

Bayesian Model Fusion: Large-Scale Performance Modeling of Analog and Mixed-Signal Circuits by Reusing Early-Stage Data

Fa Wang, *Student Member, IEEE*, Paolo Cachecho, Wangyang Zhang, *Member, IEEE*, Shupeng Sun, *Student Member, IEEE*, Xin Li, *Senior Member, IEEE*, Rouwaida Kanj, *Senior Member, IEEE*, and Chenjie Gu, *Member, IEEE*

Abstract—Efficient performance modeling of today’s analog and mixed-signal circuits is an important yet challenging task, due to the high-dimensional variation space and expensive circuit simulation. In this paper, we propose a novel performance modeling algorithm that is referred to as Bayesian model fusion (BMF) to address this challenge. The key idea of BMF is to borrow the information collected from an early stage (e.g., schematic level) to facilitate efficient performance modeling at a late stage (e.g., post layout). Such a goal is achieved by statistically modeling the performance correlation between early and late stages through Bayesian inference. Furthermore, to make the proposed BMF method of practical utility, four implementation issues, including: 1) prior mapping; 2) missing prior knowledge; 3) fast solver; and 4) prior and hyper-parameter selection, are carefully considered in this paper. Two circuit examples designed in a commercial 32 nm CMOS silicon on insulator process demonstrate that the proposed BMF method achieves up to 9× runtime speed-up over the traditional modeling technique without surrendering any accuracy.

Index Terms—Analog and mixed-signal (AMS) circuits, Bayesian model fusion (BMF), performance modeling, process variation.

I. INTRODUCTION

THE AGGRESSIVE scaling of integrated circuits leads to large-scale process variations that cannot be easily

Manuscript received July 24, 2015; revised October 1, 2015; accepted November 6, 2015. Date of publication November 29, 2015; date of current version July 15, 2016. This work was supported in part by the Division of Computing and Communication Foundations, National Science Foundation, under Contract CCF-1316363, and in part by the Intel Corporation. The work of P. Cachecho was supported by the University Research Board at the American University of Beirut. This paper was presented in part at the Design Automation Conference in 2013 [1]. This paper was recommended by Associate Editor P. Maffezzoni.

F. Wang, S. Sun, and X. Li are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: fwang1@ece.cmu.edu; shupengs@ece.cmu.edu; xinli@ece.cmu.edu).

P. Cachecho and R. Kanj are with the Department of Electrical and Computer Engineering, American University of Beirut, Beirut 1107 2020, Lebanon (e-mail: paolocach@gmail.com; rouwaida.kanj@gmail.com).

W. Zhang was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Cadence Design Systems, Inc., Pittsburgh, PA 15238 USA (e-mail: zwy677@gmail.com).

C. Gu is with Intel Corporation, Hillsboro, OR 97124 USA (e-mail: chenjie.gu@intel.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2015.2504329

reduced by foundries. Process variations manifest themselves as the uncertainties associated with the geometrical and electrical parameters of semiconductor devices. These device-level variations significantly impact the parametric yield of analog and mixed-signal (AMS) circuits and, hence, must be appropriately modeled, analyzed, and optimized at all levels of design hierarchy [2]–[6].

To address this variability issue, various techniques for performance modeling have been developed during the past two decades [7]–[16]. The objective is to approximate the circuit-level performance (e.g., gain of an analog amplifier) as an analytical (e.g., linear, quadratic, etc.) function of device-level variations (e.g., ΔV_{TH} , ΔT_{OX} , etc.). Once such a performance model is available, it can be applied to a number of important applications such as estimating parametric yield [17], extracting worst-case corner [18], optimizing circuit design [19]–[23], etc.

While performance modeling was extensively studied in the past, the evolution of today’s AMS circuits has posed a number of new challenges in this area. In particular, the recent adoption of several emerging design methodologies (e.g., reconfigurable analog design, adaptive post-silicon tuning, etc.) leads to highly complex AMS systems that integrate numerous nanoscale devices. The remarkable increase of AMS circuit size results in a twofold consequence.

- 1) *High-Dimensional Variation Space*: A large number of device-level random variables must be used to model the process variations associated with a large-scale AMS system. For example, about 40 independent random variables are required to model the device mismatches of a single transistor for a commercial 32 nm CMOS silicon on insulator (SOI) process. If an AMS system contains 10^4 transistors, there are about 4×10^5 random variables in total to capture the corresponding device-level variations, resulting in a high-dimensional variation space. In addition, it is extremely difficult, if not impossible, to preselect a subset of these random variables for variation analysis, since the impact of device mismatches is circuit- and performance-dependent.
- 2) *Expensive Circuit Simulation*: The computational cost of circuit simulation substantially increases, as the AMS circuit size becomes increasingly large. For instance, it may take a few days or even a few weeks to run the

transistor-level simulation of a large AMS circuit such as phase-locked loop or high-speed link.

These recent trends of today's AMS circuits make performance modeling extremely difficult. On one hand, a large number of simulation samples must be generated in order to fit a high-dimensional model. On the other hand, creating a single sampling point by transistor-level simulation can take a large amount of computational time. The challenging issue here is how to make performance modeling computationally affordable for today's large-scale AMS circuits. This fundamental issue has not been appropriately addressed by the state-of-the-art performance modeling techniques (e.g., the recent sparse regression algorithms based on orthogonal matching pursuit (OMP) [13] or elastic net regularization [15]).

In this paper, we propose a new Bayesian model fusion (BMF) technique to facilitate large-scale performance modeling of AMS circuits. The proposed BMF method is motivated by the fact that today's AMS circuits are often designed via a multistage flow. Namely, an AMS design often spans three core stages: 1) schematic design; 2) layout design; and 3) chip manufacturing and testing. At each stage, simulation or measurement data are collected to validate the circuit design, before moving to the next stage. The traditional performance modeling techniques rely on the data at a single stage only and they completely ignore the data that are generated at other stages. The key idea of BMF, however, is to reuse the early-stage data when fitting a late-stage performance model. As such, the performance modeling cost can be substantially reduced.

Mathematically, the proposed BMF method is derived from the theory of Bayesian inference [24]. Starting from a set of early-stage (e.g., schematic-level) sampling points, BMF first approximates an early-stage performance model based on these samples. The early-stage model is used as a template to define our prior knowledge for late-stage (e.g., post-layout) performance modeling. Specifically, a prior distribution is statistically defined for the late-stage model coefficients. The prior knowledge is then combined with very few late-stage sampling points to solve the late-stage model coefficients via Bayesian inference. From this point of view, by fusing the early-stage and late-stage performance models through Bayesian inference, we only need a small number of late-stage sampling points to fit a high-dimensional late-stage model, thereby significantly reducing the computational cost for performance modeling.

BMF was previously proposed for parametric yield estimation of AMS circuits [25] where Bayesian inference was used to estimate the probability distribution of AMS performance metrics. In this paper, we further extend the idea of BMF to performance modeling. Compared to other traditional performance modeling methods, BMF reduces the number of required sampling points by up to $9\times$ without surrendering any accuracy, as will be demonstrated by our experimental results in Section V.

The remainder of this paper is organized as follows. In Section II, we review the important background on performance modeling, and then describe our proposed BMF method in Section III. Several implementation issues are discussed

in Section IV to further improve the modeling accuracy and reduce the computational cost of BMF. The efficacy of BMF is demonstrated by several circuit examples in Section V. Finally, we conclude in Section VI.

II. BACKGROUND

A. Performance Model

Given an AMS circuit (e.g., an analog amplifier), its performance (e.g., gain) may vary due to process variations. In the process design kit, a set of independent random variables with standard normal distribution

$$\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_R]^T \quad (1)$$

are usually used to model device-level process variations. The objective of performance modeling is to approximate the circuit performance as an analytical function of the device-level variations

$$f(\mathbf{x}) \approx \sum_{m=1}^M \alpha_m \cdot g_m(\mathbf{x}) \quad (2)$$

where f represents the performance of interest, $\{\alpha_m; m = 1, 2, \dots, M\}$ contains the model coefficients, $\{g_m(\mathbf{x}); m = 1, 2, \dots, M\}$ contains the basis functions, and M is the total number of basis functions. In this paper, we adopt orthonormal polynomials [26] as our basis functions. Namely, the basis functions $\{g_m(x); m = 1, 2, \dots, M\}$ are normalized and orthogonal (i.e., orthonormal)

$$\int_{-\infty}^{+\infty} g_i(\mathbf{x}) \cdot g_j(\mathbf{x}) \cdot \text{pdf}(\mathbf{x}) \cdot d\mathbf{x} = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases} \quad (3)$$

where $\text{pdf}(\mathbf{x})$ is the probability density function of x . A simple 1-D example of orthonormal polynomials with normally distributed \mathbf{x} can be expressed as [13], [26]

$$g_1(x) = 1 \quad g_2(x) = x \quad g_3(x) = \frac{1}{\sqrt{2}} \cdot (x^2 - 1) \dots \quad (4)$$

The 1-D case in (4) can be further extended to the 2-D case

$$\begin{aligned} g_1(x_1, x_2) &= 1 & g_2(x_1, x_2) &= x_1 \\ g_3(x_1, x_2) &= x_2 & g_4(x_1, x_2) &= \frac{1}{\sqrt{2}} \cdot (x_1^2 - 1) \\ g_5(x_1, x_2) &= x_1 \cdot x_2 & & \dots \end{aligned} \quad (5)$$

More details about orthonormal polynomials can be found in [26].

The performance model in (2), once available, can be applied to several important applications such as estimating parametric yield [17], extracting worst-case corner [18], optimizing circuit design [19]–[23], etc.

B. Least-Squares Fitting

In order to determine the performance model in (2), we need to find the model coefficients $\{\alpha_m; m = 1, 2, \dots, M\}$. Toward this goal, the traditional least-squares fitting method first generates a set of sampling points and then solves the model coefficients from the following linear equation [27]:

$$\mathbf{G} \cdot \boldsymbol{\alpha} = \mathbf{f} \quad (6)$$

where

$$\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_M]^T \quad (7)$$

$$\mathbf{f} = [f^{(1)} \ f^{(2)} \ \cdots \ f^{(K)}]^T \quad (8)$$

$$\mathbf{G} = \begin{bmatrix} g_1(\mathbf{x}^{(1)}) & g_2(\mathbf{x}^{(1)}) & \cdots & g_M(\mathbf{x}^{(1)}) \\ g_1(\mathbf{x}^{(2)}) & g_2(\mathbf{x}^{(2)}) & \cdots & g_M(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(\mathbf{x}^{(K)}) & g_2(\mathbf{x}^{(K)}) & \cdots & g_M(\mathbf{x}^{(K)}) \end{bmatrix}. \quad (9)$$

In (7)–(9), $\mathbf{x}^{(k)}$ and $f^{(k)}$ are the values of \mathbf{x} and $f(\mathbf{x})$ at the k th sampling point, respectively, and K represents the total number of sampling points. The number of sampling points (i.e., K) should be greater than the number of unknown coefficients (i.e., M). As such, the linear equation in (6) is overdetermined and the unknown model coefficients $\{\alpha_m; m = 1, 2, \dots, M\}$ are determined by solving its least-squares solution.

When the aforementioned least-squares fitting method is applied to high-dimensional performance modeling with many unknown model coefficients, it requires a large number of sampling points to form the overdetermined linear equation in (6). Note that each sampling point is generated by running an expensive transistor-level simulation. It, in turn, implies that the least-squares fitting approach can be extremely expensive for high-dimensional performance modeling.

C. Sparse Regression

Recently, several sparse regression algorithms have been developed to address the complexity issue associated with least-squares fitting [11]–[16], [28]. The key idea is not to solve an overdetermined linear equation. Instead, the unknown model coefficients are uniquely determined by solving an underdetermined linear equation. This goal is achieved by exploiting the fact that most model coefficients of a high-dimensional performance model are close to zero. In other words, the unknown model coefficients carry a unique sparse pattern. The sparse regression algorithms were particularly developed to solve these sparse coefficients from a small number of sampling points. As such, the simulation cost of generating the required sampling points is greatly reduced.

OMP is a commonly used sparse regression technique [13]. OMP applies a greedy algorithm to identify a set of important basis functions and use them to approximate the performance metric of interest. In particular, OMP repeatedly selects a single most important basis function at each iteration step. Such a basis function is selected by maximizing the correlation between the basis function and the current modeling residual. The aforementioned iteration steps continue until a sufficiently large number of basis functions are chosen to accurately approximate the performance metric. More details about OMP can be found in [13].

While sparse regression has been successfully applied to many practical applications, it still requires a large number of (e.g., 10^3) sampling points to fit a high-dimensional performance model [12], [13]. Therefore, it remains ill-equipped for modeling large-scale AMS circuits where running a single transistor-level simulation to generate one sampling point may

take a few days or even a few weeks. Motivated by this observation, we will propose a new BMF technique in this paper to further reduce the number of required simulation samples and, hence, the computational cost for large-scale performance modeling.

III. BAYESIAN MODEL FUSION

Unlike the traditional sparse regression approach that fits the sparse performance model based on the simulation data at a single stage only (e.g., post-layout simulation data), BMF attempts to identify the underlying pattern of the unknown model coefficients by reusing the early-stage data (e.g., schematic-level simulation data) in order to efficiently fit a late-stage (e.g., post-layout) performance model. In particular, BMF consists of the following two major steps: 1) statistically defining the prior knowledge for the unknown model coefficients based on the early-stage simulation data and 2) optimally determining the late-stage performance model by combining the prior knowledge and very few late-stage simulation samples. In this section, we will discuss the mathematical formulation of these two steps and highlight the novelty.

A. Prior Knowledge Definition

We consider two different performance models: 1) the early-stage model $f_E(\mathbf{x})$ and 2) the late-stage model $f_L(\mathbf{x})$

$$f_E(\mathbf{x}) \approx \sum_{m=1}^M \alpha_{E,m} \cdot g_m(\mathbf{x}) \quad (10)$$

$$f_L(\mathbf{x}) \approx \sum_{m=1}^M \alpha_{L,m} \cdot g_m(\mathbf{x}) \quad (11)$$

where $\{\alpha_{E,m}; m = 1, 2, \dots, M\}$ and $\{\alpha_{L,m}; m = 1, 2, \dots, M\}$ represent the early-stage and late-stage model coefficients, respectively. In (10) and (11), we assume that the early-stage model $f_E(x)$ and the late-stage model $f_L(\mathbf{x})$ share the same basis functions. More complicated cases where $f_E(\mathbf{x})$ and $f_L(\mathbf{x})$ are approximated by different basis functions will be further discussed in Section IV.

The early-stage model $f_E(\mathbf{x})$ is fitted from the early-stage simulation data. In practice, the early-stage simulation data are collected to validate the early-stage design, before we move to the next stage. For this reason, we should already know the early-stage model $f_E(\mathbf{x})$ before fitting the late-stage model $f_L(\mathbf{x})$. Namely, we assume that the early-stage model coefficients $\{\alpha_{E,m}; m = 1, 2, \dots, M\}$ are provided as the input to our proposed BMF method for late-stage performance modeling.

Given the early-stage model $f_E(\mathbf{x})$, we first extract the prior knowledge that can be used to facilitate efficient late-stage modeling. To this end, we propose to learn the underlying pattern of the late-stage model coefficients $\{\alpha_{L,m}; m = 1, 2, \dots, M\}$ based on the early-stage model coefficients $\{\alpha_{E,m}; m = 1, 2, \dots, M\}$. Remember that both the early-stage and late-stage models are fitted for the same performance metric of the same circuit. Their model coefficients should be similar. We statistically represent such prior knowledge as a probability density function (PDF) that is referred to as the prior distribution [24]. In particular, we consider the

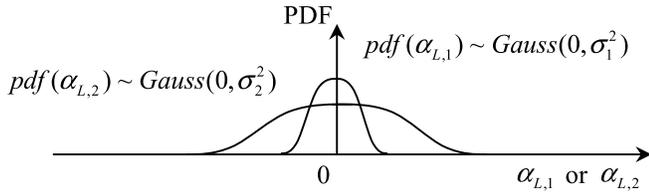


Fig. 1. Simple example of our proposed zero-mean prior distribution is shown for two model coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$. The coefficient $\alpha_{L,1}$ is possibly close to zero, since its prior distribution is narrowly peaked around zero. The coefficient $\alpha_{L,2}$ can possibly be far away from zero, since its prior distribution widely spreads over a large range.

following two different cases to define the prior distribution: 1) zero-mean prior distribution and 2) nonzero-mean prior distribution.

1) *Zero-Mean Prior Distribution*: If the early-stage model coefficient $\alpha_{E,m}$ has a large (or small) magnitude, it is likely that the late-stage model coefficient $\alpha_{L,m}$ also has a large (or small) magnitude. Such prior knowledge can be mathematically encoded as a zero-mean Gaussian distribution

$$\text{pdf}(\alpha_{L,m}) \sim \text{Gauss}(0, \sigma_m^2) \quad (m = 1, 2, \dots, M) \quad (12)$$

where the standard deviation σ_m is a parameter that encodes the magnitude information of the model coefficient $\alpha_{L,m}$. If the standard deviation σ_m is small, the prior distribution $\text{pdf}(\alpha_{L,m})$ is narrowly peaked around zero, implying that the coefficient $\alpha_{L,m}$ is possibly close to zero. Otherwise, if the standard deviation σ_m is large, the prior distribution $\text{pdf}(\alpha_{L,m})$ widely spreads over a large range and the coefficient $\alpha_{L,m}$ can possibly take a value that is far away from zero. Fig. 1 shows a simple example of our proposed zero-mean prior distribution for two model coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$ where σ_1 is small and σ_2 is large.

Given (12), we need to appropriately determine the standard deviation σ_m to fully specify the prior distribution $\text{pdf}(\alpha_{L,m})$. The value of σ_m should be optimized so that the probability distribution $\text{pdf}(\alpha_{L,m})$ correctly represents our prior knowledge. In other words, by appropriately choosing the value of σ_m , the prior distribution $\text{pdf}(\alpha_{L,m})$ should take a large value (i.e., a high probability) at the location where the actual late-stage model coefficient $\alpha_{L,m}$ occurs. However, we only know the early-stage model coefficient $\alpha_{E,m}$, instead of the late-stage model coefficient $\alpha_{L,m}$, at this moment. Remember that $\alpha_{E,m}$ and $\alpha_{L,m}$ are expected to be similar. Hence, the prior distribution $\text{pdf}(\alpha_{L,m})$ should also take a large value at $\alpha_{L,m} = \alpha_{E,m}$. Based on this criterion, the optimal prior distribution $\text{pdf}(\alpha_{L,m})$ can be found by maximizing the probability for $\alpha_{E,m}$ to occur

$$\max_{\sigma_m} \text{pdf}(\alpha_{L,m} = \alpha_{E,m}) \quad (m = 1, 2, \dots, M). \quad (13)$$

Namely, given the early-stage model coefficient $\alpha_{E,m}$, the optimal standard deviation σ_m is determined by the maximum likelihood estimation in (13).

To solve σ_m from (13), we consider the following first-order optimality condition:

$$\frac{d}{d\sigma_m} \text{pdf}(\alpha_{L,m} = \alpha_{E,m}) = 0 \quad (m = 1, 2, \dots, M). \quad (14)$$

Substituting (12) into (14) yields

$$\frac{1}{\sqrt{2\pi} \cdot \sigma_m} \cdot \exp\left(-\frac{\alpha_{E,m}^2}{2 \cdot \sigma_m^2}\right) \cdot \left(\frac{\alpha_{E,m}^2}{\sigma_m^3} - \frac{1}{\sigma_m}\right) = 0 \quad (m = 1, 2, \dots, M). \quad (15)$$

The optimal value of σ_m is equal to

$$\sigma_m = |\alpha_{E,m}| \quad (m = 1, 2, \dots, M). \quad (16)$$

Equation (16) reveals an important fact that the optimal standard deviation σ_m is simply equal to the absolute value of the early-stage model coefficient $\alpha_{E,m}$. This observation is consistent with our intuition. Namely, if the early-stage model coefficient $\alpha_{E,m}$ has a large (or small) magnitude, the late-stage model coefficient $\alpha_{L,m}$ should also have a large (or small) magnitude and, hence, the standard deviation σ_m should be large (or small), as shown in Fig. 1.

To complete the definition of the prior distribution for all late-stage model coefficients $\{\alpha_{L,m}; m = 1, 2, \dots, M\}$, we further assume that these coefficients are statistically independent. Their joint distribution is represented as

$$\begin{aligned} \text{pdf}(\alpha_L) &= \prod_{m=1}^M \text{pdf}(\alpha_{L,m}) \\ &= \frac{1}{(\sqrt{2\pi})^M \cdot \prod_{m=1}^M |\alpha_{E,m}|} \cdot \exp\left(-\sum_{m=1}^M \frac{\alpha_{L,m}^2}{2 \cdot \alpha_{E,m}^2}\right) \end{aligned} \quad (17)$$

where

$$\alpha_L = [\alpha_{L,1} \ \alpha_{L,2} \ \dots \ \alpha_{L,M}]^T \quad (18)$$

contains all late-stage model coefficients. The independence assumption in (17) simply implies that we do not know the correlation information among these coefficients as our prior knowledge. The correlation information will be learned from the late-stage simulation data, when the posterior distribution is calculated by the Bayesian inference in Section III-B.

2) *Nonzero-Mean Prior Distribution*: An alternative approach of prior definition is to construct a nonzero-mean Gaussian distribution for each late-stage model coefficient $\alpha_{L,m}$

$$\text{pdf}(\alpha_{L,m}) \sim \text{Gauss}(\alpha_{E,m}, \lambda^2 \cdot \alpha_{E,m}^2) \quad (m = 1, 2, \dots, M) \quad (19)$$

where $\alpha_{E,m}$ and $\lambda^2 \cdot \alpha_{E,m}^2$ are the mean and variance, respectively, and λ is a hyper-parameter that can be determined by cross-validation as will be discussed in detail in Section IV-D. Fig. 2 shows a simple example of our proposed nonzero-mean prior distribution for two model coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$ where $\alpha_{E,1}$ is small and $\alpha_{E,2}$ is large.

The prior distribution in (19) has a twofold meaning. First, the Gaussian distribution $\text{pdf}(\alpha_{L,m})$ is peaked at its mean value $\alpha_{L,m} = \alpha_{E,m}$, implying that the early-stage coefficient $\alpha_{E,m}$ and the late-stage coefficient $\alpha_{L,m}$ are likely to be similar. In other words, since the Gaussian distribution $\text{pdf}(\alpha_{L,m})$ exponentially decays with $(\alpha_{L,m} - \alpha_{E,m})^2$, it is unlikely to observe

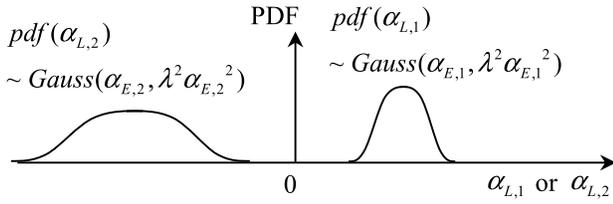


Fig. 2. Simple example of our proposed nonzero-mean prior distribution is shown for two model coefficients $\alpha_{L,1}$ and $\alpha_{L,2}$. The coefficient $\alpha_{L,1}$ possibly takes a small magnitude, since its prior distribution is narrowly peaked around a small value. The coefficient $\alpha_{L,2}$ possibly takes a large magnitude, since its prior distribution widely spreads over a large value.

a late-stage coefficient $\alpha_{L,m}$ that is extremely different from the early-stage coefficient $\alpha_{E,m}$. Second, the standard deviation of the prior distribution $\text{pdf}(\alpha_{L,m})$ is proportional to $|\alpha_{E,m}|$. It means that the absolute difference between the late-stage coefficient $\alpha_{L,m}$ and the early-stage coefficient $\alpha_{E,m}$ can be large (or small), if the magnitude of the early-stage coefficient $|\alpha_{E,m}|$ is large (or small). Restating in words, each late-stage coefficient $\alpha_{L,m}$ has been provided with a relatively equal opportunity to deviate from the corresponding early-stage coefficient $\alpha_{E,m}$.

Similar to (17), we again assume that all late-stage model coefficients $\{\alpha_{L,m}; m = 1, 2, \dots, M\}$ are statistically independent and their joint distribution is represented as

$$\text{pdf}(\alpha_L) = \prod_{m=1}^M \text{pdf}(\alpha_{L,m}) = \frac{1}{(\sqrt{2\pi} \cdot \lambda)^M \cdot \prod_{m=1}^M |\alpha_{E,m}|} \cdot \exp\left(-\sum_{m=1}^M \frac{(\alpha_{L,m} - \alpha_{E,m})^2}{2 \cdot \lambda^2 \cdot \alpha_{E,m}^2}\right). \quad (20)$$

For a given performance modeling problem, it is important to determine whether a nonzero-mean or zero-mean prior distribution is preferred. Intuitively, a nonzero-mean prior distribution provides stronger prior knowledge than a zero-mean prior distribution. The nonzero-mean prior distribution encodes both the sign and the magnitude information about the late-stage model coefficients, while the zero-mean prior distribution encodes the magnitude information only. From this point of view, a nonzero-mean prior distribution is preferred, if the early-stage and late-stage model coefficients are extremely close and, hence, the prior knowledge learned from the early-stage model coefficients is highly accurate. On the other hand, if the early-stage and late-stage model coefficients are substantially different, we should not pose an overly strong prior distribution and, hence, a zero-mean prior distribution is preferred in this case. The aforementioned problem of prior selection problem will be further discussed in detail in Section IV-D.

B. Maximum-A-Posteriori Estimation

Once the prior distribution $\text{pdf}(\alpha_L)$ is defined by (17) or (20), we will combine $\text{pdf}(\alpha_L)$ with K late-stage simulation samples $\{(\mathbf{x}^{(k)}, f_L^{(k)}); k = 1, 2, \dots, K\}$, where $\mathbf{x}^{(k)}$ and $f_L^{(k)}$ are the values of \mathbf{x} and $f_L(\mathbf{x})$ at the k th sampling point, respectively,

to solve the late-stage model coefficients α_L by maximum-a-posteriori (MAP) estimation. The key idea of MAP is to find the posterior distribution [24], i.e., the conditional PDF $\text{pdf}(\alpha_L | \mathbf{f}_L)$ where

$$\mathbf{f}_L = [f_L^{(1)} f_L^{(2)} \dots f_L^{(K)}]^T \quad (21)$$

contains all late-stage simulation samples that are collected. Intuitively, the posterior distribution $\text{pdf}(\alpha_L | \mathbf{f}_L)$ indicates the remaining uncertainty of α_L , after we observe K late-stage simulation samples. Here, since α_L is a random variable, it is described by a probability distribution, instead of a deterministic value. MAP attempts to find the optimal value of α_L to maximize the posterior distribution $\text{pdf}(\alpha_L | \mathbf{f}_L)$. Namely, it aims to find the solution α_L that is most likely to occur according to the posterior distribution.

Based on Bayes' theorem, the posterior distribution $\text{pdf}(\alpha_L | \mathbf{f}_L)$ is proportional to the prior distribution $\text{pdf}(\alpha_L)$ multiplied by the likelihood function $\text{pdf}(\mathbf{f}_L | \alpha_L)$ [24]

$$\text{pdf}(\alpha_L | \mathbf{f}_L) \propto \text{pdf}(\alpha_L) \cdot \text{pdf}(\mathbf{f}_L | \alpha_L). \quad (22)$$

The prior distribution $\text{pdf}(\alpha_L)$ is already defined by (17) or (20). To derive the likelihood function $\text{pdf}(\mathbf{f}_L | \alpha_L)$, we further assume that the error for the late-stage performance model $f_L(\mathbf{x})$ follows a zero-mean Gaussian distribution and, hence, (11) can be rewritten as:

$$f_L(\mathbf{x}) = \sum_{m=1}^M \alpha_{L,m} \cdot g_m(\mathbf{x}) + \varepsilon_L \quad (23)$$

where ε_L denotes the modeling error with the distribution

$$\text{pdf}(\varepsilon_L) = \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \cdot \exp\left(-\frac{\varepsilon_L^2}{2 \cdot \sigma_0^2}\right) \sim N(0, \sigma_0^2). \quad (24)$$

In (24), the standard deviation σ_0 controls the magnitude of the modeling error.

Given (23) and (24), since the modeling error at the k th simulation sample $(\mathbf{x}^{(k)}, f_L^{(k)})$ is simply one sampling point of the random variable ε_L , it follows the Gaussian distribution:

$$f_L^{(k)} - \sum_{m=1}^M \alpha_{L,m} \cdot g_m(\mathbf{x}^{(k)}) \sim N(0, \sigma_0^2). \quad (25)$$

Therefore, the probability of observing the k th sampling point is

$$\begin{aligned} \text{pdf}(f_L^{(k)} | \alpha_L) &= \frac{1}{\sqrt{2\pi} \cdot \sigma_0} \\ &\times \exp\left\{-\frac{1}{2 \cdot \sigma_0^2} \cdot \left[f_L^{(k)} - \sum_{m=1}^M \alpha_{L,m} \cdot g_m(\mathbf{x}^{(k)})\right]^2\right\}. \end{aligned} \quad (26)$$

Assume that all sampling points are independently generated, we can write the likelihood function $\text{pdf}(\mathbf{f}_L | \alpha_L)$ as

$$\text{pdf}(\mathbf{f}_L | \alpha_L) = \prod_{k=1}^K \text{pdf}(f_L^{(k)} | \alpha_L) \quad (27)$$

which is a multivariate Gaussian distribution. It is straightforward to prove that if both pdf(α_L) and pdf($\mathbf{f}_L|\alpha_L$) follow Gaussian distributions, the posterior distribution pdf($\alpha_L|\mathbf{f}_L$) also follows a Gaussian distribution [29]. In what follows, we will derive the posterior distribution pdf($\alpha_L|\mathbf{f}_L$) and, consequently, the MAP estimation of the late-stage model coefficients α_L for two different cases: 1) zero-mean prior distribution and 2) nonzero-mean prior distribution.

1) *Zero-Mean Prior Distribution*: Combining (17), (22), (26), and (27), we can derive the mean vector μ_L and the covariance matrix Σ_L of pdf($\alpha_L|\mathbf{f}_L$) as

$$\Sigma_L = \left[\sigma_0^{-2} \cdot \mathbf{G}^T \cdot \mathbf{G} + \text{diag}(\sigma_1^{-2}, \sigma_2^{-2}, \dots, \sigma_M^{-2}) \right]^{-1} \quad (28)$$

$$\mu_L = \sigma_0^{-2} \cdot \Sigma_L \cdot \mathbf{G}^T \cdot \mathbf{f}_L \quad (29)$$

where \mathbf{G} and \mathbf{f}_L are defined by (9) and (21), respectively, and $\text{diag}(\cdot)$ represents the operator to construct a diagonal matrix. The hyper-parameter σ_0 can be optimally determined by using the cross-validation technique discussed in Section IV-D. Since the Gaussian PDF pdf($\alpha_L|\mathbf{f}_L$) reaches its maximum at the mean value, the MAP solution α_L is equal to the mean vector μ_L

$$\alpha_L = \sigma_0^{-2} \cdot \Sigma_L \cdot \mathbf{G}^T \cdot \mathbf{f}_L. \quad (30)$$

2) *Nonzero-Mean Prior Distribution*: Combining (20), (22), (26), and (27), we can derive the mean vector μ_L and the covariance matrix Σ_L of pdf($\alpha_L|\mathbf{f}_L$) as

$$\Sigma_L = \left[\eta \cdot \text{diag}(\alpha_{E,1}^{-2}, \alpha_{E,2}^{-2}, \dots, \alpha_{E,M}^{-2}) + \mathbf{G}^T \cdot \mathbf{G} \right]^{-1} \quad (31)$$

$$\mu_L = \Sigma_L \cdot \left[\eta \cdot \text{diag}(\alpha_{E,1}^{-2}, \alpha_{E,2}^{-2}, \dots, \alpha_{E,M}^{-2}) \cdot \alpha_E + \mathbf{G}^T \cdot \mathbf{f}_L \right] \quad (32)$$

where

$$\alpha_E = [\alpha_{E,1} \quad \alpha_{E,2} \quad \dots \quad \alpha_{E,M}]^T \quad (33)$$

$$\eta = \frac{\sigma_0^2}{\lambda^2}. \quad (34)$$

Similar to (29), the MAP solution α_L is equal to the mean vector μ_L

$$\alpha_L = \Sigma_L \cdot \left[\eta \cdot \text{diag}(\alpha_{E,1}^{-2}, \alpha_{E,2}^{-2}, \dots, \alpha_{E,M}^{-2}) \cdot \alpha_E + \mathbf{G}^T \cdot \mathbf{f}_L \right]. \quad (35)$$

Studying (35) reveals an important observation that we only need to determine η , instead of the individual parameters σ_0 and λ , in order to find the MAP solution α_L . Similar to the case of zero-mean prior distribution, the hyper-parameter η can be optimally determined by using the cross-validation technique discussed in Section IV-D.

While the basic idea of prior knowledge definition and MAP estimation is illustrated in this section, several implementation issues must be carefully considered in order to make BMF of practical utility. These implementation details will be further discussed in the next section.

IV. IMPLEMENTATION ISSUES

To make the proposed BMF method of practical utility, four implementation issues, including: 1) prior mapping; 2) missing prior knowledge; 3) fast solver; and 4) prior and hyper-parameter selection, must be carefully considered. In this section, we will discuss these implementation issues in detail.

A. Prior Mapping

In Section III-A, we assume that the early-stage model $f_E(\mathbf{x})$ and the late-stage model $f_L(\mathbf{x})$ share the same basis functions, as shown in (10) and (11). The prior distribution pdf(α_L) is then defined accordingly as (17) or (20). However, the aforementioned assumption does not always hold in practice. In this paper, we consider two important scenarios where this assumption is not valid. First, additional basis functions may be required in the later stage due to the post-layout extraction of multifinger transistors. In this case, we can still learn the design knowledge from the early stage by prior mapping. Second, additional basis functions may be required in the late stage where the prior information for these extra basis functions is not available. In this case, we must set up the proposed BMF framework with missing prior knowledge. In this section, we focus on the first scenario, while the second scenario will be further discussed in Section IV-B.

We consider the schematic-level circuit as the early-stage design and the post-layout circuit as the late-stage design. In this scenario, multifinger transistors lead to different basis functions in these two stages. Due to the extra random variables modeling device mismatches, we have to apply additional basis functions containing these new random variables for performance modeling.

For illustration purpose, we consider a simple differential pair example. Its input offset voltage (V_{OS}) is strongly dependent on the threshold voltage of the two input transistors. Here, we denote the random variables modeling the threshold voltage mismatches of the first and second input transistors as x_1 and x_2 , respectively. The schematic-level performance model of the input offset voltage can be expressed as

$$f_E(x_1, x_2) \approx \alpha_{E,1} \cdot x_1 + \alpha_{E,2} \cdot x_2 + \alpha_{E,3} \quad (36)$$

where $\alpha_{E,1}$, $\alpha_{E,2}$, and $\alpha_{E,3}$ denote the schematic-level model coefficients.

We further assume that each input transistor contains two fingers at the post-layout stage. After post-layout extraction, the threshold voltage mismatch of each finger should be modeled as an independent random variable. Without loss of generality, we define $x_{1,1}$ and $x_{1,2}$ as the random variables associated with the two fingers of the first input transistor, and $x_{2,1}$ and $x_{2,2}$ as the random variables associated with the two fingers of the second input transistor. The post-layout performance model of the input offset voltage can be expressed as

$$\begin{aligned} f_L(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}) \approx & \alpha_{L,1,1} \cdot x_{1,1} + \alpha_{L,1,2} \cdot x_{1,2} \\ & + \alpha_{L,2,1} \cdot x_{2,1} + \alpha_{L,2,2} \cdot x_{2,2} + \alpha_{L,3} \end{aligned} \quad (37)$$

where $\alpha_{L,1,1}, \alpha_{L,1,2}, \alpha_{L,2,1}, \alpha_{L,2,2}$, and $\alpha_{L,3}$ denote the post-layout model coefficients.

In general, we define the following independent random variables for the post-layout stage:

$$\mathbf{x}^* = [x_{1,1} \cdots x_{1,W_1} \ x_{2,1} \cdots x_{2,W_2} \cdots x_{R,1} \cdots x_{R,W_R}]^T \quad (38)$$

where W_r denotes the number of fingers associated with the schematic-level random variable x_r ($r = 1, 2, \dots, R$). We further define the post-layout performance model $f_L(\mathbf{x}^*)$ as

$$f_L(\mathbf{x}^*) \approx \sum_{m=1}^M \sum_{t=1}^{T_m} \alpha_{L,m,t} \cdot g_{m,t}(\mathbf{x}^*) \quad (39)$$

where $\{\alpha_{L,m,t}; m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ contains the post-layout model coefficients corresponding to the basis functions $\{g_{m,t}(\mathbf{x}^*); m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$. Equation (39) is similar to (11) but with a different representation. In (39), due to the multifinger devices, each schematic-level basis function $g_m(\mathbf{x})$ ($m = 1, 2, \dots, M$) is mapped to a set of post-layout basis functions $\{g_{m,t}(\mathbf{x}^*); t = 1, 2, \dots, T_m\}$. The post-layout basis functions $\{g_{m,t}(\mathbf{x}^*); m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ are orthonormal. Furthermore, based on the completeness condition [26], the basis function set is permutation-invariant. It means that for any vector \mathbf{x}_s^* obtained by permuting the variables in \mathbf{x}^* , the new basis function set $\{g_{m,t}(\mathbf{x}_s^*); m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ should be identical to the original set $\{g_{m,t}(\mathbf{x}^*); m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$.

To intuitively illustrate the permutation-invariant property, we consider the simple example in (37) where we have

$$\begin{aligned} \mathbf{x}^* &= [x_{1,1} \ x_{1,2} \ x_{2,1} \ x_{2,2}]^T \\ g_{1,1}(\mathbf{x}^*) &= \mathbf{x}_1^* = x_{1,1} \quad g_{1,2}(\mathbf{x}^*) = \mathbf{x}_2^* = x_{1,2} \\ g_{2,1}(\mathbf{x}^*) &= \mathbf{x}_3^* = x_{2,1} \quad g_{2,2}(\mathbf{x}^*) = \mathbf{x}_4^* = x_{2,2}. \end{aligned} \quad (40)$$

$$g_{2,1}(\mathbf{x}^*) = \mathbf{x}_3^* = x_{2,1} \quad g_{2,2}(\mathbf{x}^*) = \mathbf{x}_4^* = x_{2,2}. \quad (41)$$

In this example, if we consider the following permuted vector:

$$\mathbf{x}_s^* = [x_{1,2} \ x_{1,1} \ x_{2,1} \ x_{2,2}]^T \quad (42)$$

we can construct the following basis functions:

$$\begin{aligned} g_{1,1}(\mathbf{x}_s^*) &= \mathbf{x}_{s,1}^* = x_{1,2} \quad g_{1,2}(\mathbf{x}_s^*) = \mathbf{x}_{s,2}^* = x_{1,1} \\ g_{2,1}(\mathbf{x}_s^*) &= \mathbf{x}_{s,3}^* = x_{2,1} \quad g_{2,2}(\mathbf{x}_s^*) = \mathbf{x}_{s,4}^* = x_{2,2}. \end{aligned} \quad (43)$$

It is straightforward to verify that the two sets of basis functions in (41) and (43) are identical.

The schematic-level model coefficients $\{\alpha_{E,m}; m = 1, 2, \dots, M\}$ in (10) are already known, the goal of our prior mapping is to determine the prior distribution for the post-layout model coefficients $\{\alpha_{L,m,t}; m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ in (39). To this end, we first define the following schematic-level multifinger representation $h_E(\mathbf{x}^*)$:

$$h_E(\mathbf{x}^*) \approx \sum_{m=1}^M \sum_{t=1}^{T_m} \beta_{E,m,t} \cdot g_{m,t}(\mathbf{x}^*) \quad (44)$$

where $\{\beta_{E,m,t}; m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ stands for the model coefficients. The model $h_E(\mathbf{x}^*)$ in (44) captures the same schematic-level performance function as $f_E(\mathbf{x})$ in (10), and shares the same basis functions as $f_L(\mathbf{x}^*)$ in (39).

Given that (10) and (44) are different representations of the same performance model, they should capture the same performance variability

$$\text{VAR}[\alpha_{E,m}^2 \cdot g_m(\mathbf{x})] = \text{VAR}\left[\sum_{t=1}^{T_m} \beta_{E,m,t}^2 \cdot g_{m,t}(\mathbf{x}^*)\right] \quad (m = 1, 2, \dots, M) \quad (45)$$

where $\text{VAR}(\cdot)$ denotes the variance of a random variable. Since the basis functions in (45) are orthonormal, we have

$$\alpha_{E,m}^2 = \sum_{t=1}^{T_m} \beta_{E,m,t}^2 \quad (m = 1, 2, \dots, M). \quad (46)$$

We further assume that different fingers of the same device have the same impact on performance variability. In practice, this assumption does not always hold, because different fingers may be subject to different systematic variations. However, these systematic variations cannot be easily modeled for prior definition at the schematic level without knowing the layout details; instead, they will be taken into account when calculating the posterior distribution based on the post-layout simulation data.

Given the aforementioned assumption and the permutation-invariant property of basis functions, we can derive the following equation for the model coefficients in (44):

$$\beta_{E,m,1} = \beta_{E,m,2} = \cdots = \beta_{E,m,T_m} \quad (m = 1, 2, \dots, M). \quad (47)$$

Combining (46) and (47), we have

$$\begin{aligned} \alpha_{E,m}^2 &= T_m \cdot \beta_{E,m,1}^2 = T_m \cdot \beta_{E,m,2}^2 \\ &= \cdots = T_m \cdot \beta_{E,m,T_m}^2 \quad (m = 1, 2, \dots, M). \end{aligned} \quad (48)$$

If the basis functions $\{g_{m,t}(\mathbf{x}^*); m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ are properly defined, the model coefficients $\alpha_{E,m}$ ($m = 1, 2, \dots, M$) and $\{\beta_{E,m,t}; t = 1, 2, \dots, T_m\}$ should have the same sign, yielding

$$\begin{aligned} \beta_{E,m,1} &= \beta_{E,m,2} = \cdots = \beta_{E,m,T_m} \\ &= \alpha_{E,m} / \sqrt{T_m} \quad (m = 1, 2, \dots, M). \end{aligned} \quad (49)$$

Since the schematic-level multifinger representation $h_E(\mathbf{x}^*)$ in (44) and the post-layout model $f_L(\mathbf{x}^*)$ in (39) are associated with the same performance metric and share the same basis functions, we expect that the early-stage model coefficients $\{\beta_{E,m,t}; m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ are similar to the late-stage model coefficients $\{\alpha_{L,m,t}; m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$. Hence, once the early-stage model coefficients $\{\beta_{E,m,t}; m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ are calculated by (49), the prior distribution of the late-stage model coefficients $\{\alpha_{L,m,t}; m = 1, 2, \dots, M; t = 1, 2, \dots, T_m\}$ can be defined by following (12)–(20).

B. Missing Prior Knowledge

In this section, we consider the scenario where additional basis functions are required in the late stage and the prior information for these extra basis functions is not available. In practice, this scenario can happen because the early-stage model does not necessarily capture all detailed behaviors of

a circuit. For instance, it is well-known that layout parasitics will be added to the post-layout netlist (i.e., late stage) during layout extraction. The variations of these parasitics must be modeled by a number of new random variables that are completely ignored at the schematic level (i.e., early stage). The late-stage post-layout model $f_L(\mathbf{x})$ should contain additional basis functions corresponding to the new random variables that are not found from the early-stage schematic model $f_E(\mathbf{x})$. In this case, the early-stage model $f_E(\mathbf{x})$ does not carry any prior knowledge about the late-stage model coefficients associated with these additional basis functions. In other words, the prior knowledge for these late-stage model coefficients is missing.

To appropriately handle the scenario with missing prior knowledge, we revisit the prior distribution $\text{pdf}(\alpha_{L,m})$ defined in Section III-A. In particular, we consider the following two different cases: 1) zero-mean prior distribution and 2) nonzero-mean prior distribution.

1) *Zero-Mean Prior Distribution*: As mentioned in Section III-A, the standard deviation σ_m of the Gaussian distribution $\text{pdf}(\alpha_{L,m})$ in (12) encodes the magnitude information of the late-stage model coefficient $\alpha_{L,m}$. If there is no prior knowledge available for $\alpha_{L,m}$, it implies that the late-stage model coefficient $\alpha_{L,m}$ can possibly take any value with equal probability. Hence, the standard deviation σ_m should be set to $+\infty$

$$\sigma_m = +\infty \quad (50)$$

so that the prior distribution is nearly constant over a wide range. Note that when calculating the posterior distribution in (28) and (29), only the value of σ_m^{-1} is needed. Hence, the infinite standard deviation in (50) would not cause any numerical problem for solving the late-stage model coefficients.

2) *Nonzero-Mean Prior Distribution*: In this case, the magnitude information of the late-stage model coefficient $\alpha_{L,m}$ is encoded by the early-stage model coefficient $\alpha_{E,m}$ using the Gaussian distribution $\text{pdf}(\alpha_{L,m})$ in (19). Note that the standard deviation of $\text{pdf}(\alpha_{L,m})$ is also controlled by $\alpha_{E,m}$ as shown in (19). Similar to the zero-mean prior case, we set

$$\alpha_{E,m} = +\infty \quad (51)$$

so that the prior distribution is almost uniformly distributed over a wide range. We rewrite (32) as

$$\boldsymbol{\mu}_L = \boldsymbol{\Sigma}_L \cdot \left[\eta \cdot \left[\alpha_{E,1}^{-1} \ \alpha_{E,2}^{-1} \ \cdots \ \alpha_{E,M}^{-1} \right]^T + \mathbf{G}^T \cdot \mathbf{f}_L \right]. \quad (52)$$

Therefore, only the value of $\alpha_{E,m}^{-1}$ is needed to calculate the posterior distribution in (31) and (52), similar to the zero-mean prior case.

C. Fast Solver

The late-stage model coefficients are solved by MAP estimation in (28)–(35) where a linear equation must be solved. In practice, if a large number of basis functions are used to model the circuit performance function, a large number of model coefficients must be solved in (30) or (35), thereby resulting in expensive computational cost. To address this issue, we propose a novel low-rank update algorithm to efficiently solve the linear equation posted by MAP estimation.

Studying (28)–(35), we observe that the matrix $\mathbf{G} \in \mathfrak{R}^{K \times M}$ has more columns than rows, and thus $\mathbf{G}^T \cdot \mathbf{G} \in \mathfrak{R}^{M \times M}$ is not full-rank. The rank of $\mathbf{G}^T \cdot \mathbf{G}$ is equal to K (e.g., 10^2 – 10^3) is substantially less than its size M (e.g., 10^3 – 10^5). Therefore, instead of directly solving the linear equation by a direct solver (e.g., Cholesky decomposition [30]), we apply a computationally efficient low-rank update to find all model coefficients based on the Sherman–Morrison–Woodbury formula [30]. In what follows, we derive our proposed low-rank solver for the following two cases: 1) zero-mean prior distribution and 2) nonzero-mean prior distribution.

1) *Zero-Mean Prior Distribution*: We rewrite the matrix $\boldsymbol{\Sigma}_L$ in (28) as

$$\boldsymbol{\Sigma}_L = \left[\sigma_0^{-2} \cdot \mathbf{G}^T \cdot \mathbf{G} + \mathbf{A}_Z \right]^{-1} = \mathbf{A}_Z^{-1} - \mathbf{A}_Z^{-1} \cdot \mathbf{G}^T \cdot \left(\sigma_0^2 \cdot \mathbf{I} + \mathbf{G} \cdot \mathbf{A}_Z^{-1} \cdot \mathbf{G}^T \right)^{-1} \cdot \mathbf{G} \cdot \mathbf{A}_Z^{-1} \quad (53)$$

where $\mathbf{I} \in \mathfrak{R}^{K \times K}$ denotes an identity matrix and $\mathbf{A}_Z \in \mathfrak{R}^{M \times M}$ is diagonal

$$\mathbf{A}_Z = \text{diag}(\sigma_1^{-2}, \sigma_2^{-2}, \dots, \sigma_M^{-2}). \quad (54)$$

Substituting (53) into (30) yields

$$\boldsymbol{\alpha}_L = \sigma_0^{-2} \cdot \mathbf{A}_Z^{-1} \cdot \mathbf{G}^T \cdot \mathbf{f}_L - \sigma_0^{-2} \cdot \mathbf{A}_Z^{-1} \cdot \mathbf{G}^T \times \left(\sigma_0^2 \cdot \mathbf{I} + \mathbf{G} \cdot \mathbf{A}_Z^{-1} \cdot \mathbf{G}^T \right)^{-1} \cdot \mathbf{G} \cdot \mathbf{A}_Z^{-1} \cdot \mathbf{G}^T \cdot \mathbf{f}_L. \quad (55)$$

In (55), the inverse matrix \mathbf{A}_Z^{-1} can be easily calculated since \mathbf{A}_Z is diagonal. On the other hand, the matrix $\sigma_0^2 \cdot \mathbf{I} + \mathbf{G} \cdot \mathbf{A}_Z^{-1} \cdot \mathbf{G}^T \in \mathfrak{R}^{K \times K}$ is substantially smaller than the matrix $\boldsymbol{\Sigma}_L \in \mathfrak{R}^{M \times M}$. Hence, solving the linear equation in (55) is more computationally efficient than directly solving (30).

2) *Nonzero-Mean Prior Distribution*: Similar to the previous case, we rewrite the matrix $\boldsymbol{\Sigma}_L$ in (31) as

$$\boldsymbol{\Sigma}_L = \left[\eta \cdot \mathbf{A}_N + \mathbf{G}^T \cdot \mathbf{G} \right]^{-1} = \eta^{-1} \cdot \mathbf{A}_N^{-1} - \eta^{-1} \cdot \mathbf{A}_N^{-1} \cdot \mathbf{G}^T \left(\eta \cdot \mathbf{I} + \mathbf{G} \cdot \mathbf{A}_N^{-1} \cdot \mathbf{G}^T \right)^{-1} \cdot \mathbf{G} \cdot \mathbf{A}_N^{-1} \quad (56)$$

where $\mathbf{A}_N \in \mathfrak{R}^{M \times M}$ is diagonal

$$\mathbf{A}_N = \text{diag}(\alpha_{E,1}^{-2}, \alpha_{E,2}^{-2}, \dots, \alpha_{E,M}^{-2}). \quad (57)$$

Substituting (56) into (35) yields

$$\boldsymbol{\alpha}_L = \eta^{-1} \cdot \mathbf{A}_N^{-1} \cdot (\eta \cdot \mathbf{A}_N \cdot \boldsymbol{\alpha}_E + \mathbf{G}^T \cdot \mathbf{f}_L) - \eta^{-1} \cdot \mathbf{A}_N^{-1} \cdot \mathbf{G}^T \times \left(\eta \cdot \mathbf{I} + \mathbf{G} \cdot \mathbf{A}_N^{-1} \cdot \mathbf{G}^T \right)^{-1} \cdot \mathbf{G} \cdot \mathbf{A}_N^{-1} \cdot (\eta \cdot \mathbf{A}_N \cdot \boldsymbol{\alpha}_E + \mathbf{G}^T \cdot \mathbf{f}_L). \quad (58)$$

Since the matrix $\eta \cdot \mathbf{I} + \mathbf{G} \cdot \mathbf{A}_N^{-1} \cdot \mathbf{G}^T \in \mathfrak{R}^{K \times K}$ is substantially smaller than the matrix $\boldsymbol{\Sigma}_L \in \mathfrak{R}^{M \times M}$, solving the linear equation in (58) is more computationally efficient than directly solving (35).

It is important to note that the proposed low-rank update in (55) and (58) is able to find the exact solution of $\boldsymbol{\alpha}_L$ without any approximation, while substantially reducing the computational cost. As will be demonstrated by the experimental

Algorithm 1 BMF

1. Starting from the early-stage performance model $f_E(\mathbf{x})$ in (10), define the prior distribution for the late-stage model coefficients based on (17) (i.e., zero-mean Gaussian distribution) or (20) (i.e., nonzero-mean Gaussian distribution). The hyper-parameter (i.e. σ_0 or η) should be optimally determined by cross-validation.
2. For the additional late-stage basis functions posed by multi-finger transistors, define the prior distribution for the late-stage model coefficients by prior mapping discussed in Section IV.A.
3. For the additional late-stage basis functions without prior information, define the prior distribution for the late-stage model coefficients by using (50) or (51).
4. Collect K late-stage simulation samples $\{(\mathbf{x}^{(k)}, f_L^{(k)}); k = 1, 2, \dots, K\}$.
5. Use the fast solver to calculate the late-stage model coefficients by MAP estimation based on (55) or (58).

results in Section V, the proposed fast solver achieves up to $600\times$ runtime speed-up compared to the conventional solver based on Cholesky decomposition [30].

D. Prior and Hyper-Parameter Selection

As mentioned in Section III, we must appropriately choose the prior distribution (i.e., zero-mean or nonzero-mean Gaussian distribution) and the corresponding hyper-parameter value (i.e., σ_0 or η), when applying BMF. The objective here is to find the optimal prior distribution and hyper-parameter value to minimize the modeling error. Toward this goal, we need to estimate the modeling error for different prior distributions and hyper-parameter values, and then choose the optimal setting with minimal error.

To quantitatively estimate the modeling error for a given prior distribution and hyper-parameter value, we adopt the idea of N -fold cross-validation [24]. In particular, we partition the entire data set into N nonoverlapping groups. Modeling error is estimated from N independent runs. In each run, one of the N groups is selected to estimate the modeling error and all other groups are used to calculate the model coefficients. Since the training data for coefficient calculation and the testing data for error estimation are not overlapped, overfitting can be easily detected. Furthermore, different groups are used for error estimation in different runs. As such, each run gives an estimated error value $e_n (n = 1, 2, \dots, N)$ based on a unique group of testing data. The final modeling error is computed as $e = (e_1 + e_2 + \dots + e_N)/N$.

E. Summary

Algorithm 1 summarizes the major steps of our proposed BMF method. It consists of two major components: 1) prior distribution definition and 2) MAP estimation. BMF appropriately determines the hyper-parameter (i.e., σ_0 or η) based on cross-validation. The hyper-parameter value controls the weight of the prior information when the late-stage model coefficients are solved by MAP estimation. If the prior information is not highly accurate, a small weight should be assigned to it so that the late-stage model coefficients are not biased by the inaccurate prior information.

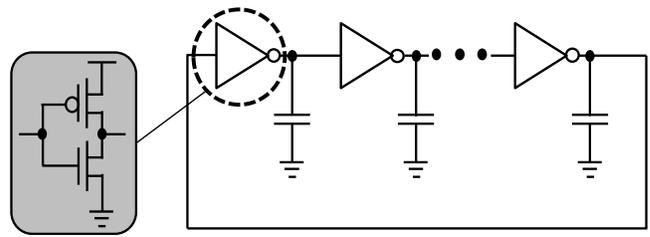


Fig. 3. Simplified circuit schematic is shown for an RO designed in a commercial 32 nm CMOS SOI process.

V. NUMERICAL EXAMPLES

In this section, two circuit examples designed in a commercial 32 nm CMOS SOI process are used to demonstrate the efficacy of the proposed BMF approach. The objective here is to build late-stage performance models for the two circuits, where the schematic stage is considered as the early stage and the post-layout stage is considered as the late stage. When applying BMF, we use the schematic-level performance model to define our prior knowledge for post-layout performance modeling. The schematic-level performance model is fitted by applying the OMP algorithm to 3000 random samples generated from transistor-level Monte Carlo simulation at the schematic stage.

For testing and comparison purposes, four different performance modeling techniques are implemented: 1) the traditional sparse regression approach based on OMP [13]; 2) the BMF method with zero-mean prior distribution (BMF-ZM); 3) the BMF method with nonzero-mean prior distribution (BMF-NZM); and 4) the proposed BMF method with prior selection (BMF-PS). Here the OMP algorithm is selected for comparison because it is one of the state-of-the-art techniques in the literature. The OMP algorithm does not consider any prior information from the schematic stage. The BMF-ZM and BMF-NZM methods rely on preselected prior distributions; they are compared against the BMF-PS method in order to highlight the benefit of prior selection.

In our experiments, two nonoverlapping data sets, referred to as the training set and the testing set, respectively, are generated from post-layout transistor-level Monte Carlo simulation with random sampling. The training set is used for coefficient fitting, including cross-validation as illustrated in Algorithm 1. The testing set contains 300 independent random samples that are used to estimate the modeling error. All numerical experiments are run on a 2.53 GHz Linux server with 16 GB memory.

A. Ring Oscillator

Fig. 3 shows the simplified circuit schematic of an RO. In this example, there are totally 7177 independent random variables to model device-level process variations, including both interdie variations and random mismatches at the post-layout stage. Our objective is to approximate three post-layout performance metrics (i.e., power, phase noise, and frequency) as linear functions of these random variables.

To build the performance models of interest, we collect a number of Monte Carlo samples for power, phase noise,

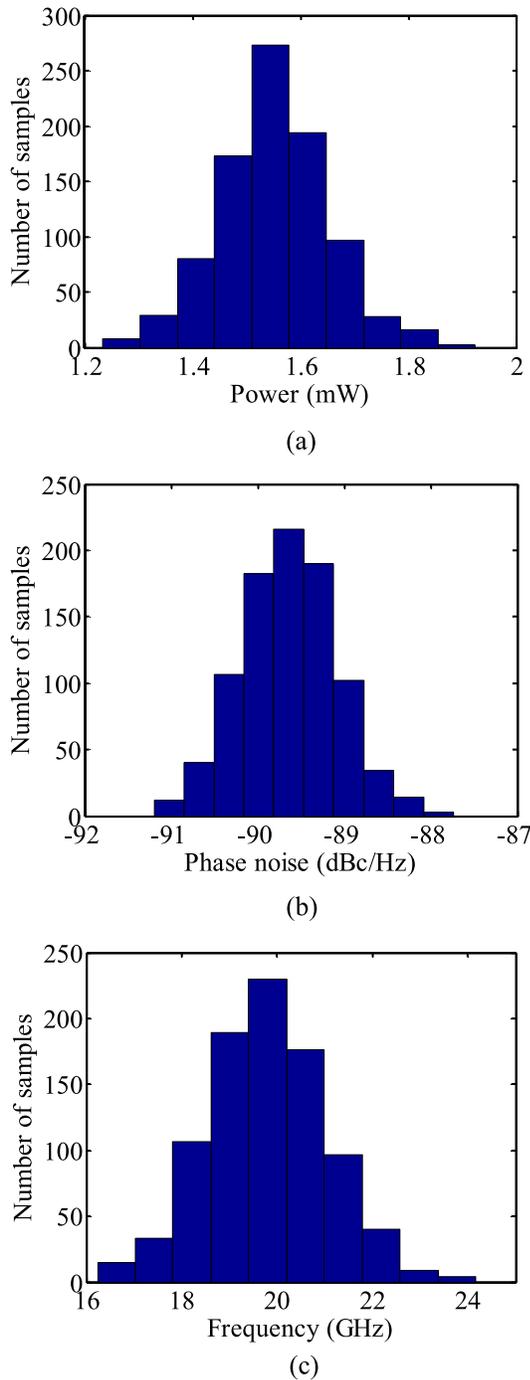


Fig. 4. Histograms of post-layout simulation samples are shown for (a) power, (b) phase noise, and (c) frequency of the RO.

and frequency by running post-layout simulation. Fig. 4 shows the histograms of these simulation samples. Based on the simulation data, we fit the performance models using four different approaches: 1) OMP; 2) BMF-ZM; 3) BMF-NZM; and 4) BMF-PS.

Tables I–III summarize the relative modeling error as a function of the number of post-layout training samples. Here the relative modeling error is defined as

$$\left\| \tilde{\mathbf{f}}_L - \mathbf{f}_L \right\|_2 / \left\| \mathbf{f}_L \right\|_2 \quad (59)$$

TABLE I
RELATIVE MODELING ERROR (%) OF POWER FOR RO

Number of samples	OMP	BMF-ZM	BMF-NZM	BMF-PS
100	2.7187	0.7466	0.5558	0.5558
200	1.3645	0.6032	0.5253	0.5253
300	1.0390	0.5411	0.5078	0.5110
400	0.9644	0.5055	0.4922	0.4925
500	0.9281	0.4848	0.4810	0.4848
600	0.9049	0.4719	0.4716	0.4736
700	0.8879	0.4622	0.4636	0.4640
800	0.8738	0.4544	0.4567	0.4546
900	0.8671	0.4501	0.4525	0.4518

TABLE II
RELATIVE MODELING ERROR (%) OF PHASE NOISE FOR RO

Number of samples	OMP	BMF-ZM	BMF-NZM	BMF-PS
100	0.2871	0.1033	0.0974	0.0982
200	0.1594	0.1006	0.0924	0.0925
300	0.1289	0.0984	0.0909	0.0909
400	0.1175	0.0948	0.0887	0.0887
500	0.1145	0.0916	0.0869	0.0869
600	0.1110	0.0893	0.0857	0.0857
700	0.1087	0.0876	0.0848	0.0848
800	0.1068	0.0863	0.0839	0.0839
900	0.1053	0.0849	0.0830	0.0830

TABLE III
RELATIVE MODELING ERROR (%) OF FREQUENCY FOR RO

Number of samples	OMP	BMF-ZM	BMF-NZM	BMF-PS
100	1.8346	0.5800	0.6664	0.6069
200	1.0677	0.4080	0.4905	0.4080
300	0.9081	0.3311	0.3674	0.3311
400	0.8592	0.2954	0.3062	0.2954
500	0.8166	0.2781	0.2841	0.2779
600	0.7948	0.2672	0.2705	0.2672
700	0.7794	0.2589	0.2609	0.2590
800	0.7667	0.2530	0.2544	0.2530
900	0.7471	0.2487	0.2500	0.2487

where $\tilde{\mathbf{f}}_L$ and \mathbf{f}_L are two vectors containing the predicted and actual post-layout performance values, respectively. The relative modeling error is averaged from 50 repeated runs with different training and testing sets.

Note that the training set used for prior selection and the testing set used for error evaluation are different. Therefore, the prior selection step may not choose the optimal prior distribution that minimizes the modeling error over the testing set. In our experiments, BMF-ZM, BMF-NZM, and BMF-PS share the same training and testing sets and, therefore, the modeling error of BMF-PS should be equal to the modeling error of either BMF-ZM or BMF-NZM for a single run. The results reported in Tables I–III, however, are averaged from 50 repeated runs. Hence, the error values of BMF-PS are not always identical to those of either BMF-ZM or BMF-NZM.

Studying Tables I–III reveals two important observations. First, the modeling error decreases as the number of simulation samples increases. Given the same number of samples, the proposed BMF-PS achieves significantly higher accuracy than OMP. Second, BMF-ZM is less accurate than BMF-NZM for power, but is more accurate than BMF-NZM for frequency.

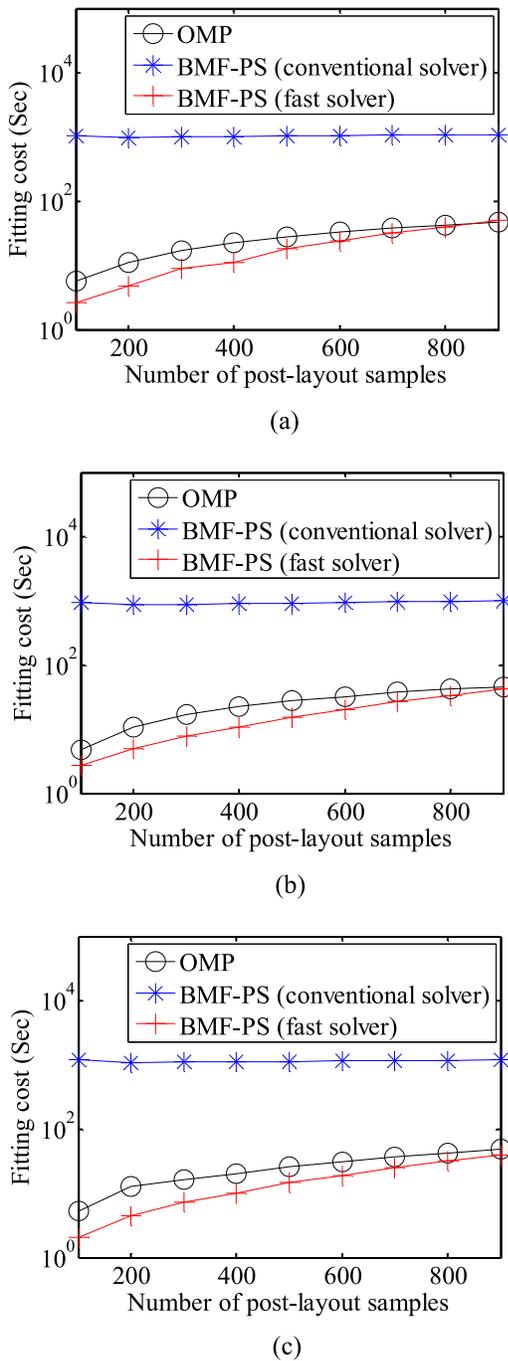


Fig. 5. Fitting cost of different modeling approaches is shown for three performance metrics. (a) Power. (b) Phase noise. (c) Frequency of the RO.

In other words, the optimal prior distribution can vary from case to case in practice. In this example, BMF-PS is able to achieve superior modeling accuracy over all other approaches.

To study the performance modeling cost for different approaches, we partition the total modeling cost into two portions: 1) simulation cost (i.e., the cost of running post-layout transistor-level simulation to generate all the samples in the training set) and 2) fitting cost (i.e., the cost of solving all unknown post-layout model coefficients). Fig. 5 shows

TABLE IV
RELATIVE MODELING ERROR AND COST FOR RO

	OMP	BMF-PS (fast solver)
# of post-layout training samples	900	100
Modeling error for power	0.8671%	0.5558%
Modeling error for phase noise	0.1053%	0.0982%
Modeling error for frequency	0.7471%	0.6069%
Simulation cost (Hour)	12.58	1.40
Fitting cost (Second)	140.31	7.42
Total modeling cost (Hour)	12.62	1.40

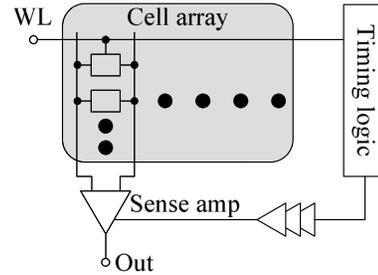


Fig. 6. Simplified circuit schematic for an SRAM read path designed in a commercial 32 nm CMOS SOI process.

the fitting cost for three different algorithms: 1) OMP; 2) BMF-PS with the conventional solver based on Cholesky decomposition [30]; and 3) BMF-PS with our proposed fast solver based on low-rank update. In this example, our fast solver achieves up to $600\times$ runtime speed-up over the conventional solver. As the problem size further increases, the efficacy of the proposed fast solver is expected to be more pronounced.

Table IV further compares the performance modeling error and cost for OMP and BMF-PS with fast solver. Note that the overall modeling cost is dominated by the simulation cost. In this example, BMF-PS achieves $9\times$ runtime speed-up over OMP with superior accuracy.

B. SRAM Read Path

Fig. 6 is the simplified circuit schematic of an SRAM read path. In this example, each SRAM column contains 128 bit cells. There are totally 66 117 independent random variables to model device-level process variations. The read delay from the wordline to the output of the sense amplifier (Out) is our circuit performance of interest. It is approximated as a linear function of the 66 117 random variables.

To build the performance model for read delay, we collect a number of Monte Carlo samples by running post-layout simulation. Fig. 7 shows the histogram of these simulation samples. Based on these simulation data, we fit the performance model using four different approaches: 1) OMP; 2) BMF-ZM; 3) BMF-NZM; and 4) BMF-PS.

Table V summarizes the relative modeling error as a function of the number of post-layout training samples, where the relative modeling error is defined as in (59). The relative modeling error is averaged from 50 repeated runs with different training and testing sets. Similar to the RO example, two important observations can be made by studying Table V. First, given the same number of post-layout training samples,

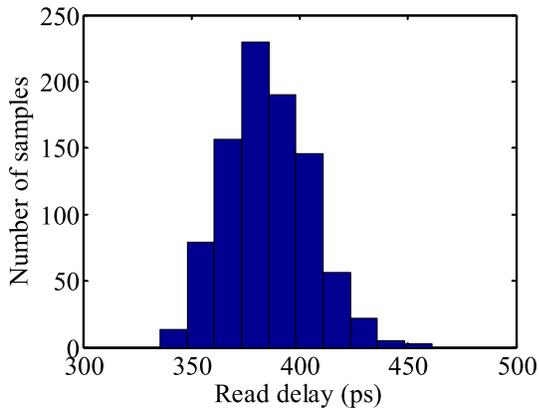


Fig. 7. Histogram of post-layout simulation samples for read delay of the SRAM read path.

TABLE V
RELATIVE MODELING ERROR (%) OF READ
DELAY FOR SRAM READ PATH

Number of samples	OMP	BMF-ZM	BMF-NZM	BMF-PS
100	3.2320	1.0592	1.1130	1.0804
200	1.8538	0.9645	0.9512	0.9630
300	1.3691	0.9055	0.8643	0.8791
400	1.1330	0.8573	0.8141	0.8250
500	1.0669	0.8156	0.7833	0.7916
600	1.0319	0.7777	0.7582	0.7609
700	1.0174	0.7455	0.7323	0.7344
800	1.0081	0.7216	0.7159	0.7174
900	0.9974	0.6986	0.6958	0.6989

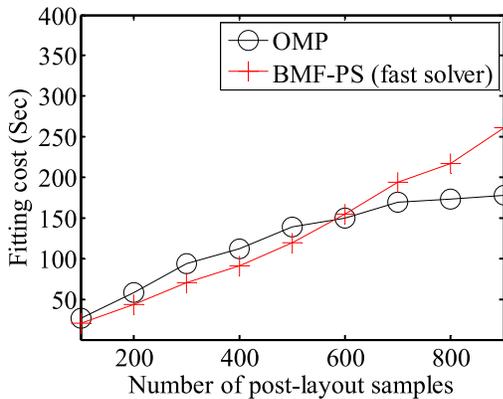


Fig. 8. Fitting cost of different modeling approaches is shown for read delay of the SRAM read path.

the proposed BMF-PS is able to achieve substantially higher accuracy than OMP. Second, BMF-NZM is less accurate than BMF-ZM with 100 samples; however, it is more accurate than BMF-ZM when a large number of samples are available. In other words, the optimal prior distribution can vary even for the same performance metric in practice. In this example, BMF-PS is able to achieve superior modeling accuracy over all other approaches.

Fig. 8 shows the fitting cost for two different algorithms: 1) OMP and 2) BMF-PS with our proposed fast solver based on low-rank update. In this example, the conventional solver based on Cholesky decomposition becomes computationally

TABLE VI
RELATIVE MODELING ERROR AND COST FOR SRAM READ PATH

	OMP	BMF-PS (fast solver)
# of post-layout training samples	400	100
Modeling error for read delay	1.1330%	1.0804%
Simulation cost (Hour)	38.77	9.69
Fitting cost (Second)	112.53	20.79
Total modeling cost (Hour)	38.80	9.70

infeasible due to the large problem size. Hence, its fitting cost is not reported here.

Table VI further compares the performance modeling error and cost for OMP and BMF-PS with fast solver. In this example, BMF-PS achieves 4 \times runtime speed-up over OMP without surrendering any accuracy.

It is important to note that we consider linear performance models only for both the RO and SRAM examples. However, the proposed BMF framework is not limited to linear performance modeling. BMF can be applied to orthonormal basis functions in (3) where high-order basis functions are included. When building a nonlinear performance model in practice, we must efficiently fit the early-stage model by applying a number of heuristics (e.g., the variable pruning approach described in [13]).

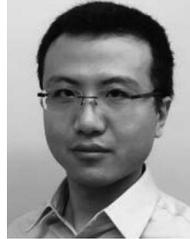
VI. CONCLUSION

In this paper, a novel BMF algorithm is developed for efficient high-dimensional performance modeling of AMS circuits with consideration of process variations. BMF optimally combines the early-stage (e.g., schematic-level) information and a small number of late-stage (e.g., post-layout) training samples by encoding them as the prior distribution and likelihood function, respectively, through Bayesian inference. Next, the late-stage model coefficients are accurately determined by maximizing the posterior distribution. Several implementation issues are carefully considered to make the proposed BMF method of practical usage. As is demonstrated by two circuit examples designed in a commercial 32 nm CMOS SOI process, the proposed BMF method is able to achieve up to 9 \times runtime speed-up compared to the traditional modeling approach.

REFERENCES

- [1] F. Wang, W. Zhang, S. Sun, X. Li, and C. Gu, "Bayesian model fusion: Large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," in *Proc. Design Autom. Conf.*, Austin, TX, USA, 2013, pp. 1–6.
- [2] Semiconductor Industry Associate. (2011). *International Technology Roadmap for Semiconductors*. [Online]. Available: www.itrs.net/Links/2011ITRS/Home2011.htm
- [3] X. Li, J. Le, and L. Pileggi, *Statistical Performance Modeling and Optimization*. Boston, MA, USA: Now, 2007.
- [4] L. Yu *et al.*, "Remembrance of transistors past: Compact model parameter extraction using Bayesian inference and incomplete new measurements," in *Proc. Design Autom. Conf.*, San Francisco, CA, USA, 2014, pp. 1–6.
- [5] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1533–1545, Oct. 2013.

- [6] Z. Zhang, X. Yang, I. V. Oseledets, G. E. Karniadakis, and L. Daniel, "Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 1, pp. 63–76, Jan. 2015.
- [7] Z. Feng and P. Li, "Performance-oriented statistical parameter reduction of parameterized systems via reduced rank regression," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2006, pp. 868–875.
- [8] A. Singhee and R. A. Rutenbar, "Beyond low-order statistical response surfaces: Latent variable regression for efficient, highly nonlinear fitting," in *Proc. Design Autom. Conf.*, San Diego, CA, USA, 2007, pp. 256–261.
- [9] A. Mitev, M. Marefat, D. Ma, and J. M. Wang, "Principle Hessian direction based parameter reduction with process variation," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2007, pp. 632–637.
- [10] T. McConaghy and G. G. E. Gielen, "Template-free symbolic performance modeling of analog circuits via canonical-form functions and genetic programming," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1162–1175, Aug. 2009.
- [11] X. Li and H. Liu, "Statistical regression for efficient high-dimensional modeling of analog and mixed-signal performance variations," in *Proc. Design Autom. Conf.*, Anaheim, CA, USA, 2008, pp. 38–43.
- [12] X. Li, "Finding deterministic solution from underdetermined equation: Large-scale performance modeling by least angle regression," in *Proc. Design Autom. Conf.*, San Francisco, CA, USA, 2009, pp. 364–369.
- [13] X. Li, "Finding deterministic solution from underdetermined equation: Large-scale performance variability modeling of analog/RF circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 11, pp. 1661–1668, Nov. 2010.
- [14] W. Zhang, T.-H. Chen, M. Y. Ting, and X. Li, "Toward efficient large-scale performance modeling of integrated circuits via multi-mode/multi-corner sparse regression," in *Proc. Design Autom. Conf.*, Anaheim, CA, USA, 2010, pp. 897–902.
- [15] T. McConaghy, "High-dimensional statistical modeling and analysis of custom integrated circuits," in *Proc. Custom Integr. Circuits Conf.*, San Jose, CA, USA, 2011, pp. 1–8.
- [16] X. Li, W. Zhang, and F. Wang, "Large-scale statistical performance modeling of analog and mixed-signal circuits," in *Proc. Custom Integr. Circuits Conf.*, San Jose, CA, USA, 2012, pp. 1–8.
- [17] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi, "Asymptotic probability extraction for nonnormal performance distributions," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 1, pp. 16–37, Jan. 2007.
- [18] M. Sengupta *et al.*, "Application-specific worst case corners using response surfaces and statistical models," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 9, pp. 1372–1380, Sep. 2005.
- [19] Z. Wang and S. W. Director, "An efficient yield optimization method using a two step linear approximation of circuit performance," in *Proc. Eur. Design Test Conf.*, Paris, France, 1994, pp. 567–571.
- [20] A. Dharchoudhury and S. M. Kang, "Worst-case analysis and optimization of VLSI circuit performance," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 4, pp. 481–492, Apr. 1995.
- [21] G. Debyser and G. Gielen, "Efficient analog circuit synthesis with simultaneous yield and robustness optimization," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 1998, pp. 308–311.
- [22] F. Schenkel *et al.*, "Mismatch analysis and direct yield optimization by spec-wise linearization and feasibility-guided search," in *Proc. Design Autom. Conf.*, Las Vegas, NV, USA, 2001, pp. 858–863.
- [23] X. Li, P. Gopalakrishnan, Y. Xu, and L. T. Pileggi, "Robust analog/RF circuit design with projection-based performance modeling," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 1, pp. 2–15, Jan. 2007.
- [24] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [25] X. Li, W. Zhang, F. Wang, S. Sun, and C. Gu, "Efficient parametric yield estimation of analog/mixed-signal circuits via Bayesian model fusion," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2012, pp. 627–634.
- [26] G. Sansone, *Orthogonal Functions*. New York, NY, USA: Dover, 2004.
- [27] R. H. Myers and D. C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New York, NY, USA: Wiley, 2002.
- [28] W. Zhang *et al.*, "Virtual probe: A statistical framework for low-cost silicon characterization of nanoscale integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 12, pp. 1814–1827, Dec. 2011.
- [29] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.
- [30] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.



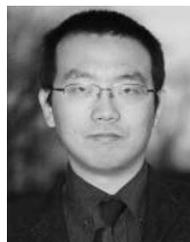
Fa Wang (S'13) received the B.S. degree in automation from Tsinghua University, Beijing, China, in 2009. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.

His current research interests include statistical design for analog and mixed-signal circuit and self-healing circuit.



Paolo Cachecho received the B.E. degree in electrical and computer engineering from the American University of Beirut, Beirut, Lebanon, in 2015.

He is currently a Research Assistant with the American University of Beirut. His current research interests include statistical modeling and performance modeling.



Wangyang Zhang (S'10–M'13) received the B.S. and M.S. degrees in computer science from Tsinghua University, Beijing, China, in 2006 and 2008, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2012.

He is currently with Cadence Design Systems, Inc., Pittsburgh.

Dr. Zhang was a recipient of the Best Paper Awards from DAC and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and the SIGDA Outstanding Ph.D. Award.



Shupeng Sun (S'11) received the B.S. degree in automation from Tsinghua University, Beijing, China, in 2010. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.

His current research interests include memory yield verification and performance modeling.



Xin Li (S'01–M'06–SM'10) received the B.S. and M.S. degrees in electronics engineering from Fudan University, Shanghai, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2005.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Carnegie Mellon University. In 2005, he co-founded Xigmix Inc., Pittsburgh. From 2009 to 2012, he was the Assistant Director for FCRP Focus Research Center for Circuit and System Solutions, Durham, NC, USA. From 2014 to 2015, he was the Assistant Director for the Center for Silicon System Implementation at Carnegie Mellon. His current research interests include integrated circuit and signal processing.

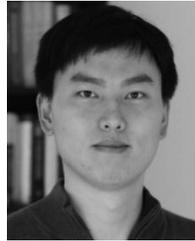
Dr. Li was a recipient of the NSF Faculty Early Career Development Award in 2012, the IEEE Donald O. Pederson Best Paper Award in 2013, the Best Paper Award from Design Automation Conference in 2010, two IEEE/ACM William J. McCalla ICCAD Best Paper Awards in 2004 and 2011, and the Best Paper Award from International Symposium on Integrated Circuits in 2014. He has been an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the *ACM Transactions on Design Automation of Electronic Systems*, the IEEE Design & Test, and the *Journal of Low Power Electronics*. He served on the Executive Committee of ACM SIGDA.



Rouwaida Kanj (M'99–SM'10) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA.

She joined the IBM Research Laboratories as a Research Staff Member from June 2004 to 2012. She led algorithmic development of statistical analysis tools and methodologies for memory designs. She is currently an Assistant Professor with the American University of Beirut, Beirut, Lebanon. She holds over 20 U.S. patents and several pending patents. Her work has resulted in over 40 publications.

Dr. Kanj was a recipient of the prestigious IEEE William McCalla Best Paper Award at ICCAD 2009, two ISQED Best Paper Award in 2006 and 2014, an Outstanding Technical Achievement Award from IBM, and six invention plateau awards.



Chenjie Gu (M'11) received the Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 2011.

He is currently a Research Scientist with Intel Corporation, Hillsboro, OR, USA. His current research interests include system/circuit design, dynamical system modeling, and statistical learning algorithms.