

Learning of Multi-Dimensional Analog Circuits Through Generative Adversarial Network (GAN)

Rahul Dutta¹, Salahuddin Raju¹, Ashish James¹, Chemmada John Leo, Yong-Joon Jeon, Balagopal Unnikrishnan, Chuan Sheng Foo, Zeng Zeng, Kevin Tshun Chuan Chai, and Vijay R. Chandrasekhar
Agency for Science, Technology and Research (A*STAR)
Fusionopolis Way, Singapore 138632
Email: salahuddin_raju@ime.a-star.edu.sg

Abstract—Analog circuits are strictly designed under operational, functional and technology constraints. Together, these bounds create a sparse multi-dimensional design optimization space with the scarcity of labeled analog training data making supervised learning methods ineffective. Accurate approximation of multi-target analog circuits, therefore, requires generation of labeled data around dominant bias and with relevant variance. With such an approach, we explore state-of-the-art semi-supervised, generative adversarial network (GAN) towards analog performance modeling. We report on various multi-target analog circuit classification experiments and demonstrate stable GAN performance achieving 2-5% higher accuracy and utilizing only 10% fully simulated manually annotated labeled data against supervised learning methods.

Keywords— *Analog and mixed-signal circuits, machine learning, semi-supervised learning, generative adversarial network.*

I. INTRODUCTION

The process of designing the next-generation analog circuit in the realm of shrinking technologies and integration needs is mostly guided by theoretical design experience where simulation and verification are carried out by electronic design automation tools (EDA) and optimizers. Further, multi-target performance evaluation is getting more difficult and time consuming, pertaining to new technologies where effects such as device parasitics, mismatch, noise floor, short channel effects, thermal noise and advanced manufacturing effects such as LDE, multicolor mask shifts, and statistical variance creates such wide set of design issues to cross optimize and verify. There have been many efforts to accelerate analog design creation in these scenarios, which are classified as circuit sizing and circuit topology synthesis. Out of which circuit sizing is the most achievable design task, and can be accelerated by space mapping surrogate modeling, genetic algorithms and recently by Neural Networks [1] with simulation-based synthesis like Bayesian optimization and by Pareto Front [2] posteriori preference techniques. Also, there are methods reported using data-driven, machine learning optimization where Neural Network weights are back fitted to physical measurements to better approximate, than modeling only, through first principals [3,4].

Most of the recent works using feedforward neural networks require a large amount of training data to develop non-parametric learning model, complex learning tasks (inference) through these models are slow on convergence and larger training data (high variance) is required to achieve acceptable performance. On the other hand, models based on specific

learning tasks (high bias) are likely to have high estimation errors as getting the right biases from input data is often complex in multi-target optimization scenarios. To our knowledge, towards analog multi-target prediction, none of the works above has attempted to utilize variance and bias in modeling and reducing the high dimensionality of input data.

Towards our first contribution, we demonstrate the use of statistical sensitivity analysis prior to machine modeling using bivariate covariance to identify the variables and ranges of importance to generate training data with less dimensionality and variance. Our second contribution is to demonstrate natural learning techniques like semi-supervised learning. Here much of the knowledge is extracted from a large amount of unlabeled data (augmented by theoretical knowledge, technology models) and very few labeled (fully simulated samples guided by covariance) input data. We explored the use of deep generative adversarial networks (GAN) classifiers rather than a committee of classifiers, known as ensemble method [4], first to learn the overall distribution density and sub densities in high dimensions (manifolds), second to improve classifier test error by means of regularization for smoothness around decision boundaries.

The overall objective of this paper is to address bias and variance right from the creation of good quality training data. Where the training data, is usually the starting point of any machine modeling and getting good and sufficient labeled data is always a challenge. Coupled with semi-supervised GAN classifier, which utilizes few labeled data samples and model low dimensional manifolds to interpolate and estimate multi-target objectives without loss of accuracy compared to supervised learning.

II. TRAINING DATA GENERATION AND ADVERSARIAL NEURAL NETWORK (GAN)

Machine models approximating analog circuits require good quality training data to best model the non-ideality of circuit behavior. The goal of machine learning algorithms is to learn the non-linear relationships between inputs and outputs as there are no linear or quadratic relationships termed as non-parametric learning [5] and to accurately predict (infer) the response with respect to an unseen input vector during execution. The inference, which is conditioned around training data is built on a statistical probability of estimating an output as regression or classification.

The learned probability distribution from training data, forms the underline mechanism of any neural network learning

¹ Equal contributing author.

model, here the learning problem is to construct a function f based on input-output pair (x, y) such that its minimized for (x_N, y_N) training samples, loss function $L_{FF}(1)$ of an idealized feed forward network with synaptic weights is represented by sum of observed square errors:

$$L_{FF} = \sum_{i=1}^N (y_i - f(x_i))^2 \quad (1)$$

As a generalization, the measure of effectiveness of f as predictor of y is the expectation E with respect to probability measured as regression (mean value of output y , $E[y|x]^2$) of all functions of x , it highlights *variance* (2) of y given x that does not depend on data or on estimator f :

$$\sigma_{y|x} = E[(y - E[y|x])^2 | x] \quad (2)$$

Further, training data may not always be balanced amongst various classes thus (3), highlights bias where f can be represented as $f(x;D)$, and D is the dependence of f on training data, here mean square error of f as an estimator E based on regression on training dataset, D is represented as:

$$E_D = [(f(x;D) - E[y|x])^2] \quad (3)$$

Modeling through machine learning could generate many scenarios where $f(x;D)$ may be an accurate approximation with an optimal predictor of y . Also however it may be a case when $f(x;D)$ has quite different dependency when using other training data sets and results far away from regression or classification estimator $E[y|x]$ thus is considered as biased on training dataset. The problem of bias and variance becomes paramount and complex, as feature dimensionality and multi-target goals increase with design complexity. For modeling multi objective circuits, bivariate probability covariance between input feature and output performance target pairs reveals a rich set of information about the circuit under model and can be utilized for guiding sampling rather than sampling through parametric sweep. Bivariate probability covariance, known as Pearson correlation coefficient (PCC) r for a sampled input-output pair (X, Y) with n data sample pairs $\{(x_1, y_1) \dots (x_n, y_n)\}$ is represented as (4),

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{(n-1)s_x s_y} \quad (4)$$

Where $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ is the mean of individual sample set, analogous for \bar{y} , a dimensionless signed standard score $s_{x|y}$ has the form (5),

$$s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (5)$$

defined for each input feature and output target pair with positive and negative notation above and below mean for a bivariate covariance. It is observed that for $(x_i - \bar{x})(y_i - \bar{y})$ is positive only if x_i and y_i lie on the same side of their respective mean value else for vice-versa is negative, denoting that larger absolute value on either side can be identified as strong positive or strong negative correlation [6].

Towards exploration of circuit's functional sensitivity, the goal is to minimize strong bivariate correlation and maximize near zero correlation areas. Though it is an optimization problem in itself, we use sample selection technique guided by

standard score against previous reported works of using probabilistic methods such as covariate shift and sample section bias [7] to make judgement of sample selection. This simplified method makes sampling task efficient and relatively easy to integrate with any off the shelf EDA simulator. The time and computational complexity of this method is far less than a probabilistic model to guide sample selection or through generation of carefully selected training samples by analog designer.

We explore the use of semi supervised GAN, to address the challenge of generating large quantity of fully simulated labeled data, typically used for non-linear supervised learning approaches for modeling analog circuits like SVM [8]. It is well known that supervised learning approaches exploit huge labeled dataset, like in computer vision, ImageNet [9] and are successful as minimization of training error cost function is performed using stochastic gradient descent (SGD) which is highly effective. Getting large quantity of unlabeled input data is computationally cheap, in the context to analog circuits, it refers to the various input perturbations of active and passive device parameters (features) in a design space bounded by technology and design constraints, as compared to getting fully simulated labeled data for training is computationally expensive and costly. Towards semi supervised / unsupervised learning, the cost function has to take advantage and learn from limited labeled dataset and propagate labels to unlabeled dataset based on similarity heuristics, transfer learning or through joint objective [10].

Deep generative semi supervised models learn this non-linear parametric mapping, by a feed forward multilayer perceptron (MLP) neural network called as a training *generator* (G) network which models a distribution p_g trained and tested on a set of L labeled dataset $\{(x_i, y_i)\}_{i=1}^l$. This baseline model, facilitates sample generation by transforming noise vector z using an U unlabeled dataset $\{x_j\}_{j=l+1}^{j=l+u}$ distribution, such that $x = G(z; \theta_g)$ with a probability distribution,

$$p_z : Z \xrightarrow{\dim(z) \ll \dim(x)} X$$

where θ_g are MLP parameters. GAN frameworks first train the generator G , which tries to minimize the divergence D_{JS} (6), measure between p_g and the real labeled data distribution p_x as an optimization function:

$$D_{JS}(P||Q) = \frac{1}{2} D_{KL}(P||\frac{1}{2}(P+Q)) + \frac{1}{2} D_{KL}(Q||\frac{1}{2}(P+Q)) \quad (6)$$

Here. D_{JS} and D_{KL} are the Jensen-Shannon, Kullback-Leibler divergence and P, Q are labeled data and generator estimated data distribution respectively. A second feed forward MLP neural network introduces the key concept in GAN training, which is the *discriminator* (D) providing a feedback training signal to G by maintain an adversary $D(G(z)) \rightarrow [0,1]$ and is trained to distinguish the real samples from $x \sim p_x$ vs the

generated samples $x_g \sim p_g$. The generator is trained to challenge discriminator in a min-max game with a value function $V(G, D)$ (7):

$$\begin{aligned} & \min_G \max_D V(G, D) \\ &= E_{x \sim p_{data}(x)} [\log D(x)] \\ &+ E_{z \sim p_{z(z)}} [\log(1 - D(g(z)))] \end{aligned} \quad (7)$$

Semi supervised learning frameworks generate labeled data according to probability distribution function P on $X \times \mathbb{R}$, where U unlabeled dataset has $x \in X$ drawn according to marginal distribution P_x , of P . The knowledge from marginal distribution P_x can be exploited for better (classification) learning tasks under certain specific assumptions, that is, it is assumed, if two input data points $x_1, x_2 \in X$ are close in the intrinsic geometry of P_x , then standard distribution $P(y|x_1)$ and $P(y|x_2)$ are similar, therefore the geometric structure of marginal distribution P_x is smooth in P [11]. As P_x is unknown (generated from unlabeled samples) which lies in a compact sub manifold $\mathcal{M} \subset \mathbb{R}$ the optimization function f (8), gets represented as:

$$\|f\|^2 = \int_{x \in \mathcal{M}} \|\nabla_{\mathcal{M}} f(x)\|^2 dP_x(x) \quad (8)$$

Where $\nabla_{\mathcal{M}}$ is the gradient of f along manifold \mathcal{M} and integral is taken over marginal distribution P_x . Recent applications of GAN on image data has shown excellent proximity of generative samples to real image samples, mostly GAN training and convergence is based on finding Nash equilibrium, in addition to determining real vs fake generated samples, GAN discriminators have been extended to determine specific class of the generated samples based on low dimensional feature representations. It has also been reported, that GAN classifiers can interpolate between points in low dimensional space [12] and suggest that a classifier generalization given a conditional distribution $p_i(x_i, y_i)$ for a feature vector x_i and its performance target y_i can be smoothed using manifold \mathcal{M} regularization [13]. Classifier smoothness is necessary along the decision boundaries because local perturbations injected by $p(z)$ may result in non convergent unstable GAN classifiers with high inference errors.

There are various reported smoothing techniques that use divergence approximations through regularizers such as f -GAN, WGAN, adversarial direction [14] and explicit data transformation [15]. Our work uses Monte-Carlo approximations of Laplacian norm [16] to perform label propagation from labeled dataset to unlabeled dataset based on feature matching, here the generator is bounded by only generating data, which matches the statistics of a real data and the discriminator is training on feature statistics that are worth matching. This is achieved, by tapping internal layers of the discriminator rather than training on overall discriminative approaches from preventing overtraining of the discriminator. The optimization function $\|f\|^2$ thus can be approximated using graph Laplacian approaches (9) and the gradient $\nabla_{\mathcal{M}}$

represents a Jacobian matrix J with respect to latent representation instead of computing tangent directions explicitly in the form for $z^{(i)}$ samples drawn from latent space of G .

$$\|f\|^2 = \frac{1}{n} \sum_{i=1}^n \|J_z f(g(z^{(i)}))\|_F^2 \quad (9)$$

We have used stochastic finite difference to approximate final Jacobian regularizer to the discriminator to feature match, thus the overall generator loss function (10) is:

$$L_G = \left\| \mathbb{E}_{x \sim p_{data}} h(x) - \mathbb{E}_{z \sim p_{z(z)}} h(g(z)) \right\| \quad (10)$$

Here, $h(x)$ are the activation on an intermediate layer of discriminator. Where the loss function of the discriminator is

$$L = L_{supervised} + L_{unsupervised} \quad (11)$$

$$\begin{aligned} L_{supervised} &= -\mathbb{E}_{x,y \sim p_{data}(x,y)} [\log p_f(y|x, y < K+1)] \\ L_{unsupervised} &= -\mathbb{E}_{x \sim p_{data}(x)} [\log [1 - p_f(y = K+1|x)] - \\ &\quad \mathbb{E}_{x \sim g} [\log [p_f(y = K+1|x)]] + \\ &\quad \gamma_{\mathcal{M}} \mathbb{E}_{z \sim U(z), \delta \sim N(\delta)} \|f(g(z)) - f(g(z + \varepsilon\bar{\delta}))\|^2 \end{aligned}$$

Here, $\|f(g(z)) - f(g(z + \varepsilon\bar{\delta}))\|^2$ is the stochastic finite difference, where $\bar{\delta} = \frac{\delta}{\|\delta\|}$, $\delta \sim \mathcal{N}(0, I)$ and $\gamma_{\mathcal{M}}$ are the manifold regularization parameters.

III. IMPLEMENTATION

We first illustrate our method to generate training data based on PCC, all the analog circuit features (active devices (A_D), passive devices (P_D) and bias current and voltages B_{CV}) are parameterised and are represented as an input training tuple:

$$\begin{aligned} A_D &= \{W, L, Number\ of\ Fingers\} \\ P_D &= \{C, R, L\} \\ B_{CV} &= \{Bias\ Current, Bias\ Voltage\} \end{aligned} \quad (12)$$

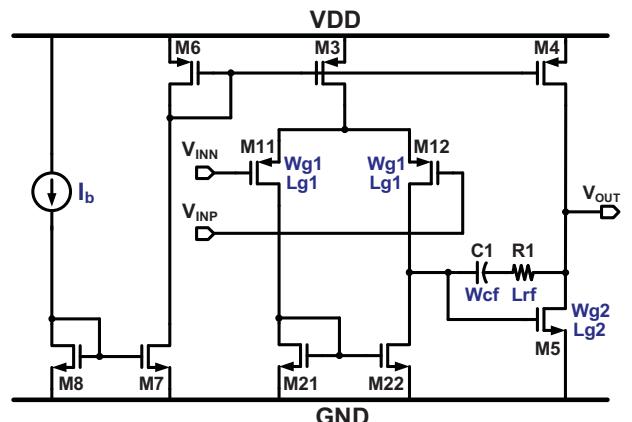


Fig. 1. Schematic of a two-stage operational amplifier.

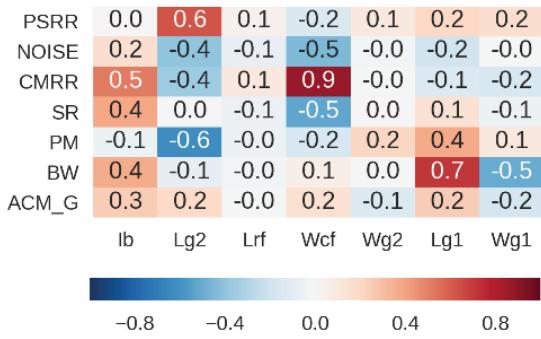


Fig. 2. Covariance matrix heat map plot of a two-stage amplifier.

The input tuple (12), containing sizing and bias parameters have complex correlation amongst themselves and towards multi-target performance outputs which is a n dimension vector. We generate first few random samples within the functional design space using parametric sweep and rely on designers circuit knowledge to fix the upper and lower bounds to these variables. This random batch simulation forms the seed of the initial bivariate covariance matrix PCC having a standard score $s_{x|y}$. Fig.1, illustrates this idea, where input features such as bias current, 1st and 2nd stage transistor widths, lengths and passive device features have been identified as having dominant correlation.

Seed PCC matrix is shown in Fig. 2, here input features (x -axes) plotted against performance targets (y -axes), here 2nd stage transistor length (L_{g2}), load capacitor width (W_{cf}) and 1st stage length (L_{g1}) shows strong positive/negative correlation for various performance targets, Gain (ACM_G), Phase Margin (PM) Common Mode Rejection Ratio (CMRR) and Bandwidth (BW) respectively. To elaborate a bit more, the bias current I_b has a positive covariance with BW and ACM_G suggesting that higher BW and ACM_G can be achieved by having higher I_b .

Next, a guided set of tuple training sample is generated and simulated on the identified features, which report high standard score of $-0.4 \leq 0.4$ and vice versa for low score towards output targets. A random vector x_r is drawn from a training sample available under $\{x_1 \dots x_n\} \in X$ such that $x_r = \{x_r \in X \mid x_r \notin X_e\}$ where X_e is excluded vector set $\{x_e_1 \dots x_e_m\} \in X_e$ for which the standard score is known. Here the stopping criterion is when, the kernel density distribution reports less than 5% variation between simulation runs. Fig. 3, depicts the overall setup for training data generation, in a progressive manner, as the most dominant knobs and ranges gets identified and the seed PCC matrix gets overridden.

For classification, we define two classes namely $\{\text{Pass}, \text{Fail}\}$, a Pass label is allocated to simulation run, if all multi target specifications are met, these classes are a weighted sum of various target specifications sampled at a design frequency, time or performance attribute. More weightage is given to certain output targets, such as in the case of the two stage operational amplifier, bandwidth is most weighted output performance target than others to be classified as Pass or Fail.

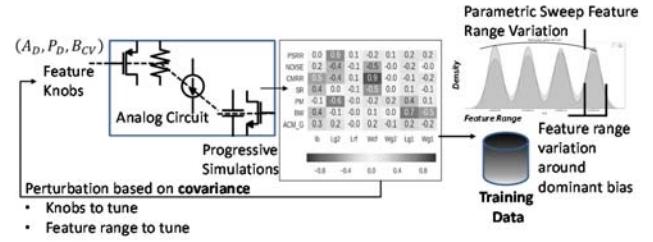


Fig. 3. Progressive sampling around dominant feature knobs and ranges to generate training data.

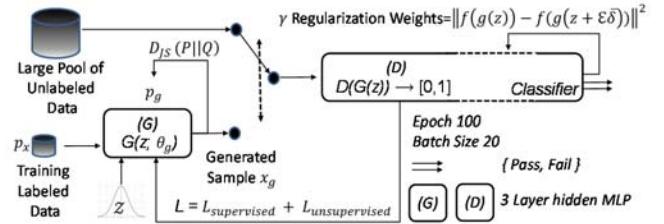


Fig. 4. Semi supervised GAN algorithm framework.

Next, guided set of tuple training sample is generated on the identified features which report very less standard score $-0.4 \leq 0.4$ towards the output targets, such that a random vector x is drawn from a training sample available under $\{x_1 \dots x_n\} \in X$ such that $x = \{x \in X \mid x \notin X_e\}$ where X_e is excluded vector set $\{x_e_1 \dots x_e_m\} \in X_e$ for which the standard score is known. The main motivation towards such sampling is to induce good bias and variance in the training data as during semi supervised modelling a small fraction of labelled dataset is used to generate a baseline parametric model which approximates the probability distribution of a real dataset. Getting an accurate predictive model with large unlabelled data thus require the initial training data to be representative of dominant circuit behaviour than generic sweep analysis and designer's guidance. For semi supervised classification, we designed two identical adversarial networks G and D . These are three hidden layer fully connected dense networks with attributes mentioned under, Table 1-2.

The overall GAN algorithm framework, Fig. 4 depicts training of G using latent space perturbation z with small number of labelled samples having p_x distribution the inference x_g batch samples are fed to the D network which also receives batch samples from large pool of unlabelled data. Laplacian regularization for accuracy and stability is enforced within D which maintains an classification adversary. A feedback loop to G , improves its inference ability to forge data for min-max game. At the end of training and validation D network is deployed to do performance benchmarking.

IV. EXPERIMENTAL RESULTS

We experimented and tested aforementioned sampling and semi supervised GAN approaches on four varying complexity analog circuits, designed on 55-nm CMOS technology. All experiments are performed on quad core, multi-threaded, 3.1GHz, RHEL 7.0 Linux. Various machine learning packages

used are as follows, (1) python implementations of Google tensorflow 1.8.0, (2) random forest (RF) and decision tree package (DT) available under anaconda 3.0, (3) XGBoost (XGB) available under scikit, (4) covariance framework from seaborn and pandas, and EDA simulation framework using Cadence Spectre 15.10 with OCEAN SKILL interface.

Table 1: GAN architecture we used for our experiments.

Generator	Discriminator
Latent space 5 (uniform noise)	64 Dense, lReLU
64 Dense, ReLU	Dropout, $\rho = 0.6$
128 Dense, ReLU	128 Dense, lReLU
256 Dense, ReLU	Dropout, $\rho = 0.5$
No. of inputs Dense, tanh	256 Dense, lReLU
	Dropout, $\rho = 0.5$

Table 2: Hyperparameters of GAN models.

Hyperparameters	Experiments
γ regularization weight	10^{-5}
ϵ latent z perturbation	10^{-5}
lReLU Slope (D)	0.2
Learning rate decay (D)	Linear decay to 0 after 90 epochs
Optimizer (G, D)	ADAM ($\alpha = 10^{-4}, \beta_1 = 0.5$)
NN Weight Initialization (G, D)	Isotropic Gaussian ($\mu = 0, \sigma = 0.05$)
NN Bias Initialization	Constant(0)

Two stage operational amplifier Fig. 1, is a high gain differential amplifier with 7 input features and 7 output performance targets, t-SNE plot Fig. 6 highlights similarity scatter plots of the {Pass, Fail} classifications which is the nature of the training data, using learning rate of 10 on 3420 fully simulated dataset. As it can be inferred from Fig. 6, that the decision boundaries are highly non-linear.

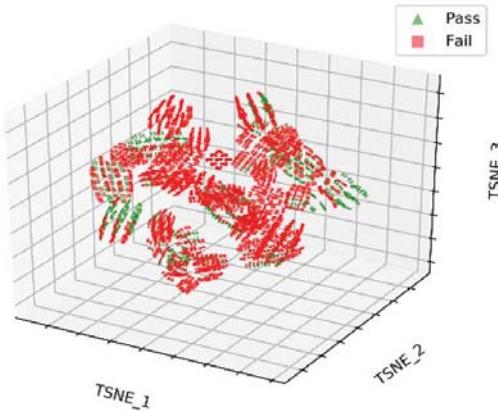


Fig. 6. T-distributed Stochastic Neighbour Embedding (t-SNE) visualization using KL divergence, mapped to low dimensional (2D) plots highlighting non-linear distributions that GAN discriminator learns to generate decision boundaries.

Performance modelling comparison between our GAN algorithm against Supervised Neural Networks (SNN), DT, RF and XGB is shown in Fig.7. It can be observed, that our method of sampling, is able to get high accuracy with an overall less labelled training data, this shows a novel use of covariance based sampling even for supervised methods together with semi-supervised approaches. Our GAN algorithm is able to outperform with an average improvement in accuracy of 3.04%, when using only 10% fully simulated labelled training dataset. Progressively GAN converges to the results of supervised learning which suggests that semi-supervised learning methods can be adopted without the loss of accuracy to model analog circuits. GAN based semi-supervised algorithm can be applied, in circuit design scenarios where the circuit designs have limited training dataset due to high EDA tool cost to simulate and high engineering cost for labelling datasets. A more detailed performance benchmark summary is presented later in this section with even lower percentage (between 1-1.5%) of labelled dataset to highlight the effectiveness of our approach. For all benchmarking, we have used 80/20 bin ratio for training vs validation, and use Epoch as 100 with a batch size of 20 for pooling unlabeled dataset.

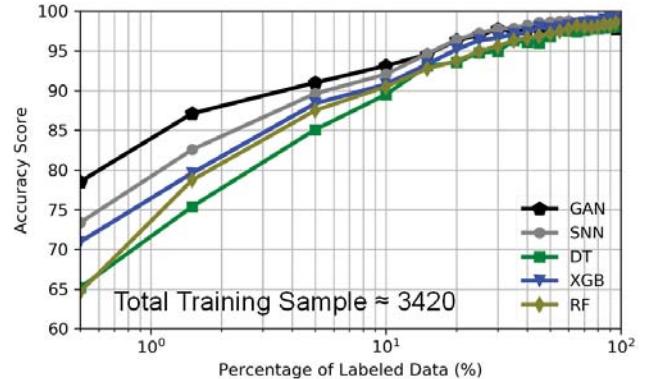


Fig. 7. Performance modelling comparison of a two stage operational amplifier, our robust method, generating high quality sampling and regularized GAN performs better than supervised methods.

Table 3: Classification Accuracy Summary.

Type of Algorithm	Accuracy at 1.5% of training data
SNN	82.55
DT	75.38
RF	78.74
XGB	79.61
GAN w/o manifold regularization	83.74
GAN with manifold regularization	87.12

GAN accuracy with/without regularization compared against all supervised benchmark algorithms is presented in Table 3. According to the case studies on the design experiments, GAN requires significantly less number of labelled training data. Our methodology thus saves compute,

EDA and human labelling cost to make it widely adopted for IC design.

V. CONCLUSION

Towards machine learning applications on circuit design and optimization, in this paper, we explored a knowledge-based methodology to generate high-quality training data using bivariate covariance between high-dimensional input features and the output performance targets as a prior and its subsequent use to demonstrate semi-supervised GAN learning as an analog performance classifier. We trained a stable GAN to model analog circuit data distribution and applied geometric Laplacian regularization to improve classifier accuracy. Demonstrated, on the analog circuit blocks our methods surpass supervised learning methods in the areas where far less labeled training data is available, which is typical in the field of circuit design and optimization. In the future, we would like to extend this methodology to digital standard cell based IPs and regression estimators. We also plan to apply in areas of mixed-signal design optimization, physical design, and digital verification and explore together with active learning frameworks.

REFERENCES

- [1] N. Takai and M. Fukuda, "Prediction of element values of OPAMP for required specifications utilizing deep learning," 2017 International Symposium on Electronics and Smart Devices (ISESD), Yogyakarta, 2017, pp. 300-303.
- [2] W. Lyu, F. Yang, C. Yan, D. Zhou and X. Zeng, "Multi-objective Bayesian Optimization for Analog/RF Circuit Synthesis," 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), San Francisco, CA, 2018, pp. 1-6.
- [3] F. Wang, W. Zhang, S. Sun, X. Li and C. Gu, "Bayesian Model Fusion: Large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2013, pp. 1-6.
- [4] Hongzhou Liu, A. Singhee, R. A. Rutenbar and L. R. Carley, "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," Proceedings 2002 Design Automation Conference (IEEE Cat. No.02CH37324), New Orleans, LA, USA, 2002, pp. 437-442.
- [5] S. Geman, E. Bienenstock and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," in Neural Computation, vol. 4, no. 1, pp. 1-58, Jan. 1992.
- [6] Mahmoud Parsian, Data Algorithms, Recipes for scaling up with Hadoop and Spark, Pearson's Correlation Coefficient, 2012, O'reilly Publications.
- [7] Steffen Bickel, Michael Brückner, and Tobias Scheffer, "Discriminative Learning Under Covariate Shift," J. Mach. Learn. Res. 10 (December 2009), pp. 2137-2155.
- [8] F. De Bernardinis, M. I. Jordan and A. SangiovanniVincentelli, "Support vector machines for analog circuit performance representation," Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451), Anaheim, CA, 2003, pp. 964-969.
- [9] ImageNet: A large-scale hierarchical image database, Link: http://vision.stanford.edu/pdf/ImageNet_CVPR2009.pdf.
- [10] K. Tuyls and G. Weiss, "Multiagent Learning: Basics, Challenges, and Prospects", AIMag, vol. 33, no. 3, p. 41, Sep. 2012.
- [11] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples," J. Mach. Learn. Res. 7 (December 2006), pp. 2399-2434.
- [12] Abhishek Kumar, Prasanna Sattigeri, and P. Thomas Fletcher, 'Semi-supervised learning with GANs: manifold invariance with improved inference,' in Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), pp. 5540-5550.
- [13] Mikhail Belkin and Partha Niyogi, "Semi-Supervised Learning on Riemannian Manifolds," Mach. Learn. 56, 1-3 (June 2004), pp. 209-239.
- [14] T. Miyato, S. Maeda, M. Koyama and S. Ishii, "Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 8, pp. 1979-1993, 1 Aug. 2019.
- [15] Patrice Simard, Bernard Victorri, Yann Le Cun, and John Denker, "Tangent prop: a formalism for specifying selected invariances in an adaptive network," in Proceedings of the 4th International Conference on Neural Information Processing Systems (NIPS'91), pp. 895-903.
- [16] L. Bruno, F.Chuan-Sheng, Z. Houssam, C. Vijay, "Manifold regularization with GANs for semi-supervised learning," arXiv:1807.04307, 2018, Link: <https://arxiv.org/abs/1807.04307>.