Spatial Estimation of Wafer Measurement Parameters Using Gaussian Process Models

Nathan Kupp*, Ke Huang†, John Carulli‡, and Yiorgos Makris†
*Department of Electrical Engineering, Yale University, New Haven, CT 06511
†Department of Electrical Engineering, The University of Texas at Dallas, Richardson, TX 75080
‡Texas Instruments Inc., 12500 TI Boulevard, MS 8741, Dallas, TX 75243

***Abstract*—In the course of semiconductor manufacturing, various e-test measurements (also known as inline or kerf measurements) are collected to monitor the health-of-line and to make wafer scrap decisions preceding final test. These measurements are typically sampled spatially across the surface of the wafer from between-die scribe line sites, and include a variety of measurements that characterize the wafer's position in the process distribution. However, these measurements are often only used for wafer-level characterization by process and test teams, as the sampling can be quite sparse across the surface of the wafer. In this work, we introduce a novel methodology for extrapolating sparsely sampled e-test measurements to every die location on a wafer using Gaussian process models. Moreover, we introduce radial variation modeling to address variation along the wafer center-to-edge radius. The proposed methodology permits process and test engineers to examine e-test measurement outcomes at the die level, and makes no assumptions about wafer-to-wafer similarity or stationarity of process statistics over time. Using high volume manufacturing (HVM) data from industry, we demonstrate highly accurate cross-wafer spatial predictions of e-test measurements on more than 8,000 wafers.**

I. Introduction

In modern high-volume semiconductor manufacturing, uncontrollable process variations are increasingly challenging due to decreasing feature sizes. These variations often result in device failures, impacting yield and the number of marketable devices produced per wafer. Therefore, understanding and monitoring these process variations is critical to the production of semiconductor devices.

A traditional mechanism for monitoring process variation effects during semiconductor manufacturing is via e-test measurements, also known as inline or kerf measurements. These e-test measurements consist of low-level circuit component data sampled from test structures across the surface of each wafer. The test structures are typically constructed in the wafer scribe lines (that is, the areas of the wafer destroyed during wafer dicing) although some on-die test structures may also be employed. These measurements are typically associated with wafer scrap limits to catch unacceptable process variations early, before wafer processing is completed and expensive fabrication steps are wasted. Consequently, e-test measurements provide an indirect measure of the health of the wafer under inspection. They are also frequently used to inspect problematic stages of the fabrication process via wafer-level commonality analysis against fabrication tools or chambers.

Despite the usefulness of these e-test measurements in providing a measure of circuit performance, they are only sparsely sampled across each wafer, with only a handful of sites explicitly measured on a wafer that may have many hundreds or thousands of die. Explicitly collecting e-test measurements from scribe line sites adjacent to every reticle shot would be prohibitively time consuming, and more importantly, would only marginally increase the amount of information provided about circuit performance for wafer characterization.

In this work, we demonstrate that sparse wafer-level spatial sampling of e-test measurements does not limit us to only performing wafer-level correlation analyses. By employing a regression technique known as Gaussian process modeling, originating from the field of geostatistics, we introduce a methodology for accurately extrapolating e-test measurement observations from sampled scribe line structures to every die location. The proposed method also incorporates domain-specific knowledge via radial variation modeling. Our results are demonstrated on HVM data, using measurements from more than 8,000 production wafers.

The remainder of this paper is organized as follows. In Section II, we discuss the importance of spatial interpolation, in Section III, we discuss existing work on statistical modeling for semiconductor manufacturing, and in Section IV we describe the e-test measurements used to construct spatial models. Section V introduces Gaussian process models and their relevance to semiconductor manufacturing. In Section VI, we provide experimental results, and conclusions are drawn in Section VII.

II. Spatial Interpolation of e-Test Measurements

Without accurate die-level measurements, performing die-level analysis and constructing die-level statistical models is not possible. Consequently, the sparsity of e-test measurement sampling prevents us from readily modeling relationships at this level. As process variation increases with smaller geometries at each technology node, this problem becomes even more pronounced. Approximations can be made using simple linear interpolation or k-nearest-neighbor methods, and sampled e-test measurements may subsequently be extrapolated using such models to obtain die-level estimates. For measurements with approximately linear or constant variation across the wafer, this may be sufficient to obtain low prediction error.

978-1-4673-1595-1/12/$31.00

However, in the case when e-test parameters show non-linear or radial variations, simplistic approaches will be subject to a high degree of error. We find theoretical justification for this in the domain of decision theory [1]–[3]. Consider the concept of *capacity* [1], defined informally as the complexity of a model; a model with high capacity will outperform a model with low capacity in modeling non-trivial correlations. Using the Vapnik-Chervonenkis (VC) dimension [1] as a capacity measure on various function classes, we can pursue a modeling strategy known as Structural Risk Minimization (SRM), whereby we rank order proposed models by considering a nested sequence of function classes with strictly increasing VC dimension:

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_n \tag{1}$$

For example, perhaps we consider an ordering where $\mathcal{F}_1$ is the set of all constant models, $\mathcal{F}_2$ is the set of all linear models, $\mathcal{F}_3$ is the set of all quadratic models, and so on. By balancing generalization capability against capacity during training, we can make an appropriate choice for $\mathcal{F}$ from the nested sequence of Equation 1 which minimizes the empirical risk. Gaussian process models, as described in Section V, have sufficient capacity to handle very complex, non-linear behavior in the training set, while elegantly accounting for noise in the training data to avoid over fitting.

An overview of the proposed approach is shown in Figure 1. We are provided some arbitrary, spatially-sampled data from a wafer or set of wafers. From this data, we would like to generate approximations of the sampled data extrapolated to every die location. To do this, we build a statistical model using our small set of samples, as shown in the center of Figure 1, and use it to predict across the surface of the wafer, as shown on the right of Figure 1.

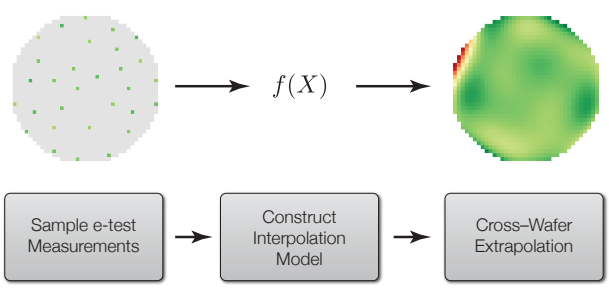

Fig. 1. Overview of Proposed System

III. Prior Work

There is a great deal of literature involving statistical modeling of semiconductor manufacturing test data. The alternate test approach [4], [5] uses a spline regression model to link inexpensive test measurements to high-cost counterparts during final test, thereby reducing overall test cost. Machine learning-based low-cost testing [6], [7] solves a similar problem, via construction of ontogenic neural networks to predict final test outcomes. Both of these approaches address the high cost of semiconductor device testing by reducing the test set or replacing it with inexpensive alternative measurements, and rely on statistical models to recover the original test set or predict pass/fail labels. However, neither of these approaches involve spatial correlation, as test cost reduction involves solving a slightly different problem that does not presently target cross-wafer variation statistics.

In terms of spatial interpolation of semiconductor statistics, the most prominent methodology in the literature is an approach known as "Virtual Probe" [8], [9]. In [9], the authors extend Virtual Probe to address the same test cost reduction problem targeted by alternate test or machine learning-based low cost testing, but by performing wafer-level spatial sampling of expensive tests instead of completely removing them, to find a different tradeoff between test cost and the test escape or test error rate. In general, the spatial modeling problem addressed by Virtual Probe has similar properties to the problem addressed in this work. However, it takes a fundamentally different algorithmic approach, reasoning from the domain of compressed sensing rather than geostatistics. The core modeling approach of Virtual Probe is a discrete cosine transform (DCT) that projects spatial statistics into the frequency domain. The main assumption of this approach is the spatial patterns of process variations are smooth and they can be represented by a few dominant DCT coefficients at low frequencies [10]. In this work, we employ Gaussian process models that perform a more general projection via kernel functions. A complete empirical comparison of the proposed methodology and Virtual Probe is provided in the experimental results.

A predecessor of this work can be found in [11], where the author lays the groundwork for applying Gaussian Process models (also known as "kriging") to spatial interpolation of semiconductor data based on Generalized Least Squares fitting and a structured correlation function. The computational method combines empirical data fitting and unconstrained optimization. In this work, we extend the key ideas of [11] by introducing modeling of radial variation and by introducing a function-space derivation of Gaussian process models. Moreover, we demonstrate our experimental results on the largest industrial case study of semiconductor spatial modeling to-date.

IV. E-Test Measurements

E-test measurements are a set of process characterization parameters collected from scribe line test structures. These test structures are drawn in the areas of the wafer that are destroyed during wafer dicing, as shown in Figure 2, and therefore can only be measured before circuit packaging, at the wafer test stage or earlier. A subset of e-test measurements is typically collected from these structures after completing a layer or two of metallization, and the remainder are collected later, during wafer acceptance testing. The e-test structure is often associated with several die within a reticle, as it typically is drawn to the full height of the reticle. For example, the illustration of Figure 2 shows a 2x2 die reticle, and consequently the

e-test structure spans two die in the y-dimension. The e-test measurements collected from these scribe-line structures are designed to capture a very broad set of process statistics, and generally include parameters such as:

1) V_{th}, T_{ox}, G_m, I_{off}, measurements for all types of NMOS/PMOS transistors (e.g., high V_{th}, low V_{th}, etc.)
2) N-well / poly resistor measurements
3) Diode V_f and parasitics
4) Capacitor unit cell measurements and parasitics
5) Inductor L and Q and parasitics
6) General parasitics
7) Physical process dimensions

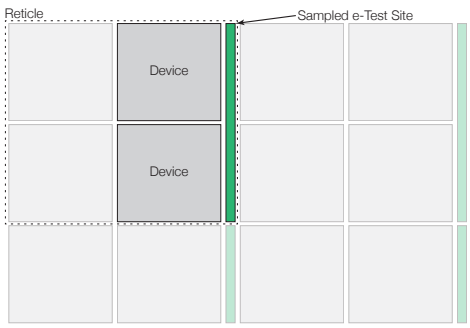

Fig. 2. e-Test Measurement Site Diagram

Clearly, these types of measurements capture a great deal of information about the health of the process and particular wafers. However, most semiconductor manufacturing usage is traditionally limited to collecting a few samples of these measurements from around the wafer. As shown in Figure 3, the e-test sites may be sampled only one or two dozen times on a wafer with several thousand die. This has effectively limited the scope of statistical models that can be applied using the data to wafer-level correlation, or to zonal analysis of the wafer, grossly batching sets of die with the nearest neighbor e-test sites. In this work, we demonstrate a novel methodology to generate highly accurate extrapolations of e-test measurements from a small set of sample sites, vis-à-vis Figure 3, to every die location on a wafer, without making unnecessary assumptions about manufacturing process stationarity or wafer-to-wafer statistics.

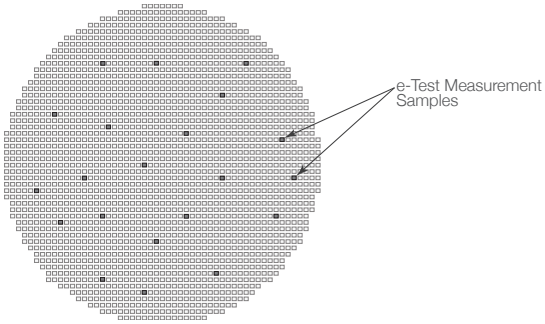

Fig. 3. e-Test Measurements

V. Gaussian Process Models

Gaussian process regression modeling [12] is an inductive regression approach targeted at extrapolating a function over a Gaussian random field based on limited data observations. Gaussian process models have their foundations in Bayesian statistics and share a common basis in kernel theory with Support Vector Machines [1], [13], [14]. In this section, we describe the theoretical basis for Gaussian process models, and develop our Gaussian process regression-based methodology for extrapolating e-test measurements to every die location on a wafer.

Consider the monolithic linear regression formulation $t = f(\mathbf{x}) + \varepsilon$, where $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}$ and ε represents independent and identically distributed (i.i.d.) additive noise. In this formulation, $\mathbf{w}$ is a vector of weights associated with each dimension of $\mathbf{x}$; that is, larger elements of $\mathbf{w}$ corresponds to more "important" features in the model. Note that by enforcing such a model, we impose structure on the problem space and reduce model variance at the cost of increased model bias.

Such a model performs well when the true generative model happens to be linear, but this is often not the case. To see why this can be problematic, consider the following scenario. Suppose we attempt to use a linear model to extrapolate an e-test measurement t over Cartesian wafer coordinates $\mathbf{x} = [x, y]$. By adopting a linear model, we are making the assumption that the underlying generative function is fully parameterized by linear coefficients, and can be represented as a simple 2D plane. If the true generative function is non-linear, the predictions made by this model will present high bias and consequently high prediction error.

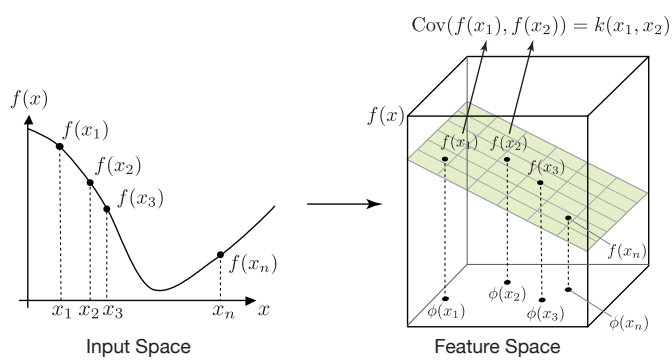

Fig. 4. Overview of Gaussian Process modeling

With Gaussian processes, we do not presume the output $f(\mathbf{x})$ is necessarily of linear form, as shown by the one-dimensional input space curve on the left side of Figure 4. Instead, we define a Gaussian process as a collection of random variables $f(\mathbf{x})$ associated with points $\mathbf{x}$, such that every finite set of n functions $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}$ is jointly Gaussian-distributed[1]. To derive a Gaussian process model for inductive regression, we first consider a noise-free

[1]In this section, we adopt notation similar to [12] for convenience.

linear model, shown by the right side of Figure 4, which has the form:

$$t = f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w} \tag{2}$$

where $\phi(\mathbf{x})$ is a function of the inputs mapping the input columns into some high dimensional feature space, shown by the bottom plane on the right side of Figure 4. For example, a scalar input $\mathbf{x}$ could be projected into the feature space: $\phi(\mathbf{x}) = (1, \mathbf{x}, \mathbf{x}^2)^\top$. We assign a Bayesian prior on the weights such that $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$. As the realizations of the Gaussian process at points $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}$ are jointly Gaussian, we can fully specify the Gaussian process with mean and covariance functions:

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}] = 0, \tag{3}$$

$$\begin{aligned}\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] &= \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top]\phi(\mathbf{x}') \\ &= \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}')\end{aligned} \tag{4}$$

A. Kernel Covariance Functions

Recall that our ultimate goal of building a Gaussian process-based regression model is to somehow capture spatial variation in t as a function of the coordinates $\mathbf{x}$. The following discussion demonstrates how we can accomplish this task by modeling our data as drawn from a process with a covariance function that depends on spatial location. By taking this approach, proximal data points are modeled as being highly covariant, and distant points are modeled with low covariance. This codifies our intuition and *a priori* knowledge of the domain: we expect the variation of e-test measurement data to strongly correlate to spatial coordinates. In Section V-D we explain how we can easily extend this reasoning to non-Cartesian coordinate spaces; in particular, to model radial variation of e-test parameters.

Consider the covariance function specified in Equation 4. Now, since covariance matrices are by definition positive semi-definite (see proof in the appendix), we can redefine Σ_p as $(\Sigma_p^{1/2})^2$, and rewrite Equation 4 as:

$$\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}') \tag{5}$$

$$= \phi(\mathbf{x})^\top (\Sigma_p^{1/2})^\top \Sigma_p^{1/2} \phi(\mathbf{x}') \tag{6}$$

We now introduce the parameter $\psi(\mathbf{x})$ by defining $\psi(\mathbf{x}) = \Sigma_p^{1/2}\phi(\mathbf{x})$, and subsequently rewrite the covariance of Equation 4 as:

$$\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \phi(\mathbf{x})^\top (\Sigma_p^{1/2})^\top \Sigma_p^{1/2} \phi(\mathbf{x}') \tag{7}$$

$$= \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle \tag{8}$$

Crucially, this covariance function is formed as an inner product, permitting us to leverage the kernel trick [15] and express Equation 8 as a kernel function $k(\mathbf{x}, \mathbf{x}')$. In other words, the covariance between any outputs can be written as a function of the inputs using the kernel function without needing to explicitly computing $\phi(\mathbf{x})$, as shown in Figure 4. Many kernel functions exist, and any function $k(\cdot, \cdot)$ that satisfies Mercer's condition [1] is a valid kernel function. However, only a handful of kernels see widespread use. Among these common kernels, the most prevalent is the squared exponential, also known as the radial basis function kernel. In this work, we employed a squared exponential kernel of the form:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2l^2}|\mathbf{x} - \mathbf{x}'|^2\right) \tag{9}$$

where l is some characteristic length-scale of the squared exponential kernel, $|\cdot|$ denotes the Euclidean distance. Employing this kernel is equivalent to training a linear regression model with an infinite-dimensional feature space. Substituting our squared-exponential covariance function into the definition of the Gaussian process, we arrive at a Gaussian process formulation as:

$$t = f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \tag{10}$$

The following section describes how to employ this process to derive predictive distributions, as well as how to manage the inclusion of additive noise in the model.

B. Training and Prediction

Suppose that we are provided a training set of n data points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ observed in an N-dimensional space, e.g., each vector in X is $\mathbf{x}_i = \{x_1, x_2, \ldots, x_N\}$. So, X is an $n \times N$ matrix of inputs. With these input points, we are also given a set of predictive targets, $\mathbf{t} = \{t_1, t_2, \ldots, t_n\}$. Now, we wish to model the observed data as a noise-free Gaussian process and define, as before, $y = f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$.

To derive the predictive distribution of this Gaussian process, we first write the joint distribution of the training set targets and a new test function value as:

$$\begin{bmatrix} \mathbf{t} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & \mathbf{k}_* \\ \mathbf{k}_*^\top & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right) \tag{11}$$

Where $\mathbf{x}_*$ is a location we wish to extrapolate to, and where we have defined $K = K(X, X')$ as the matrix of the kernel function $k(\mathbf{x}, \mathbf{x}')$ evaluated at all pairs of training locations. We have also defined $\mathbf{k}_* = K(X, \mathbf{x}_*)$ as the column vector of kernel evaluations between the test point and the entire set training points, and lastly, $k(\mathbf{x}_*, \mathbf{x}_*)$ as the variance of the test function value at the observation point $\mathbf{x}_*$. With this distribution, we can condition the test function value on the observed data to obtain the predictive distribution (we omit the derivation for brevity):

$$\begin{aligned} f_*|X, \mathbf{t}, \mathbf{x}_* \sim \mathcal{N}(&\mathbf{k}_*^\top K^{-1}\mathbf{t}, \\ &k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top K^{-1}\mathbf{k}_* \end{aligned} \tag{12}$$

In this work, we primarily concern ourselves with point predictions, and so we use simply the distribution mean $\bar{f}_* = \mathbf{k}_*^\top K^{-1}\mathbf{t}$ to generate a point prediction from the predictive distribution. This corresponds to decision-theoretic risk minimization [1], [3], [13] using a squared-loss function.

C. Regularization

To avoid overfitting, a technique known as regularization [3] is often employed in decision-theoretic empirical risk minimization. In a traditional linear regression model, regularization typically involves penalizing the L_1 or L_2 norm of the model coefficient estimates $\hat{\beta}$ to ensure the "slope" of the model is not too large. This ensures that extrapolative predictions are not too extreme.

Gaussian process models handle regularization somewhat differently. Instead of penalizing coefficients, we consider our predictive targets $\mathbf{t} = \{t'_1, t'_2, \ldots, t'_n\}$ as affected by additive noise such that $t'_i = t_i + \varepsilon$, where we make the usual i.i.d. assumptions about the additive noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. To incorporate this into our Gaussian process model, we update Equation 10 to model additive noise in the observations:

$$y = f(\mathbf{x}) + \varepsilon \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}) \tag{13}$$

where $\delta_{\mathbf{x},\mathbf{x}'}$ is the Kronecker delta function. This, in turn, affects the joint distribution of Equation 11:

$$\begin{bmatrix} \mathbf{t} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K + \sigma_n^2 I & \mathbf{k}_* \\ \mathbf{k}_*^\top & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right) \tag{14}$$

as well as the predictive distribution:

$$f_* | X, \mathbf{t}, \mathbf{x}_* \sim \mathcal{N}(\mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{t}, \\ k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{k}_* \tag{15}$$

resulting in a point prediction for new observations of $\bar{f}_* = \mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{t}$. This constrains the fitted model to avoid extreme predictions. For example, consider the univariate fit of Figure 5, shown with 4 monotonically increasing noise parameters $\sigma_n^2 = \{0, 0.0001, 0.01, 0.5\}$. The blue line is the fit model, the red dots are the original data, and the dotted red line is the true generative function. As this noise parameter increases, the model gradually flattens, and for very large σ_n^2, approaches a constant fit. Applying a model with a $\sigma_n^2 = 0$ is equivalent to the hypothesis that our observations are noise-free. Therefore, employing a non-zero σ_n^2 captures our intuition that real-world data measurements are affected by noise, and that we should not expect to fit a model exactly through each observed data point. As a practical matter, we have found empirically that $\sigma_n^2 = 0.1$ works well for our data. In the general case, this parameter should be adjusted to the particular application using a hold-out set of data.

In Figure 6, we show the effects of incorporating additive noise on example wafer data, with $\sigma_n^2 = \{0, 0.00001, 0.01, 0.1\}$. As can be seen from the figure, modeling observations as noise-free leads to extreme variation in the model as it fits the response surface exactly through each point observation, and relaxing this constraint leads to smoother response surfaces.

D. Modeling Radial Variation

A key contribution of this work is the extension of Gaussian process modeling over Cartesian coordinates to a joint Cartesian-radius space, capturing our intuition that wafer variance is often radial. By including a radius feature, we canonicalize the notion that any set of die drawn from a wafer-centered ring should present similar e-test measurement profiles.

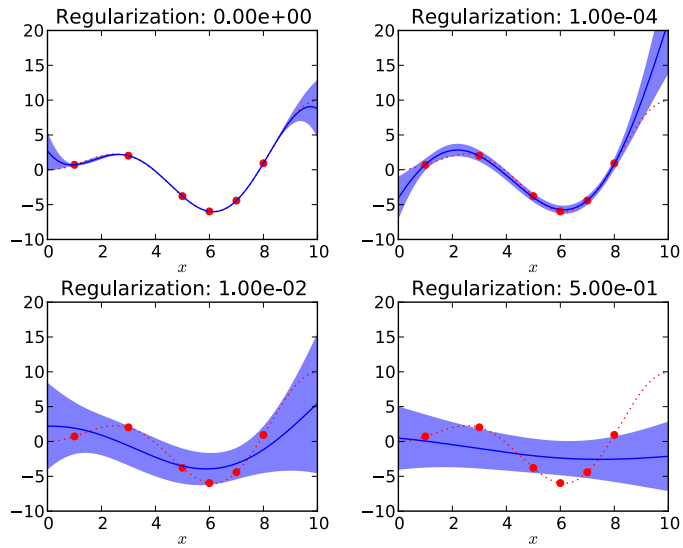

Fig. 5. Regularization Example

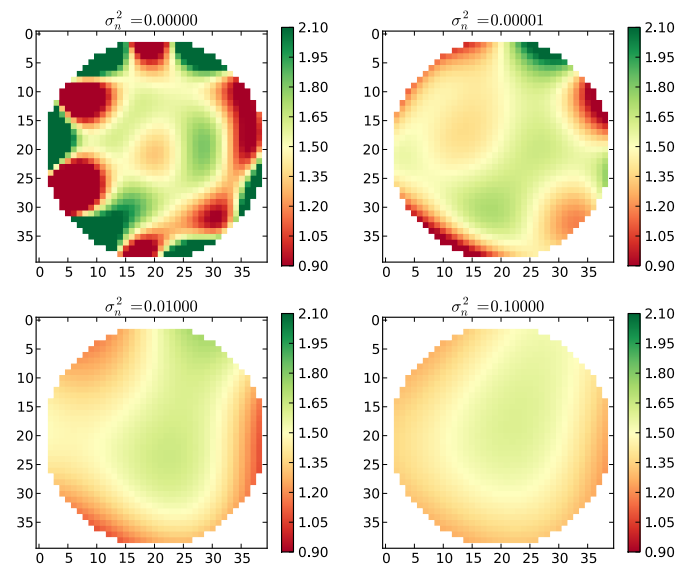

Fig. 6. Wafer Regularization

An advantage of using Gaussian process regression is the ability to apply a Gaussian process over arbitrary index sets. Thus far, we have been describing a Gaussian process implementation that estimates e-test parameters over a 2D Cartesian plane, but we are free to use any other field. As noted above, many parameters will manifest radial variation patterns due to the physical realities of semiconductor manufacturing. To accommodate this in our Gaussian process model, we can simply update our coordinates from $\mathbf{x} = [x, y]$ to include a

radius $r = \sqrt{x^2 + y^2}$:

$$\mathbf{x} = \left[x, y, \sqrt{x^2 + y^2}\right]$$

Now, applying the Gaussian process regression model over this space will result in a model that takes radial variation patterns into account. In Figure 7, we show the effect this has on the prediction outcomes.

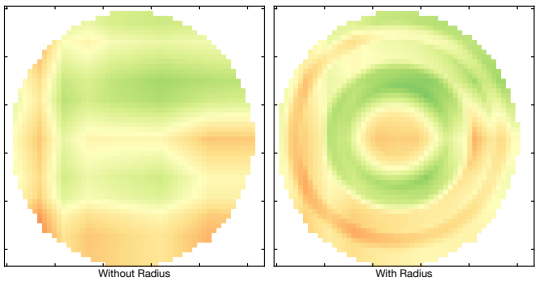

Fig. 7. Radial Modeling

E. Gaussian Process Models for e-test Interpolation

Our objective with using Gaussian process regression for e-test parameter inductive interpolation is to build per-wafer models that accurately estimate e-test parameter outcomes at previously unobserved wafer $[x, y]$ locations. By modeling variation on a per-wafer basis, we sidestep the need for the "median polishing" methodology of [11].

Our Gaussian process implementation was designed to record spatial prediction error across the surface wafer and provide a reasonable metric of prediction quality in terms of percent error. The output of our Gaussian process implementation is a $j \times k$ matrix E composed of prediction errors ϵ_{jk}, where ϵ_{jk} is the prediction error of the model for the j-th wafer and the k-th e-test parameter.

To train the Gaussian process model for a particular wafer j, we employ leave-one-out cross validation. Specifically, consider observations of the k-th e-test parameter at sites $t_k^{(i)},\ i \in \{1, 2, \ldots, n\}$, where each site has an associated $\mathbf{x}_i = [x, y]$ location consisting of Cartesian e-test site coordinates. We select the l-th example $t_k^{(l)}$ as a test case, and subsequently train a model on the remaining observations $t_k^{(i)}, i \neq l$. This model is then used to produce an estimate:

$$\hat{t}_k^{(l)} = \bar{f}_* = \mathbf{k}_*{}^\top (K + \sigma_n^2 I)^{-1} \mathbf{t} \tag{16}$$

This process is then repeated for every e-test site, leaving out a single observation as a test set and training on the rest to produce an estimate for the test observation. Given these predictions, we compute test error as the mean absolute percent error across all predictions:

$$\epsilon_{jk} = \frac{1}{n} \sum_{i=1}^{n} \left| (\hat{t}_k^{(i)} - t_k^{(i)}) / t_k^{(i)} \right| \tag{17}$$

Thus, ϵ_{jk} represents the mean percent error of predicting the k-th e-test parameter for all sites on a particular wafer j. Applying the model in this fashion for all wafers and all e-test parameters, we populate the matrix E that completely characterizes the performance of the Gaussian process models on the dataset at hand. We can also summarize mean prediction error for a particular e-test parameter by computing the mean error over all wafers:

$$\epsilon_k = \frac{1}{N_{wafers}} \sum_{i=1}^{N_{wafers}} \epsilon_{ik} \tag{18}$$

VI. Experimental Results

In this work, we demonstrate results on e-test data collected from industry HVM. Our dataset has in total 8,440 wafers, and each wafer has 269 e-test measurements collected from 21 sites randomly sampled across the wafer. Clearly, increasing the number of sites per wafer would provide more information about the health of the process and particular wafers, at the expense of longer testing time. Leave-one-out cross validation was used to characterize the prediction error at each of the 21 sites, and the mean cross-validation error across all 21 sites was collected for each e-test measurement and each wafer, as described in Section V-E. Consequently, the resultant matrix of errors E had $8,440 \times 269$ elements.

A. Virtual Probe

As a baseline reference, we first provide experimental results for Virtual Probe. In Figure 8, we present a histogram of mean prediction errors across all wafers, as computed via Equation 18. The black line overlaid on the histogram represents the cumulative percent of e-test measurements included in each successive bin. Virtual Probe performs quite well, with more than 85% of the e-test measurements realizing less than 4% mean prediction error.

In Figure 11 we present an overview of the prediction errors with 10%–90% error bars shown for all 269 e-test measurements, sorted by median Virtual Probe prediction error. Each index on the x-axis corresponds to a single e-test measurement, and the y-axis shows the prediction error in percent incurred by Virtual Probe.

B. Proposed Method: Gaussian Process Models

In Figure 9, we display a histogram of the mean (across all wafers) e-test measurement prediction errors, again as computed via Equation 18. As can be seen from the figure, the Gaussian process model prediction errors are even lower than Virtual Probe, with more than 90% of e-test measurement predictions below 4% error.

In Figure 12 we present the Gaussian process model prediction errors with 10%–90% error bars. Again, a qualitative comparison to Figure 11 shows that the errors are generally lower and the error bars tighter than for Virtual Probe. The error ranges are quite small across the majority of e-test measurements, demonstrating that the variance of the errors is low over the entire set of 8,440 wafers. Importantly, this shows that the models are insensitive to process shifts over time, a result that is attributable to the fact that we train and deploy our models on a per-wafer basis.

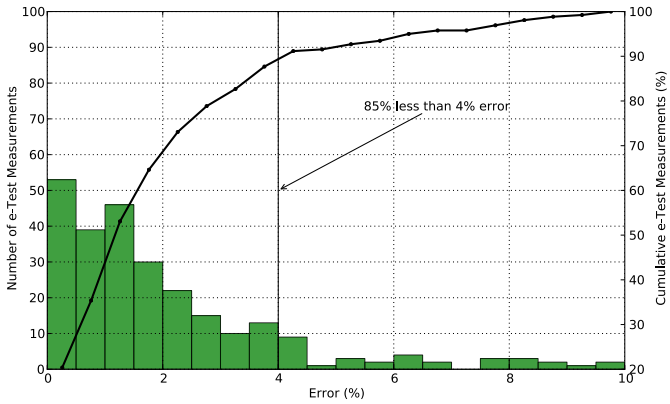

Fig. 8. Virtual Probe prediction error across all wafers

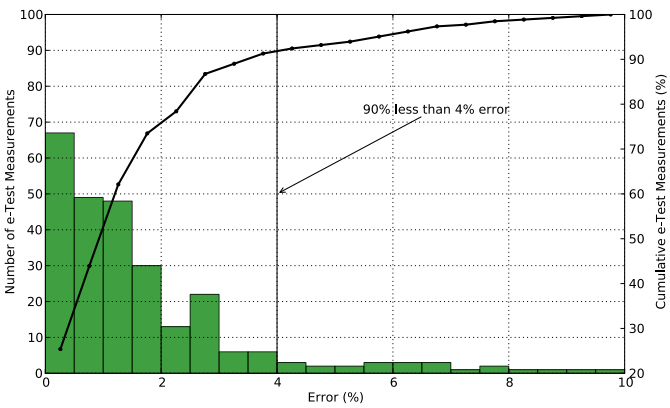

Fig. 9. Gaussian process model prediction error across all wafers

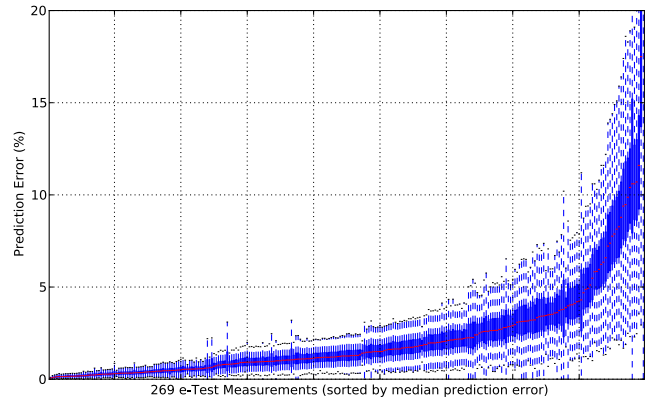

Fig. 11. Virtual Probe prediction error for each e-test measurement

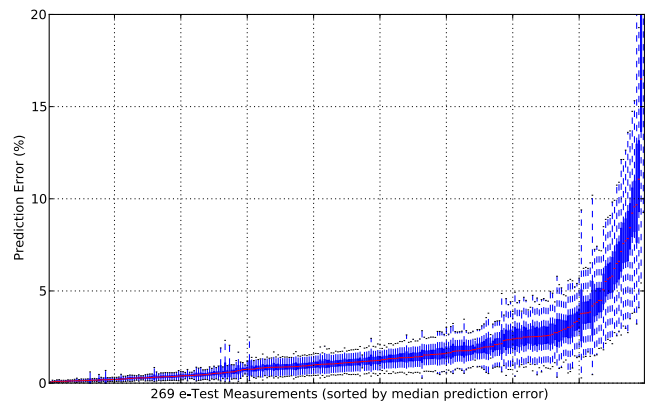

Fig. 12. Gaussian process prediction error for each e-test measurement

Figure 10 presents a zoomed-in version of Figure 12 with the best-predicted 30 measurements, e.g., the left-hand side of Figure 12. The x-axis shows the 30 e-test parameters; these include a mixture of diode, capacitor, and NMOS/PMOS transistor parameters. The y-axis is again percent error—for these 30 measurements, the mean prediction error is less than 0.2%, indicating that we can very successfully interpolate these parameters across the surface of the wafer.

In Figure 13 we present another zoomed-in version of Figure 12, in this case with the worst-predicted 30 e-test measurements, or the right-hand side of Figure 12. The parameters shown here are the most challenging for us to predict with our Gaussian process models, in some cases reaching more than 40% prediction error. These parameters are largely comprised of a) G_{ds} drain-source conductance measurements of various transistors, and b) various resistance measurements. Both tend to have high cross-wafer variation, or even die-to-die variation. Consequently, high prediction error for these parameters is relatively unsurprising. However, for most of these worst-case prediction errors the error is under 10%, which is still within an acceptable range for most use cases. A possibility to improve the prediction errors for these parameters is to increase the number of samples during the training.

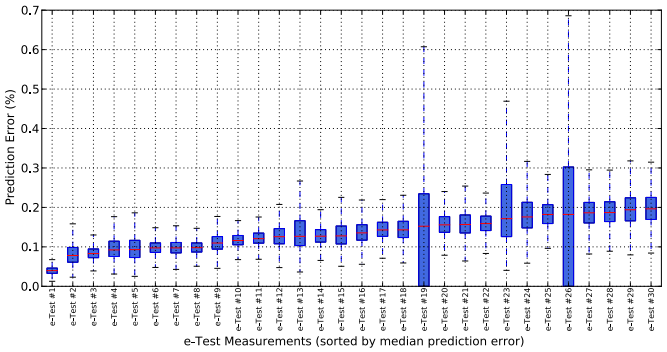

Fig. 10. Prediction error for best 30 e-test measurements

C. Comparison to Virtual Probe

In Figure 14 we present a comparison of the two methodologies, with Virtual Probe set as the baseline at 0%. The proposed methodology consistently outperforms Virtual Probe by an average of 0.5%, and in some cases, by almost 5%. The overall mean prediction error of Virtual Probe across all e-test measurements and all wafers is 2.68%, and the overall mean prediction error for Gaussian process-based spatial models is 2.09%.

A tabular comparison of the proposed Gaussian process model approach versus Virtual Probe is provided in Table

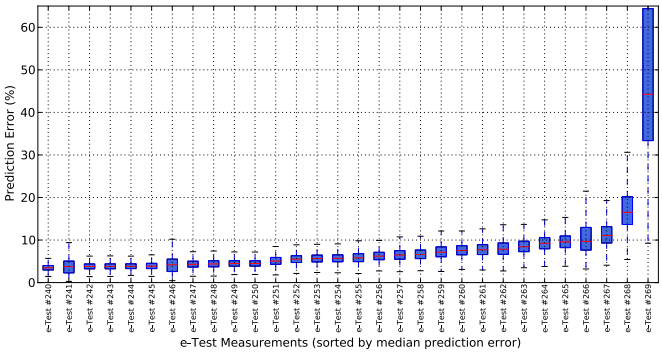

Fig. 13. Prediction error for worst 30 e-test measurements

I, with overall mean error reported alongside mean training and prediction time per wafer across all e-test measurements. The timing measurements were collected on a 2010 Core i5 2.4GHz, and represent the mean total time required to construct and predict with all $269 \times 21 = 5,649$ models for a given wafer. Note that the proposed methodology consistently has lower error than Virtual Probe, while incurring an order of magnitude less runtime to construct and evaluate the predictive models.

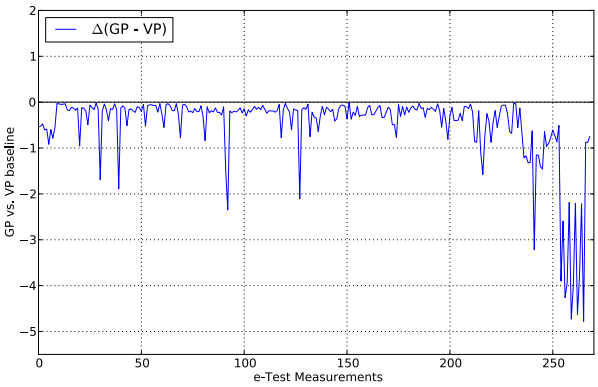

Fig. 14. Comparison of Gaussian Process Regression vs. Virtual Probe

Method	Overall Mean Percent Error	Avg. Running Time (per wafer)
Virtual Probe	2.68%	116.2s
Gaussian Process Model	2.09%	16.43s

TABLE I
VIRTUAL PROBE & GAUSSIAN PROCESS MODELS COMPARISON

VII. CONCLUSION

In this work, we presented a Gaussian process model-based method for generating spatial estimates of e-test parameters across the surface of wafers, enabling extraction of per-die estimates of e-test parameters. In general, our Gaussian process model is able to generate extremely accurate predictions for e-test performances across more than 8,000 HVM wafers. For 90% of the parameters, the Gaussian process model-based methodology demonstrates less than 4% error, and for the majority of the parameters the prediction error is much lower still. Moreover, the distribution of prediction errors is tightly clustered across all of the wafers, indicating that our models are not affected by process shifts over time. Lastly, our Gaussian process model-based approach consistently outperforms Virtual Probe, on average by 0.5%, and in certain cases, by a significant margin of almost 5%, while requiring an order of magnitude less runtime to evaluate on each wafer.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

[2] V. Cherkassky and F. Mulier, *Learning from Data*, John Wiley & Sons, 1998.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2001.

[4] P. N. Variyam, S. Cherubal, and A. Chatterjee, "Prediction of analog performance parameters using fast transient testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 3, pp. 349–361, 2002.

[5] S. S. Akbay and A. Chatterjee, "Fault-based alternate test of RF components," in *Proc. of International Conference on Computer Design*, 2007, pp. 517–525.

[6] H.-G. D. Stratigopoulos and Y. Makris, "Error moderation in low-cost machine learning-based analog/RF testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 2, pp. 339–351, 2008.

[7] H.-G. D. Stratigopoulos, P. Drineas, M. Slamani, and Y. Makris, "Non-RF to RF test correlation using learning machines: A case study," in *Proc. of VLSI Test Symposium*, 2007, pp. 9–14.

[8] X. Li, R. R. Rutenbar, and R. D. Blanton, "Virtual probe: A statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits," in *Proc. of International Conference on Computer-Aided Design*, 2009.

[9] H.-M. Chang, K.-T. Cheng, W. Zhang, X. Li, and K. Butler, "Test cost reduction through performance prediction using virtual probe," in *Proc. of International Test Conference*, 2011.

[10] W. Zhang, Xin. Li, F. Liu, E. Acar, R. A. Rutenbar, and R. D. Blanton, "Virtual probe: A statistical framework for low-cost silicon characterization of nanoscale integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 12, pp. 1814–1827, 2011.

[11] F. Liu, "A general framework for spatial correlation modeling in VLSI design," in *Proc. of Design Automation Conference*, 2007, pp. 817–822.

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[13] V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, Inc., 1998.

[14] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2001.

[15] M. A. Aizerman, E. A. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," in *Automation and Remote Control*, 1964, vol. 25, pp. 821–837.

ACKNOWLEDGEMENTS

This research has been carried out with the support of the National Science Foundation (NSF CCF-1149463) and the Semiconductor Research Corporation (SRC-1836.073). The first author is supported by an IBM/GRC (Global Research Collaboration) graduate fellowship.