IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

IEEE ELECTRONICS PACKAGING SOCIETY SECTION

Received 19 June 2026, accepted 3 July 2026, date of publication 6 July 2026, date of current version 17 July 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3710627

RESEARCH ARTICLE

Small-Data Modeling of Electromagnetic Structures via Compressed Vector-Valued Kernel Ridge Regression

NAZANIN SOLEIMANI, (Graduate Student Member, IEEE), IGOR S. STIEVANO, (Senior Member, IEEE), AND RICCARDO TRINCHERO, (Member, IEEE)
EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy

Corresponding author: Riccardo Trinchero (riccardo.trinchero@polito.it)

ABSTRACT This paper presents a data-efficient surrogate modeling framework for electromagnetic structures and components based on a compressed formulation of vector-valued kernel ridge regression (VV-KRR). The method is intended for small-data regimes, where training samples generated through full-wave or circuit simulations are computationally expensive and therefore limited in number. By combining a compressed spectral formulation with a fully data-driven output-kernel construction, the proposed approach enables accurate multi-output modeling with reduced memory footprint, lower training cost, and improved computational scalability. The framework retains the ability of VV-KRR to capture cross-output correlations and to model high-dimensional responses, while featuring a simple training procedure and good numerical robustness. It is particularly attractive for simulation-driven design workflows in high-speed interconnect, RF, and microwave applications, where data generation is often the dominant computational cost. The method is validated on three representative electromagnetic modeling problems, demonstrating that accurate and robust surrogate models can be obtained even in data-constrained scenarios.

INDEX TERMS Data-efficient learning, small-data modeling, vector-valued kernel ridge regression, RF systems, high-speed interconnects, surrogate modeling, parametric modeling.

I. INTRODUCTION

Machine learning (ML) and data-driven techniques have become increasingly important in the parametric modeling of electromagnetic structures arising in electronic packaging, high-speed interconnect, RF, and microwave applications. Approaches such as neural networks [1], [2], [3], [4], [5], [6], [7], [8], [9], inverse modeling [10], [11], [12], transfer learning [13], [14], knowledge-based methods [15], [16], [17], [18], [19], kernel-based regression [20], [21], [22], [23], [24], [25], [26], Bayesian learning [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], and parametric macromodeling [33], [34] have shown remarkable effectiveness in building accurate and computationally efficient surrogate models. Once trained, these models provide an explicit approximation of the underlying input-output mapping and can be evaluated extremely quickly, making them attractive for applications such as optimization and uncertainty quantification, where repeated evaluations of the original computational model would otherwise be prohibitively expensive [35].

In this application domain, however, the practical use of surrogate modeling is constrained by a fundamental bottleneck: the generation of training data. Unlike fields such as image recognition, where large datasets are often readily available, each training sample in electromagnetic and circuit applications typically requires a costly full-wave or circuit-level simulation. As a consequence, the amount of available data is often severely limited, and the design of data-efficient learning strategies becomes not only advantageous but necessary [36], [37], [38], [39]. In many practical cases,

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiguo Qian.

training data generation remains the dominant cost in terms of both computational time and resources.

These considerations highlight the need for surrogate modeling techniques that are simultaneously accurate, data-efficient, and computationally lightweight. Ideally, such methods should achieve high predictive accuracy even in small-data regimes,[1] while keeping both training and deployment affordable on standard computing platforms [38].

Kernel-based methods are strong candidates in this respect, since they offer an effective balance between modeling flexibility, robustness, and training efficiency [23], [44], [45]. Compared with deep neural networks (DNNs), including more specialized architectures such as convolutional neural networks (CNNs), kernel methods usually lead to convex coefficient-estimation problems for fixed kernel hyperparameters, since the model is linear with respect to the unknown coefficients. This avoids issues related to local minima and typically provides more stable coefficient estimation, which is particularly attractive in small-data regimes. By contrast, neural-network-based surrogates may achieve high accuracy when suitable architectures are adopted, but their performance can depend strongly on architecture design, regularization, and hyperparameter optimization, especially when only limited training data are available [16], [39], [46]. However, this advantage comes at the price of a less flexible model structure and of reduced scalability when plain kernel methods are applied to very large training datasets. Moreover, while neural networks can naturally handle multiple outputs through a multi-output layer, the use of conventional scalar kernel methods generally requires either the training of separate models for each output component or a dedicated vector-valued formulation. Training separate scalar models increases computational and storage costs, limits scalability, and neglects the correlations that may exist across the output dimensions [46], [47].

Dimensionality reduction techniques such as principal component analysis (PCA) [48] have been proposed to alleviate these issues in electronic and electromagnetic applications [24], [49], [50], [51], [52], [53]. While effective in many cases, they may introduce an additional source of approximation error, since the regression is carried out in a reduced space and the final response must then be reconstructed in the original output space after an unsupervised projection. Furthermore, because PCA is intrinsically unsupervised, the reduced basis is identified without directly accounting for the final regression accuracy. As a result, the selection of the number of retained components is less naturally connected to the supervised learning objective, making cross-validation and hyperparameter tuning less straightforward. In addition, the resulting framework still relies on the training and tuning of multiple scalar surrogate models, which may lead to a large number of hyperparameters and does not naturally allow the use of a single common regularization parameter, thus complicating automation and reducing scalability [46], [47], [54].

Vector-valued kernel ridge regression (VV-KRR) provides a natural alternative for handling multi-output problems [55], [56], [57], [58]. It retains the main advantages of scalar kernel methods, including a convex training formulation, a linear predictor structure, and a rigorous theoretical foundation based on the vector-valued extension of the representer theorem [57]. At the same time, VV-KRR can account for correlations among the output components in a structurally consistent way. Its practical application, however, is hindered by the complexity of the associated training problem, whose size grows with both the number of training samples and the output dimensionality, leading to large memory requirements and high computational cost [46], [47], [55]. In addition, the design of a suitable output kernel remains a nontrivial task, and standard kernel choices may not always be well matched to the structure of the problem [26], [27], [28].

To address these limitations, this paper proposes a compressed and data-efficient formulation of VV-KRR specifically tailored to small-data settings and computationally constrained environments. The main contributions of this work are as follows:

1) a fully data-driven procedure for constructing the output kernel, which simplifies the training process, reduces manual tuning, and provides a structurally coherent multi-output alternative to PCA combined with independent scalar regressions;
2) a compressed training scheme that significantly reduces both the computational cost and the memory footprint of VV-KRR through a reduced spectral formulation, thereby enabling scalable training even in the presence of high-dimensional outputs.

The proposed framework is assessed against several baselines, including PCA combined with scalar regression models and neural-network-based surrogates. In particular, a structured CNN baseline is included to account for neural architectures that exploit the frequency-domain organization of the output response.

The remainder of this paper is organized as follows. Section II introduces the mathematical background of VV-KRR and highlights its main training challenges. Section III presents the proposed compressed spectral training scheme. Section IV discusses the kernel choices considered in this work, including both analytic kernels and a data-driven output-kernel construction. Section V discusses data availability, output-kernel selection, and computational aspects related to hyperparameter tuning. Section VI assesses the effectiveness of the proposed approach on three application examples characterized by different numbers of input parameters and training-set sizes. Finally, conclusions are drawn in Section VII.

[1] Here, small-data is meant in a relative, problem-dependent sense; a quantitative definition is given in Sec. V-A.

II. BACKGROUND: VECTOR-VALUED KERNEL RIDGE REGRESSION

A. MATHEMATICAL FORMULATION

Consider the problem of learning a vector-valued function from a training set $\mathcal{S} = \{(\boldsymbol{x}_l, \boldsymbol{y}_l)\}_{l=1}^{L}$, where $\boldsymbol{x}_l \in \mathcal{X} \subset \mathbb{R}^p$ denotes the input feature vector associated with the l-th sample, and $\boldsymbol{y}_l = [y_l^{(1)}, \ldots, y_l^{(D)}]^T \in \mathcal{Y} \subset \mathbb{R}^D$ collects the corresponding D output components. Equivalently, one may view the problem as learning D coupled scalar functions

$$\hat{f}^{(d)} : \mathcal{X} \to \mathbb{R}, \tag{1}$$

for $d = 1, \ldots, D$.

In the vector-valued setting, the training problem can be formulated as the regularized least-squares problem

$$\hat{\boldsymbol{f}} = \arg\min_{\tilde{\boldsymbol{f}} \in \mathcal{H}} \sum_{d=1}^{D} \sum_{l=1}^{L} \left(y_l^{(d)} - \tilde{f}^{(d)}(\boldsymbol{x}_l) \right)^2 + \lambda \|\tilde{\boldsymbol{f}}\|_{\mathcal{H}}^2, \tag{2}$$

where $\mathcal{H}$ is a vector-valued reproducing kernel Hilbert space and $\lambda > 0$ is the regularization parameter.

According to the representer theorem for VV-KRR [57], the solution of (2) admits the finite-dimensional representation

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{l=1}^{L} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)\boldsymbol{c}_l, \tag{3}$$

where $\mathbf{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{D\times D}$ is a matrix-valued kernel and $\boldsymbol{c}_l \in \mathbb{R}^D$ are vectors collecting the regression coefficients.

For a generic output component d, (3) can be written as

$$\begin{aligned}\hat{f}^{(d)}(\boldsymbol{x}) &= \sum_{l=1}^{L} [\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)]_{[d,:]}\boldsymbol{c}_l \\ &= \sum_{d'=1}^{D} \sum_{l=1}^{L} [\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)]_{[d,d']} c_{d',l},\end{aligned} \tag{4}$$

which is equivalently expressed through the scalar kernel

$$\hat{f}^{(d)}(\boldsymbol{x}) = \sum_{d'=1}^{D} \sum_{l=1}^{L} k((\boldsymbol{x}, d), (\boldsymbol{x}_l, d'))c_{d',l}, \tag{5}$$

where $k((\boldsymbol{x}, d), (\boldsymbol{x}_l, d')) = [\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)]_{[d,d']}$ accounts simultaneously for similarities in the input space and across the output components [56].

A particularly convenient and widely used construction is the separable kernel [55], [57], defined as

$$k((\boldsymbol{x}, d), (\boldsymbol{x}_l, d')) = k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}_l)k_o(d, d'), \tag{6}$$

where $k_{\boldsymbol{x}}$ models similarity in the input space and k_o models similarity across the output components.

Under this assumption, the matrix-valued kernel in (3) can be written as

$$\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l) = k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}_l)\mathbf{B}, \tag{7}$$

where $\mathbf{B} \in \mathbb{R}^{D\times D}$ is the Gram matrix induced by the output kernel, namely

$$[\mathbf{B}]_{[d,d']} = k_o(d, d'), \tag{8}$$

for $d, d' \in \{1, \ldots, D\}$.

Substituting (7) into (3) yields the separable VV-KRR model

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{l=1}^{L} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}_l)\boldsymbol{c}_l = \sum_{l=1}^{L} k_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}_l)\mathbf{B}\boldsymbol{c}_l, \tag{9}$$

where the same input kernel $k_{\boldsymbol{x}}$ and output Gram matrix $\mathbf{B}$ are shared by all samples and output components.

B. MATRIX FORMULATION AND CONVENTIONAL TRAINING ALGORITHMS

For compactness, the training set can be represented in matrix form as $\mathcal{S} = \{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L]^T \in \mathbb{R}^{L\times p}$ collects the input samples and $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]^T \in \mathbb{R}^{L\times D}$ collects the corresponding outputs.

Using this notation, the VV-KRR learning problem can be written as

$$\min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{F}(\mathbf{X}; \mathbf{C})\|_F^2 + \lambda \|\boldsymbol{f}\|_{\mathcal{H}}^2, \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathbf{C} \in \mathbb{R}^{L\times D}$ collects the unknown regression coefficients, and $\mathbf{F}(\mathbf{X}; \mathbf{C}) \in \mathbb{R}^{L\times D}$ contains the model predictions evaluated at the training inputs, namely

$$\mathbf{F}(\mathbf{X}; \mathbf{C}) = [\hat{\boldsymbol{f}}(\boldsymbol{x}_1), \ldots, \hat{\boldsymbol{f}}(\boldsymbol{x}_L)]^T. \tag{11}$$

For separable matrix-valued kernels, the RKHS norm of the model can be written as

$$\|\boldsymbol{f}\|_{\mathcal{H}}^2 = \langle \mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}, \mathbf{B}\rangle_F. \tag{12}$$

By evaluating the separable model in (9) at each training point, one obtains

$$[\mathbf{F}(\mathbf{X}; \mathbf{C})]_{[j,:]} = \sum_{l=1}^{L} k_{\boldsymbol{x}}(\boldsymbol{x}_j, \boldsymbol{x}_l)\boldsymbol{c}_l^T\mathbf{B}, \tag{13}$$

where the symmetry of the real-valued output Gram matrix, $\mathbf{B} = \mathbf{B}^T$, has been used.

Collecting all rows together gives the compact matrix form

$$\mathbf{F}(\mathbf{X}; \mathbf{C}) = k_{\boldsymbol{x}}(\mathbf{X}, \mathbf{X})\mathbf{C}\mathbf{B} = \mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}, \tag{14}$$

where $\mathbf{K}_{\boldsymbol{x}} \in \mathbb{R}^{L\times L}$ is the input Gram matrix with entries $[\mathbf{K}_{\boldsymbol{x}}]_{[i,j]} = k_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Substituting (14) into (10), the training problem becomes

$$\min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B}\|_F^2 + \lambda\langle \mathbf{C}^T\mathbf{K}_{\boldsymbol{x}}\mathbf{C}, \mathbf{B}\rangle_F, \tag{15}$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product.

The corresponding first-order optimality condition leads to the discrete Sylvester equation [46], [55]

$$\mathbf{K}_{\boldsymbol{x}}\mathbf{C}\mathbf{B} + \lambda\mathbf{C} = \mathbf{Y}, \tag{16}$$

which constitutes the core training problem for separable VV-KRR.

A direct way to solve (16) is to vectorize the unknown matrix $\mathbf{C}$ and exploit the Kronecker-product identity, yielding

$$\mathrm{vec}(\mathbf{C}) = (\mathbf{B} \otimes \mathbf{K}_{\boldsymbol{x}} + \lambda \mathbf{I}_{LD})^{-1} \mathrm{vec}(\mathbf{Y}), \tag{17}$$

where $\otimes$ denotes the Kronecker product.

Although exact, this formulation becomes computationally demanding when either the number of training samples L or the output dimension D is large. Indeed, the plain solution of the fully coupled system in (17) has complexity $\mathcal{O}(L^3D^3)$ [46], [55], which may severely limit scalability and hinder the practical use of richer kernel models and extensive hyperparameter tuning.

The computational burden of (17) can be alleviated to some extent by using iterative solvers, such as gradient-based schemes of the type considered in [46], whose complexity scales approximately as $(LD)^2$ per iteration. However, such methods may still suffer from convergence and accuracy issues, especially for large-scale problems. These limitations motivate the search for more efficient training schemes, which is the topic of the next section.

III. COMPRESSED DIAGONALIZATION-BASED TRAINING SCHEME

This section introduces the proposed training strategy for the efficient solution of the discrete Sylvester equation (16). The method is based on a compressed spectral representation of both the input and output kernel matrices, and can be regarded as an extension of the plain diagonalization strategy presented in [60]. In contrast to that approach, the proposed formulation explicitly supports simultaneous compression of both the input and output spaces.

The starting point is a truncated eigendecomposition of the two positive semidefinite Gram matrices $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$:

$$\mathbf{K}_{\boldsymbol{x}} \approx \mathbf{U}_r \mathbf{N}_r \mathbf{U}_r^T \quad \text{and} \quad \mathbf{B} \approx \mathbf{T}_n \mathbf{M}_n \mathbf{T}_n^T, \tag{18}$$

where $\mathbf{U}_r \in \mathbb{R}^{L\times r}$ and $\mathbf{T}_n \in \mathbb{R}^{D\times n}$ collect the leading r and n eigenvectors of $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$, respectively, namely

$$\mathbf{U}_r = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r] \quad \text{and} \quad \mathbf{T}_n = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n]. \tag{19}$$

The associated diagonal matrices of retained eigenvalues are

$$\mathbf{N}_r = \mathrm{diag}(n_1, \ldots, n_r), \; \mathbf{M}_n = \mathrm{diag}(m_1, \ldots, m_n), \tag{20}$$

with $n_1 \geq n_2 \geq \cdots \geq n_r \geq 0$ and $m_1 \geq m_2 \geq \cdots \geq m_n \geq 0$. Since $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$ are symmetric positive semidefinite Gram matrices, their eigenvalues are real and nonnegative by construction [56]. Consequently, their singular values coincide with their eigenvalues, i.e., $\sigma_i(\mathbf{K}_{\boldsymbol{x}}) = \lambda_i(\mathbf{K}_{\boldsymbol{x}})$ and $\sigma_i(\mathbf{B}) = \lambda_i(\mathbf{B})$, so that the truncated eigendecompositions in (18) are equivalent to truncated singular value decompositions.

The truncation orders r and n are selected according to the cumulative normalized spectral energy. Assuming that $\{n_i\}_{i=1}^L$ and $\{m_i\}_{i=1}^D$ denote the complete eigenvalue spectra of $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$, respectively, sorted in descending order, r and n are chosen as the smallest integers such that

$$\frac{\sum_{i=1}^r n_i}{\sum_{i=1}^L n_i} \geq 1 - \mathtt{tol}, \qquad \frac{\sum_{i=1}^n m_i}{\sum_{i=1}^D m_i} \geq 1 - \mathtt{tol}, \tag{21}$$

where $\mathtt{tol} \in [0, 1)$ is a user-defined tolerance controlling the discarded spectral energy. In this way, the retained eigenspaces preserve at least a fraction $1 - \mathtt{tol}$ of the total spectral energy, while the discarded part is limited to at most $\mathtt{tol}$. The truncation rule therefore determines automatically the minimum number of retained modes needed to meet the prescribed accuracy level, without requiring the ranks to be fixed a priori. In the numerical examples, $\mathtt{tol}$ is chosen sufficiently small, e.g., $\mathtt{tol} = 10^{-6}$, corresponding to an almost complete retention of the spectral energy. When needed, different tolerances can be assigned to $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$.

The reduced eigenspaces defined above are now used to derive a compressed form of the original Sylvester equation. By substituting the low-rank approximations of $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$ into (16), one obtains

$$(\mathbf{U}_r \mathbf{N}_r \mathbf{U}_r^T) \mathbf{C} (\mathbf{T}_n \mathbf{M}_n \mathbf{T}_n^T) + \lambda \mathbf{C} = \mathbf{Y}. \tag{22}$$

The coefficient matrix $\mathbf{C}$ is then approximated in the reduced spectral bases as

$$\mathbf{C} \approx \mathbf{U}_r \tilde{\mathbf{C}}_{r\times n} \mathbf{T}_n^T, \tag{23}$$

where $\tilde{\mathbf{C}}_{r\times n} \in \mathbb{R}^{r\times n}$ is the reduced coefficient matrix.

Substituting (23) into (22) yields

$$(\mathbf{U}_r \mathbf{N}_r \mathbf{U}_r^T)(\mathbf{U}_r \tilde{\mathbf{C}}_{r\times n} \mathbf{T}_n^T)(\mathbf{T}_n \mathbf{M}_n \mathbf{T}_n^T) + \lambda \mathbf{U}_r \tilde{\mathbf{C}}_{r\times n} \mathbf{T}_n^T = \mathbf{Y}. \tag{24}$$

By left-multiplying by $\mathbf{U}_r^T$ and right-multiplying by $\mathbf{T}_n$, and using the orthonormality relations $\mathbf{U}_r^T \mathbf{U}_r = \mathbf{I}_r$ and $\mathbf{T}_n^T \mathbf{T}_n = \mathbf{I}_n$, one obtains

$$\mathbf{N}_r \tilde{\mathbf{C}}_{r\times n} \mathbf{M}_n + \lambda \tilde{\mathbf{C}}_{r\times n} = \mathbf{U}_r^T \mathbf{Y} \mathbf{T}_n. \tag{25}$$

Defining $\tilde{\mathbf{Y}}_{r\times n} = \mathbf{U}_r^T \mathbf{Y} \mathbf{T}_n$, the reduced problem can be written as

$$\mathbf{N}_r \tilde{\mathbf{C}}_{r\times n} \mathbf{M}_n + \lambda \tilde{\mathbf{C}}_{r\times n} = \tilde{\mathbf{Y}}_{r\times n}. \tag{26}$$

The diagonal structures of $\mathbf{N}_r$ and $\mathbf{M}_n$ allow the above matrix equation to decouple entrywise. Indeed,

$$[\mathbf{N}_r \tilde{\mathbf{C}}_{r\times n} \mathbf{M}_n]_{[i,j]} = n_i \tilde{c}_{ij} m_j, \tag{27}$$

for $i = 1, \ldots, r$ and $j = 1, \ldots, n$. Hence, (26) reduces to the scalar equations

$$n_i \tilde{c}_{ij} m_j + \lambda \tilde{c}_{ij} = \tilde{y}_{ij}, \tag{28}$$

whose explicit solution is

$$\tilde{c}_{ij} = \frac{\tilde{y}_{ij}}{n_i m_j + \lambda}, \tag{29}$$

for $i = 1, \ldots, r$ and $j = 1, \ldots, n$.

This coefficient-estimation step is numerically robust, since the regularization term λ prevents instabilities associated with very small eigenvalues. In particular, when $n_i m_j \approx 0$, the corresponding coefficient reduces to

$$\tilde{c}_{ij} \approx \frac{\tilde{y}_{ij}}{\lambda}, \tag{30}$$

thus preventing excessively large coefficient values and stabilizing the overall solution.

After computing the reduced matrix $\tilde{\mathbf{C}}_{r\times n}$, an approximation of the coefficient matrix $\mathbf{C}$ is obtained through (23). The compressed representation can then be used directly in the VV-KRR predictor to evaluate the model at a generic test point $\boldsymbol{x}^* \in \mathcal{X}$:

$$\begin{aligned}\hat{\boldsymbol{f}}(\boldsymbol{x}^*)^T &= \sum_{l=1}^{L} k_{\boldsymbol{x}}(\boldsymbol{x}^*, \boldsymbol{x}_l)\, \boldsymbol{c}_l^T \mathbf{B} = \mathbf{K}_{\boldsymbol{x}}(\boldsymbol{x}^*, \mathbf{X})\mathbf{C}\mathbf{B} \\ &\approx \mathbf{K}_{\boldsymbol{x}}(\boldsymbol{x}^*, \mathbf{X})\mathbf{U}_r \tilde{\mathbf{C}}_{r\times n}\mathbf{T}_n^T \mathbf{B},\end{aligned} \tag{31}$$

where $\mathbf{K}_{\boldsymbol{x}}(\boldsymbol{x}^*, \mathbf{X})$ is the $1 \times L$ row vector collecting the kernel evaluations between the test point and the training samples.

Equation (31) can be rewritten in the compact form

$$\hat{\boldsymbol{f}}(\boldsymbol{x}^*)^T = \tilde{\mathbf{K}}_{\boldsymbol{x}}(\boldsymbol{x}^*, \mathbf{X})\tilde{\mathbf{C}}_{r\times n}\tilde{\mathbf{B}}, \tag{32}$$

where

$$\tilde{\mathbf{K}}_{\boldsymbol{x}}(\boldsymbol{x}^*, \mathbf{X}) = \mathbf{K}_{\boldsymbol{x}}(\boldsymbol{x}^*, \mathbf{X})\mathbf{U}_r, \tag{33}$$

$$\tilde{\mathbf{B}} = \mathbf{T}_n^T \mathbf{B}, \tag{34}$$

are the projected input and output kernel factors associated with the reduced spectral representation. This expression highlights that the prediction stage can also be carried out in the compressed spaces defined by the truncated eigendecompositions, thereby reducing the computational complexity of model evaluation.

The above derivation also highlights several structural properties of the proposed formulation. Unlike the plain diagonalization-based approach reported in [60], the present scheme naturally enables model compression, since it does not require the transformation matrices $\mathbf{U}$ and $\mathbf{T}$ to be full-rank in order to construct the coefficient matrix from its reduced counterpart $\tilde{\mathbf{C}}_{r\times n}$. In fact, the training scheme in [60] is not inherently compressive, since exact reconstruction from the transformed coefficient matrix would require $\mathbf{U}_r\mathbf{U}_r^T = \mathbf{I}_L$ and $\mathbf{T}_n\mathbf{T}_n^T = \mathbf{I}_D$, which only hold in the full-rank case, namely when $r = L$ and $n = D$.

The proposed compressed formulation therefore discards redundant spectral components while preserving the dominant input-output couplings. In particular, components associated with vanishing eigenvalues need not be explicitly computed, since their contribution is effectively controlled by the regularization term λ. This property is consistent with the entrywise solution in (26), where small values of the product $n_i m_j$ do not give rise to numerical instabilities, but instead lead to coefficients that remain bounded by the regularization term.

From a computational viewpoint, the overall cost of the proposed compressed training scheme scales as

$$\mathcal{O}(L^3 + D^3 + r^2 n + r n^2), \tag{35}$$

where the cubic terms are associated with the eigendecompositions of $\mathbf{K}_{\boldsymbol{x}}$ and $\mathbf{B}$, while the remaining terms arise from the solution of the reduced problem.

IV. KERNEL DESIGN FOR VECTOR-VALUED REGRESSION

This section discusses two families of analytic kernels, namely the squared exponential (SE) and Matérn kernels, which can be used for both $k_{\boldsymbol{x}}$ and k_o. It then introduces a data-driven construction of the output Gram matrix $\mathbf{B}$ in (9), based on the empirical correlation of the training outputs.

TABLE 1. Common Matérn Kernels.

ν	Kernel Name	$k_\nu(\xi, \xi')$
$\frac{1}{2}$	Exponential kernel/RBF	$\sigma_f^2 \exp(-r)$
$\frac{3}{2}$	Matérn 3/2	$\sigma_f^2 \left(1 + \sqrt{3}r\right) \exp(-\sqrt{3}r)$
$\frac{5}{2}$	Matérn 5/2	$\sigma_f^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp(-\sqrt{5}r)$

A. ANALYTIC KERNELS

Kernels commonly used in Gaussian-process regression [63], such as the SE and Matérn families, can be directly adopted in the present VV-KRR framework for both the input and output kernels. Their definition is conveniently expressed in terms of a scaled distance.

Let

$$r = \sqrt{(\xi - \xi')^T \mathbf{L}^{-1} (\xi - \xi')}, \tag{36}$$

where ξ and ξ' denote either the input vectors $(\boldsymbol{x}, \boldsymbol{x}')$ when evaluating the input kernel $k_{\boldsymbol{x}}$, or the output indices (d, d') when evaluating the output kernel k_o.

The matrix $\mathbf{L}$ encodes the kernel length-scale hyperparameters. In the isotropic case, $\mathbf{L} = \ell^2\mathbf{I}$, so that a single length scale ℓ is shared by all dimensions. In the anisotropic case, hereafter used only for the input kernel $k_{\boldsymbol{x}}$, $\mathbf{L} = \mathrm{diag}(\ell_1^2, \ldots, \ell_p^2)$, meaning that each input dimension is assigned an independent length scale. The latter case corresponds to the well-known Automatic Relevance Determination (ARD) formulation.

1) SQUARED EXPONENTIAL / RBF KERNEL

The squared exponential (SE) kernel, also referred to as the radial basis function (RBF) or Gaussian kernel, is one of the most widely used stationary kernels. In terms of the scaled distance r in (36), it is defined as

$$k_{\mathrm{SE/RBF}}(\xi, \xi') = \sigma_f^2 \exp\left(-\frac{1}{2}r^2\right), \tag{37}$$

where σ_f is the kernel scale. The same expression applies to both isotropic and ARD versions, depending on the choice of $\mathbf{L}$.

2) MATÉRN KERNEL

The Matérn family provides a broader class of kernels than the RBF kernel by introducing an explicit smoothness parameter ν, which controls the regularity of the target function. For specific values of ν, closed-form expressions are available and can be used efficiently in practice. Table 1 reports the most common choices.

Larger values of ν correspond to smoother functions. In the limit $\nu \to \infty$, the Matérn kernel converges to the RBF kernel.

B. CORRELATION-BASED OUTPUT KERNEL

As an alternative to the analytic kernels introduced above, and inspired by PCA-based approaches [24], [48], [53], the output Gram matrix $\mathbf{B}$ in (9) can be estimated directly from the training data by computing the correlation across the output dimensions:

$$\mathbf{B} = \mathrm{corr}(\mathbf{Y}), \tag{38}$$

where $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]^T \in \mathbb{R}^{L\times D}$ collects the training outputs. In this interpretation, $\mathbf{Y}$ is viewed as a set of L realizations of D random variables, with each column corresponding to one output component and each row to one realization.

The entries of the correlation matrix in (38) are given by

$$[\mathbf{B}]_{[i,j]} = \frac{\mathbb{E}_l\left[\left(y_l^{(i)} - \mu_y^{(i)}\right)\left(y_l^{(j)} - \mu_y^{(j)}\right)\right]}{\sigma_y^{(i)}\sigma_y^{(j)}}, \tag{39}$$

where $y_l^{(i)}$ denotes the i-th output component of the l-th realization, while $\mu_y^{(i)}$ and $\sigma_y^{(i)}$ are the sample mean and sample standard deviation of the i-th output component, respectively. The operator $\mathbb{E}_l[\cdot]$ denotes averaging over the L available samples.

If the output matrix is standardized columnwise, i.e., if one considers a matrix $\bar{\mathbf{Y}}$ whose columns have zero mean and unit standard deviation, then the above expression simplifies to

$$[\mathbf{B}]_{[i,j]} = \mathbb{E}_l\left[\bar{y}_l^{(i)}\bar{y}_l^{(j)}\right] = \frac{1}{L}\sum_{l=1}^{L}\bar{y}_l^{(i)}\bar{y}_l^{(j)}, \tag{40}$$

which leads directly to the matrix identity

$$\mathbf{B} = \frac{1}{L}\bar{\mathbf{Y}}^T\bar{\mathbf{Y}}. \tag{41}$$

This construction allows the model to infer the correlation structure among the output components directly from the training data, without introducing additional assumptions or kernel hyperparameters, since the resulting output kernel is hyperparameter free.

V. DATA AVAILABILITY, OUTPUT-KERNEL SELECTION, AND COMPUTATIONAL COST

This section collects methodological and practical aspects related to the proposed framework. First, it introduces a data-availability index for quantifying the small-data regime in vector-valued surrogate modeling problems. Then, it discusses output-kernel selection and regularization in data-limited settings, and analyzes the impact of hyperparameter tuning on the computational cost of the proposed VV-KRR formulation.

A. DATA-AVAILABILITY INDEX

The notion of small-data modeling is problem dependent and cannot be defined only in terms of an absolute number of samples. In the small-data machine-learning literature, data scarcity is typically related to the cost and difficulty of data acquisition, the complexity of the learning task, and the reliability of generalization [40], [41], [42]. A simple ratio such as L/p, where L is the number of training samples and p is the number of input parameters, provides only a rough indication of input-space sampling density. However, it is not a general definition of a small-data regime. For instance, in a surrogate-modeling problem with one or two input variables, the condition $L < p$ would imply that zero or one sample corresponds to a small-data regime, which is not meaningful for constructing a reliable surrogate model. Conversely, even when $L > p$, the problem may still be data-limited if the response is highly nonlinear, resonant, multi-scale, or if the required accuracy is demanding.

In this work, the data-limited character of a problem is quantified at the level of the surrogate-modeling task, independently of the specific regression technique used to solve it. The goal is therefore not to define a complexity measure of the proposed VV-KRR model, but rather to introduce a problem-level indicator of how many high-fidelity samples are available with respect to the effective input-output dimensionality of the data.

For vector-valued surrogate models, the output dimension must also be considered, since multi-output regression is a specific learning setting in which dependencies among output variables play an important role [43]. A direct use of the raw output dimension D may be overly conservative for frequency-domain responses, since D depends on the chosen frequency resolution. Increasing the number of output samples does not necessarily increase the intrinsic number of independent output degrees of freedom. For instance, increasing the number of frequency points used to describe a frequency response does not necessarily increase the number of independent output components, because adjacent frequency samples are often strongly correlated. For this reason, an effective measure of the intrinsic output dimensionality is introduced through $d_{\mathrm{out,eff}}$. In this work, $d_{\mathrm{out,eff}}$ is defined as the number of dominant spectral components of the output correlation matrix required to retain a prescribed fraction of the output variance, in analogy with PCA-based output compression [48], [53]. This choice should be regarded as a practical indicator rather than as a unique or optimal definition, since the resulting value may depend on the adopted tolerance and may be affected by noise, especially for very small datasets.

If m_i are the eigenvalues of the output correlation matrix, sorted in descending order, like in (21), the effective output dimension is defined as

$$d_{\text{out,eff}}(\text{tol}) = \min\left\{ n : \frac{\sum_{i=1}^{n} m_i}{\sum_{i=1}^{D} m_i} \geq 1 - \text{tol} \right\}. \quad (42)$$

In the numerical examples, tol $= 10^{-4}$ is used, corresponding to 99.99% retained output variance.

The data-availability index is then defined as

$$\eta_{\text{SD}} = \frac{L}{p + d_{\text{out,eff}}}. \quad (43)$$

This index represents the number of available high-fidelity samples per effective input-output degree of freedom. Accordingly, the empirical condition

$$\eta_{\text{SD}} \lesssim 10 \quad (44)$$

is used as a practical indicator of a small-data regime. If $1 < \eta_{\text{SD}} \lesssim 10$, the problem is considered data-limited, whereas values $\eta_{\text{SD}} \leq 1$ indicate a strongly data-limited regime.

The index η_{SD} should be interpreted as a practical data-availability indicator rather than as a complete measure of the nonlinear sample complexity of the problem. Indeed, $d_{\text{out,eff}}$ quantifies output-space compressibility, but it does not directly measure the difficulty of learning the dependence of the response on the input parameters. Such difficulty may depend on nonlinearities, parameter interactions, resonant shifts, noise level, and target accuracy. Therefore, the proposed index is used only as a first-order quantitative diagnostic and is complemented by the predictive results reported in the application examples.

B. OUTPUT-KERNEL SELECTION IN DATA-LIMITED REGIMES

The choice of the output kernel is important in the proposed VV-KRR formulation, since it determines how correlations among output components are modeled. The correlation-based output kernel introduced in Sec. IV-B is attractive because it is fully data-driven and hyperparameter-free. For deterministic simulation-based datasets, where the outputs are not affected by measurement noise, the empirical correlation matrix can provide an effective estimate of the output dependencies.

However, in strongly data-limited regimes or in the presence of noisy observations, the empirical correlation matrix may include spurious output couplings, which can affect both prediction accuracy and spectral compression. Regularized alternatives may therefore be useful when the empirical correlation estimate is unreliable. For example, a shrinkage estimator $\mathbf{B}_\rho = (1 - \rho)\,\text{corr}(\mathbf{Y}) + \rho\mathbf{I}$, with $0 \leq \rho \leq 1$, shrinks the output kernel toward a diagonal matrix and reduces the influence of off-diagonal correlations. However, it also introduces an additional hyperparameter and may weaken the eigenvalue decay exploited by the proposed compression strategy.

In the compressed VV-KRR framework, shrinkage regularization also interacts with the spectral truncation step. Indeed, applying shrinkage before compression or to the already compressed approximation generally leads to different compressed models, since the shrinkage term modifies the eigenvalue distribution and the retained output subspace. The choice may also affect the effective output dimension used in the small-data index in (43). For these reasons, and since the benchmarks considered in this work are based on deterministic circuit or full-wave simulations rather than noisy measurements, no shrinkage regularization is adopted in the reported results. A systematic study of shrinkage-based output-kernel regularization is left for future work.

Analytic output kernels, such as RBF and Matérn kernels, provide a structured alternative to the empirical correlation matrix. They impose a smoother output-correlation model controlled by kernel hyperparameters and may provide additional regularization when the empirical correlation estimate is unreliable or affected by noise [47].

C. COMPUTATIONAL COST IN CROSS-VALIDATION

The computational cost reported in Sec. III refers to one direct training stage after the kernel hyperparameters have been selected. When hyperparameter tuning is included, the total cost depends on whether the output kernel is analytic or data-driven.

If an analytic kernel is used, such as an RBF or Matérn kernel, the output kernel depends on tunable hyperparameters. Therefore, its eigendecomposition may need to be recomputed during the hyperparameter optimization procedure. For N optimization steps and k-fold cross-validation, the total cost scales as

$$\mathcal{O}\Big(Nk\left(L^3 + D^3 + r^2 n + rn^2\right)\Big). \quad (45)$$

For the data-driven output kernel, the output kernel is hyperparameter-free. Therefore, its eigendecomposition is performed only once and is not repeated at each cross-validation fold or optimization step. In this case, the total cost can be written as

$$\mathcal{O}\Big(D^3 + Nk\left(L^3 + r^2 n + rn^2\right)\Big), \quad (46)$$

where the first term denotes the one-time output-kernel eigendecomposition. In this case, the repeated cost during hyperparameter tuning is dominated by the input-kernel operations, making the data-driven output kernel more attractive for high-dimensional output spaces.

For very large output dimensions, direct dense eigendecompositions should be avoided. More efficient truncated spectral decomposition techniques, such as randomized SVD [51], [64] or block iterative eigensolvers [49], [52], can be integrated into the proposed framework to compute only the dominant modes required by the compressed formulation. However, their numerical parameters and their impact on the final regression accuracy require a dedicated study and are left as future work.

VI. APPLICATION EXAMPLES

This section assesses the proposed framework on three representative examples and compares its performance with that of selected state-of-the-art alternatives. The considered benchmarks cover different levels of input dimensionality, output dimensionality, and data availability. They include the modeling of the magnitude of the frequency response of a high-speed link with 11 input parameters, the modeling of the real and imaginary parts of the scattering parameters of a low-noise amplifier with 25 input parameters, and the modeling of the mixed-mode scattering parameters of a common-mode suppression filter with 6 input parameters. All datasets, training scripts, and validation procedures are publicly available in an online repository.[2]

Before discussing the individual examples, the common model setup, hyperparameter-tuning strategy, and baseline methods are summarized below.

A. MODEL SETUP AND VALIDATION PROTOCOL

In all test cases, the proposed VV-KRR model was trained using 5-fold cross-validation with a truncation threshold `tol` $= 10^{-6}$. The hyperparameters were estimated through the MATLAB function `fminunc`, employing a quasi-Newton algorithm with a maximum of 300 iterations and 3000 function evaluations. Central finite differences were used for numerical gradient approximation, with a finite-difference step size equal to 10^{-2}. All hyperparameters were optimized in logarithmic scale. In particular, the initial guess was chosen by setting the logarithms of all kernel lengthscales, both for k_o and $k_{\boldsymbol{x}}$, equal to $\log_{10}(10)$, while the logarithm of the regularization parameter was initialized to $\log_{10}(10^{-6})$. All computations were performed in MATLAB on a MacBook Air equipped with an Apple M3 chip, featuring an 8-core CPU and 24 GB of unified memory.

For comparison purposes, the proposed VV-KRR framework is benchmarked against a PCA-based surrogate model combined with Gaussian-process regression (GPR) [63] in all examples. The PCA-based model uses a truncated principal-component expansion of the training outputs, followed by the independent regression of the retained latent coefficients by means of scalar kernel regressors implemented through the MATLAB GPR toolbox. In the reported results, the PCA truncation threshold is fixed to `tol` $= 10^{-3}$. This choice is consistent with the value `tol` $= 10^{-6}$ adopted in the proposed VV-KRR formulation, since PCA compression is based on the singular values of the output matrix, whereas the proposed compressed scheme truncates the eigenvalues of the corresponding covariance matrix, which are proportional to the squared singular values.

For Example 1, two neural-network baselines were implemented in MATLAB: a fully connected DNN for multi-output regression and a structured CNN inspired by frequency-domain convolutional surrogate modeling approaches [5]. The DNN consists of a z-score-normalized feature input layer, two fully connected hidden layers with ReLU or leaky-ReLU activations and dropout regularization, and a final regression layer. Its main architectural and training hyperparameters, including the number of hidden units, dropout rates, learning rate, mini-batch size, and activation functions, were selected by Bayesian optimization.

The CNN exploits the frequency-domain structure of the output response by repeating the input parameters along the frequency axis and augmenting them with a normalized frequency-coordinate channel. Thus, each sample is represented by a tensor of size $D \times 1 \times (p+1)$, where D is the number of frequency samples and p is the number of input parameters, and the network learns the map $(\mathbf{x}, f) \mapsto y(\mathbf{x}, f)$ through convolutional filters along the frequency direction. The CNN hyperparameters, including the number of filters, kernel size, learning rate, regularization parameter, and mini-batch size, were also selected by Bayesian optimization.

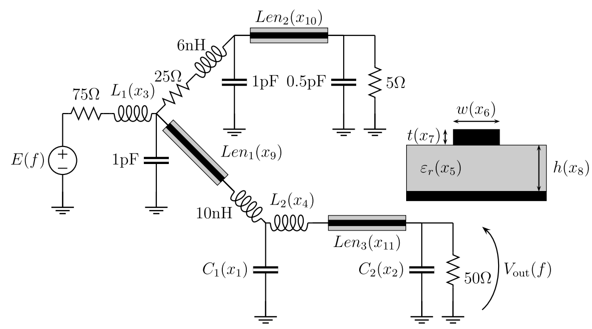

FIGURE 1. Schematic of the high-speed link considered in Sec. VI-B.

Both networks were trained using the Adam optimizer and a 20% hold-out validation set, with up to 1000 epochs for the DNN and 1500 epochs for the final CNN training. The reported training times include both Bayesian hyperparameter optimization and final retraining with the selected hyperparameters. Although the adopted neural-network baselines were extensively tuned through Bayesian optimization, they should be interpreted as indicative auxiliary comparisons rather than strict upper bounds on the accuracy achievable by neural-network-based surrogate models.

B. EXAMPLE 1: HIGH-SPEED LINK

As a first benchmark, we consider the high-speed-link example originally presented in [46] (see Sec. VI), involving $p = 11$ uniformly distributed input parameters. The interconnect structure is shown in Fig. 1, while the nominal values and variation ranges of the parameters are reported in Table 2. The goal is to predict the magnitude of the frequency response $|H(j\omega; \boldsymbol{x})|$, defined between the voltage source E and the output voltage $V_{\rm out}$, at $D = 100$ frequency points in the band from 1 MHz to 5 GHz. As in [46], the dataset consists of $L = 300$ training samples and 1000 test samples generated by latin hypercube sampling (LHS). For this application, the data-availability index defined in (43) is equal to $\eta_{\rm SD} = 7.1$, obtained with $L = 300$, $p = 11$,

[2]https://github.com/RiccardoTrinchero/small-data-compressed-vv-krr

TABLE 2. Nominal values and relative variation ranges of the parameters of the high-speed interconnect considered in Sec. VI-B.

Parameter	Mean Value	Uniform Variation
$C_1(x_1)$	1 pF	$\pm 20\%$
$C_2(x_2)$	0.5 pF	$\pm 20\%$
$L_1(x_3)$	10 nH	$\pm 20\%$
$L_2(x_4)$	10 nH	$\pm 20\%$
$\varepsilon_r(x_5)$	4.1	$\pm 1\%$
$w(x_6)$	252 μm	$\pm 1\%$
$t(x_7)$	35 μm	$\pm 1\%$
$h(x_8)$	60 μm	$\pm 1\%$
$Len_1(x_9)$	5 cm	$\pm 5\%$
$Len_2(x_{10})$	3 cm	$\pm 5\%$
$Len_3(x_{11})$	3 cm	$\pm 5\%$

and an effective output dimension $d_{\text{out,eff}} = 31$. This value satisfies the empirical condition in (44), confirming that the benchmark falls within the data-limited regime considered in this work.

Table 3 reports the complete comparison for the first benchmark example, including the reference methods from [46], all the VV-KRR configurations obtained with the proposed compressed training scheme, the PCA-based baselines, and the DNN and CNN baselines. Overall, the proposed VV-KRR models improve over the original implementations of [46] in terms of both ℓ_2 and ℓ_∞ test errors. This gain can be attributed to the direct solution of the reduced Sylvester equation, which avoids iterative approximations, together with the lower computational cost of the compressed formulation, which makes hyperparameter tuning more effective and enables the practical use of richer kernel configurations. Among all the tested models, the best overall performance is achieved by model #4, which combines a data-driven output kernel with an ARD Matérn 5/2 input kernel. This configuration attains one of the lowest ℓ_2 errors, equal to 0.5%, and the lowest ℓ_∞ error, equal to 5.1%. Comparable ℓ_2 values are also obtained by models #10 and #16, both based on analytic output kernels and ARD Matérn 5/2 input kernels. However, model #4 provides the best compromise between predictive accuracy and model compactness, since it retains 13,850 coefficients, whereas model #16 requires 28,900 coefficients. Model #10 is more compact, but it yields a slightly larger ℓ_∞ error.

More generally, the results confirm the importance of anisotropic input kernels in this benchmark. Configurations based on ARD Matérn 5/2 or ARD RBF kernels consistently provide the lowest errors, whereas isotropic input kernels lead to a substantial loss of accuracy. This behavior indicates that the 11 input parameters do not contribute equally to the frequency response and that the use of independent length scales improves the ability of the model to capture the parametric dependence.

The comparison with the PCA-based baselines further confirms the effectiveness of the proposed VV-KRR framework. The best PCA-based model, namely model #24, reaches an ℓ_2 error of 0.6% and an ℓ_∞ error of 6.3%, which remain slightly worse than those obtained by model #4. Moreover, the PCA-based approach relies on the training of multiple independent scalar regressors, whereas the proposed VV-KRR formulation learns a coupled multi-output model. At the same time, the PCA-based baseline represents the most relevant practical reference among the alternative approaches considered in this example, since it provides a favorable compromise between predictive accuracy, model compactness, and training cost.

The neural-network results provide an additional perspective. The fully connected DNN benchmark, although tuned by Bayesian optimization, remains less accurate than the best kernel-based alternatives and requires a significantly larger training time. The structured CNN substantially improves the neural-network accuracy, reaching an ℓ_2 error of 0.9% and an ℓ_∞ error of 6.1%. This confirms that exploiting the frequency-domain structure of the output response is beneficial for neural-network-based surrogate modeling. However, this improved accuracy is obtained at the price of a much larger training time, equal to 1662 s, mainly due to the larger number of architectural and training hyperparameters to be optimized and to the higher complexity of the training procedure. Therefore, although the CNN baseline is competitive in terms of accuracy, it is less attractive from the viewpoint of the accuracy–training-cost tradeoff.

It is also worth noting that model #4 requires the tuning of only 12 hyperparameters in total, corresponding to 11 ARD length scales and one regularization parameter λ. In contrast, the corresponding PCA-based surrogate combined with nonlinear regression requires a separate set of hyperparameters for each retained PCA component. In this example, where 42 components are retained, the total number of hyperparameters rises to $12 \times 42 = 504$. This difference stems from the fact that the PCA basis is constructed in an unsupervised way, whereas the associated scalar regressors are trained independently and each requires its own kernel and set of hyperparameters.

From the viewpoint of model size, the proposed formulation with a data-driven output kernel requires a number of coefficients of the same order as the PCA-based models, while preserving a coupled representation of the output response. This is particularly evident for model #4, which achieves the best accuracy with a coefficient count comparable to that of the PCA-based baselines. By contrast, when analytic kernels are adopted for the output kernel k_o, the spectral compression may become less effective, because the resulting output Gram matrix can exhibit a slower eigenvalue decay. As a consequence, more coefficients may need to be retained, together with a larger training cost. This effect is visible, for example, by comparing model #4 with model #16: both models attain the same ℓ_2 error, but model #16 requires more than twice as many retained coefficients.

Figure 2 compares the scatter plots obtained from model #1, trained according to [46], and from model #4 trained with the proposed approach. The left panel clearly highlights the gain in prediction accuracy achieved by the proposed training strategy. For completeness, the right panel

TABLE 3. Comparison of the performance of the various models considered for the example in Section VI-B. Reported metrics include the relative ℓ_2 and ℓ_∞ errors, the training time, and the number of coefficients (when available). Best predictive results are highlighted in bold.

ID	Algorithm	k_o Type	k_x Type	ℓ_2 [%]	ℓ_∞ [%]	$T_{\rm tr}$ [s]	# Coeff.
1	[46] (Sec. IV-A)	Kronecker delta	RBF	2.8*	23.1*	15**	30,000
2	[46] (Sec. IV-B)	RBF	RBF	4.1*	20.3*	961**	30,000
3	Proposed	Data-Driven	Matérn 5/2	2.6	18.7	3.7	13,100
4	Proposed	Data-Driven	ARD Matérn 5/2	**0.5**	**5.1**	33.0	13,850
5	Proposed	Data-Driven	RBF	2.7	17.5	8.2	7,900
6	Proposed	Data-Driven	ARD RBF	0.6	6.8	106.7	14,300
7	Proposed	Data-Driven	Matérn 1/2	3.2	21.0	3.7	14,950
8	Proposed	Data-Driven	ARD Matérn 1/2	0.9	6.7	61.7	14,950
9	Proposed	RBF	Matérn 5/2	2.7	20.2	6.3	26,200
10	Proposed	RBF	ARD Matérn 5/2	**0.5**	5.5	67.9	2,890
11	Proposed	RBF	RBF	2.8	20.0	3.3	17,700
12	Proposed	RBF	ARD RBF	0.6	6.9	97.7	27,400
13	Proposed	RBF	Matérn 1/2	3.2	21.0	6.6	29,800
14	Proposed	RBF	ARD Matérn 1/2	0.9	6.7	69.3	30,000
15	Proposed	Matérn 5/2	Matérn 5/2	2.7	20.2	4.4	26,200
16	Proposed	Matérn 5/2	ARD Matérn 5/2	**0.5**	5.5	85.1	28,900
17	Proposed	Matérn 5/2	RBF	2.8	20.0	3.6	17,700
18	Proposed	Matérn 5/2	ARD RBF	0.6	6.9	111.2	27,700
19	Proposed	Matérn 5/2	Matérn 1/2	3.2	21.0	6.4	29,800
20	Proposed	Matérn 5/2	ARD Matérn 1/2	0.9	6.7	68.7	30,000
21	PCA+Reg.	PCA	Matérn 5/2	2.6	18.5	8.5	12,600
22	PCA+Reg.	PCA	ARD Matérn 5/2	0.9	8.5	24.3	12,600
23	PCA+Reg.	PCA	RBF	2.6	18.2	5.1	12,600
24	PCA+Reg.	PCA	ARD RBF	0.6	6.3	23.2	12,600
25	DNN***	–	–	2.2	11.5	303.7	–
26	CNN***	–	–	0.9	6.1	1662	–

* Different from [46], the errors are computed on data in dB.
** Updated training times reflect execution on new hardware.
*** DNN/CNN hyperparameters selected by Bayesian optimization. Reported times include tuning and final retraining.

reports the scatter plot obtained with the best PCA-based baseline, namely model #24.

Figure 3 shows the predictions obtained from model #4 for five random configurations of the input parameters in the test set, together with the corresponding reference curves. The inset illustrates the correlation matrix learned from the training data and used as the output kernel **B**. These plots provide a graphical illustration of the accuracy achieved by the best-performing model.

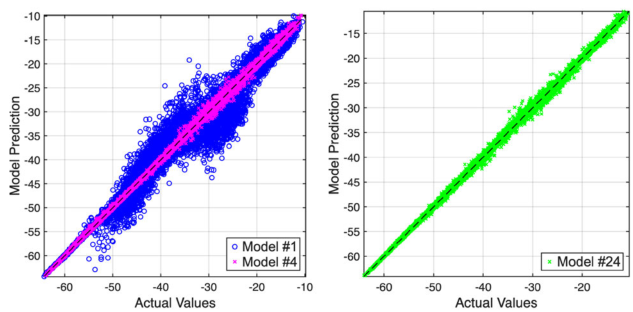

FIGURE 2. Scatter plots for $|H(j\omega; x)|$ for the example in Section VI-B, computed on the test set across all frequency points. Left panel: comparison between model #1 from [46] (blue dots) and model #4 trained via the proposed approach (magenta crosses). Right panel: scatter plot obtained with the PCA+regression baseline, model #24.

C. EXAMPLE 2: LOW-NOISE AMPLIFIER

As a second example, the proposed approach is used to model the real and imaginary parts of the S_{11} and S_{21} parameters of the 2-GHz low-noise amplifier shown in Fig. 4. The amplifier depends on $p = 25$ uniformly distributed parameters affecting the BJT parasitics, the forward current gain, the lumped circuit elements, and the widths of the microstrip lines. All parameters are assumed to vary within ±20% around their nominal values. A detailed description of the considered parameters is provided in Section 7.3 of [29].

The test case is implemented in HSPICE through a parametric small-signal AC analysis, which provides S_{11} and S_{21} at 201 frequency points between 1 GHz and 3 GHz. Three training sets with $L = 100$, 200, and 400 samples are generated via a LHS, and the corresponding surrogate models are evaluated on the same 500-sample test set.

Since the responses are complex-valued, each output is split into its real and imaginary parts and rearranged into a single real-valued vector in order to comply with the proposed VV-KRR formulation. A 10-sample zero padding is inserted between the real and imaginary parts, and the same strategy is used to concatenate the responses associated with S_{11}

TABLE 4. Comparison of the performance of the various models considered for the example in Section VI-C, with increasing training-set sizes. Reported metrics include the relative ℓ_2 and ℓ_∞ errors, the training time T_{tr}, and the number of coefficients. Best predictive results are highlighted in bold.

ID	Alg.	k_o Type	k_x Type	$L=100$				$L=200$				$L=400$			
				ℓ_2 [%]	ℓ_∞ [%]	T_{tr} [s]	Coeff.	ℓ_2 [%]	ℓ_∞ [%]	T_{tr} [s]	Coeff.	ℓ_2 [%]	ℓ_∞ [%]	T_{tr} [s]	Coeff.
1	Proposed	Data-Driven	Matérn 5/2	1.8	7.3	2.0	1,980	1.4	5.7	2.6	3,920	0.7	3.1	10.1	8,085
2	Proposed	Data-Driven	ARD Matérn 5/2	1.2	5.3	10.6	1,860	0.6	3.1	29.8	3,580	**0.3**	1.6	220.6	7,665
3	Proposed	Data-Driven	RBF	1.9	7.4	1.2	1,960	1.5	6.0	1.8	3,980	0.6	2.6	23.3	8,190
4	Proposed	Data-Driven	ARD RBF	**1.1**	4.3	13.1	1,900	0.6	2.5	30.2	3,540	**0.3**	**1.5**	150.8	7,665
5	Proposed	Data-Driven	Matérn 1/2	3.8	12.0	1.2	1,940	2.3	8.7	2.0	3,660	1.7	6.7	7.3	8,337
6	Proposed	Data-Driven	ARD Matérn 1/2	2.5	9.7	12.8	2,000	1.8	7.2	29.5	4,000	1.2	5.6	176.9	8,379
7	Proposed	RBF	Matérn 5/2	1.8	7.5	20.2	82,566	1.4	5.7	27.1	163,464	0.8	3.2	85.8	324,870
8	Proposed	RBF	ARD Matérn 5/2	1.2	4.9	200.5	79,008	0.7	3.2	237.5	158,339	**0.3**	1.7	537.5	312,750
9	Proposed	RBF	RBF	1.8	7.3	23.1	82,566	1.4	5.7	38.8	164,298	0.7	2.8	29.7	326,190
10	Proposed	RBF	ARD RBF	**1.1**	**4.0**	287.9	79,230	0.6	2.6	317.8	145,950	**0.3**	1.6	362.4	304,410
11	Proposed	RBF	Matérn 1/2	3.8	11.9	28.0	80,413	2.3	8.7	19.0	165,966	1.7	6.7	29.7	331,098
12	Proposed	RBF	ARD Matérn 1/2	2.5	9.5	287.8	83,400	1.8	7.2	309.5	166,800	1.2	5.6	479.7	333,600
13	Proposed	Matérn 5/2	Matérn 5/2	1.8	7.3	26.6	82,566	1.4	5.7	25.3	163,464	0.8	3.2	25.6	327,762
14	Proposed	Matérn 5/2	ARD Matérn 5/2	1.2	4.7	265.6	73,392	0.6	3.1	341.8	150,954	**0.3**	1.7	522.3	294,402
15	Proposed	Matérn 5/2	RBF	1.8	7.3	18.7	82,566	1.4	5.7	8.3	164,298	0.6	2.8	26.3	324,426
16	Proposed	Matérn 5/2	ARD RBF	**1.1**	4.2	298.6	79,230	0.6	2.6	318.5	145,950	**0.3**	1.6	470.7	297,738
17	Proposed	Matérn 5/2	Matérn 1/2	3.8	11.9	35.3	79,734	2.3	8.7	50.2	136,512	1.7	6.7	66.2	331,932
18	Proposed	Matérn 5/2	ARD Matérn 1/2	2.5	9.6	300.5	83,400	1.8	7.2	324.2	166,800	1.2	5.6	494.4	333,600
19	PCA+Reg.	PCA	Matérn 5/2	1.8	7.5	1.5	1,800	1.5	6.0	3.1	3,600	0.7	2.8	5.2	7,200
20	PCA+Reg.	PCA	ARD Matérn 5/2	1.2	4.7	5.5	1,800	0.6	2.7	15.8	3,600	**0.3**	1.6	46.7	7,200
21	PCA+Reg.	PCA	RBF	1.8	7.4	0.4	1,800	1.5	5.9	2.0	3,600	0.6	2.6	4.6	7,200
22	PCA+Reg.	PCA	ARD RBF	1.6	5.0	4.5	1,800	**0.5**	**2.4**	11.3	3,600	**0.3**	1.6	44.0	7,200

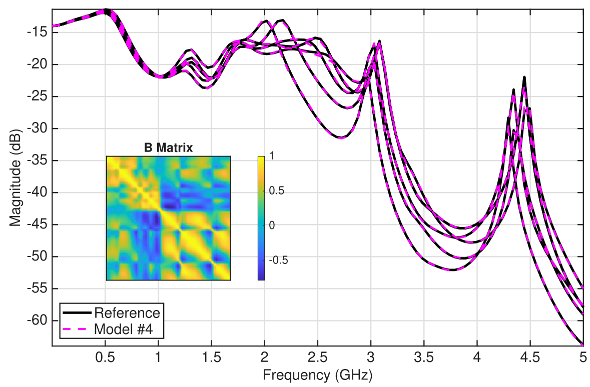

FIGURE 3. Predictions obtained from model #4 for five random configurations of the input parameters in the test set, compared with the corresponding reference curves. The inset shows the correlation matrix learned from the training data and used as the output kernel **B**.

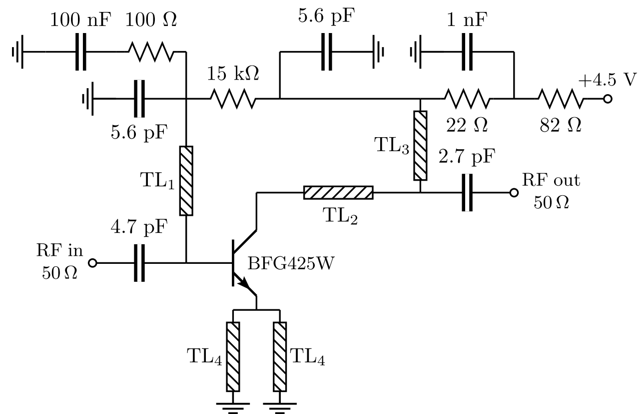

FIGURE 4. Schematic of the low-noise amplifier considered in Sec. VI-C [29].

and S_{21}. This preprocessing makes it possible to train a single real-valued multi-output model that jointly represents both scattering parameters while preserving the correlations among all output components. The same preprocessing is also adopted for the PCA-based surrogate models used for comparison.

The data-availability index in (43) can also be used to quantify the limited-data character of this example. For $p = 25$, $d_{\mathrm{out,eff}} = 15$, and $L = 100$, 200, and 400, the corresponding values are $\eta_{\mathrm{SD}} = 2.5$, 5, and 10, respectively. These values satisfy the empirical condition in (44), confirming that all three training configurations fall within the data-limited regime considered in this work.

Table 4 reports the complete comparison for the second benchmark example, including all the considered VV-KRR configurations and the PCA-based baselines for increasing training-set sizes. The reported errors are computed on the reconstructed complex-valued responses obtained from the predicted real and imaginary parts of S_{11} and S_{21}, using both the relative ℓ_2 and ℓ_∞ metrics over the test set.

As expected, the relative ℓ_2 error of all the models improves as the number of training samples increases, confirming that both the proposed VV-KRR formulation and the PCA-based surrogates are able to exploit the additional information provided by larger datasets. The improvement is particularly evident for the ARD-based configurations, which consistently provide the best predictive accuracy across the three training-set sizes.

For $L = 100$, the most accurate results are obtained by models based on anisotropic input kernels. In particular, models #4, #10, and #16 attain the lowest ℓ_2 error, equal to 1.1%. The lowest ℓ_∞ error, equal to 4.0%, is obtained by model #10, which combines an RBF output kernel with an ARD RBF input kernel. The data-driven model #4 remains very competitive, with an ℓ_2 error of 1.1% and an ℓ_∞ error of 4.3%, while requiring only 1900 retained coefficients, compared with 79,230 coefficients for model #10 and 79,230 coefficients for model #16.

For $L = 200$, the best accuracy is achieved by the PCA-based model #22, which provides the lowest ℓ_2 and ℓ_∞ errors, equal to 0.5% and 2.4%, respectively. The proposed VV-KRR models remain very close, with the best configurations reaching $\ell_2 = 0.6\%$ and ℓ_∞ values between 2.5% and 2.6%. Among them, the data-driven model #4 provides an excellent compromise between accuracy, model compactness, and training cost.

For $L = 400$, both the proposed VV-KRR models and the PCA-based baselines reach very high accuracy. The best ℓ_2 error, equal to 0.3%, is achieved by several ARD-based configurations, including the data-driven VV-KRR models #2 and #4 and the PCA-based models #20 and #22. The lowest ℓ_∞ error, equal to 1.5%, is obtained by model #4, which combines the data-driven output kernel with an ARD RBF input kernel. This confirms that the proposed VV-KRR formulation can match the accuracy of the strongest PCA-based baselines while preserving a coupled multi-output representation.

From a computational viewpoint, the data-driven VV-KRR models are particularly attractive. They generally require a number of retained coefficients comparable to that of the PCA-based surrogates, while avoiding the need to train multiple independent scalar regressors. For example, for $L = 400$, the data-driven models require approximately 7.7×10^3–8.4×10^3 coefficients, which is close to the 7200 coefficients used by the PCA-based baselines. By contrast, several configurations based on analytic output kernels require more than 3×10^5 coefficients. This confirms that the empirical output kernel leads to a much more effective spectral compression of the output space. This behavior is clearly visible in models #7–#18, especially for the larger training sets. Moreover, when the analytic output kernel depends on hyperparameters, the output Gram matrix must be recomputed and diagonalized during hyperparameter tuning, which further increases the computational cost.

Overall, the results show that the proposed VV-KRR framework remains highly competitive across all the considered training-set sizes. The data-driven output kernel provides the best tradeoff between accuracy, model compactness, and computational cost, while the use of ARD input kernels is essential to achieve the best predictive performance in this 25-parameter benchmark. The proposed formulation also preserves a fully coupled multi-output representation, directly accounting for the correlations among the real and imaginary parts of S_{11} and S_{21}, as well as across the frequency samples.

Figure 5 shows the scatter plots of the real and imaginary parts of S_{11} computed on the test set across all frequency points using one of the most accurate VV-KRR models trained with $L = 400$, namely model #2. Moreover, Fig. 6 presents the corresponding parametric plots for five random configurations of the input parameters in the test set. The inset illustrates the structure of the correlation matrix learned

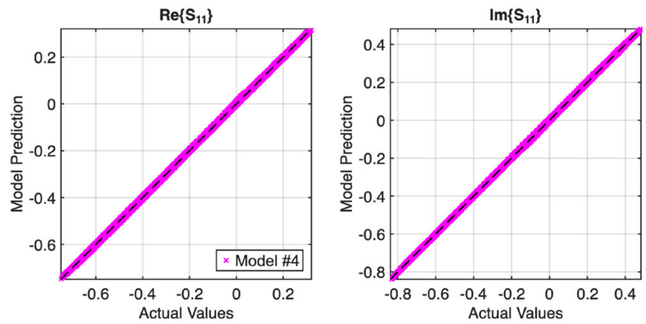

FIGURE 5. Scatter plots of the real and imaginary parts of S_{11} for the example considered in Section VI-C, computed on the test set across all frequency points using model #2.

from the available training data and used as the output kernel $\mathbf{B}$ within the VV-KRR model. These plots provide a graphical illustration of the accuracy achieved by the proposed approach.

D. EXAMPLE 3: COMMON-MODE SUPPRESSION FILTER

In the final example, the proposed modeling scheme is assessed on a particularly challenging small-data scenario, in which only 290 samples are available. The task concerns the parametric modeling of the mixed-mode scattering parameters of the compact common-mode suppression filter shown in Fig. 7 [12], [65]. The filter is characterized by 6 geometrical parameters, namely the line width w_i, spacing s_i, and length l_i of the coupled microstrip lines for $i = 1, 2$, whose nominal values are reported in Table 5 and taken from [65]. The structure is implemented on a Rogers RO4350B substrate with $h = 1.524$ mm, $\epsilon_r = 3.66$, $\tan\delta = 0.003$, and $\sigma = 4.1 \times 10^7$ S/m.

The proposed VV-KRR method is employed to construct surrogate models that capture the impact of a ±20% uniform variation around the nominal values of the $p = 6$ geometrical parameters on the mixed-mode scattering parameters S_{dd21} and S_{cd21}, defined with respect to the single-ended ports shown in Fig. 7 (additional details are reported in [12] and [65]). The responses are evaluated at $D = 600$ frequency points in the range from 10 MHz to 6000 MHz.

A dataset with 290 samples is generated through full-wave simulations in ADS Momentum using an LHS scheme. The proposed VV-KRR approach is then used to train two independent surrogate models, one for S_{dd21} and one for S_{cd21}. The dataset is split into a fixed training set and a fixed test set. In particular, 190 samples are used for training and hyperparameter selection, while the remaining 100 samples are reserved exclusively for final testing. This protocol provides a standard assessment of the model performance on a common hold-out test set, while preserving the small-training-data condition considered in this work. For each training set, the model hyperparameters are selected by the same 5-fold cross-validation procedure adopted in the previous examples.

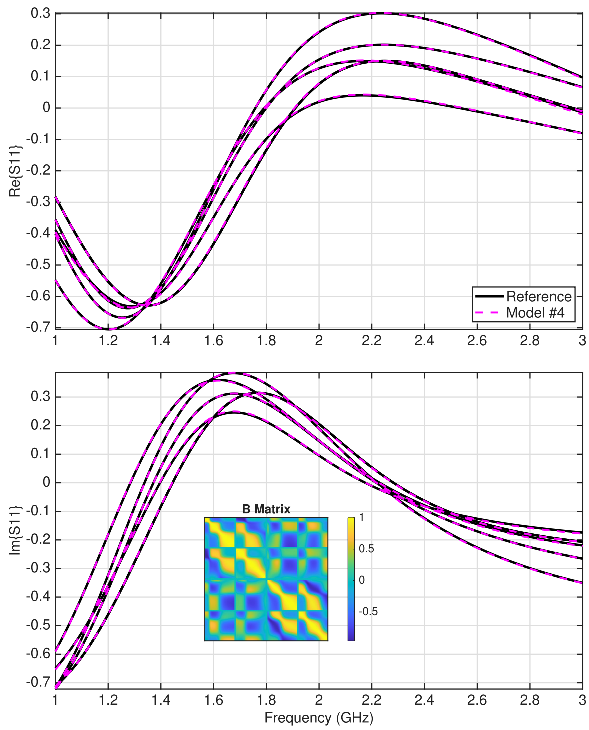

FIGURE 6. Parametric plots comparing the predictions of the real and imaginary parts of S_{11} for the example in Section VI-C, obtained using model #2, with the corresponding reference curves for five random test configurations of the input parameters. The inset shows the correlation matrix learned from the training data and used as the output kernel **B**, which accounts for the correlation between the real and imaginary parts of the output across frequency.

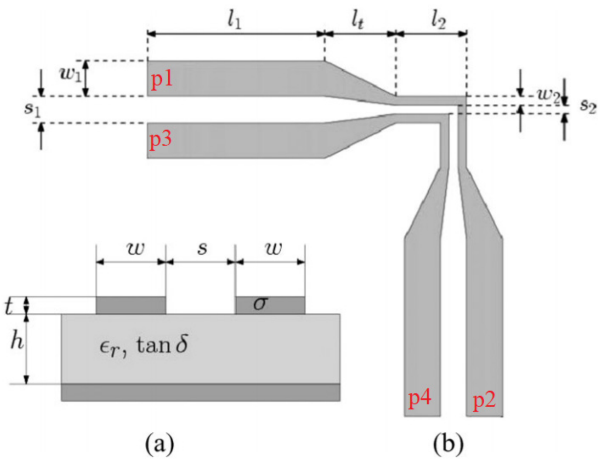

FIGURE 7. Common-mode suppression filter and its geometrical parameters considered in Sec. VI-D. Panel (a): cross-section. Panel (b): top view (adapted from [12], [65]).

Also in this case, the small-data regime can be assessed through the data-availability index η_{SD}. For $p = 6$, $L = 190$, and effective output dimensions $d_{\text{out,eff}} = 22$ for S_{dd21} and $d_{\text{out,eff}} = 15$ for S_{cd21}, the corresponding values are $\eta_{\text{SD}} = 6.8$ and $\eta_{\text{SD}} = 9.0$, respectively. These values satisfy the empirical condition in (44), confirming that this benchmark

TABLE 5. Nominal parameters of the common-mode suppression filter in Sec. VI-D.

w_1	1.8 mm	s_1	0.7 mm	l_1	28.55 mm
w_2	0.6 mm	s_2	0.2 mm	l_2	10 mm

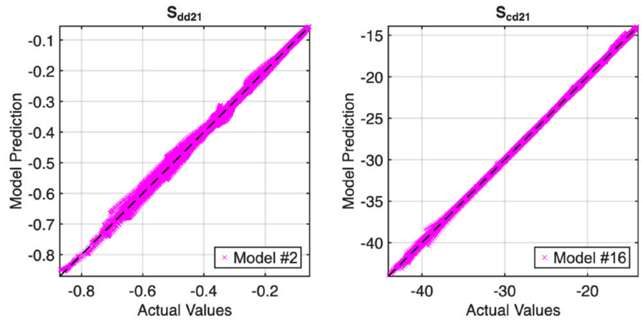

FIGURE 8. Scatter plots for S_{dd21} and S_{cd21} for the example in Sec. VI-D, computed on the fixed test set across all frequency points using model #2 for S_{dd21} and model #16 for S_{cd21}.

falls within the data-limited regime considered in this work.

Table 6 reports the complete comparison for the third benchmark example, including all the considered VV-KRR configurations and the PCA-based baselines for both S_{dd21} and S_{cd21}. The results are computed on the same fixed test set of 100 samples and are reported in terms of relative ℓ_2 and ℓ_∞ errors, training time, and number of retained coefficients.

For S_{dd21}, the best overall performance is obtained by model #2, which combines the data-driven output kernel with an ARD Matérn 5/2 input kernel. This configuration achieves an ℓ_2 error of 1.6% and the lowest ℓ_∞ error, equal to 5.0%, while retaining only 7,904 coefficients. The PCA-based baseline also provides competitive results, especially models #20 and #22, which reach ℓ_2 errors of 1.6% and 1.7%, respectively. However, model #2 provides a better ℓ_∞ error than the PCA-based alternatives and preserves a coupled multi-output representation of the response.

For S_{cd21}, the lowest ℓ_2 error is equal to 0.3% and is achieved by models #8, #14, and #16, all based on analytic output kernels and ARD input kernels. The lowest ℓ_∞ error, equal to 2.7%, is obtained by model #16, which combines a Matérn 5/2 output kernel with an ARD RBF input kernel. The data-driven output-kernel configurations remain close in accuracy, with models #2 and #4 reaching ℓ_2 errors of 0.4% and ℓ_∞ errors of 3.5% and 3.3%, respectively. These results indicate that analytic output kernels can provide a slight accuracy advantage for this response, at the price of a much larger number of retained coefficients.

From a computational viewpoint, the data-driven VV-KRR models provide the most favorable compromise between accuracy, training time, and model size. For both S_{dd21} and S_{cd21}, the data-driven configurations require only a few thousand retained coefficients, whereas several analytic-output-kernel models require more than 8×10^4 or even

TABLE 6. Comparison of the performance of the various models considered for the example in Section VI-D. Reported metrics include the relative ℓ_2 and ℓ_∞ errors, the training time T_{tr}, and the number of coefficients. The results are computed using a fixed training/test split. Best predictive results are highlighted in bold.

ID	Algorithm	k_o Type	k_x Type	S_{dd21}				S_{cd21}			
				ℓ_2 [%]	ℓ_∞ [%]	T_{tr} [s]	# Coeff.	ℓ_2 [%]	ℓ_∞ [%]	T_{tr} [s]	# Coeff.
1	Proposed	Data-Driven	Matérn 5/2	2.2	6.4	1.6	8,060	0.7	3.9	1.2	4,928
2	Proposed	Data-Driven	ARD Matérn 5/2	**1.6**	**5.0**	7.2	7,904	0.4	3.5	5.3	4,384
3	Proposed	Data-Driven	RBF	2.4	6.0	1.1	6,656	0.9	4.6	1.8	2,272
4	Proposed	Data-Driven	ARD RBF	3.6	11.2	2.3	3,120	0.4	3.3	5.1	2,656
5	Proposed	Data-Driven	Matérn 1/2	4.5	17.6	1.5	8,580	0.8	3.7	0.4	5,664
6	Proposed	Data-Driven	ARD Matérn 1/2	3.6	16.9	5.8	8,684	0.6	3.1	3.4	5,280
7	Proposed	RBF	Matérn 5/2	2.2	6.2	5.4	92,747	0.6	4.3	2.6	86,488
8	Proposed	RBF	ARD Matérn 5/2	1.7	6.5	16.6	91,040	**0.3**	3.2	10.3	72,263
9	Proposed	RBF	RBF	2.2	6.3	5.4	79,091	0.7	4.7	11.4	64,866
10	Proposed	RBF	ARD RBF	1.8	6.4	71.2	67,142	0.4	3.0	23.6	44,951
11	Proposed	RBF	Matérn 1/2	4.5	17.6	4.4	93,885	0.8	3.6	2.4	100,144
12	Proposed	RBF	ARD Matérn 1/2	3.6	16.8	32.8	94,454	0.6	3.1	11.6	97,299
13	Proposed	Matérn 5/2	Matérn 5/2	2.2	6.2	5.0	92,747	0.6	4.3	2.8	86,488
14	Proposed	Matérn 5/2	ARD Matérn 5/2	1.7	6.9	33.9	91,040	**0.3**	3.0	44.3	70,556
15	Proposed	Matérn 5/2	RBF	2.3	7.2	18.5	84,212	0.8	4.2	12.7	33,571
16	Proposed	Matérn 5/2	ARD RBF	3.0	8.7	16.5	56,331	**0.3**	**2.7**	31.2	54,055
17	Proposed	Matérn 5/2	Matérn 1/2	4.5	17.6	4.0	94,454	0.8	3.6	2.6	100,144
18	Proposed	Matérn 5/2	ARD Matérn 1/2	3.6	16.8	30.2	95,592	0.6	3.1	12.9	97,299
19	PCA+Reg.	PCA	Matérn 5/2	2.2	7.0	3.6	8,360	0.6	4.2	1.9	4,750
20	PCA+Reg.	PCA	ARD Matérn 5/2	**1.6**	5.5	5.8	8,360	0.5	4.9	3.3	4,750
21	PCA+Reg.	PCA	RBF	2.4	6.5	2.2	8,360	0.6	4.0	1.3	4,750
22	PCA+Reg.	PCA	ARD RBF	1.7	5.2	5.0	8,360	0.5	3.3	2.3	4,750

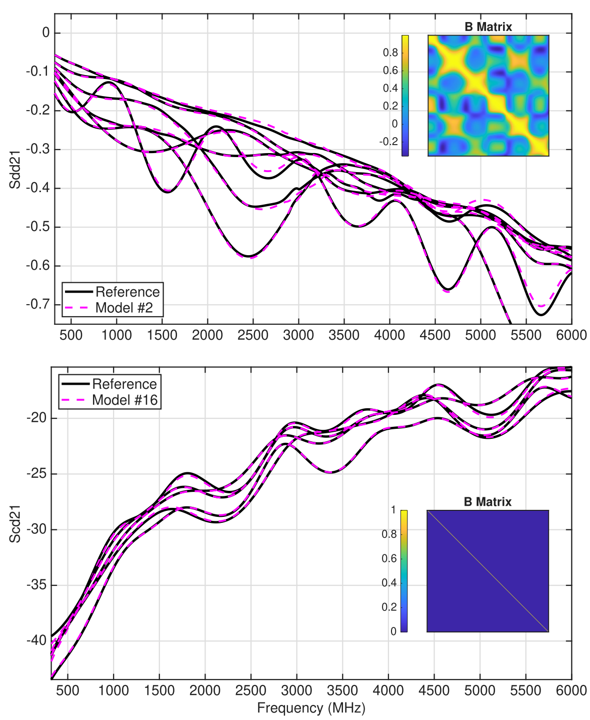

FIGURE 9. Parametric plots comparing the predictions of S_{dd21} and S_{cd21} for the example in Section VI-D, obtained using model #2 for S_{dd21} and model #16 for S_{cd21}, with the corresponding reference curves for five representative test configurations. The insets show the output-kernel matrices used by the selected VV-KRR models.

10^5 coefficients. This difference confirms that the empirical output kernel leads to a more effective spectral compression of the output space in this example.

By contrast, analytic output kernels often lead to less effective compression, with a significantly larger number of retained coefficients and longer training times. This behavior is particularly evident in models #7–#18. Nevertheless, analytic output kernels can still provide competitive or slightly better accuracy for some configurations, especially for S_{cd21}. Overall, the results confirm that the proposed compressed VV-KRR formulation remains effective in a strongly data-constrained regime and highlights the tradeoff between predictive accuracy, model compactness, and training cost.

Figure 8 shows the scatter plots obtained with the best-performing VV-KRR configurations selected from Table 6, namely model #2 for S_{dd21} and model #16 for S_{cd21}. Figure 9 shows the corresponding parametric responses for five representative configurations selected from the test set. The insets show the output-kernel matrices used by the selected models.

VII. CONCLUSION

This paper presented a data-efficient surrogate modeling framework based on a compressed formulation of vector-valued kernel ridge regression (VV-KRR) for electromagnetic structures. The proposed training scheme exploits spectral diagonalization and low-rank kernel approximations to derive a compressed solution of the original coupled learning problem. In this way, the computational cost and memory requirements of VV-KRR can be significantly reduced, while preserving its fully coupled multi-output nature and maintaining high predictive accuracy.

The notion of data-limited modeling was also discussed and quantified through a practical data-availability index that accounts for the number of training samples, the number of input parameters, and the effective dimensionality of the output response. This allowed the considered examples to be interpreted within a common small-data framework, while also clarifying that such an index should be regarded as a first-order diagnostic rather than as a complete measure of nonlinear sample complexity.

The numerical results obtained on three benchmark examples confirm the effectiveness of the proposed approach. In the first example, the compressed VV-KRR formulation outperformed the original VV-KRR implementations from the literature, as well as the PCA-based baselines considered in this work. The comparison with neural-network baselines also showed that a structured CNN can achieve promising accuracy when the frequency-domain organization of the output response is exploited, but at the price of a substantially larger training time due to the more complex hyperparameter optimization and training procedure. In the second example, the proposed method remained competitive with the best PCA-based surrogates and, in several cases, achieved comparable or better accuracy. In the third and most data-limited example, the assessment based on a fixed training/test split confirmed the effectiveness of the proposed formulation when only a limited number of full-wave simulations is available for training, providing accurate predictions for both considered mixed-mode scattering parameters.

The results also highlighted the role of the output-kernel choice. The data-driven output kernel provided an attractive tradeoff between accuracy, model compactness, and training cost, especially because it is hyperparameter-free and led to effective spectral compression in the considered deterministic simulation-based examples. At the same time, analytic output kernels were shown to be competitive in some cases, suggesting that structured output-correlation models may be useful when the empirical correlation matrix is less reliable or when additional regularization is needed. Shrinkage-based regularization of the output kernel was identified as a possible extension, particularly relevant for noisy data or poorly conditioned empirical correlation estimates, but its interaction with the spectral compression step requires a dedicated analysis.

Overall, the proposed compressed VV-KRR framework provides a good balance between predictive accuracy, model compactness, and training cost, even when only limited training data are available. These characteristics make the proposed framework attractive for surrogate-assisted design, optimization, and variability analysis in RF and interconnect applications, where only limited simulation data can often be afforded. The encouraging performance of the proposed vector-valued deterministic approach also motivates future extensions toward probabilistic vector-valued modeling strategies, such as multi-output Gaussian-process models and intrinsic coregionalization models [55], [66].

Future work will focus on the integration of the proposed models into circuit and electromagnetic simulation tools, on the use of more efficient truncated spectral decomposition techniques to further accelerate the training phase, and on the investigation of alternative and adaptive strategies for kernel compression, output-kernel regularization, and compression-tolerance selection.

ACKNOWLEDGMENT

The authors used ChatGPT-5.4 for language editing and as an aid in generating portions of the code for the deep neural network and the convolutional neural network used in Example 1. All generated material was reviewed, corrected where needed, and validated by them.

REFERENCES

[1] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q.-J. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 4140–4155, Oct. 2019.

[2] J. Zhang, J. Chen, Q. Guo, W. Liu, F. Feng, and Q.-J. Zhang, "Parameterized modeling incorporating MOR-based rational transfer functions with neural networks for microwave components," *IEEE Microw. Wireless Compon. Lett.*, vol. 32, no. 5, pp. 379–382, May 2022.

[3] W. Zhang, W. Liu, S. Yan, F. Feng, J. Zhang, J. Jin, and Q.-J. Zhang, "Surrogate-assisted multistate tuning-driven EM optimization for microwave tunable filter," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 4, pp. 2015–2030, Apr. 2022.

[4] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microw. Mag.*, vol. 11, no. 3, pp. 105–118, May 2010.

[5] H. M. Torun, H. Yu, N. Dasari, V. C. K. Chekuri, A. Singh, J. Kim, S. K. Lim, S. Mukhopadhyay, and M. Swaminathan, "A spectral convolutional net for co-optimization of integrated voltage regulators and embedded inductors," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2019, pp. 1–8.

[6] H. M. Torun, A. C. Durgun, K. Aygun, and M. Swaminathan, "Causal and passive parameterization of S-parameters using neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 10, pp. 4290–4304, Oct. 2020.

[7] M. Swaminathan, H. M. Torun, H. Yu, J. A. Hejase, and W. D. Becker, "Demystifying machine learning for signal and power integrity problems in packaging," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 8, pp. 1276–1295, Aug. 2020.

[8] O. Akinwande, S. Erdogan, R. Kumar, and M. Swaminathan, "Surrogate modeling with complex-valued neural nets for signal integrity applications," *IEEE Trans. Microw. Theory Techn.*, vol. 72, no. 1, pp. 478–489, Jan. 2024.

[9] K. C. Sahu, S. Koziel, and A. Pietrenko-Dabrowska, "Surrogate modeling of passive microwave circuits using recurrent neural networks and domain confinement," *Sci. Rep.*, vol. 15, no. 1, Apr. 2025, Art. no. 13322.

[10] W. Na, T. Bai, D. Jin, H. Xie, W. Zhang, and Q.-J. Zhang, "Advanced neural space mapping-based inverse modeling method for microwave filter design," *IEEE Microw. Wireless Technol. Lett.*, vol. 35, no. 1, pp. 12–15, Jan. 2025.

[11] M. Swaminathan, O. W. Bhatti, Y. Guo, E. Huang, and O. Akinwande, "Bayesian learning for uncertainty quantification, optimization, and inverse design," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4620–4634, Nov. 2022.

[12] M. Sedaghat, R. Trinchero, Z. H. Firouzeh, and F. G. Canavero, "Compressed machine learning-based inverse model for design optimization of microwave components," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 7, pp. 3415–3427, Jul. 2022.

[13] O. Akinwande, X. Jia, A. Deroo, Y. Lu, H. Lin, B.-C. Tseng, and M. Swaminathan, "Transfer learning framework for 3-D electromagnetic applications," *IEEE Trans. Microw. Theory Techn.*, vol. 73, no. 8, pp. 4490–4500, Aug. 2025, doi: 10.1109/TMTT.2025.3525986.

[14] S. Guglani, A. K. Jakhar, A. Dasgupta, and S. Roy, "Combining prior knowledge with transfer learning (PKID-TL) for fast neural network enabled uncertainty quantification of graphene on-chip interconnects," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 15, no. 1, pp. 85–93, Jan. 2025.

[15] R. Kumar, S. S. L. Narayan, S. Kumar, S. Roy, B. K. Kaushik, R. Achar, and R. Sharma, "Knowledge-based neural networks for fast design space exploration of hybrid copper-graphene on-chip interconnect networks," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 1, pp. 182–195, Feb. 2022.

[16] S. Kushwaha, N. Soleimani, F. Treviso, R. Kumar, R. Trinchero, F. G. Canavero, S. Roy, and R. Sharma, "Comparative analysis of prior knowledge-based machine learning metamodels for modeling hybrid copper–graphene on-chip interconnects," *IEEE Trans. Electromagn. Compat.*, vol. 64, no. 6, pp. 2249–2260, Dec. 2022.

[17] J. Cui, L. Zhang, H. Kabir, Z. Zhao, R. Sweeney, and Q.-J. Zhang, "Knowledge-based extrapolation of neural network model for transistor modeling," *IEEE Microw. Wireless Technol. Lett.*, vol. 35, no. 6, pp. 812–815, Jun. 2025.

[18] F. Feng, V.-M.-R. Gongal-Reddy, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric modeling of microwave components using adjoint neural networks and pole-residue transfer functions with EM sensitivity analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 1955–1975, Jun. 2017.

[19] Z. Zhao, F. Feng, W. Zhang, J. Zhang, J. Jin, and Q.-J. Zhang, "Parametric modeling of EM behavior of microwave components using combined neural networks and hybrid-based transfer functions," *IEEE Access*, vol. 8, pp. 93922–93938, 2020.

[20] L.-Y. Xiao, W. Shao, X. Ding, and B.-Z. Wang, "Dynamic adjustment kernel extreme learning machine for microwave component design," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 10, pp. 4452–4461, Oct. 2018.

[21] 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.

[22] K. S. Sidhu and R. Khazaka, "Nested Latin hypercube-based sampling for efficient uncertainty quantification using sensitivity-assisted least squares SVM," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 15, no. 1, pp. 75–84, Jan. 2025.

[23] R. Trinchero, M. Larbi, H. M. Torun, F. G. Canavero, and M. Swaminathan, "Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters," *IEEE Access*, vol. 7, pp. 4056–4066, 2019.

[24] N. Soleimani and R. Trinchero, "Compressed complex-valued least squares support vector machine regression for modeling of the frequency-domain responses of electromagnetic structures," *Electronics*, vol. 11, no. 4, p. 551, Feb. 2022.

[25] M. Atlante, R. Trinchero, I. S. Stievano, M. Telescu, and N. Tanguy, "IC modeling via machine learning regressions: A data-driven approach to SPICE integration," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 15, no. 9, pp. 1814–1822, Sep. 2025, doi: 10.1109/TCPMT.2025.3584470.

[26] F. Treviso, R. Trinchero, and F. G. Canavero, "Multiple delay identification in long interconnects via LS-SVM regression," *IEEE Access*, vol. 9, pp. 39028–39042, 2021.

[27] Y. Lindemans, T. Ullrick, I. Couckuyt, D. Deschrijver, and T. Dhaene, "Bayesian optimization of microwave filters: A physics-informed approach using the Szegö kernel," in *Proc. IEEE 29th Workshop Signal Power Integrity (SPI)*, Gaeta, Italy, May 2025, pp. 1–4.

[28] F. Garbuglia, D. Deschrijver, and T. Dhaene, "On the role of Bayesian learning for electronic design automation: A survey," *IEEE Electromagn. Compat. Mag.*, vol. 12, no. 4, pp. 77–84, 4th Quart., 2023.

[29] P. Manfredi and R. Trinchero, "Nonparametric formulation of polynomial chaos expansion based on least-square support-vector machines," *Eng. Appl. Artif. Intell.*, vol. 133, Jul. 2024, Art. no. 108182.

[30] P. Manfredi, "Probabilistic uncertainty quantification of microwave circuits using Gaussian processes," *IEEE Trans. Microw. Theory Techn.*, vol. 71, no. 6, pp. 2360–2372, Jun. 2023.

[31] S. Koziel, A. Pietrenko-Dabrowska, and M. Al-Hasan, "Design-oriented two-stage surrogate modeling of miniaturized microstrip circuits with dimensionality reduction," *IEEE Access*, vol. 8, pp. 121744–121754, 2020.

[32] S. Koziel, "Low-cost data-driven surrogate modeling of antenna structures by constrained sampling," *IEEE Antennas Wireless Propag. Lett.*, vol. 16, pp. 461–464, 2017.

[33] T. Bradde, S. Grivet-Talocia, A. Zanco, and G. C. Calafiore, "Data-driven extraction of uniformly stable and passive parameterized macromodels," *IEEE Access*, vol. 10, pp. 15786–15804, 2022.

[34] A. Zanco, S. Grivet-Talocia, T. Bradde, and M. De Stefano, "Uniformly stable parameterized macromodeling through positive definite basis functions," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 11, pp. 1782–1794, Nov. 2020.

[35] K. T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*. New York, NY, USA: Taylor & Francis, 2016.

[36] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. A. Mohamed, M. H. Bakr, K. H. Madsen, and J. Søndergaard, "Space mapping: The state of the art," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 337–361, Jan. 2004.

[37] R. Trinchero and F. Canavero, "Machine learning regression techniques for the modeling of complex systems: An overview," *IEEE Electromagn. Compat. Mag.*, vol. 10, no. 4, pp. 71–79, 4th Quart., 2021.

[38] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A surrogate modeling and adaptive sampling toolbox for computer based design," *J. Mach. Learn. Res.*, vol. 11, no. 68, pp. 2051–2055, Jul. 2010.

[39] D. Özese, M. G. Baydoğan, A. C. Durgun, and K. Aygün, "Tree-based boosting for efficient estimation of S-parameters for package electrical analysis," in *Proc. IEEE 33rd Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Toronto, ON, Canada, Oct. 2024, pp. 1–3.

[40] P. Xu, X. Ji, M. Li, and W. Lu, "Small data machine learning in materials science," *npj Comput. Mater.*, vol. 9, no. 1, Mar. 2023, Art. no. 42.

[41] I. Kraljevski, Y. C. Ju, D. Ivanov, C. Tschöpe, and M. Wolff, "How to do machine learning with small data? A review from an industrial perspective," 2023, *arXiv:2311.07126*.

[42] J. Bornschein, F. Visin, and S. Osindero, "Small data, big decisions: Model selection in the small-data regime," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2020, pp. 1035–1044.

[43] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *WIREs Data Mining Knowl. Discovery*, vol. 5, no. 5, pp. 216–233, Sep. 2015, doi: 10.1002/widm.1157.

[44] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed., New York, NY, USA: Springer, 1999.

[45] J. A. K. Suykens, *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.

[46] N. Soleimani, R. Trinchero, and F. G. Canavero, "Bridging the gap between artificial neural networks and kernel regressions for vector-valued problems in microwave applications," *IEEE Trans. Microw. Theory Techn.*, vol. 71, no. 6, pp. 2319–2332, Jun. 2023.

[47] N. Soleimani and R. Trinchero, "Efficient implementation of the vector-valued kernel ridge regression for the parametric modeling of the frequency-response of a high-speed link," in *Proc. 35th Gen. Assem. Sci. Symp. Int. Union Radio Sci. (URSI GASS)*, Aug. 2023, pp. 1–4.

[48] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., New York, NY, USA: Springer, 2002.

[49] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *IEEE Microw. Wireless Compon. Lett.*, vol. 18, no. 6, pp. 383–385, Jun. 2008.

[50] S. B. Olivadese and S. Grivet-Talocia, "Compressed passive macromodeling," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 2, no. 8, pp. 1378–1388, Aug. 2012.

[51] A. Carlucci, S. Grivet-Talocia, S. Kulasekaran, and K. Radhakrishnan, "Structured model order reduction of system-level power delivery networks," *IEEE Access*, vol. 12, pp. 18198–18214, 2024.

[52] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, 2nd ed., Philadelphia, PA, USA: SIAM, 2011.

[53] P. Manfredi and R. Trinchero, "A data compression strategy for the efficient uncertainty quantification of time-domain circuit responses," *IEEE Access*, vol. 8, pp. 92019–92027, 2020.

[54] A. Rudi, L. Carratino, and L. Rosasco, "Falkon: An optimal large scale kernel method," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3891–3901.

[55] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Found. Trends Mach. Learn.*, vol. 4, no. 3, pp. 195–266, Jun. 2012.

[56] A. C. Micchelli and M. Pontil, "Kernels for multi-task learning," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA, 2004, pp. 921–928.

[57] C. A. Micchelli and M. Pontil, "On learning vector-valued functions," *Neural Comput.*, vol. 17, no. 1, pp. 177–204, Jan. 2005.

[58] L. Baldassarre, L. Rosasco, A. Barla, and A. Verri, "Multi-output learning via spectral filtering," *Mach. Learn.*, vol. 87, pp. 259–301, Jun. 2012.

[59] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
[60] N. Soleimani, P. Manfredi, and R. Trinchero, "Efficient implementation of the vector-valued kernel ridge regression for the uncertainty quantification of the scattering parameters of a 2-GHz low-noise amplifier," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2023, pp. 143–146.
[61] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial," 2019, *arXiv:1905.12787*.
[62] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 2960–2968.
[63] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[64] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," 2010, *arXiv:0909.4061*.
[65] C. Gazda, I. Couckuyt, H. Rogier, D. Vande Ginste, and T. Dhaene, "Constrained multiobjective optimization of a common-mode suppression filter," *IEEE Trans. Electromagn. Compat.*, vol. 54, no. 3, pp. 704–707, Jun. 2012.
[66] E. V. Bonilla, K. M. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 153–160.

NAZANIN SOLEIMANI (Graduate Student Member, IEEE) received the M.Sc. degree in control engineering from the Azad University of Gonabad, Gonabad, Iran, in 2016. She is currently pursuing the Ph.D. degree with the EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy. Her research interests include modeling techniques and machine-learning methods for the parametric modeling of electronic and electromagnetic systems and devices.

IGOR S. STIEVANO (Senior Member, IEEE) received the master's degree in electronic engineering and the Ph.D. degree in electronics and communication engineering from Politecnico di Torino, Turin, Italy, in 1996 and 2001, respectively. He is currently a Professor of electrical engineering with the Department of Electronics and Telecommunications, Politecnico di Torino. He has authored or co-authored more than 140 papers published in international journals and conference proceedings. His research interests include modeling and simulation of electrical and electronic systems, with an emphasis on behavioral modeling of digital integrated circuits, signal, and power integrity, energy networks, statistical, and machine learning methods.

RICCARDO TRINCHERO (Member, IEEE) received the M.Sc. and Ph.D. degrees in electronics and communication engineering from the Politecnico di Torino, Turin, Italy, in 2011 and 2015, respectively. He is currently an Associate Professor with the EMC Group, Department of Electronics and Telecommunications, Politecnico di Torino. His research interests include the analysis of switching DC–DC converters, machine learning, and statistical simulation of circuits and systems.