Comparative Study of Surrogate Modeling Methods for Signal Integrity and Microwave Circuit Applications

Thong Nguyen, Bobi Shi, Hanzhi Ma, *Graduate Student Member, IEEE*, Er-Ping Li, *Fellow, IEEE*, Xu Chen, *Member, IEEE*, Andreas C. Cangellaris, *Life Fellow, IEEE*, and José Schutt-Ainé, *Fellow, IEEE*

***Abstract*— With a short product cycle as we see today, fast and accurate modeling methods are becoming crucial for the development of new generation of electronics devices. Furthermore, increased complexity in circuitry and integration compounds design iteration and the associated, high-dimensional sensitivity analysis and performance optimization studies. Therefore, black-box surrogate models replacing the actual circuitry offer an attractive alternative for more efficient design iteration, optimization, and even direct Monte Carlo analysis. In this article, surrogate models built using nonparametric Gaussian process (GP) are presented. A robust framework based on probabilistic programming is used for training GP models. Other methods, such as partial least-square regression, support vector regression, and polynomial chaos, are used to compare with the performance of GP. Three design applications, a high-speed channel, a millimeter-wave filter, and a low-noise amplifier are used to demonstrate the robustness of the proposed GP-based surrogate models.**

***Index Terms*— Bayesian modeling, Gaussian process (GP), microwave circuits, nonintrusive method, sensitivity analysis, signal integrity (SI), stochastic analysis, surrogate model, variability analysis.**

I. Introduction

EXPEDIENT design iteration and performance optimization and design verification of state-of-the-art electronic devices and systems are hindered by the ever-increasing functionality integration. In the quest for computationally efficient methods capable of handling the high-dimensional design space of such devices and systems, machine learning (ML) methods are being explored recently for modeling and design optimization applications. In the following paragraphs, we provide an overview of recent and ongoing research pursuits in this that are most pertinent to the work reported in this article.

Using neural network for modeling, for frequency-domain analysis on passive linear time-invariant (LTI) systems, the works [1]–[5] use feedforward neural networks (FNNs) to learn the mapping from geometry parameters to electrical measure such as S-parameter. More recently, Torun *et al.* [6] introduced special layers into their NN, namely, causality enforcement layer (CEL) and passivity enforcement layer (PEL) to enforce local causality and passivity for NN-based models. For time-domain modeling, modeling methods using nonlinear autoregressive network with exogenous input type of recurrent neural network (NARX-RNN) were presented in [7]–[12], while

Elman RNN (ERNN) to model electrostastic discharge (ESD) circuits is used in [13]–[17], digital high-speed link, and so on without an explicit feedback connection in the model construction. Nguyen *et al.* [17] had a thorough review about different NN-based approaches for time-domain circuit simulation and detailed explanation about different types of RNN models and their advantages as well as disadvantages. Li *et al.* [18] combined RNN models with system identification to improve the prediction accuracy up to 99%. Other efforts have also been applied to predict performance of high-speed systems via measurements of the eye diagram. The work in [19] and [20], which use FNN-based models to solve the forward problem, is worth noting. A forward problem in this article refers to a problem where a set of design parameters is given and the electrical performance of a circuit is desired. For example, in a high-speed link, the channel geometry and equalization settings are inputs, while the eye openings (eye width and eye height) are the output. A comparative study between different ML methods, such as support vector regression (SVR) and FNN, has been reported in [21]–[23] focusing on predicting performance of a high-speed link system. The work reported in [24] uses an SVR-based model to address the design process as an inverse problem instead of an optimization problem. In [25] and [26], SVR and active subspace [27] are combined to perform reduced dimensionality regression. This results in

Manuscript received March 30, 2021; revised June 29, 2021; accepted July 15, 2021. Date of publication July 21, 2021; date of current version September 10, 2021. This work supported in part by the National Science Foundation under Grant CNS 16-24810, in part by the U.S. Army Small Business Innovation Research (SBIR) Program Office, in part by the U.S. Army Research Office under Contract W911NF-16-C-0125, and in part by the National Science Foundation (Center for Advanced Electronics through Machine Learning (CAEML) and its industry members) under Grant CNS16-24811. The work of Hanzhi Ma was supported by the Zhejiang University Academic Award for Outstanding Doctoral Candidates under Grant 202035. Recommended for publication by Associate Editor K. Aygun upon evaluation of reviewers' comments. *(Corresponding author: Thong Nguyen.)*

Thong Nguyen, Bobi Shi, Xu Chen, Andreas C. Cangellaris, and José Schutt-Ainé are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA (e-mail: tnnguye3@illinois.edu; bobishi2@illinois.edu; xuchen1@illinois.edu; cangella@illinois.edu; jesa@illinois.edu).

Hanzhi Ma and Er-Ping Li are with the College of Information Science and Electronic Engineering, ZJU-UIUC Institute, Zhejiang University, Hangzhou 310027, China (e-mail: mahanzhi@zju.edu.cn; liep@zju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCPMT.2021.3098666.

Digital Object Identifier 10.1109/TCPMT.2021.3098666

a speedup of the fitting process and facilitates the solution of the sensitivity analysis and design optimization problem. Nguyen and Schutt-Aine [28] used the Gaussian process (GP) building a surrogate model to study the behavior of a bandpass filter under the variation of its design parameters. Also, solving the forward problem, Larbi *et al.* [29], [30], Trinchero and Canavero [31], and Trinchero *et al.* [32] used partial least-square regression (PLS) and least-square support vector machine (LS-SVM) to perform not only predictions but sensitivity analysis on design problems with as many as 30 design parameters. In addition, realizing that LS-SVM has a deterministic nature, Trinchero and Canavero [31] proposed to combine LS-SVM and GP to create a fully statistical model that, in addition to predictive modeling, provides a confidence measure for its predictions.

In the uncertainty quantification (UQ) regime, in general, we are interested in knowing how a variation in the input would affect the output. Robust, adaptive, and computationally cheap methods for efficient stochastic analyses are favorable over naive Monte Carlo (MC) (brute-force) analysis due to the prohibitive cost of MC, especially for high-dimensional problems. There are various studies and reports on this problem using polynomial chaos expansion (PCE) methods [33], [34]. PCE methods can be categorized into intrusive methods, which requires the reformulation of the problem at hand to insert the randomness seeking for a PC representation of the solution [35], [36], and nonintrusive methods, which leave the deterministic model of interest untouched and use, instead, a sampling strategy to sample the data and fit the PCE [34], [37], [38]. Most surrogate modeling methods are suitable for nonintrusive uncertainty propagation, and once the input–output mapping is learned, propagating uncertainty from input to output can be obtained simply by running MC simulation on the surrogate model. Hence, the UQ problem can also be reduced to obtaining a very accurate and fast surrogate model.

Unlike the aforementioned methods, GP is stochastic in nature. A GP is not just a possible mapping that explains the seen data but a distribution of possible such mappings. Full Bayesian treatment applied to GP parameters during training is what makes it stand out from other surrogate modeling methods. This article reviews and compares different surrogate modeling methods, including GP, PLS, SVR, and PC in Section II. Then, Section III presents three examples to benchmark the performance of the models. As seen later, GP, more specifically, multioutput GP (MOGP), overall performs consistently well in both experiments; therefore, it offers an attractive option for input–output black-box modeling and for efficient uncertainty propagation and sensitivity analysis. Section IV concludes the experiments and points out potential extensions to further improve surrogate modeling techniques.

II. Surrogate Modeling Methods

In this section, different regression methods, which are compared against each other, are briefly reviewed. GP is presented in much more detailed as it is the main focus of this article. Linear regression (LR) and polynomial regression (PR) are also recalled as they are used as the baseline for the indicator of the nonlinearity level in the examples that are presented in Section II-A.

A. Gaussian Process Regression (GPR)

The understanding of GPR cannot be separated from that of Bayesian regression. In the following, nonlinear regression will be revisited in the context of Bayesian learning. Then, the connection between Bayesian regression and GP is established.

1) Bayesian Parametric Regression: Consider the probabilistic view of a regression problem: given a set of data $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)}), i = 1, 2, \ldots, N\}$ of N pairs of d-dimensional vector-valued input $\boldsymbol{x}^{(i)} \in \mathcal{X} = \mathbb{R}^d$ and function-valued output $y^{(i)} \in \mathbb{R}$ such that

$$y = f(\boldsymbol{x}) + \epsilon \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is independent and identically distributed (i.i.d.) [39] Gaussian noise, and we seek for the conditional mean of the output, y_*, at test input $\boldsymbol{x}_*$, namely $E(y_*|\boldsymbol{x}_*, \mathcal{D}) = f(\boldsymbol{x}_*)$. The Bayesian LR assumes a parametric form of f as $\exists \boldsymbol{\theta} \in \mathbb{R}^d$ such that

$$f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\theta}. \tag{2}$$

Imposing a prior on $\boldsymbol{\theta}$: $\boldsymbol{\theta} \sim \mathcal{N}(0, (1/d)\Sigma_\theta)$, the parameter's ($\boldsymbol{\theta}$'s) posterior is given by Bayes' rule

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\displaystyle\int_\theta p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}. \tag{3}$$

For a test (unseen) input $\boldsymbol{x}_*$, the predicted output y_* can be sampled from the posterior predictive distribution calculated by marginalizing $\boldsymbol{\theta}$ out

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}) = \int_\theta p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\mathrm{d}\boldsymbol{\theta}. \tag{4}$$

Usually, (3) is intractable due to the integral in the denominator. However, due to θ's prior and the noise being Gaussian, the posterior predictive distribution, (4), is also Gaussian. Hence, what is left is to find its mean and variance, which is a straightforward process. A Bayesian nonlinear regressor introduces a nonlinear transformation, often called the feature map, to transform a d-dimensional vector-valued input $\boldsymbol{x}$ to a d'-dimensional vector-valued feature z, $\boldsymbol{\varphi} : \mathbb{R}^d \longmapsto \mathbb{R}^{d'}$; (2) now becomes

$$f(\boldsymbol{x}) = \boldsymbol{\varphi}(\boldsymbol{x})^T \boldsymbol{\theta}. \tag{5}$$

The posterior predictive distribution now involves the term $\boldsymbol{\varphi}(\boldsymbol{x})^T \Sigma_\theta \boldsymbol{\varphi}(\boldsymbol{x}')$ instead of just simply $\boldsymbol{x}^T \Sigma_\theta \boldsymbol{x}'$ where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are 2 arbitrary inputs in either the training or the prediction points. Since $\Sigma_\theta \succ 0$, let

$$k(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{d}\boldsymbol{\varphi}(\boldsymbol{x})^T \Sigma_\theta \boldsymbol{\varphi}(\boldsymbol{x}') \tag{6}$$

be the covariance function. It characterizes the similarity and correlation between the features $\varphi(\boldsymbol{x})$ and, as we shall see in detail later, it gives rise to the conditional predictive distribution.

2) Nonparametric GP: A GP is a random process, i.e., a collection of random variables whose any finite set obeys a multivariate Gaussian distribution. This means that a GP can be fully characterized by a mean function $m(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$. We say that $f(\boldsymbol{x}) \sim \mathcal{GP}(m, k)$ iff for all $M \in \mathbb{N}$

$$\mathbf{f} = \left[f\left(\boldsymbol{x}^{(1)}\right) \;\; f\left(\boldsymbol{x}^{(2)}\right) \;\; \cdots \;\; f\left(\boldsymbol{x}^{(M)}\right) \right] \sim \mathcal{N}\left(\bar{\boldsymbol{f}}, \boldsymbol{K}\right) \quad (7)$$

where

$$\bar{\boldsymbol{f}}_i = m\left(\boldsymbol{x}^{(i)}\right) \quad (7a)$$

$m(\cdot)$ is, typically, without loss of generality, chosen to be $\mathbf{0}$ and

$$\boldsymbol{K_{ij}} = \mathrm{Cov}\left(f\left(\boldsymbol{x}^{(i)}\right), f\left(\boldsymbol{x}^{(j)}\right)\right) = k\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) \quad (7b)$$

which is also referred to as the Gram(ian) matrix.

Classical literature on this topic is [40]. GP is known as a probability distribution over a family of functions. A sample from a GP is a function or, more precisely, a finite set of (N) evaluations of a function. The kernel function gives rise to the covariance matrix of a multivariate Gaussian distribution, which needs to be positive semidefinite (PSD). This puts a constraint on the type of kernel functions that are valid to describe a GP.

There are many popular kernel functions that can be used to specify a GP prior: squared exponential, rational quadratic, Matern, radial basis function (RBF), periodic, and so on [40]. Kernel functions can also be combined together to represent a complex prior; the following kernel is the sum of q basic kernels:

$$k_a\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \sum_{i=1}^{q} k_i\left(\boldsymbol{x}, \boldsymbol{x'}\right). \quad (8)$$

Since $k_i(\boldsymbol{x}, \boldsymbol{x'})$'s are PSD, their sum, $k_a(\boldsymbol{x}, \boldsymbol{x'})$, is also PSD.

In addition, Duvenaud *et al.* [41] proposed additive kernels that act differently on each dimension of the input $\boldsymbol{x}$, which enriches the class of functions that can be represented by GP even more. Furthermore, such an approach can also capture different degrees of interactions between the input dimensions. The scope of this article will focus on basic kernels. Applications of GP involving additive kernels have been reported in [42] and [43].

3) Relationship Between GP and Bayesian Regression: Without loss of generality, we will use d as the dimension of the feature space in this section. In parametric Bayesian regression, (5) assumes that the model is a d-component basis expansion and could be rewritten as

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} \boldsymbol{\varphi_i}(\boldsymbol{x})\theta_i \quad (9)$$

and the kernel function is written as

$$k\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \frac{\sigma^2}{d} \sum_{i=1}^{d} \boldsymbol{\varphi_i}(\boldsymbol{x})\boldsymbol{\varphi_i}\left(\boldsymbol{x'}\right). \quad (10)$$

If we take $d \to \infty$, then (10) becomes a Riemann sum and

$$k\left(\boldsymbol{x}, \boldsymbol{x'}\right) = \int_{-\infty}^{\infty} \boldsymbol{\varphi_c}(\boldsymbol{x})\boldsymbol{\varphi_c}\left(\boldsymbol{x'}\right)\mathrm{d}c. \quad (11)$$

This is a remarkable result. First, it indicates that each feature map in Bayesian regression is corresponding to a kernel function in GP. For example, the squared exponential kernel function

$$k_{SE}\left(x, x'\right) = \sigma_{SE}^2 \exp\left[-\frac{\left(x - x'\right)^2}{2\ell_{SE}^2} \right] \quad (12a)$$

was derived from the feature map

$$\varphi_i(x) = \exp\left[-\frac{(x - c_i)^2}{2\ell^2} \right] \quad (12b)$$

and the prior imposed on the model coefficients

$$\theta_i \sim \mathcal{N}\left(0, \frac{\sigma^2}{d}\right) \quad (12c)$$

with $\sigma_{SE}^2 = (\pi)^{1/2}\ell\sigma^2$ and $\ell_{SE} = (2)^{1/2}\ell$

Second, instead of choosing a set of finite feature map basis $\varphi_i(\boldsymbol{x})$, an infinite order model ($d \to \infty$) is formed. The model can learn as much as available data. Its complexity increases as the data provided increases and does not depend on the complexity of the hypothesis, i.e., the number of parameters. It is especially clear when the GP model making predictions at unseen points, and the posterior predictive distribution requires (4) to marginalize the model parameters $\boldsymbol{\theta}$ out. Third, a reproducing kernel Hilbert space (RKHS), $k(\boldsymbol{x}, \boldsymbol{x'})$, is all needed to build a powerful nonparametric model from the data. Topologically speaking, when using feature maps to convert the original input space $(\boldsymbol{x})$ into the feature space, $\varphi(\boldsymbol{x})$, we only need to be able to compute the dot product of the feature maps, i.e., evaluating $k(\boldsymbol{x}, \boldsymbol{x'})$ for any pair of input $\boldsymbol{x}, \boldsymbol{x'}$, the transformation itself, $\varphi(\cdot)$, is not needed [39], [44]. Any valid GP kernel function must have a corresponding feature map in Bayesian regression point of view. Thus, in doing regression by GP, instead of putting a prior on the expansion coefficients of a specific basis function, we put a GP prior on the mapping we are interested in and compute the posterior distribution as the data becomes available. A thorough treatment of Hilbert space and RKHS with more details can be found in [44].

It can be seen that for large d and small N, performing regression by GP is preferred over using a feature map, while the reverse is true if the problem at hand has small d but N is large.

4) Inference in GP Models: GP regression is implemented by the following steps: first, a kernel is chosen, and this is equivalent to choosing a nonlinear feature map in classical regression. In this article, Matern-3/2 kernel [45] is chosen as the default kernel for all experiments. Kernel hyperparameters are optimized by minimizing the negative log likelihood (NLL) using the Adam optimizer [46], a stochastic gradient-based optimization method. The NLL can be analytically derived as

$$\mathrm{NLL}(\boldsymbol{\theta}) = \frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{K_{rr}}| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{K_{rr}^{-1}}\boldsymbol{y} \quad (13)$$

where $|\boldsymbol{K_{rr}}|$ is the determination of matrix $\boldsymbol{K_{rr}}$. The subscript r indicates that the Gram matrix $\boldsymbol{K_{rr}}$ is calculated from (7b) using training data. The gradient of NLL can also be

calculated analytically for the optimizer to iteratively optimize the hyperparameters

$$\nabla_{\theta_j}\mathrm{NLL}(\theta) = -\boldsymbol{y}^T\overline{\boldsymbol{K}}^{-1}\frac{\partial\overline{\boldsymbol{K}}}{\partial\theta_i}\overline{\boldsymbol{K}}^{-1}\boldsymbol{y} - \mathrm{tr}\left(\overline{\boldsymbol{K}}^{-1}\frac{\partial\overline{\boldsymbol{K}}}{\partial\theta_i}\right) \quad (14)$$

where $\overline{\boldsymbol{K}} = \boldsymbol{K}_{rr} + \sigma^2\boldsymbol{I}$. Once the training completes, as testing data, $\boldsymbol{x}_*$, are fed into the model, prediction, $\boldsymbol{y}_*$, can be made by sampling from the posterior distribution

$$p(\boldsymbol{y}_*|\boldsymbol{x}_*, \mathcal{D}) = \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{K}_*) \quad (15)$$

where

$$\boldsymbol{\mu}_* = \boldsymbol{K}_{tr}\overline{\boldsymbol{K}}^{-1}\boldsymbol{y} \quad (15a)$$

$$\boldsymbol{K}_* = \boldsymbol{K}_{tt} - \boldsymbol{K}_{tr}\overline{\boldsymbol{K}}^{-1}\boldsymbol{K}_{rt}. \quad (15b)$$

The subscript t stands for testing data. $\boldsymbol{K}_{tr}$ is the covariance submatrix between test and train data, i.e., $(\boldsymbol{K}_{tr})_{ij} = k(\boldsymbol{x}_t^{(i)}, \boldsymbol{x}_r^{(j)})$.

An m-component output GP, generally referred to as MOGP in this article, can be constructed by different methods. An extensive review of these methods can be found in [45]. As pointed out in [40] and [45], the complexity of the kernel in GP will affect the training time, and therefore, in order to remain computationally cheap while achieving robustness for multioutput cases, the MOGP kernel in this article is created by using a linear mixture of multiple single-output GPs. For example, if the output has two features, i.e., $m = 2$, the model can be constructed as

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} u^{(1)} \\ u^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a_1} & \boldsymbol{a_2} \end{bmatrix}\begin{bmatrix} u^{(1)} \\ u^{(2)} \end{bmatrix} \quad (16)$$

where $u^{(1)}$ and $u^{(2)}$ are output of two latent single-output GP, $\boldsymbol{a_1} = [\,a_{11}\ a_{12}\,]^T$, and $\boldsymbol{a_2} = [\,a_{21}\ a_{22}\,]^T$. The covariance matrix of the MOGP relates to that of the latent GPs, $\boldsymbol{K_u}$, by

$$\mathrm{cov}\big(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')\big) = \left(\sum_{i=1}^{m}\boldsymbol{a}_i\boldsymbol{a}_i^T\right)\boldsymbol{K_u}. \quad (17)$$

The coefficient a_{ij}'s are also considered hyperparameters, included in $\boldsymbol{\theta}$. As pointed out in [45], when there is no correlation between outputs (referred to as tasks in [45]), the a_{ij}, $i \neq j$, coefficients are zero. An m-component MOGP is equivalent to m independent GPs. This phenomenon is known as autokrigeability.

B. PLS Regression

PLS regression relies on the idea of principal component analysis (PCA). Principal component regression (PCR) involves the PCA in which the input space is reduced to the principal component space; then, an interpolation is carried out between a few significant principal components and the output. Assume a multi-input–multioutput system, i.e., $y \in \mathbb{R}^q$. Let

$$\boldsymbol{X} = \boldsymbol{V}\boldsymbol{P}^T \quad (18a)$$

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{Q}^T \quad (18b)$$

be the principal decomposition of $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ and $\boldsymbol{Y} \in \mathbb{R}^{N\times q}$, and $\boldsymbol{V}, \boldsymbol{P}, \boldsymbol{U}$, and $\boldsymbol{Q}$ are of appropriate dimensions. PCR perform regression on V and $\boldsymbol{U}$. We can see that though $\boldsymbol{V}$ best describes inputs and $\boldsymbol{U}$ best describes outputs as PCA was applied to both input and output, it was applied separately. PLS fixes this limitation, and it iteratively projects input and output onto the most significant components, but the projection happens in a leapfrog scheme so that there is cross-information exchange between input and output while doing projections. Process details can be found in [29] and [30]. After L projections, we obtain an L-component decomposition of $\boldsymbol{X}$ and $\boldsymbol{Y}$, $\boldsymbol{V}, \boldsymbol{U} \in \mathbb{R}^{N\times L}$ and $\boldsymbol{P} \in \mathbb{R}^{d\times L}$, $\boldsymbol{Q} \in \mathbb{R}^{q\times L}$. Now, a regression model can be created using $\boldsymbol{U}$ and $\boldsymbol{V}$

$$\boldsymbol{U} = \boldsymbol{V}\boldsymbol{\theta}. \quad (19)$$

Predictions can be obtained by

$$\boldsymbol{Y} = \boldsymbol{U}\,\boldsymbol{Q}^T = \boldsymbol{V}\boldsymbol{\theta}\,\boldsymbol{Q}^T = \boldsymbol{X}\boldsymbol{P}\boldsymbol{\theta}\,\boldsymbol{Q}^T. \quad (20)$$

For single-output case, the process finding $\boldsymbol{V}$ and $\boldsymbol{U}$ becomes a one-step calculation, while for multioutput case, it is iterative. In this article, single-output and multioutput PLS are used as two different approaches and will also be benchmarked against each other. To find the optimal L, a cross-validation scheme is used, multiple PLS models are built as L varies, and the optimal L is chosen when the corresponding model achieves the lowest fitting error.

C. Support Vector Regression

SVR [47], an important branch of SVM [48], aims to solve the regression prediction problem by finding a regressor, $h(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x}) + b$, that approximates y, i.e., $y \approx h(\boldsymbol{x})$. SVR predictive function can be expressed as

$$y \approx h(\boldsymbol{x}) = \sum_{1\leq i\leq N}\big(\alpha_i' - \alpha_i\big)\kappa\big(\boldsymbol{x}, \boldsymbol{x}^{(i)}\big) + b \quad (21)$$

where $\alpha_i \geq 0$ and $\alpha_i' \geq 0$ are introduced as Lagrange multipliers and we select the Gaussian kernel in this problem [21], [26]

$$\kappa\big(\boldsymbol{x}, \boldsymbol{x}^{(i)}\big) = \exp\left(\frac{-\big\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\big\|^2}{2\sigma^2}\right) \quad (22)$$

where $\sigma > 0$ is the width of Gaussian kernel.

The SVR algorithm is implemented by MATLAB Statistics and Machine Learning Toolbox, whose hyperparameters, e.g., σ, are calculated by Bayesian optimization method in order to minimize the cross-validation error and provide accurate prediction results with robustness.

D. Polynomial Chaos

The PC method estimates an arbitrary random variable of interest as a function of other random variables with a given distribution as an orthonormal polynomial expansion. The general form of multidimensional polynomials is estimated as [49]

$$y \approx \sum_{i=0}^{P} c_i\Phi_i(\boldsymbol{x}) \quad (23)$$

where c_i denotes the unknown polynomial coefficients to be determined and $\Phi_i(\boldsymbol{x})$ represents multidimensional orthonormal polynomials, constructed using the product of the 1-D orthonormal polynomials, via

$$\boldsymbol{\Phi}_i = \prod_{k \in \mathcal{K}_i} \phi_k \tag{24}$$

where $\mathcal{K}_i$ is multi-index set for 1-D orthonormal polynomials

$$\mathcal{K}_i = \{k_{i1}, \ldots, k_{id}\}, \sum k_{ij} \leq M. \tag{25}$$

The number of basis is given by $P = ((M+d)!/(M!d!))$, where M is the polynomial order. Depending on the distribution of input variables $\boldsymbol{x}$, the polynomial basis $\phi(\cdot)$ is chosen accordingly to make the bases orthogonal to each other. Among them, Hermite polynomials [50] are used to represent Gaussian random inputs but have also been proved to be effective for other distribution [49]. Therefore, PC models in this article are implemented using Hermite polynomials, regardless of the input distribution type. The expansion coefficients, c_i's, are linear w.r.t y, and hence, for a set of N samples, they can be found by a simple least-square solution.

E. Polynomial Regression

PR is the simplest regression method but sometimes appears to be very effective in modeling real-world data. PR assumes that the mapping between the input $\boldsymbol{x}$ and the output y is given by

$$y = \sum_{i=0}^{P-1} \beta_i \prod_{j=1}^{d} x_j^{k_j} \tag{26}$$

where P is the number of terms, given by $P = ((M+d)!/(M!d!))$, m is the polynomial order, and x_j is the jth component of $\boldsymbol{x} \in \mathbb{R}^d$ or the jth input variable. The monomial power $k_j \geq 0$ must satisfy the condition

$$\sum_{j=1}^{d} k_j \leq M. \tag{27}$$

When $M = 1$, the regression model is linear, hence, referred to as LR. For example, if $M = 3$ and $d = 2$, $P = 6$ and the PR model reads

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 \tag{28}$$

and the LR model only has linear terms

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2. \tag{29}$$

PR and LR are included in the study as the baseline to indicate how much nonlinearity exists in the problem. For a problem that LR performs well on, the input–output mapping must be linear and vice versa.

III. Examples

In this section, different examples in high-speed and microwave circuit designs will be used to investigate the performance of various surrogate models. Training data are collected by simulations using commercial solvers from Keysight (for circuit simulations) and Ansys (for electromagnetics simulations). Each experiment is designed to build a predictive model starting with $N = 10$ randomly distributed training samples. Once a model is generated, it is validated by calculating the coefficient of determination, or R^2 score, between the true values $\hat{y}$ and the predicted values y, defined as

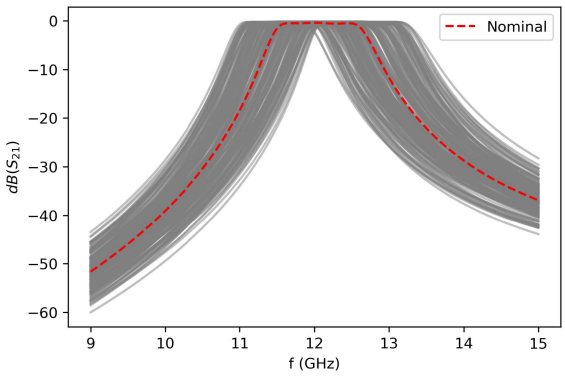

Fig. 1. Filter insertion loss variations.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left\| y^{(i)} - \hat{y}^{(i)} \right\|^2}{\sum_{i=1}^{N} \left\| \hat{y}^{(i)} - \bar{y} \right\|^2} \tag{30}$$

where $\bar{y} = (1/N) \sum_{i=1}^{N} \hat{y}^{(i)}$. An R^2 score of 1.0 means that the predicted and the true values are in perfect agreement. If the validation R^2 score of a model reaches 0.99, the training for that particular model may be stopped, and this article will sometimes refer to this fact as the convergence of a model. If the model has not reached convergence, i.e., $R^2 < 0.99$, a number of training samples is added and the model is retrained using this new set of training data. It is worth noting that in cases where the output is multidimensional, when using single-output models such as PLS or SVR, we have to create multiple-independent such models. Since MOPLS, MOGP, and PC are multioutput models, only one model is needed. In the following, training R^2 scores for each output feature are reported separately as they are different for single-output models, but one should expect a single R^2 score to report for multioutput models.

As demonstrated in the following, the PLS method, though fast, is inherently linear and, hence, unable to capture nonlinear input–output mapping correctly. PC method, on the other hand, appears to overhandle the nonlinearity and, hence, struggles to achieve a fast convergence rate when the underlying mapping is linear.

A. Milimeter-Wave Filter

The first example is a 12-GHz coupled-line bandpass filter [28]. There are 12 geometry-related design variables, such as lengths, widths, and separations of the coupled lines, and stack-up features such as dielectric height, permittivity, and loss tangent. They are all varied by 15% (±7.5%) from their nominal values. Fig. 1 shows the insertion loss of the filter as design variables vary. To quantify the insertion loss of the filter, the center frequency (y_0), bandwidth (y_1), and shape

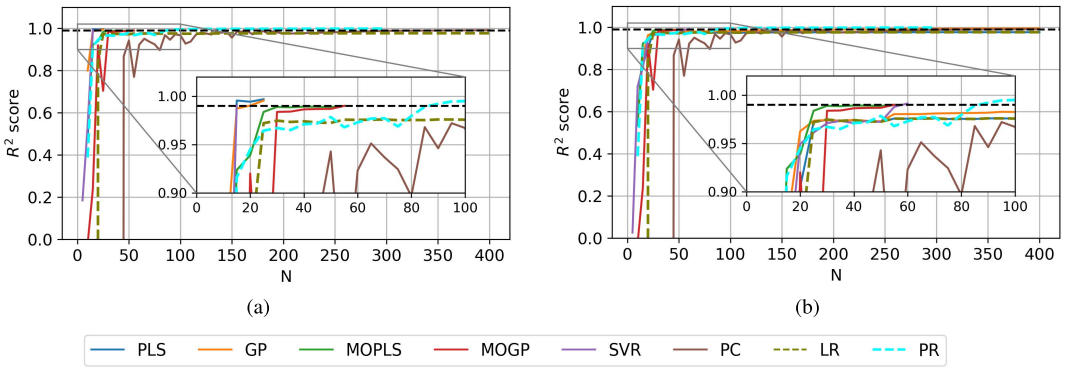

Fig. 2. Validation R^2 score during training with different numbers of training samples (N). The black dashed line represents $R^2 = 0.99$. (a) Center frequency as output. (b) Shape factor as output.

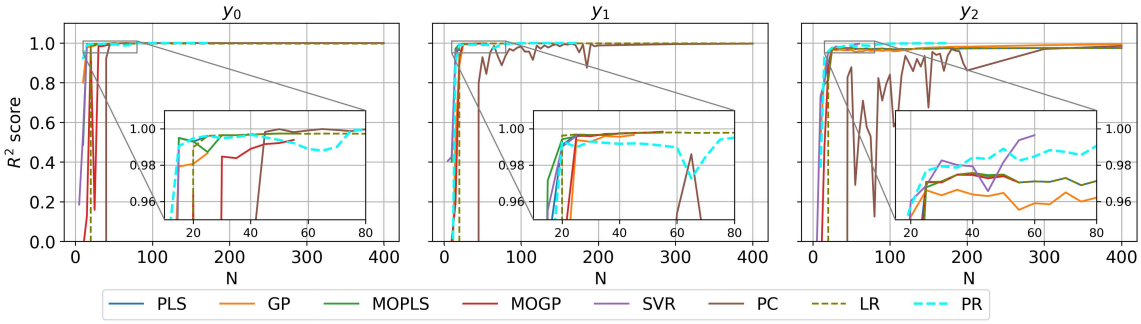

Fig. 3. Test performance predicting center frequency (y_0), bandwidth (y_1), and shape factor (y_2) of an millimeter-wave bandpass filter; input variables are independent Gaussian distributed.

factor (y_2) are calculated. Surrogate models were created to predict these three figures of merit (FOMs).

Fig. 2 shows the training process for each FOM. First, most models converge quite fast, and only the PC model requires a large number of samples to reach a validation R^2 score of 0.99. Second, single-output models have more difficulties in learning the shape factor than the center frequency and the bandwidth as shown in Fig. 2(b), and PLS and SVR models require more training samples to converge than MOPLS or MOGP. As mentioned before, MOPLS, MOGP, and PC are multioutput models, and there is a single R^2 score to determine the convergence of them. Table I(a) shows the minimum training sample required for each model to reach 0.99 validation R^2 score. Though LR models did not reach 0.99 validation R^2 score even when all others have ($N = 400$), the fact that it was able to achieve a validation R^2 score higher than 0.96 for all three outputs indicates that even though this is a high-dimensional problem, the mapping between the design variables and the insertion loss FOMs is mostly linear.

Fig. 3 shows the test R^2 score for each output using different models trained by different values of N. The test performance of the models is consistent with their validation R^2 score, and most models achieve test R^2 scores above 0.96.

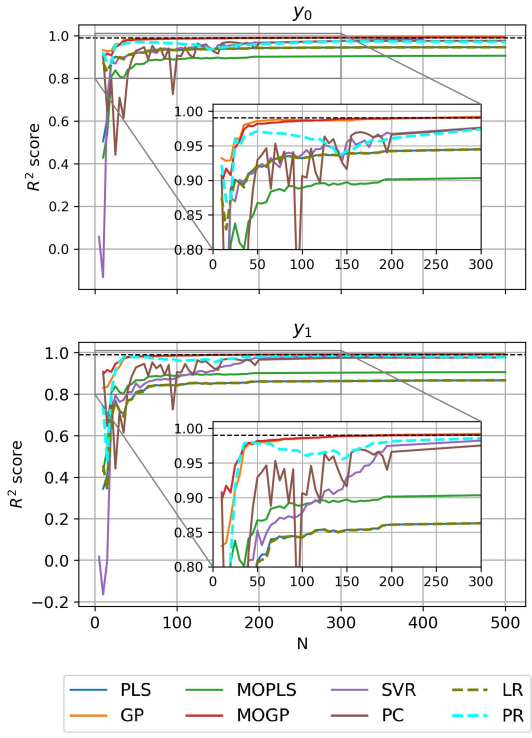

Fig. 4. Validation R^2 score during training when varying N. Black dashed line represents $R^2 = 0.99$.

B. High-Speed Link

We consider a chip-to-chip, high-speed serial link model, which involves 16 geometry-related design parameters associated with the stack-up and transmission lines [25], [26]. The link performance is quantified by looking at the eye opening at the receiver (RX), after equalization.

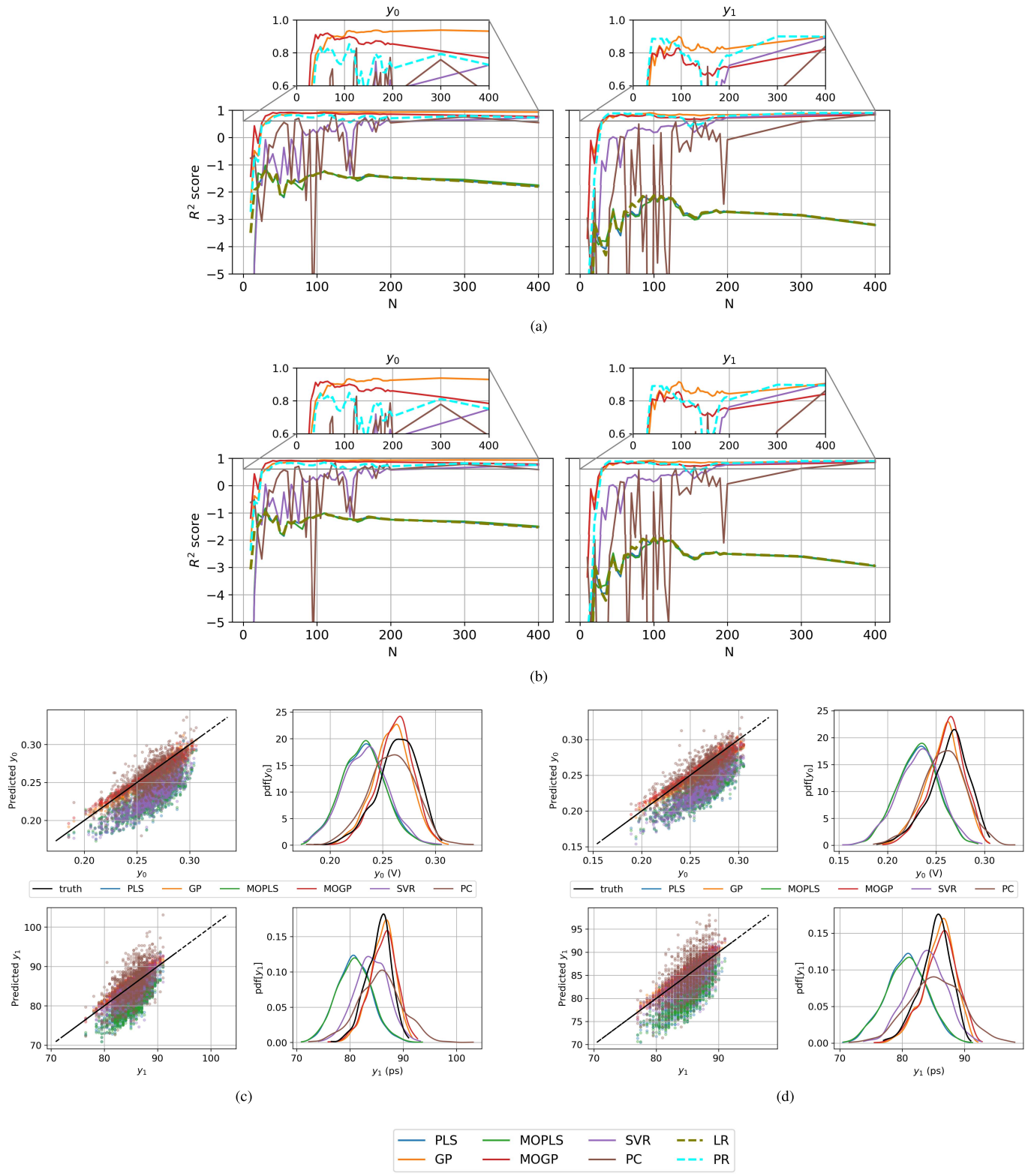

Fig. 5. Performance of trained models predicting eye height (y_0) and eye width (y_1). (a) Test performance with independent Gaussian distributed input variables. (b) Test performance with correlated Gaussian distributed input variables. (c) Predictive distribution by models trained with $N = 50$; input variables are independent Gaussian distributed. (d) Predictive distribution by models trained with $N = 50$; input variables are correlated Gaussian distributed.

These 16 parameters constitute an input parameter space of relatively high dimensionality such that a brute-force parameter sweep is intractable. A surrogate model that can quickly generate the eye openings from the geometric inputs is, therefore, imperative as it can be used for design optimization or uncertainty propagation.

TABLE I

MINIMUM TRAINING SAMPLE FOR EACH MODEL TO REACH 0.99 VALIDATION R^2 SCORE. EACH ROW IS FOR EACH OUTPUT. N/A MEANS THE MODEL DID NOT REACH 0.99 VALIDATION R^2 SCORE WITHIN SWEPT VALUES OF N

(a)

	PLS	GP	MOPLS	MOGP	SVR	PC	LR	PR
y_0	15	20	55	55	15	400	N/A	90
y_1	25	20			25			
y_2	N/A	300			60			

(b)

	PLS	GP	MOPLS	MOGP	SVR	PC	LR	PR
y_0	N/A	300	N/A	300	N/A	N/A	N/A	N/A
y_1	N/A	185			800			

(c)

	PLS	GP	MOPLS	MOGP	SVR	PC	LR	PR
y_0	N/A	30	N/A	30	30	30	N/A	45
y_1	N/A	30			20			
y_2	N/A	30			30			

Fig. 4 shows the training result for eye width (y_0) and eye height (y_1). PLS (both single-output and multioutput) models never reach the validation R^2 score of 0.99. While GP-based models quickly approach the threshold, followed by SVR model, PC model has roughly the same convergence rate as SVR.

As shown in Table I(b), when N was varied up to 900, only GP models were able to consistently reach to 0.99 validation R^2 score for all outputs. Besides PLS-based models and LR, other models though did not reach the target value 0.99, their validation R^2 scores are all well above 0.96 for $N > 200$.

The trained models are then used to perform uncertainty propagation tests. A set of 1000 test samples are collected for each test.

The inputs are sampled from two multivariate Gaussian distributions for two tests: one independent Gaussian distribution and the other one assuming correlations between the inputs. Fig. 5 shows the R^2 score for the two tests. PC models when trained with few samples appear to have generalization issue as its test R^2 score varies widely. Fig. 5(c) and (d) shows the comparison of the predictive distribution of the outputs with the true distribution when all models are trained with $N = 50$ samples. The output distributions obtained from GP-based models prediction follow the true distribution quite well for both tests. As N increases, eventually that obtained from SVR and PC models also matches the true distribution well. Even with as many training points as $N = 1000$, PLS models still could not reach acceptable prediction accuracy as others.

Unlike the previous example, LR and PLS cannot learn the data well while PR can. This suggests that the underlying mapping is nonlinear, which explains why PLS is unable to reach higher R^2 score during training, thus failing to predict the output when the input is sampled from a distribution different from the one that generated the training data.

C. Low-Noise Amplifier

Finally, a two-stage low-noise amplifier (LNA) designed for a carrier frequency of 8 GHz is studied. The design

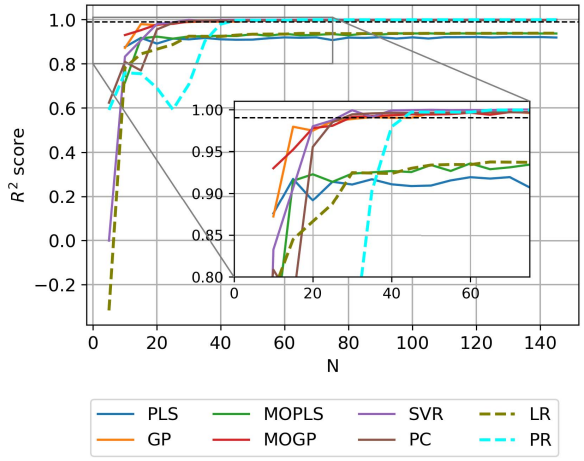

Fig. 6. Validation R^2 score during training the LNA model when varying N. Black dashed line represents $R^2 = 0.99$.

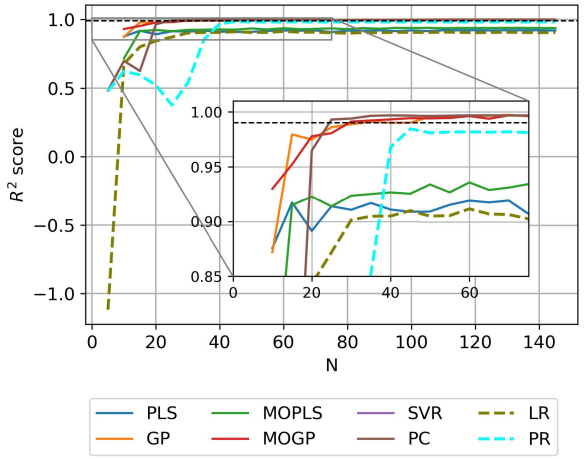

Fig. 7. Testing R^2 score for the LNA model predicting the gain. Black dashed line represents $R^2 = 0.99$.

comprises two amplifier stages sandwiched by three matching networks. For this variability analysis study, the width of a microstrip quarter-wave transformer in the input matching network, the spacing and linewidth of a spiral inductor, and the length of a shorted stub from the middle matching network and a series resistor from the second amplifying stage are varied. Three quantities of interest at operating frequency, which include total gain, output return loss, and output noise figure, are modeled. The design was fine-tuned based on an example found in Keysight ADS example directory [51].

The variability analysis was performed on the optimized design, and hence, even 20% variation from their optimal values of the design variables does not yield large variations in output quantities. Therefore, similar to the filter example, the input–output mapping is relatively simple for all models to capture with a small number of samples, as shown in Fig. 6. This is well expected because the number of samples needed to generally sufficient cover the output space depends on the variance of the outputs. However, GP models are among models that reach convergence with the fewest number of training samples compared with other traditional methods. It is worth noting that, inherently linear models, such as LR and PLS,

TABLE II
SUMMARY OF REVIEWED SURROGATE MODELS

Method	Advantages	Disadvantages
PLS MOPLS	Fast training, fast inference.	Only linear problems.
GP MOGP	Can handle nonlinear problems, Hyper-parameters learnt directly from data.	Training time scales with $\mathcal{O}\left(N^3\right)$
SVR	Can handle nonlinear problems.	Require hyper-parameters tuning (cross-validation, Bayesian optimization etc.), Training time scales with $\mathcal{O}\left(N^2\right)$ in the best case scenario
PC	Can handle nonlinear problems, straightforward implementation, no hyper-parameters	Input distribution dependence, Training time scales with $\mathcal{O}\left(N^2\right)$

can only explain the linear part of the data and hence never reach the convergence defined by the 0.99 threshold. Fig. 7 shows the test performance of the trained models on more than 3000 unseen samples. It is noticeable that the PR model, though converged during training, slightly underperforms as its prediction accuracy was below 0.99, while all other ML methods not only converged with a fewer number of samples but also yield higher accuracy in predictions.

IV. CONCLUSION AND FUTURE WORK

GP-based modeling method has been reviewed in this article. Due to the Bayesian treatment to the hyperparameters, they can be learned from the data by maximizing the data likelihood, which autotunes the hyperparameters to be most suitable for the provided data. Other methods, such as PLS, SVR, and PC, are also briefly discussed and benchmarked against GP models. Table II summarizes the key advantages and drawbacks of the methods studied in this article. Via three examples, it was demonstrated that all of the reviewed methods show good agreement with the true distribution of the test data though some methods would require more training data than others. PLS models are extremely fast and straightforward to implement, but they may fail to capture the strongly nonlinear mapping. MOGP though takes more time to train consistently yields good results while using the fewest number of training samples. Especially, GP models mainly suffer from numerical inefficiency only when a large dataset involves, making it suitable to model problems where only a small number of training data are available due to the expensive cost to collect data.

Training data are crucial to the convergence rate of surrogate models. Especially for GP models, training samples if chosen appropriately would quickly minimize the overall uncertainty in the model prediction. Potential extension of this work should investigate which sampling method provides the best convergence for GP models.

REFERENCES

[1] G. Xue-Lian, C. Zhen-Nan, F. Nan, Z. Xiao-Yu, and H. Jian-Hong, "An artificial neural network model for S-parameter of microstrip line," in *Proc. Asia–Pacific Symp. Electromagn. Compat. (APEMC)*, May 2013, pp. 1–4.

[2] X. Zhang, Y. Cao, and Q. J. Zhang, "A combined transfer function and neural network method for modeling via in multilayer circuits," in *Proc. 51st Midwest Symp. Circuits Syst.*, Aug. 2008, pp. 73–76.

[3] F. Feng, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric modeling of EM behavior of microwave components using combined neural networks and pole-residue-based transfer functions," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 60–77, Jan. 2016.

[4] 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.

[5] T. Nguyen and J. E. Schutt-Aine, "A pseudo-supervised machine learning approach to broadband LTI macro-modeling," in *Proc. IEEE Int. Symp. Electromagn. Compat. IEEE Asia–Pacific Symp. Electromagn. Compat. (EMC/APEMC)*, May 2018, pp. 1018–1021.

[6] H. M. Torun, A. C. Durgun, K. Aygun, and M. Swaminathan, "Enforcing causality and passivity of neural network models of broadband S-parameters," in *Proc. IEEE 28th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2019, pp. 1–3.

[7] Z. Aimin, Z. Hang, L. Hong, and C. Degui, "A recurrent neural networks based modeling approach for internal circuits of electronic devices," in *Proc. 20th Int. Zurich Symp. Electromagn. Compat.*, Jan. 2009, pp. 293–296.

[8] 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.

[9] Y. Fang, M. C. E. Yagoub, F. Wang, and Q.-J. Zhang, "A new macro-modeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 48, no. 12, pp. 2335–2344, Dec. 2000.

[10] Y. Cao and Q.-J. Zhang, "A new training approach for robust recurrent neural-network modeling of nonlinear circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 6, pp. 1539–1553, Jun. 2009.

[11] W. Liu, W. Na, L. Zhu, and Q.-J. Zhang, "A review of neural network based techniques for nonlinear microwave device modeling," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jul. 2016, pp. 1–2.

[12] H. Yu, T. Michalka, M. Larbi, and M. Swaminathan, "Behavioral modeling of tunable I/O drivers with preemphasis including power supply noise," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 1, pp. 233–242, Jan. 2020.

[13] T. Nguyen, T. Lu, J. Sun, Q. Le, K. We, and J. Schut-Aine, "Transient simulation for high-speed channels with recurrent neural network," in *Proc. 27rd IEEE Conf. Elect. Perform. Electron. Packag. Syst.*, Oct. 2018, pp. 303–305.

[14] A. Beg, P. W. C. Prasad, M. M. Arshad, and K. Hasnain, "Using recurrent neural networks for circuit complexity modeling," in *Proc. IEEE Int. Multitopic Conf.*, Dec. 2006, pp. 194–197.

[15] W.-T. Hsieh, C.-C. Shiue, and C.-N. J. Liu, "A novel approach for high-level power modeling of sequential circuits using recurrent neural networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 3591–3594.

[16] Z. Chen, M. Raginsky, and E. Rosenbaum, "Verilog—A compatible recurrent neural network model for transient circuit simulation," in *Proc. IEEE 26th Conf. Elect. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2017, pp. 1–3.

[17] T. Nguyen, T. Lu, K. Wu, and J. Schutt-Aine, "Fast transient simulation of high-speed channels using recurrent neural network," 2019, *arXiv:1902.02627*. [Online]. Available: http://arxiv.org/abs/1902.02627

[18] B. Li, B. Jiao, M. Huang, R. Mayder, and P. Franzon, "Improved system identification modeling for high-speed receiver," in *Proc. IEEE 28th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2019, pp. 1–3.

[19] N. Ambasana, G. Anand, D. Gope, and B. Mutnury, "S-parameter and frequency identification method for ANN-based eye-height/width prediction," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 7, no. 5, pp. 698–709, May 2017.

[20] C. H. Goay, P. Goh, N. S. Ahmad, and M. F. Ain, "Eye-height/width prediction using artificial neural networks from S-parameters with vector fitting," *J. Eng. Sci. Technol.*, vol. 13, no. 3, pp. 625–639, Mar. 2018.

[21] H. Ma, E.-P. Li, A. C. Cangellaris, and X. Chen, "Comparison of machine learning techniques for predictive modeling of high-speed links," in *Proc. IEEE 28th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2019, pp. 1–3.

[22] T. Lu, J. Sun, K. Wu, and Z. Yang, "High-speed channel modeling with machine learning methods for signal integrity analysis," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 6, pp. 1957–1964, Dec. 2018.

[23] R. Trinchero and F. G. Canavero, "Modeling of eye diagram height in high-speed links via support vector machine," in *Proc. IEEE 22nd Workshop Signal Power Integrity (SPI)*, May 2018, pp. 1–4.

[24] R. Trinchero, M. A. Dolatsara, K. Roy, M. Swaminathan, and F. G. Canavero, "Design of high-speed links via a machine learning surrogate model for the inverse problem," in *Proc. Electr. Design Adv. Packag. Syst. (EDAPS)*, Dec. 2019, pp. 1–3.

[25] H. Ma, E.-P. Li, A. C. Cangellaris, and X. Chen, "High-speed link design optimization using machine learning SVR-AS method," in *Proc. IEEE 29th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2020, pp. 1–3.

[26] H. Ma, E.-P. Li, A. C. Cangellaris, and X. Chen, "Support vector regression-based active subspace (SVR-AS) modeling of high-speed links for fast and accurate sensitivity analysis," *IEEE Access*, vol. 8, pp. 74339–74348, 2020.

[27] P. G. Constantine, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2015.

[28] T. Nguyen and J. Schutt-Aine, "Gaussian process surrogate model for variability analysis of RF circuits," in *Proc. IEEE Electr. Design Adv. Packag. Syst. (EDAPS)*, Dec. 2020, pp. 1–3.

[29] M. Larbi, R. Trinchero, F. G. Canavero, P. Besnier, and M. Swaminathan, "Analysis of parameter variability in an integrated wireless power transfer system via partial least-squares regression," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 11, pp. 1795–1802, Nov. 2020.

[30] M. Larbi, R. Trinchero, F. G. Canavero, P. Besnier, and M. Swaminathan, "Analysis of parameter variability in integrated devices by partial least squares regression," in *Proc. IEEE 24th Workshop Signal Power Integrity (SPI)*, May 2020, pp. 1–4.

[31] R. Trinchero and F. G. Canavero, "Combining LS-SVM and GP regression for the uncertainty quantification of the EMI of power converters affected by several uncertain parameters," *IEEE Trans. Electromagn. Compat.*, vol. 62, no. 5, pp. 1755–1762, Oct. 2020.

[32] R. Trinchero, M. Larbi, M. Swaminathan, and F. G. Canavero, "Statistical analysis of the efficiency of an integrated voltage regulator by means of a machine learning model coupled with Kriging regression," in *Proc. IEEE 23rd Workshop Signal Power Integrity (SPI)*, Jun. 2019, pp. 1–4.

[33] J. Slim *et al.*, "Polynomial chaos expansion method as a tool to evaluate and quantify field homogeneities of a novel waveguide RF Wien filter," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 859, pp. 52–62, Jul. 2017.

[34] M. A. Dolatsara, A. Varma, K. Keshavan, and M. Swaminathan, "A modified polynomial chaos modeling approach for uncertainty quantification," in *Proc. Int. Appl. Comput. Electromagn. Soc. Symp. (ACES)*, Apr. 2019, pp. 1–2.

[35] X. Chen, M. Qiu, J. E. Schutt-Aine, and A. C. Cangellaris, "Stochastic LIM for transient simulation of circuits with uncertainties," in *Proc. IEEE 23rd Conf. Electr. Perform. Electron. Packag. Syst.*, Oct. 2014, pp. 29–32.

[36] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1533–1545, Oct. 2013.

[37] C. Cui and Z. Zhang, "High-dimensional uncertainty quantification of electronic and photonic IC with non-Gaussian correlated process variations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 8, pp. 1649–1661, Aug. 2020.

[38] C. Cui and Z. Zhang, "Stochastic collocation with non-Gaussian correlated process variations: Theory, algorithms, and applications," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 9, no. 7, pp. 1362–1375, Jul. 2019.

[39] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," in *Springer Series in Statistics*. New York, NY, USA: Springer, 2001.

[40] C. E. Rasmussen and C. K. I. Williams, "Gaussian processes for machine learning," in *Adaptive Computation and Machine Learning*. Cambridge, MA, USA: MIT Press, 2005.

[41] D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen, "Additive Gaussian processes," in *Adv. Neural Inf. Process. Syst.*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2011, pp. 226–234. [Online]. Available: http://papers.nips.cc/paper/4221-additive-gaussian-processes.pdf

[42] H. M. Torun and M. Swaminathan, "High-dimensional global optimization method for high-frequency electronic design," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 6, pp. 2128–2142, Jun. 2019.

[43] H. M. Torun, M. Swaminathan, A. K. Davis, and M. L. F. Bellaredj, "A global Bayesian optimization algorithm and its application to integrated system design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 4, pp. 792–802, Apr. 2018.

[44] V. I. Paulsen and M. Raghupathi, "An introduction to theory reproducing kernel Hilbert spaces," in *Cambridge Studies in Advanced Mathematics*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[45] M. A. Alvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Found. Trends Mach. Learn.*, vol. 4, no. 3, pp. 195–266, 2012, doi: 10.1561/2200000036.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[47] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[48] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer, 1995.

[49] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Comput.*, vol. 24, no. 2, pp. 619–644, Jul. 2002, doi: 10.1137/S1064827501387826.

[50] R. H. Cameron and W. T. Martin, "The orthogonal development of nonlinear functionals in series of Fourier-Hermite functionals," *Ann. Math.*, vol. 48, no. 2, pp. 385–392, 1947. [Online]. Available: http://www.jstor.org/stable/1969178

[51] *ADS Example Files and Models*. Accessed: Jul. 16, 2021. [Online]. Available: https://edadocs.software.keysight.com/display/support/ADS%20Example%20F%iles%20and%20Models?

Thong Nguyen received the B.S. degree in electrical engineering from the Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam, in 2013, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign (UIUC), Urbana, IL, USA, in 2017 and 2021, respectively.

He is currently a Post-Doctoral Researcher at UIUC in 2021. He had been working at Qualcomm, Inc., San Diego, CA, USA; Hewlett-Packard Enterprise, Colorado Springs, CO, USA; and Ansys, Champaign, IL, USA, during his doctoral study. His research interests focus on macromodeling techniques, design optimization for signal integrity (SI), and more recently, machine learning methods for modeling, and cloud-based electronic design automation (EDA) algorithms and tools.

Dr. Nguyen has been serving as a Reviewer for *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* since 2019.

Bobi Shi received the M.S. degree from the University of Illinois Urbana–Champaign (UIUC), Urbana, IL, USA, in 2020, where he is currently pursuing the Ph.D. degree in electrical engineering.

He is working in the area of high-speed link simulation and jitter analysis. His current research interests include optimized methods for high-speed system modeling, SI issues in high-speed systems, and equalization design and simulation.

Hanzhi Ma (Graduate Student Member, IEEE) received the B.S. degree in engineering from Zhejiang University, Hangzhou, China, in 2017. She is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University–University of Illinois at Urbana–Champaign Institute, Zhejiang University.

Her current research interests include machine learning techniques for electromagnetic interference (EMI)/SI/power integrity (PI) analysis and SI analysis for neuromorphic chip.

Dr. Ma was a recipient of the 2020 IEEE EMC Society President's Memorial Award.

Er-Ping Li (Fellow, IEEE) is currently a Changjiang-Distinguished Professor with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, and the Dean of the Joint Institute of Zhejiang University–University of Illinois at Urbana–Champaign. Since 1989, he has served as a Research Fellow, a Principal Research Engineer, an Associate Professor, and the Technical Director at Singapore Research Institute and University, Singapore. In 2000, he joined the Institute of High Performance Computing, A*STAR, Singapore, as a Principal Scientist and the Director. He has authored or coauthored over 300 articles published in the referred international journals and authored two books published by John-Wiley–IEEE Press and Cambridge University Press. He holds and has filed a number of patents at the U.S. patent office. His research interests include electrical modeling and design of microscale/nanoscale integrated circuits, 3-D electronic package integration, and nanoplasmonic technology.

Dr. Li is a Fellow of the MIT Electromagnetics Academy, USA. He is also a Founding Member of the IEEE MTT-RF Nanotechnology Committee. He was a recipient of the 2015 IEEE Richard Stoddard Award on EMC, the IEEE EMC Technical Achievement Award, the Singapore IES Prestigious Engineering Achievement Award, the Changjiang Chair Professorship Award from the Ministry of Education in China, and a number of best paper awards. He was elected to the IEEE EMC Distinguished Lecturer in 2007. He has been a general chair and a technical chair for many international conferences. He was the President for the 2006 International Zurich Symposium on EMC, the Founding General Chair of the Asia-Pacific EMC Symposium, and the General Chair of 2008, 2010, 2012, and 2016 APEMC and 2010 IEEE Symposium on Electrical Design for Advanced Packaging Systems. He has been invited to give numerous invited talks and plenary speeches at various international conferences and forums. He served as an Associate Editor for the IEEE MICROWAVE AND WIRELESS COMPONENTS LETTERS from 2006 to 2008 and a Guest Editor for the 2006 and 2010 IEEE TRANSACTIONS ON ELECTROMAGNETIC COMPATIBILITY Special Issues and 2010 IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES APMC Special Issue. He is also an Associate Editor of the IEEE TRANSACTIONS ON ELECTROMAGNETIC COMPATIBILITY and IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY.

Xu Chen (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 2005, 2014, and 2018, respectively.

From 2005 to 2012, he was a Hardware Engineer with the IBM Systems Group, Poughkeepsie, NY, USA. In 2013 and 2015, he worked at Apple, Cupertino, CA, USA. Since 2018, he has been a Teaching Assistant Professor at the Department of Electrical and Computer Engineering, University of Illinois. His primary research interests are in stochastic modeling, uncertainty quantification, and machine learning techniques, especially as they apply to electromagnetics and circuit problems. He is currently a Faculty Member with the Center for Advanced Electronics through Machine Learning (CAEML).

Dr. Chen was a recipient of the Best Conference Paper award at the IEEE Electrical Design of Advanced Packaging and Systems in 2017, the Raj Mittra Outstanding Research Award, the Mavis Future Faculty Fellowship, the USNC-URSI Fellowship Grant Award, and so on.

Andreas C. Cangellaris (Life Fellow, IEEE) received the Diploma degree in electrical engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Berkeley, Berkeley, CA, USA, in 1983 and 1985, respectively.

After a two-year tenure with the Electronics Department, General Motors Research Laboratories, Detroit, MI, USA, he joined the Electrical and Computer Engineering (ECE) Department, The University of Arizona, Tucson, AZ, USA, initially as an Assistant Professor from 1987 to 1992 and then as an Associate Professor from 1992 to 1997. In 1997, he joined the Electrical and Computer Engineering Department, University of Illinois at Urbana–Champaign, Urbana, IL, USA, as a Full Professor in 1997, where he is currently the Vice Chancellor for Academic Affairs and Provost and M. E. Van Valkenburg Professor. He has authored or coauthored over 250 articles in peer-reviewed journals and conference proceedings on the topics of finite methods for electromagnetic field modeling, multiconductor transmission line modeling, and signal and PI of integrated electronic systems. His expertise and research interests are in applied and computational electromagnetics and its applications to the advancement of modeling methodologies and computer-aided design (CAD) tools in support of electrical performance analysis and noise-aware design of integrated electronic systems. His research group has conducted pioneering work in the advancement of modeling methodologies and computer tools for predictive electromagnetic response performance of the signal and the power distribution networks of integrated electronics at the board, package, and chip levels. Several of the prototype tools developed by his group have been transferred to the microelectronics industry and have contributed to the development of several EDA SI tools.

Dr. Cangellaris is an active member of the IEEE Microwave Theory and Techniques Society (IEEE MTT-S) and the IEEE Components Packaging and Manufacturing Technology Society, contributing to numerous professional activities, conferences, and symposia organized and sponsored by them. He was a recipient of the 2000 University of Illinois, ECE Department Outstanding Faculty Teaching Award and the 2005 Alexander von Humboldt Research Award for outstanding contributions to electromagnetic theory. He was the Co-Founder of the IEEE Topical Meeting on Electrical Performance of Electronic Packaging in 1991. He is also an Editor of the IEEE PRESS SERIES ON ELECTROMAGNETIC WAVE THEORY. From 2010 to 2012, he was a Distinguished Lecturer of the IEEE MTT-S.

José Schutt-Ainé (Fellow, IEEE) received the B.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1981, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign (UIUC), Urbana, IL, USA, in 1984 and 1988, respectively.

He joined the Hewlett-Packard Technology Center, Santa Rosa, CA, USA, as an Application Engineer, where he was involved in research on microwave transistors and high-frequency circuits. In 1983, he joined UIUC and then joined the Electrical and Computer Engineering Department as a member of the Electromagnetics and Coordinated Science Laboratories, where he is currently involved in research on SI for high-speed digital and high-frequency applications. He is a consultant for several corporations. His current research interests include the study of SI and the generation of computer-aided design tools for high-speed digital systems.

Dr. Schutt-Ainé was a recipient of several research awards, including the 1991 National Science Foundation (NSF) MRI Award, the NASA Faculty Award for Research in 1992, the NSF MCAA Award in 1996, and the UIUC-National Center for Superconducting Applications Faculty Fellow Award in 2000. He served as Coeditor-in-Chief for the IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY (TCPMT) from 2007 to 2018.