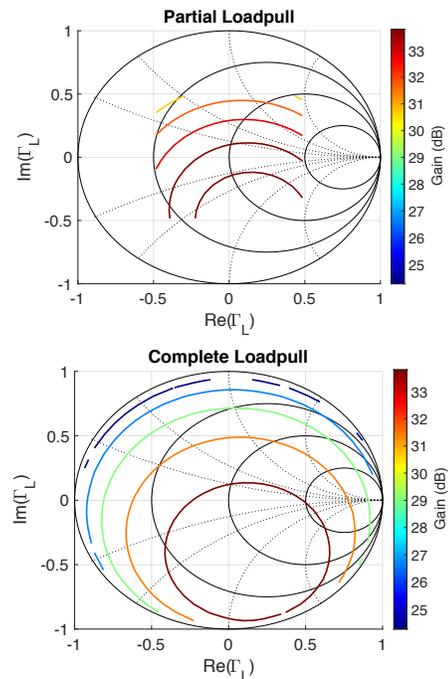


Fig. 1. Illustration of load-pull extrapolation objective. Given the partial set of load-pull measurements (top), it is desired to produce the complete set of contours (bottom) without additional measurement.

II. EXTRAPOLATION METHOD

To achieve the partial load-pull extrapolation results of this paper, a gradient-based image completion technique is applied to a generative adversarial network (GAN) trained on known load-pull contours. GANs consist of two adversarial neural networks trained with opposing goals [5]. The first network (discriminator) is trained to classify instances of data as either belonging to some defined dataset or not belonging to the dataset. The second network (generator) is trained to utilize random input values to produce instances of data that are misclassified by the discriminator. These networks are trained in an alternating fashion to allow each to adapt to weaknesses in the other's behavior. Once a GAN is successfully trained on a specific dataset, the generator can be used to synthesize elements that belong to the dataset, despite not being in the explicit subset used for training. Given a functional GAN, image completion can be performed by searching for an input to the generator that produces a result that closely agrees with the known portion of the image undergoing completion. As the GAN is pre-trained prior to image completion, this approach provides a good basis for quickly evaluating an unknown device.

A. Implemented GAN Architecture

This work utilizes a variation of the traditional GAN architecture known as Wasserstein GAN (WGAN) [6]. This architecture uses the Wasserstein distance (also called Earth Mover distance, representing the amount of work required to transfer the mass of one probability distribution to another) as the loss metric when training the generator and critic (WGAN discriminator) networks. This distance metric avoids the metric saturation that can occur when one of the networks performs too well (such as if the critic is never wrong). Avoiding this saturation greatly improves the stability and robustness of the network training process by ensuring that the gradient calculations used to update the generator network do not vanish if the critic becomes too performant. The WGAN system used in this work is trained according to the approach presented by [7] utilizing gradient penalties (WGAN-GP) as opposed to the weight clipping methods of the initial WGAN architecture [6]. The implementation (in Tensorflow 2.0), including network layer topology, is adapted from [8].

B. Image Completion Process

The trained WGAN system is then extended for image completion using the techniques of [9]. Although [9] originally utilizes a Deep Convolution GAN (DCGAN) architecture, the image completion process readily translates to other GAN architectures, such as WGAN.

The goal of image completion is to find an input \hat{z} to the generator network $G(\cdot)$ that produces an image $G(\hat{z})$ that is similar to the known partial image \hat{x} and fits the overall

target dataset. To complete an image, a mask M is first specified that encodes what portion of the full image size is provided by \hat{x} . The mask is specified as

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

where n specifies pixels within the image. This mask is used in (2) to determine which portions of the generated image should be used when comparing the quality of the generated image to the provided partial image.

Two loss metrics are used to determine the quality of $G(\hat{z})$: contextual loss and perceptual loss. Contextual loss describes how similar the generated and partial images are to each other, and is defined as:

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

Perceptual loss describes how closely the generated image resembles members of the trained dataset according to the critic network $C(\cdot)$, and is defined as:

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

These two losses are combined with a hyperparameter λ that weights the relative importance of the two metrics. (Reference [9] recommends $\lambda = 0.1$; this work instead uses $\lambda = 1$.) The final loss is then:

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

with the optimization searching for \hat{z} that minimizes $L(\hat{z})$. The results presented in this work are completed using 1000 iterations of the Adam optimization technique [10], with learning rate 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.

Note that the method of [9] combines the generated and partial images, utilizing the mask to remove part of the generated image and replace it with the original partial image. This work forgoes this step, using only the generated image as the final output.

C. Training Dataset Generation

The data used for this project is generated in MATLAB by simulating the output power contours for randomly generated sets of amplifier S-Parameters. As such, these contours represent linear device performance. This approach is used in lieu of large-signal device behavior because of the need for a large dataset during training and the relative ease of generating S-Parameters compared to simulating non-linear models. Datasets for large-signal operation could be produced by performing load-pull simulations using existing nonlinear device models across a variety of settings (frequency, bias conditions, input power, etc.).

Given a set of amplifier S-Parameters and source and load impedances, the amplifier output power can be calculated using the methods of Gonzalez [11]. The magnitude bounds for the amplifier S-Parameters are included in Table I. These bounds were selected to avoid generating potentially unstable amplifiers. Only unconditionally stable amplifiers are used for training and

evaluation to avoid unbounded performance values, determined by the stability conditions $K > 1$ and $|\Delta| < 1$ as defined in [11].

The source impedance was fixed at 50Ω , and the load reflection coefficient's real and imaginary parts were varied over $[-1, 1]$ with 28 points each and uniform spacing, discarding values with $|\Gamma_L| > 1$.

The resulting dataset was pre-processed by standardizing to zero mean and unit variance and applying sigmoidal normalization with a hyperbolic tangent function as recommended by [12], such that

$$P'(n) = \frac{1 - e^{-P_s(n)}}{1 + e^{-P_s(n)}} = \tanh\left(\frac{P_s(n)}{2}\right) \quad (5)$$

$$P_s(n) = \frac{P(n) - \bar{P}}{\sigma_P}, \quad (6)$$

where $P'(n)$ are the normalized power samples and $P_s(n)$ are the power samples standardized from the mean \bar{P} and variance σ_P^2 of the original samples $P(n)$, represented in Watts. Neglecting to apply the sigmoidal normalization results in the networks failing to predict optimized powers far from the mean observed power. This preprocessing can be inverted to recover predicted powers as

$$P_{pred}(n) = \left(2 \tanh^{-1}\left(P'_{pred}(n)\right) * \sigma_P\right) + \bar{P}. \quad (7)$$

For computational efficiency, each load-pull was rendered as 32×32 pixel greyscale images for use in the load-pull extrapolation system.

TABLE I
DATASET S-PARAMETER GENERATION BOUNDS

S-Parameter	Bound (Linear Magnitude)
S_{11}	[0, 0.8]
S_{12}	[0.0001, 0.001]
S_{21}	[2, 20]
S_{22}	[0, 0.8]

III. RESULTS

The load-pull extrapolation system was trained using 100,000 simulated amplifier load-pulls. The resulting system was then tested by presenting incomplete load-pulls dictated by the mask of (1). Three mask sizes were selected such that known values were given for regions of the Smith Chart where $|\text{Re}(\Gamma_L), \text{Im}(\Gamma_L)| < \{0.25, 0.15, 0.05\}$. These are referred to as “0.25 Mask,” “0.15 Mask,” and “0.05 Mask” and correspond to partial load-pulls of 256, 81, and 9 values, respectively. Example extrapolated load-pulls are shown in Fig. 2-4 along with the masked region used as a basis for prediction and the true load-pull contours.

Each mask was then tested with 100 different simulated amplifier load-pulls. The prediction performance for each set is summarized in Table II.

TABLE II
PREDICTION PERFORMANCE STATISTICS

	0.25 Mask	0.15 Mask	0.05 Mask
Mean Gain Error	0.162 dB	0.453 dB	0.584 dB
Median Gain Error	0.092 dB	0.241 dB	0.401 dB
Mean Γ_L Error	0.074	0.114	0.186
Median Γ_L Error	0.065	0.091	0.144

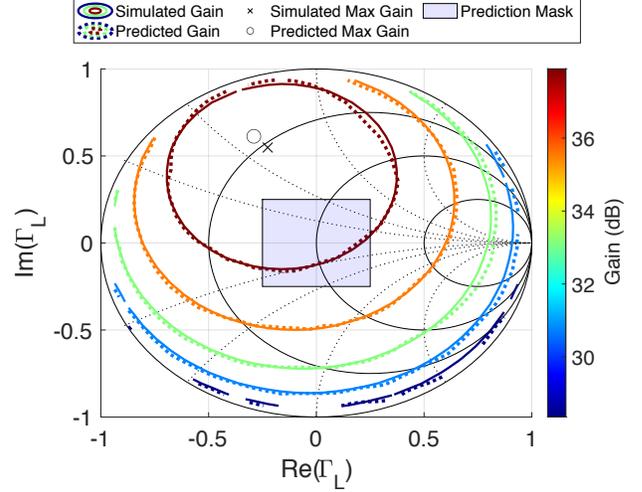


Fig. 2. Original and predicted load-pull contours based on known data in the region $|\text{Re}(\Gamma_L), \text{Im}(\Gamma_L)| < 0.25$ (shaded region). Actual maximum gain is 40.28 dBm at $\Gamma_L = 0.59 \angle 112.4^\circ$. Predicted maximum gain is 40.65 dBm at $\Gamma_L = 0.68 \angle 115.3^\circ$.

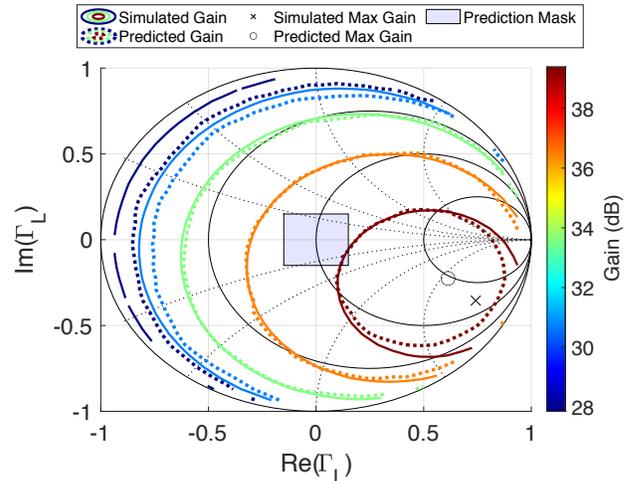


Fig. 3. Original and predicted load-pull contours based on known data in the region $|\text{Re}(\Gamma_L), \text{Im}(\Gamma_L)| < 0.15$ (shaded region). Actual maximum gain is 42.27 dBm at $\Gamma_L = 0.82 \angle -25.6^\circ$. Predicted maximum gain is 41.68 dBm at $\Gamma_L = 0.70 \angle -20.2^\circ$.

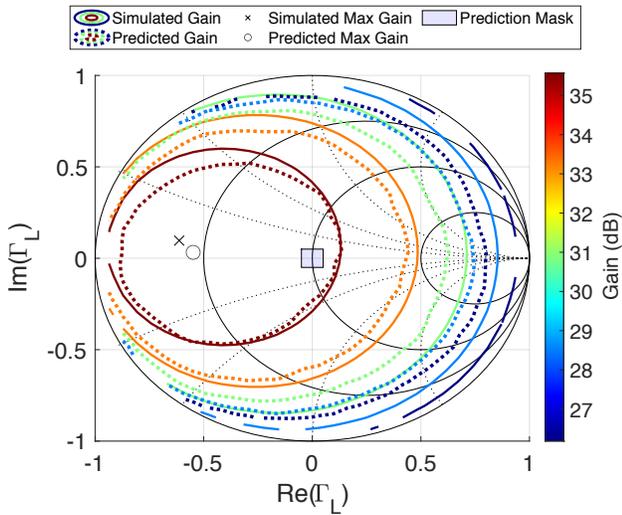


Fig. 4. Original and predicted load-pull contours based on known data in the region $|\text{Re}(\Gamma_L), \text{Im}(\Gamma_L)| < 0.05$ (shaded region). Actual maximum gain is 37.92 dBm at $\Gamma_L = 0.62 \angle 171.0^\circ$. Predicted maximum gain is 37.78 dBm at $\Gamma_L = 0.57 \angle 163.6^\circ$.

As expected, the extrapolation system performs better, on average, when given a larger initial load-pull sample (the 0.25 mask case). However, it still performs remarkably well even when given only nine measurements (0.05 Mask) as a prediction basis. Performance of optimum Γ_L predictions appears to encounter diminishing returns with increased sample size compared to predicted gain. This is likely an effect of confining Γ_L to the discrete 32×32 measurement grid used for evaluation as opposed to interpolating to a more precise maximum location.

In some cases, the prediction fails to converge to a reasonable set of contours for the given data, such as in Fig. 5. Such failures are indicated by large values of $L_{\text{percept}}(\hat{z})$, suggesting that the resulting prediction does not represent an expected set of contours, and $L_{\text{context}}(\hat{z})$, indicating the prediction doesn't match the known data. To fight convergence failure, a different initial value set \hat{z} can be chosen and another prediction can be run to obtain more realistic results. In practice, multiple predictions from various initial \hat{z} values can be aggregated to improve reliability at the cost of increased computation, or predictions could be repeated as needed according to predetermined thresholds applied to $L_{\text{percept}}(\hat{z})$ and $L_{\text{context}}(\hat{z})$. Note that these mitigations are not utilized in the results of this paper.

IV. CONCLUSION

The ability to extrapolate device performance from partial load-pull datasets utilizing deep learning image completion techniques has been demonstrated. Such an approach can be used to usefully estimate the location and value of optimum device performance with as few as nine

sample points near $\Gamma_L = 0$. These estimates can be used to inform amplifier optimization processes, such as traditional load-pulls or real-time impedance tuning techniques by eliminating large regions of the Smith Chart from consideration. Improved estimates are expected through the aggregation of multiple prediction iterations.

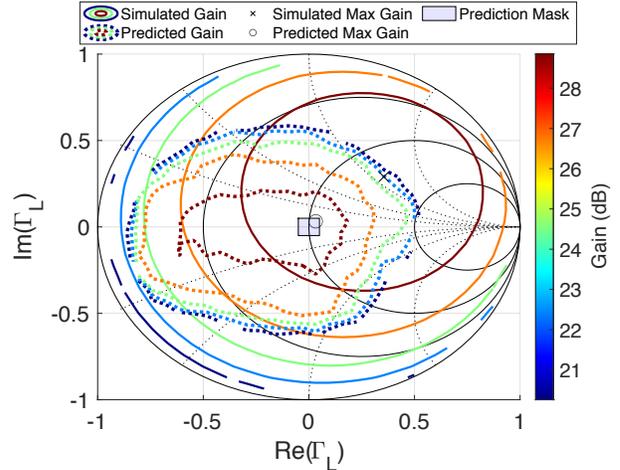


Fig. 5. Demonstration of failure to generate reasonable load-pull contours. Original and predicted load-pull contours based on known data in the region $|\text{Re}(\Gamma_L), \text{Im}(\Gamma_L)| < 0.05$ (shaded region). Actual maximum gain is 31.00 dBm at $\Gamma_L = 0.46 \angle 39.3^\circ$. Predicted maximum gain is 29.98 dBm at $\Gamma_L = 0.05 \angle 45.0^\circ$.

The authors are optimistic that this technique can be applied to large-signal operation, as similar deep learning approaches have achieved great success in much more complex applications, such as generation of full-color photographs [7] and completion of partial electron microscopy [13]. Additionally, the ability of neural networks to learn and extrapolate performance of an individual large-signal microwave device has been previously demonstrated [14], which suggests favorable outcomes in applying the techniques of this paper to this class of devices in general.

ACKNOWLEDGMENT

This work has been funded by the Army Research Laboratory (Grant No. W911NF-16-2-0054). The views and opinions expressed do not necessarily represent the opinions of the U.S. Government. The authors are grateful to John Clark of the Army Research Laboratory for assistance in development of this paper.

REFERENCES

- [1] D. Wilde, Optimum Seeking Methods, Prentice-Hall, 1964, ch. 2.
- [2] A. Howard, "How to Setup and Run Load Pull Simulations: The Basics," *YouTube*, Jan. 27, 2015 [Video file]. Available: <https://www.youtube.com/watch?v=34N-r5rPSzU>.

- [3] A.P. de Hek, "A Novel Fast Search Algorithm for an Active Load-Pull Measurement System," *GaAs Symposium*, 1998.
- [4] A. Dockendorf, *et al.*, "Fast Optimization Algorithm for Evanescent-Mode Cavity Tuner Optimization and Timing Reduction in Software-Defined Radar Implementation," *IEEE Trans. Aero. Elec. Sys.*, to be published.
- [5] I. Goodfellow, *et al.*, "Generative Adversarial Nets," *Neural Information Processing Systems 2014 (NIPS'14)*, Montreal, 2014.
- [6] M. Arjovsky, *et al.*, "Wasserstein GAN," 2017. [Online]. arXiv:1701.07875.
- [7] I. Gulrajani, *et al.*, "Improved Training of Wasserstein GANs," 2017. [Online]. arXiv:1704.00028.
- [8] D. Szurko, "TensorFlow 2.0 WGAN-GP," 2019. [Online]. <https://github.com/drewszurko/tensorflow-WGAN-GP>.
- [9] B. Amos, "Image Completion with Deep Learning in TensorFlow," 2016. [Online]. <http://bamos.github.io/2016/08/09/deep-completion/>.
- [10] D. P. Kingma, "Adam: A Method for Stochastic Optimization," 2014. [Online]. arXiv:1412.6980.
- [11] G. Gonzalez, *Microwave Transistor Amplifiers*, Second Edition, Prentice-Hall, 1997.
- [12] H. Li, *et al.*, "Data Preprocessing" in *Fuzzy Neural Intelligent Systems: Mathematical Foundation and the Applications in Engineering*, CRC Press, 2000, ch. 15.
- [13] J. M. Ede and R. Beanland, "Partial Scanning Transmission Electron Microscopy with Deep Learning," 2020. [Online]. arXiv:1905.13667.
- [14] G. Avolio, *et al.*, "GaN FET Load-Pull Data in Circuit Simulators: a Comparative Study," 2019 14th European Microwave Integrated Circuits Conference (EuMIC), Paris, France, 2019, pp. 80-83.