Active Learning Framework for Post-Silicon Variation Extraction and Test Cost Reduction

Cheng Zhuo [1], Kanak Agarwal [2], David Blaauw [1], Dennis Sylvester [1]
[1] EECS Department, University of Michigan, Ann Arbor, MI 48109
[2] IBM Research, Austin, TX 78758
[1] {czhuo, blaauw, dennis}@eecs.umich.edu
[2] kba@us.ibm.com

***Abstract*—Traditional process variation modeling is primarily focused on design-time analysis and optimization. However, with the advances of post-silicon techniques, accurate variation model is also highly desired in various post-silicon applications, such as post-silicon tuning, test vector generation, and reliability prediction. The accuracy of such post-silicon variation models is greatly improved by incorporating test measurements from each wafer or die. However, to limit test cost, the number of measurements must be reduced as much as possible. This paper proposes an active learning framework to dynamically extract post-silicon process variation models with tightened variance from measurements. The framework is composed of two stages, active training and model adaptation. Active training collects information and initializes the models to be used for the forthcoming wafers. Model adaptation stage then validates the models and optimally determines the test configuration for partial testing to reduce the test cost. Experimental results based on the measurements from two industrial processes show that the proposed framework can achieve variation models with variance reduction of ∼80% when compared with design-time variation models. Meanwhile, the average estimation error for those untested sites is well maintained at ∼2-3% using merely ∼30% available test structures for two processes.**

I. Introduction

Susceptibility to process variation has increased with the scaling of CMOS into the nano-regime of VLSI designs [1]. The application of new resolution enhancement techniques complicates an already complex manufacturing process and makes it more difficult in maintaining process uniformity [1]. As a result, efficient and accurate process variation modeling becomes essential to ensure good yield.

Traditionally process variation modeling is targeted for design-time use and guides engineers in the optimization of their chips before silicon fabrication [2], [3]. Typically, such design-time process models rely on characterizing tens to even hundreds of test wafers [4]–[8]. The characterized model is then fed to either the statistical analysis tools to estimate design yield [9], [10] or statistical optimization engine that efficiently tunes thousands of devices to achieve a robust and high yield solution [11]. However, in recent years, due to the increasingly significant variability and the inability to measure every device on a die [12], process variation models are also critical after chips are fabricated for multiple post-silicon applications, such as:

- **Post-silicon tuning** which requires an accurate understanding of current process to appropriately adjust tuning parameters [13], [14].
- **Post-silicon timing characterization** where speed binning and critical path diagnosis require efficient process variation models to reduce test vector sets. [12], [15].
- **Post-silicon reliability analysis** where accurate models can tighten the process uncertainty to improve the chip lifetime prediction and enables specific supply voltage adjustment for the chip to obtain a better lifetime/performance trade-off [16].

Since there is limited research focused on extracting variation models for post-silicon use, most post-silicon techniques still rely on design-time variation models and do not take advantage of the availability of test structure measurements from individual wafers and dies. This leads to the following two drawbacks:

- Since a design-time process variation model must capture variations across all wafers and lots, it results in a significantly more loosely distributed or pessimistic variation model than could be obtained based on even limited measurements from one or all the wafers within a lot. Figure 1 shows 4 wafers from the same 65nm process but two different lots. The wafers of lot 1 in Figure 1(a) and (b) have similar wafer-level global trends, which are quite different from the ones of lot 2 in (c) and (d). In other words, if we can model the process variation of a wafer by using information from post-fabrication measurements on the same or another wafer, the uncertainty of the model may be significantly reduced, which helps mitigate the unnecessary pessimism in post-silicon applications.
- The extraction of a design-time model assumes that the process remains constant after the model has been generated using test wafers. However, in practice, the process recipe for test wafers run at design time may not correlate well with the one for production chips [12]. In addition, process is continually optimized and the variation may change over time [17]. This is only captured by periodically running new test wafers, which is uncommon and expensive.

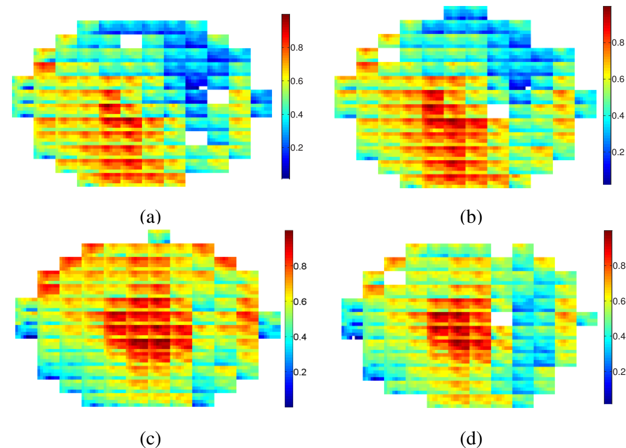


Fig. 1. Ring oscillator (RO) frequency measurements (scaled) of 4 wafers from 2 different lots in a 65nm process: (a) and (b) are wafers from lot 1; (c) and (d) are wafers from lot 2. The 2 lots have different global trends.

Instead, what is needed is a dynamic post-silicon variation model that automatically tracks and adapts to process changes using limited test measurements. Such a model is not only useful for post-silicon applications but could also be used for future designs in the same process.

However, constructing such a model is not trivial. It is common today to deploy hundreds or even thousands of test structures (*e.g.*, ring oscillators or resistor arrays) within a product chip or in the scribe line for process monitoring [17]. But the overhead to measure all structures for all dies across all wafers is clearly too high [17], [18]. It is also unclear how to reuse measured information from earlier wafers to facilitate the modeling on a different wafer or lot. To address these issues, we propose a new framework where we dynamically extract a variation model from measurements using wafers of product chips that are instrumented with small test structures. The extracted model accounts for both systematic and spatially-correlated patterns as well as random variations.

978-1-4244-8194-1/10/$26.00 ©2010 IEEE

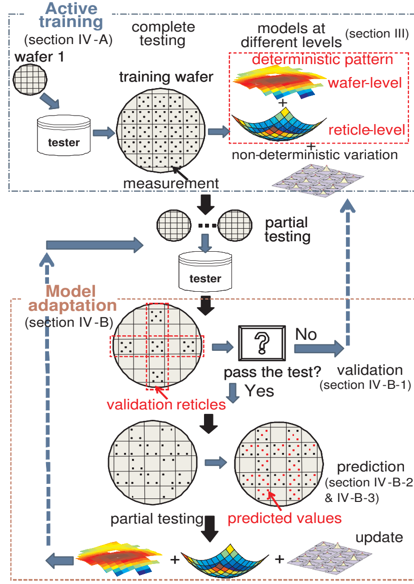


Fig. 2. Flow chart of the proposed framework

Prior works in [6] and [19] also attempt to reduce the number of measurements to monitor process but either require simulations on tens or hundreds of wafers to achieve converged results or limit their analysis only to the current wafer under test. In addition, they are fixed approaches that do not dynamically adapt to process change. The proposed active learning framework *allows the process model to evolve by reusing information from past wafers to validate and improve the model*. The flow chart in Figure 2 gives an overview of the proposed framework. In particular, initial measurements are conducted for model training with high measurement density which will then be reduced when the model fidelity increases. In this manner the number of measurements is gradually decreased over time. This allows the model to adapt and improve with the process changes while reducing the test cost to a minimum. The framework has the following key modeling contributions:

- **Hierarchical process variation modeling**. We develop a hierarchical variation model that incorporates wafer-to-wafer, across-wafer, reticle-to-reticle, across-reticle and independent variation components, accounting for systematic, spatially correlated and random variations. The variation is extracted on a reticle basis by noting the design-process interactions in lithography steps.
- **Active training**. Active training initializes the active learning models in the framework (Figure 2). This stage completely measures the initial wafer set to achieve deterministic spatial pattern models and quantify the uncertainty reduction ability of each test site.
- **Spatially correlated variation characterization**. We employ a sparse Bayesian learning method [20] to estimate the spatially correlated variation. Measurements from earlier wafers are used to identify the significance of bases to speed up the estimation.
- **Adaptive test configuration determination**. Test configuration is defined as a selective set of m out of n available test structures. Each measurement may reduce the model uncertainty to a different degree. This algorithm resolves how many and which measurements are conducted for a desirable accuracy.
- **Model validation and adjustment**. For an untested wafer, we apply several statistical tests on a selective set of reticles to justify if the existing model needs an incremental adjustment or a complete re-construction due to process drift.

The observation and experimental results in this paper are based on two industrial processes with two different types of test structures and different reticle sizes. Process 1 is a 65nm technology process and has approximately 300 wafers with embedded ring oscillators (RO). The test structures within each reticle are coarse-grained but there are approximately 100 reticles within in each wafer. Process 2 is a 130nm technology process and has 5 wafers with electrical linewidth measurement (ELM). The test structures within each reticle are fine-grained with 23 reticles in each wafer. The generality and efficiency of the proposed framework is validated on both processes for the two different test structures, reticle sizes and measurement densities. Experimental results show that the proposed framework can achieve 83% and 78% variance reduction for two processes in comparison to design-time models. Meanwhile, the average estimation error is well maintained at ∼2-3% using merely ∼30% available test structures for two processes. Compared to a recently reported approach in [19], the framework can further reduce the test cost by more than 37% to achieve the same or better accuracy.

II. Statistical Preliminaries

In this section we briefly review several statistical techniques that will be used throughout the paper. The details can be found in [20]–[22].

A. Robust Regression

The deterministic spatial pattern can be fitted from measurements to a given function. Least square fitting may be easily impacted by outliers or long-tailed error distribution. Robust regression is an alternative estimator to minimize fitting errors with the following term [21]:

$$\sum \rho(y_i - x_i^T \beta) \quad (1)$$

where $y_i - x_i^T \beta$ is the i_{th} estimation error and ρ is a weighting function to mitigate the impact of outliers. The details can be found in [21].

B. Statistical Tests

In the model validation step of Figure 2, we need to justify any model before applying it to an untested wafer. Since the non-deterministic variation for a device is typically modeled as a Gaussian random variable, we can use the following statistical tests in our framework [21]:

- *t-test* checks whether the mean of a normal distribution has a value specified in a null hypothesis. In the framework, it is used to justify if a predictor in the polynomial model is statistically significant.
- χ^2 *goodness-of-fit test* describes how well the given model fits a set of observations. This test is used to check the overall fitting goodness of an existing model, such as the fitting goodness of a wafer-level pattern model for the raw data from another wafer.
- *Kurtosis* is a measure of the peakedness of the distribution. We use this measure in our framework to justify if total variance distribution has a fundamental change and hence requires a re-fitting.

C. Sparse Bayesian Learning

This section gives a brief review of sparse Bayesian learning (SBL), the details of which can be found in [20], [22]. The basic idea of sparse Bayesian learning is to solve the following under-determined system through a Bayesian inference [20]:

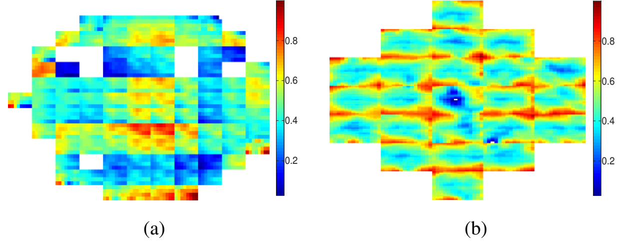
$$\mathbf{t} = \Phi \mathbf{w} + \epsilon \quad (2)$$

where Φ is a $m \times n$ matrix with each column called a basis and $m \leq n$, ϵ is typically considered to be zero mean Gaussian random variables with variance of σ^2. Given a $m \times 1$ vector $\mathbf{t}$ and a priori knowledge that $\mathbf{w}$ is *sparse*, SBL can find the most probable estimation of $\mathbf{w}$ and the corresponding co-variance matrix. The target vector $\mathbf{t}$ has a multi-variate Gaussian likelihood [20]:

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = (2\pi)^{-m/2} \sigma^{-m} e^{-\|t-\Phi \mathbf{w}\|^2/(2\sigma^2)} \quad (3)$$

and the prior over the parameters $\mathbf{w}$ is a zero mean Gaussian:

$$p(\mathbf{w}|\alpha) = (2\pi)^{-n/2} \prod_{i=1}^{n} \alpha_i^{1/2} e^{-\alpha_i w_i^2/2} \quad (4)$$



(a) (b)

Fig. 3. Wafer-level contours with systematic patterns removed for (a): RO measurements (scaled) for a wafer in 65nm process; (b): ELM measurements (scaled) for a wafer in 130nm process.

where $\alpha = (\alpha_1 ... \alpha_n)^T$ are n independent hyper-parameters, one per weight w_i, which represents the inverse of variance for $\mathbf{w}$ and pushes the solution to be sparse. The proof of using Gaussian priors to achieve sparsity is detailed in [20]. Given α, the posterior distribution is then a Gaussian and can be analytically written as:

$$p(\mathbf{w}|\mathbf{t}, \alpha, \sigma^2) = p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)/p(\mathbf{t}|\alpha, \sigma^2) = N(\mathbf{w}|\mu, \mathbf{\Sigma}) \quad (5)$$

with

$$\mu = \sigma^{-2}\Sigma\Phi^T\mathbf{t} \qquad \Sigma = (A + \sigma^{-2}\Phi^T\Phi)^{-1} \quad (6)$$

where A is diag$(\alpha_1, ...\alpha_n)$.

The posterior distribution can be achieved by Expectation Maximization method, which is equivalent to solving a type-II marginal likelihood maximization with respect to the hyper-parameters α [20]:

$$\begin{aligned} \mathbf{Max} \quad L(\alpha) &= logp(\mathbf{t}|\alpha, \sigma^2) = log\int p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)d\mathbf{w} \\ &= -1/2[Nlog2\pi + log|C| + \mathbf{t}^\mathbf{T}C^{-1}\mathbf{t}] \end{aligned} \quad (7)$$

with $C = \sigma^2 + \Phi A^{-1}\Phi^T$. Once the most probable α_{MP} are found, they can be plugged into (6) to get μ_{MP} and the covariance matrix Σ_{MP}.

III. Hierarchical Modeling and Characterization of Process Variation

In this section we discuss the variation modeling at different spatial levels and the characterization of deterministic and non-deterministic parts of the model when complete or partial testing is conducted.

A. Hierarchical Modeling of Process Variation

Traditionally the process variation of a device is considered to be a compound effect of inter-die, intra-die spatially correlated and random (or residual) variations [5]. Recent works investigate the origins of process and propose that a great portion of within-die spatially correlated variation is actually caused by deterministic across-wafer and across-reticle spatial patterns [7], [8]. By not recognizing systematic patterns at the reticle or wafer level, pessimism is unnecessarily increased, attributing more variation than is actually present [7], [18]. On the other hand, only extracting deterministic global trend but ignoring the non-deterministic spatially correlated variations not only obscures wafer-level trend but also leaves too much unevenly distributed across-reticle variation to the residual. Figure 3 demonstrates the wafer-level contours with systematic patterns removed using a similar methodology in [7], [8]. Either for RO frequency in Figure 3(a) from process 1 or ELM in Figure 3(b) from process 2, it can be observed the non-uniformity across the wafer and certain spatially correlated patterns within the reticle. Residual variation is supposed to be independent and evenly distributed, which is unable to explain Figure 3. Thus, it is necessary to include spatially correlated variations in the model.

Without loss of generality, we denote z as the measurable process parameter of interest, which can be either a physical parameter or a parametric quantity. We model z as a location dependent random variable including seven distinct parts:

$$z(x, y) = z_0 + z_{iw} + z_w(x, y) + z_r(x_0, y_0) + z_{ir} + z_{ar} + r \quad (8)$$

z_0 is the nominal design specification and turns out to be a constant for any device. z_{iw} is the inter-wafer variation that captures the long-term drifts in tools and process difference from wafer to wafer. $z_w(x, y)$ is the deterministic across-wafer spatial pattern and $(x,\ y)$ is the location within a wafer. Such pattern may be caused by post-exposure bake (PEB) temperature non-uniformity, or resist thickness variation. $z_r(x_0, y_0)$ is the reticle-level spatial pattern where $(x_0,\ y_0)$ is the location within a reticle. This component is primarily due to design-process interactions in the lithography steps, like lens abbreviation. Both $z_w(x, y)$ and $z_r(x_0, y_0)$ are deterministic global patterns. z_{ir} is inter-reticle variation component, which may be caused by the light source change. z_{ar} is across-reticle spatially correlated random variation. For all the devices within one reticle, z_{ar} can be understood as a zero mean multi-variate Gaussian random vector. z_{ar} may be caused by the proximity effect or coma and result in uneven within-reticle contour as in Figure 3. Finally, r is the independent residual variation caused by local random effect, and typically modeled as an independent Gaussian variable[1]. The only assumption we hold in our model is the variation type of Gaussian, which has been validated by many characterization works [8], [17].

The proposed framework extracts variation on a reticle basis, which may include one or several dies. The reticle directly interacts with the lithography steps and exhibits certain regularity from reticle to reticle. Assume there are $m_1 \times n_1$ available test structures on a reticle with each representing one random variable as defined in (8). Then an $m_1 \times n_1$ matrix, denoted as A_i for the i_{th} reticle, can be constructed, which uniquely identifies the process variation of the reticle. According to (8), A_i can be written as:

$$A_i = z_0 + z_{iw} + z_{ir,i} + A_{w,i} + A_{r,i} + A_{ar,i} + R_i \quad (9)$$

where $A_{w,i}$ and $A_{r,i}$ are the matrices for wafer- and reticle-level systematic patterns observed in the i_{th} reticle, $A_{ar,i}$ and R_i are the matrices for the non-deterministic variations representing across-reticle spatially correlated and independent residual variations, respectively. It is noted that $A_{r,i}$ is a deterministic pattern at the reticle scale and hence the same from reticle to reticle. Thus we can rewrite $A_{r,i}$ as A_r.

B. Variation Characterization at Different Spatial Scales

1) Model Identification of Systematic Spatial Patterns: Deterministic spatial patterns are extracted by completely testing a set of reticles A_is. To split the mixed effect of two patterns, it is essential to first extract the pattern at lower scale, $i.e.$ reticle-level pattern. In order to mitigate the impact of wafer-level pattern and other variation components in (8), we take the average of the matrices of A_is:

$$\overline{A} = \frac{1}{l}\sum_{i=1}^{l} A_i = z_0 + z_{iw} + \frac{1}{l}\sum_{i=1}^{l}(A_{w,i} + A_r + z_{ir,i} + A_{ar,i} + R_i) \quad (10)$$

where l is the number of reticles. By carefully analyzing the characteristics of each component, we have the following observations:

- Any device in the wafer observes the same z_0 and z_{iw}.
- Each entry in R_i is an independent Gaussian with zero mean and the same variance and hence can be cancelled after taking average.
- $z_{ir,i}$ is a constant for all the devices within one reticle. Thus, any entry in $\overline{A}$ observes the same constant $\frac{1}{l}\sum_{i=1}^{l} z_{ir,i}$.
- For an entry at a specific location of $A_{ar,i}$ across different reticles, it still follows a zero mean Gaussian. Thus, $\frac{1}{l}\sum_{i=1}^{l} A_{ar,i} \approx 0$.
- The wafer-level pattern typically has symmetric characteristics, $e.g.$ slanted or parabolic surface [8]. By carefully choosing the reticles symmetrically placed on a wafer, the difference among entries in $\frac{1}{l}\sum_{i=1}^{l} A_{w,i}$ is limited and can be approximated to a constant.

Based on those observations, (10) can be simplified to:

$$\overline{A} \approx A_r + const \quad (11)$$

[1] For measurement results, r may also include the measurement white noise.

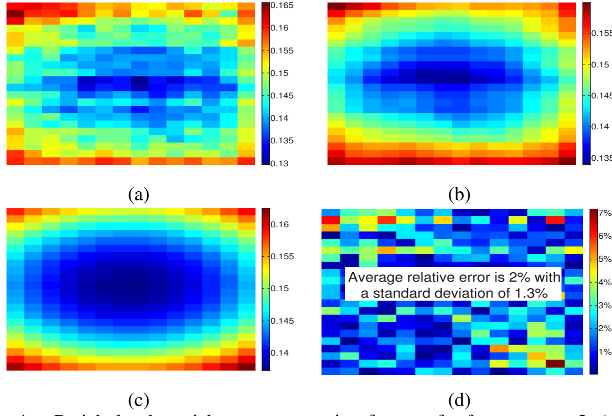


Fig. 4. Reticle-level spatial pattern extraction for a wafer from process 2. (a): the extracted reticle-level common pattern $\overline{A}$; (b): smoothed result after MA; (c): estimated model using backward elimination and robust regression; (d): relative error of the fitted result in (c) in comparison with the original pattern in (a).

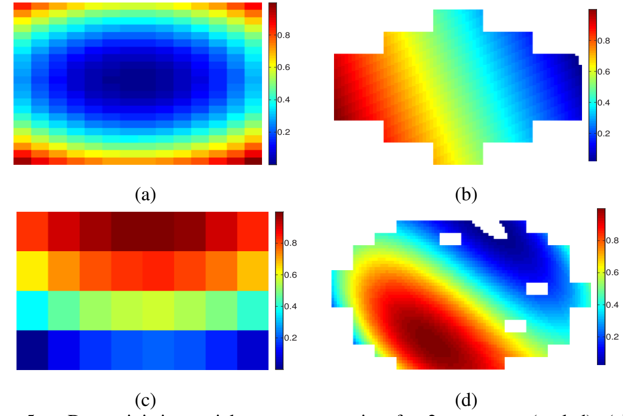


Fig. 5. Deterministic spatial pattern extraction for 2 processes (scaled). (a): reticle-level pattern and (b): wafer-level pattern (slanted) for 130nm process; (c): reticle-level pattern and (d): wafer-level pattern (cubic) for 65nm process.

where $const \approx z_0 + z_{iw} + \frac{1}{l}\sum_{i=1}^{l}(A_{w,i} + z_{ir,i})$ is a constant. Thus, $\overline{A}$ is a compound effect of reticle-level pattern and the average of other variation components. We name it as the reticle-level common pattern.

Given $\overline{A}$ and its corresponding reticle-level locations ($\mathbf{X}_0$, $\mathbf{Y}_0$), a two-dimensional closed-form function can be fitted. The impact of outlier can be mitigated by using moving average (MA) and robust regression (section II-A) [21]. Moving average helps smooth over rapidly varying features by moving a $m_0 \times n_0$ sub-block across $\overline{A}$ and replacing the original entry with the average of the sub-block. Figure 4 demonstrates a reticle-level pattern extraction example including: (a) the pattern $\overline{A}$ extracted from process 2 using 23 reticles in a wafer and (b) the change after moving average is conducted.

Before the regression is applied, we need to determine the model to fit. We here employ a backward elimination strategy and make the data find the model itself. For example, if the initial model is a 2_{nd} order full polynomial model, the fitted model by robust regression is then:

$$\mathbf{t}_{\overline{\mathbf{A}}} \sim [1, \mathbf{x}_0, \mathbf{y}_0, \mathbf{x_0 y_0}, \mathbf{x_0^2}, \mathbf{y_0^2}] \times \mathbf{p^T} \quad (12)$$

where $\mathbf{p} = [p_0, p_1 ... p_5]$ is the parameter vector to be fitted, $\mathbf{t}_{\overline{\mathbf{A}}}$, $\mathbf{x}_0$ and $\mathbf{y}_0$ are the vectorization results of matrices $\overline{A}$, $\mathbf{X}_0$ and $\mathbf{Y}_0$ respectively. A t-test is conducted on each parameter in $\mathbf{p}$ to compute the corresponding p-value. The most statistically insignificant predictor in $[1, \mathbf{x}_0, \mathbf{y}_0, \mathbf{x_0 y_0}, \mathbf{x_0^2}, \mathbf{y_0^2}]$ is then removed to simplify the model. This procedure is performed repeatedly till all the terms (predictors) in the polynomial model $f_r(\mathbf{X}_0, \mathbf{Y}_0)$ is significant. Figure 4(c) illustrates the acquired model using backward elimination and robust regression and (d) exhibits the relative error of fitting compared with the original pattern $\overline{A}$ in (a). It is observed approximately 2% average relative error, which indicates the necessity of modeling reticle-level patterns.

After reticle-level pattern is extracted and removed from the raw data, the wafer-level pattern $f_w(\mathbf{X}, \mathbf{Y})$ can be extracted in a similar way from the data at wafer-level, except that the coordinate matrices are now at wafer-level instead of reticle-level. Figure 5 demonstrates the reticle-level and wafer-level patterns for the wafers from two processes. The algorithm of deterministic pattern extraction is summarized in Figure 6.

2) Non-Deterministic Variation Estimation through SBL: After the deterministic patterns are removed from the model, the residual part is comprised of across-reticle spatially correlated and residual variations[2]:

$$A_{random} = A_i - \overline{A} - A_w(x, y) \approx A_{ar} + A_r \quad (13)$$

[2]In practice the model is unable to fit perfectly with the data. A_{ar} in the residual model of (13) is a compound effect of actual across-reticle spatially correlated variation and the residual from fitting or modeling approximation. The fitting or modeling approximation terms can be lumped into the mean of A_{ar}, and captured by SBL method afterwards. For simplicity we still use A_{ar} here to represent this compound effect.

Procedure: *Deterministic spatial pattern extraction*

Input: the raw data matrix A and the corresponding coordinate matrix $\mathbf{X}$, $\mathbf{Y}$, sub-block size $[m_0, n_0]$ for MA, initial polynomial model order n

Output: the fitted polynomial model $f_n(\mathbf{X}, \mathbf{Y})$, the confidence interval $C = [lb, ub]$ for the fitted parameters in $f_n(\mathbf{X}, \mathbf{Y})$

1: Perform MA on A to achieve the smoothed matrix A_s;
2: Vectorize A_s, $\mathbf{X}$, $\mathbf{Y}$ to $\mathbf{t}_{\mathbf{A_s}}$, $\mathbf{x}$ and $\mathbf{y}$;
3: Construct a full n_{th}-order polynomial model $f_n(\mathbf{x}, \mathbf{y})$;
4: **While TRUE**
5: Fit $\mathbf{t}_{\mathbf{A_s}} \sim f_n(\mathbf{x}, \mathbf{y})$ to achieve the parameters in $\mathbf{p}$ and achieve its confidence interval $C = [lb, ub]$;
6: Perform t-test on each parameter in $\mathbf{p}$;
7: **If** all p-values are statistically significant
8: **return** f_n and corresponding coefficients;
9: **else**;
10: find the most insignificant term and remove it from the model f_n;
11: **endif**;
12: **end while**;

Fig. 6. Algorithm for deterministic spatial pattern extraction

The non-deterministic part can be fully known only when a complete testing is conducted. Thus, a natural question is, can we characterize the reticle-level variation with certain accuracy when only partial testing is conducted? We propose to handle this problem using SBL.

Assume there are n entries within each reticle. The matrices A_{random}, A_{ar} and A_r can be vectorized to $n \times 1$ vectors, $\mathbf{t_{random}}$, $\mathbf{t_{ar}}$ and $\mathbf{t_r}$, respectively. When m measurements are conducted ($m \leq n$), we have:

$$\mathbf{t_m} = B\mathbf{t_{random}} = B\mathbf{t_{ar}} + \mathbf{t_{r,m\times 1}} \quad (14)$$

where $\mathbf{t_m}$ is a $m \times 1$ vector, and B is a $m \times n$ selection matrix. Any row of B is a unit vector e_i, with the i_{th} entry equal to 1 and the other entries equal to 0. Since independent residual variations $\mathbf{t_r}$ cannot be estimated but only bounded, the question turns out to be, given measurement of $\mathbf{t_m}$ known, how to characterize $\mathbf{t_{ar}}$ which are masked by $\mathbf{t_{r,m\times 1}}$?

We can associate (14) with SBL by applying a sparsity inducing transform Ψ^T on $\mathbf{t_{ar}}$ [20], [23]:

$$\mathbf{w} = \Psi^T \mathbf{t_{ar}} \quad (15)$$

where Ψ^T may be an orthogonal transform matrix for either discrete cosine transform (DCT) or discrete wavelet transform (DWT), *i.e.*, $\Phi^T = \Phi^{-1}$, and $\mathbf{w}$ is sparse or has a few entries that are more significant than the rest. (14) can now be written as:

$$\mathbf{t_m} = B\Psi\mathbf{w} + \mathbf{t_r} = \Phi\mathbf{w} + \mathbf{t_r} \quad (16)$$

where $\Phi = B\Psi$, and $\mathbf{w}$ is sparse with k significant entries (k usually is much smaller than m). $\mathbf{w}$ and $\mathbf{t_{ar}}$ have a canonical one-to-one

relationship as in (15). Thus if $\mathbf{w}$ is accurately estimated, we can always recover $\mathbf{t_{ar}}$ from $\mathbf{w}$. If the sparsity inducing transform Ψ is applied to $\mathbf{t_{random}}$ instead of $\mathbf{t_{ar}}$, *i.e.*, $\mathbf{t_m} = B\mathbf{t_{random}} = B\Psi\mathbf{w}$, it turns out to be a compressive sensing problem as in [19]. Compressive sensing is an alternative application of SBL [20], which also requires $\mathbf{w}$ to be sparse and can be solved by linear regression in a point estimate manner [19], [23]. However, including independent variation into the transform may induce high frequency components and hence the non-sparsity in the frequency domain, thereby limiting the efficiency of compressive sensing. As a result we separate $\mathbf{t_{ar}}$ from $\mathbf{t_{random}}$ to maintain the sparsity in $\mathbf{w}$.

In our framework, the SBL method in [22] is applied to (16) to predict $\mathbf{w}$ through the measurements of $\mathbf{t_m}$. Similar as section II-C, we have:

$$\begin{aligned}\mathbf{Max} \quad L(\alpha) &= log \int p(\mathbf{t_m}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\alpha) d\mathbf{w} \\ &= -1/2[Nlog2\pi + log|C| + \mathbf{t_m^T} C^{-1}\mathbf{t_m}]\end{aligned} \tag{17}$$

(17) can be solved by either using a variational Bayesian inference or more directly in a constructive manner by adding/deleting the candidate basis (column of Φ) into/from the solution model till the likelihood is converged [20]–[22]. Then, with the estimated $\mu_{\mathbf{MP}}$ and its covariance matrix Σ_{MP} in (6), the distribution of $\mathbf{t}_{ar}$ given $\mathbf{t_m}$ can be written as:

$$\mathbf{t}_{ar}|\mathbf{t}_m \sim N(\Psi\mu_{\mathbf{MP}}, \Omega), \quad \Omega = \Psi\Sigma_{MP}\Psi^T \tag{18}$$

However, to improve the efficiency of SBL in variation extraction, we still need to determine the best test sites to be measured (or the selection matrix B) and the order to add the bases by information collected from the test wafers, which will be detailed in the next section.

IV. Active Learning Framework for Variation Extraction

The framework is composed of two major stages, active training to learn the models and model adaptation to adjust the models (Figure 2). In this section we will discuss the key modules in those two stages.

A. Active Training

In the active training stage, the framework learns the models by densely measuring each test structure in a wafer (or training wafer set). Several tasks are supposed to be conducted by learning the features of the measurements (denoted as W), including:

- *Model identification* that identifies both the wafer- and reticle-level deterministic spatial patterns.
- *Uncertainty exploration* that exploits the uncertainty reduction each measurement can contribute and scores the contribution.
- *Basis significance ranking* that gives an initial order of adding bases into the solution when solving (17).
- *Linking variance to prediction accuracy* that sheds insight into the control of the estimation accuracy.

Once all those models are characterized in training stage, we start to process the forthcoming wafers with the learned models.

1) Model Identification: Most of the model identification details are discussed in section III-B1. Here we briefly summarize the flow to extract wafer- and reticle-level deterministic patterns in Figure 7. After the global trends are removed from W, we denote the residual raw data as W_r, which enables the exploration in the next three sections.

2) Uncertainty Exploration: Any measurement may provide a certain amount of information and hence reduce the model uncertainty. If all the available test structures are measured, the uncertainty of a model is exactly zero. At training stage we do not have any given variation information. Although it is always preferred to conduct the measurements that reduce the uncertainty to a maximum, it remains unclear how to quantitatively evaluate the uncertainty each measurement can reduce.

Here we propose a simple yet efficient method to evaluate the uncertainty reduction ability for any test site. Denote the raw data of a reticle from W_r as $A_{random,i}$. If all n test structures in the reticle

Procedure: *Model identification to extract spatial patterns*

Input: the raw data W of wafer 1 and the corresponding coordinate matrix X, Y

Output: the fitted polynomial model $f_w(X,Y)$ and $f_r(X_0,Y_0)$, confidence interval $C_w = [lb_w, ub_w]$ for the parameters in $f_w(X,Y)$ and $C_r = [lb_r, ub_r]$ for $f_r(X_0,Y_0)$

1: Compute the reticle-level coordinate matrix (X_0, Y_0);
2: Compute the averaged matrix $\overline{A}$ from all the reticles in W;
3: Extract the reticle-level common pattern model $f_r(X_0,Y_0)$ from $\overline{A}$ using the algorithm in Figure 6;
4: Compute the residual raw data with reticle-level common pattern removed, name it W_g;
5: Extract the wafer-level spatial pattern model $f_w(X,Y)$ from W_g using the algorithm in Figure 6;

Fig. 7. Algorithm to identify the wafer- and reticle-level pattern models

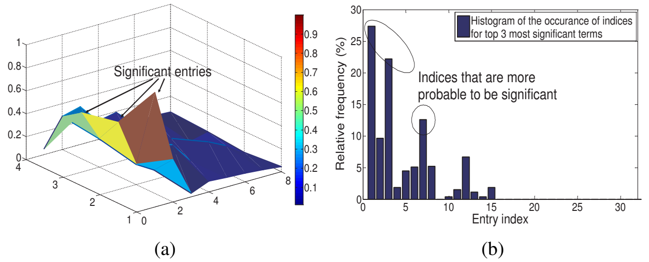


Fig. 8. (a): Scaled DCT transform for a reticle with deterministic patterns removed; (b): Histogram of indices for entries in $\mathbf{w}$ that fall into the top three most significant entry set. The results are collected from 100 reticles of a wafer.

are measured, the uncertainty of the model for this particular reticle is 0, *i.e.*, $U(A_{random,i})$=0, where $U(.)$ denotes the model uncertainty. On the other hand, if there is only one test structure (j_{th}) unmeasured, SBL in section III-B2 can estimate its value as well as a covariance matrix Ω_j as in (18). Clearly the variance terms in Ω_j is due to the unmeasured site, *i.e.* the uncertainty is attributed to the j_{th} site. Now define total variance as a measure of uncertainty,

Definition 1: Total variance is defined as the sum of variance for each test site, *i.e.*, $TV(j) = tr(\Omega_j)$, where $tr(.)$ computes the sum of diagonal entries and Ω_j is the estimation covariance matrix with j_{th} site unmeasured.

Then the uncertainty reduction by the j_{th} site can be approximated by:

$$\Delta U(j) = TV(j) \tag{19}$$

In this manner, we can check each site in the reticle and name the resulted vector as $\mathbf{\Delta}U_i$ for $A_{random,i}$. To mitigate some local effects, we will compute $\mathbf{\Delta}U_i$ for a representative set of l reticles (*e.g.*, the reticles in the middle column and middle row of the training wafer) and then take the average of them, $\overline{\mathbf{\Delta}U} = \sum \mathbf{\Delta}U_i/\mathrm{l}$, as a measure of the uncertainty reduction ability. The normalized vector $\mathbf{S}_u = \overline{\mathbf{\Delta}U}/||\overline{\mathbf{\Delta}U}||$ is considered as the uncertainty score for the sites within a reticle. A site with a higher score is always preferred to be tested first.

3) Basis Significance Ranking: SBL can solve (17) by adding/deleting the candidate bases (columns of Φ) to/from the model and then updating the corresponding hyper-parameters for the selected bases [20]. For an unselected basis, its hyper-parameter is infinity, and the corresponding entry in $\mathbf{w}$ is 0. In other words, the selected bases correspond to significant entries in $\mathbf{w}$. If we know those significant entries in advance, we can simply plug the corresponding candidate basis set into the solution model, which is beneficial for both run-time and accuracy.

Figure 8(a) illustrates the DCT transform of a reticle with global patterns extracted. It can be seen most entries in the frequency domain happen to be insignificant. On the other hand, the significant entries in $\mathbf{w}$ are limited to a small subset even from reticle to reticle. Figure 8(b) shows the histogram of top 3 most significant entries in $\mathbf{w}$, across

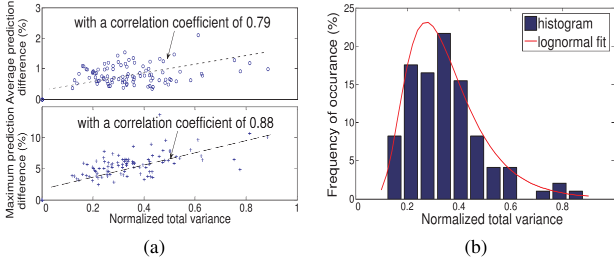


Fig. 9. (a): Strong correlation of prediction accuracy and total variance (TV); (b): Histogram of total variance across 100 reticles of a wafer for 65nm process

100 reticles within a wafer. Though there are potentially 32 entries, the entries that are probable to be very significant are actually a small subset. We then propose a score list S_s to rank the significance of each entry. The basis with a higher significance score in S_s is first selected into the solution model. In details, we first sort the significance of orthogonal transform coefficients $\mathbf{w}$ from the most to the least significant for each reticle. The rank vector, $\mathbf{r}_i$, for a reticle is then scaled and fed into a continuous score function to achieve the score for the bases:

$$\mathbf{S_{s,i}} = \exp(-p_s \times \mathbf{r}_i/max(\mathbf{r}_i)) \quad (20)$$

where p_s is a customized parameter to tune the slope of the exponential function. Then the score for basis significance is achieve by taking average of $\mathbf{S_{s,i}}$ for all the reticles, $i.e.$, $\mathbf{S_s} = \sum \mathbf{S_{s,i}}/l$.

4) Linking Variance to Prediction Accuracy: When a reticle is under partial testing, it is essential to know how accurate the prediction may be. However, the accuracy is unable to be known till all the measurements are conducted. Thus, we need to find another measure to quantify the quality of the estimation. By noting that the error is almost always positively correlated with the model uncertainty, we propose to use total variance (TV) in **Definition 1** for a given set of k_0 measurements as a measure of the prediction accuracy. Figure 9(a) shows trends of the average (top) and maximum (bottom) relative prediction error with respect to TV for 100 reticles of a wafer. Either figure exhibits a strong correlation with TV, with a correlation coefficient of 0.79 and 0.88, respectively. The relationship between TV and prediction accuracy is explored in the training stage by conducting k_0 measurements on each reticle according to $\mathbf{S_u}$ and then recording the total variance for each reticle. It is noted that we are attempting to evaluate estimation accuracy in a $qualitative$ instead of $quantitative$ way. The data collected from the training wafer helps describe the statistical behavior of TV in this variation space given k_0 measurements. Figure 9(b) shows the histogram of total variance from 100 reticles and its log-normal fit. This histogram is used to decide the number of measurements to be conducted, as in section IV-B2.

B. Model Adaptation

After models at different levels are constructed in training stage, the framework starts partial testing on the other wafers (from the same lot or different lots), which is the model adaptation stage.

1) Model Validation: It is essential to validate the models before applying them to an untested wafer. The model validation justifies if the deterministic pattern models $f_w(X,Y)$ and $f_r(X_0,Y_0)$ requires a complete reconstruction or just incremental adjustment. For the current wafer under test (WUT), the validation module selects a representative set of reticles. Then the χ^2 test in section II-B is used to evaluate the overall fitting goodness. If the overall fitting is good, a t-test is then applied to judge if any parameter in $f_w(X,Y)$ and $f_r(X_0,Y_0)$ needs adjustment. The validation flow is summarized as follows:

- Step 1: Compute the chi-square statistics χ^2 using the raw data from the representatives set of reticles [21].
- Step 2: If χ^2 is beyond the predefined tolerance bound, include the WUT into the training set and go back to the training stage. Otherwise, go to Step 3.

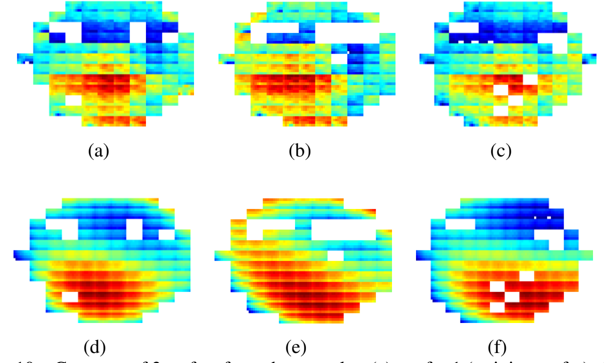


Fig. 10. Contours of 3 wafers from the same lot. (a): wafer 1 (training wafer); (b): wafer 2; (c): wafer 3; and the extracted deterministic spatial patterns (including both wafer- and reticle-level patterns) for (d): wafer 1; (e): wafer 2; (f): wafer 3.

- Step 3: Extract the wafer- and reticle-level patterns from the representative reticle set using the algorithms in section IV-A1 to achieve the comparison models $f_{w,c}(X,Y)$ and $f_{r,c}(X_0,Y_0)$.
- Step 4: Check if the parameters of $f_{w,c}(X,Y)$ (or $f_{r,c}(X_0,Y_0)$) are in the confidence intervals C_w (or C_r) for the parameters of $f_w(X,Y)$ (or $f_r(X_0,Y_0)$).
- Step 5: For those within the bounds, accept the original parameters in $f_w(X,Y)$ or $f_r(X_0,Y_0)$; for those beyond the bounds, re-fit the parameters using the representative set of reticles.

Figure 10 demonstrates the contours of three wafers from the same lot and how the deterministic pattern models evolve adaptively from wafer to wafer. It can be seen the deterministic patterns of wafer 2 and wafer 3 (Figure 10(e) and (f)) are based on the training patten (Figure 10(d)) but still capture the major features of the original wafer contours.

2) Adaptive Test Configuration Determination: The most important problem for partial test on a reticle is which and how many measurements to conduct. The uncertain score is a global evaluation across the reticles and not efficient enough for a particular reticle. To better understand the across-reticle process condition, the proposed partial test has two phases, within each, k_0 and k_1 measurements are conducted respectively.

Assume we have n_{max} available site for testing. In the first phase we conduct k_0 measurements on the reticle according to $\mathbf{S}_u$ and apply SBL to achieve the covariance matrix $\Omega_{S_{k_0}}$ [3]. k_0 is the same as the number used in section IV-A4 to characterize a TV histogram (or distribution with a cumulative density function $F(.)$). Then the total variance ($e.g.$, TV_i) for the reticle is computed from $\Omega_{S_{k_0}}$. The relative location of TV_i in the distribution $F(.)$ is mapped to the number of measurements to be conducted. In other words, we have:

$$k_1 \sim F(TV_i) \times (n_{max} - k_0) \quad (21)$$

Thus, if TV_i is large, almost all the measurements will be conducted. Meanwhile fewer measurements are required for smaller TV_i.

The second phase determines where to conduct the k_1 measurements. The underlying motivation is to maximize the uncertainty reduction by those measurements, $i.e.$,

$$\textbf{Max} \quad \Delta U_{S_{k_0} \cup S_{k_1}} - \Delta U_{S_{k_0}} \quad (22)$$

However, this formulation itself is difficult to evaluate or optimize. It is also noted that if we simply choose test sites according to S_u, most measurements may be conducted at some corner due to the spatial correlation. It is therefore desirable to conduct measurement on sites with higher S_u as well as relatively uniform distribution. The uniformity requirement is equivalent to the consideration of correlation among test sites. The closely placed sites usually have higher correlation, and hence should avoid repeated measurements. This intuition motivates a

[3]Those measurements are typically a very small subset S_{k_0} that helps understand the process condition without much extra overhead to the framework. k_0 can be determined at training stage by sweeping several typical numbers and comparing the results.

Procedure: *Summary of the active learning framework*	
Input: wafers	
Output: variation components decomposition, process parameter estimation and corresponding confidence intevals	
1:	Perform complete testing on the training wafer and actively learn models at different scales as in section IV-A;
2:	Validate the trained models for the next untested wafer as in section IV-B1;
3:	Achieve the adjusted wafer- and reticle-level spatial pattern models $f_w(X, Y)$ and $f_r(X_0, Y_0)$;
4:	Determine the test configuration for each untested reticle as in section IV-B2;
5:	Estimate the process parameter for untested devices, validate results using cross-validation and update the models;
6:	Move to the next wafer;

Fig. 11. Summary of the active learning framework

greedy search algorithm to maximize the covariance reduction instead of variance. Given the covariance matrix $\Omega_{S_{k_0}} = [\omega_1, \omega_2 ... \omega_n]$ for n test sites from phase 1 and ω_i is a column in $\Omega_{S_{k_0}}$,

- Step 1: set $S_{k_1} = \emptyset$.
- Step 2: choose the site i_0, which is:

$$i_0 = argmax_i \quad |\omega_i|^T \times [\mathbf{1}]_{n\times 1} \tag{23}$$

  where $[\mathbf{1}]_{n\times 1}$ is an $n \times 1$ all-one vector.
- Step 3: $S_{k_1} = S_{k_1} \bigcup i_0$. Remove the i_{0th} column and row in $\Omega_{S_{k_0}}$.
- Step 4: Go back to step 2 till k_1 sites are found.

3) Prediction and Model Update: After test configuration is determined, the framework moves forward to estimate the deterministic and non-deterministic variation components for any untested site. The deterministic components are calculated using those validated models, given the locations of the device in the reticle and wafer. The non-deterministic component is acquired by SBL in section III-B2. The estimated process parameter is then the sum of those components:

$$z_{esti} = f_w(x, y) + f_r(x_0, y_0) + \widetilde{z_{ar}} \tag{24}$$

where $\widetilde{z_{ar}}$ is the estimated spatially correlated variation component. The estimation also comes with a confidence interval that is computed from the covariance matrix Ω and the estimated σ^2 as in section III-B2 [20].

To ensure the quality of the prediction, we also employ a cross-validation stage by using another small subset S_{cv} of measurements to validate the results. Similar as S_{k_0}, S_{cv} is determined according to the uncertainty score $\mathbf{S}_u$ of those unmeasured sites. If the average error of the prediction results is unable to meet an error tolerance threshold θ_{cv}, more measurements will be conducted till cross validation requirement is satisfied. It is noted that we can always adjust θ_{cv} and n_{max} in section IV-B2 to make tradeoff between estimation accuracy and test cost (number of measurements).

After the prediction results are validated, several models needs updating before the framework moves to the next wafer:

- Total variance distribution. The total variance with S_{k_0} measurements can be collected from each reticle and included into the original TV data set. Kurtosis statistics is applied here to monitor if the distribution has a fundamental change. Otherwise the next wafer will be set to a training wafer to reconstruct the TV distribution.
- Basis significance score. According to the prediction values, the significance score for each reticle can be computed and then combined with the original score.
- Uncertainty score. The covariance matrix for each reticle is achieved from SBL which also sheds insight into the uncertainty reduction ability.

C. Summary of the Active Learning Framework

Here we summarize the complete algorithm for our active learning framework in Figure 11.

TABLE I
SIMULATION RESULTS OF THE PROPOSED FRAMEWORK ON TWO INDUSTRIAL PROCESSES (65NM AND 130NM)

proc.	#wafers	ave. err.	#failure	#measure	var. reduc.	time/wafer
1	288	0.8%	0.031%	45.9%	83%	20.3 sec
2	5	1.4%	0.025%	54.2%	78%	25.7 sec

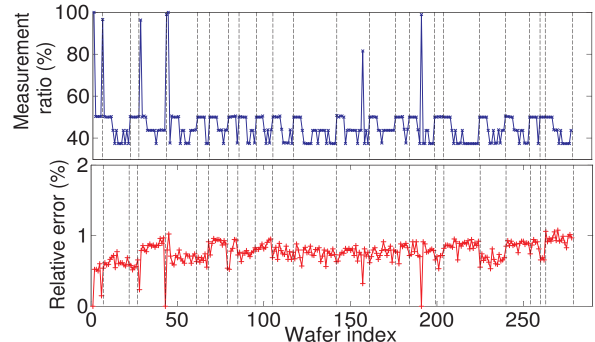


Fig. 12. Evolvement of relative average error (bottom) and measurement ratio (top) across the wafers and lots from process 1

V. EXPERIMENTAL RESULTS

In this section we demonstrate the efficiency and accuracy of the proposed framework based on the industrial measurement data from two processes in 65nm and 130nm technologies. All the experiments are conducted on a 2.0GHz Linux machine with 32GB RAM.

Table I summarizes the performance and accuracy of the framework, with a 2% average error tolerance in the cross-validation stage and with all the test structures available for measurement. Column 2 shows the total number of wafers for two processes. Column 3 is the average relative error which is computed by:

$$\frac{1}{N_t} \sum_{\text{any untested site}} \frac{|\text{estimated value} - \text{actual measurement}|}{\text{actual measurement}} \tag{25}$$

where N_t is the number of untested structures. Our framework can achieve 0.8% and 1.4% relative average error for two processes, respectively. In column 4, we present the relative failure number to evaluate the efficacy of the estimated confidence intervals. A failure is defined as an untested structure whose actual process (measurement) is beyond the estimated confidence interval. The relative failure is approximately 0.03% across all the lots. This is in good agreement with the expected failure of 0.1% from 3σ bounds. Column 5 exhibits the ratio of the number of measurements over the total number of available test structures. The test cost reduction is up to 50% to achieve an average error of $\sim$1%. The greater saving of test cost can be achieved by loosing the error tolerance, which will be presented shortly in this section. Column 6 is the average variance reduction of the proposed post-silicon variation model in comparison with the traditional design-time variation models (characterized from all the wafers). Even with half the test structures measured, the model variance can be reduced by approximately 80% (5$\times$ tighter) for either process, which may significantly reduce the pessimism in post-silicon applications. The average run time per wafer is listed in the last column and is expected to be smaller with increased number of wafers. Figure 13 further illustrates the significantly tightened post-silicon process variation models for 4 dies from 2 different wafers in 2 different lots. In the contrast, the design-time model is widely distributed because the wafer/reticle specific data is not available at design-time.

Figure 12 shows the evolution of the average relative error (bottom) and test measurement ratio (top) from wafer to wafer for all the lots of process 1. The black dashed line denotes the transition from one lot to another. The spikes of the measurement ratio at some of the transitions are due to the global pattern difference between two different lots (as shown in Figure 1). However, the framework can adapt the model to

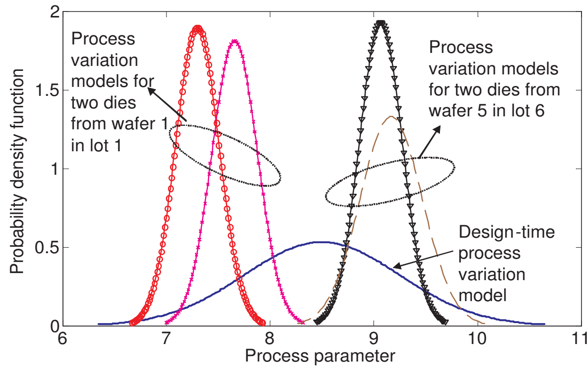


Fig. 13. Comparison of the post-silicon variation models for 4 dies from 2 wafers in 2 lots and traditional design-time process variation model

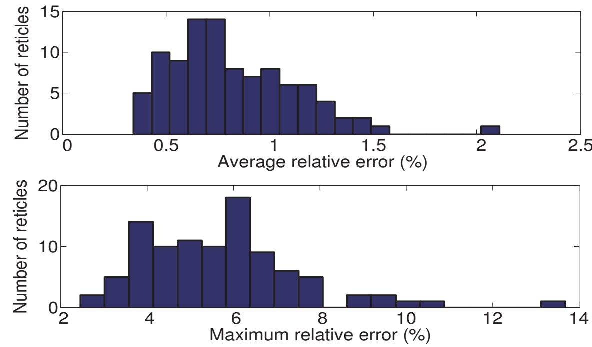


Fig. 14. Histograms of average (top) and maximum (bottom) relative error for all the reticles in a wafer

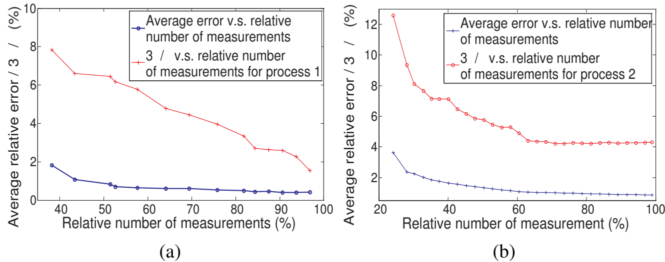


Fig. 15. Trend of average error and scaled average variance ($3\sigma/\mu$) reduction with an increased number of measurements for (a): 65nm and (b): 130nm process

capture this difference. Across the lots, the relative average error is well maintained at approximately 1%.

Figure 14 presents the histograms of average relative error and maximum relative error for 100 reticles from the same wafer. Most reticles have limited average error of approximately 1% and maximum error smaller than 10%. Figure 15 clearly demonstrates the reduction trend of both average error and scaled average variance (=$3\sigma/\mu$) with an increased number of measurements for both 65nm and 130nm processes. With approximately 30% available test structures, the framework can still achieve ∼2-3% average relative errors for two processes. The accuracy of the proposed framework is compared in Figure 16 with another two methods, the virtual probe method in [19] and a bilinear interpolation method on the wafers of 130nm process. The proposed framework can achieve a better accuracy with much fewer measurements. For the same accuracy of approximately 2% relative error, the test cost is reduced by 37% and 75% compared with [19] and bilinear interpolation, respectively.

VI. Conclusions

This paper proposes an active learning framework to extract process variation from measurements and reduce test cost. Several techniques are developed to model the variation. By reusing a priori knowledge from earlier wafers, the partial test can be conducted on the forthcoming wafers to achieve the required accuracy and test cost. Experimental results show that the framework can achieve an accuracy of ∼2-3% relative error using only ∼30% test structures for two industrial processes.

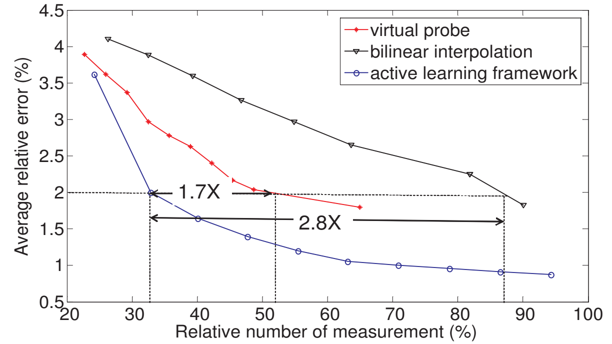


Fig. 16. Average relative error comparison with increased measurements for the wafers from process 2 using the active learning framework, virtual probe method from [19] and bilinear uniform interpolation

VII. Acknowledgement

The authors gratefully acknowledge the National Science Foundation (NSF) for supporting this work.

References

[1] S. Borkar, $et.al$. Parameter variations and impact on circuits and microarchitecture. In *Proc. DAC*, pages 338–342, 2003.

[2] J. Singh, $et.al$. Robust gate sizing by geometric programming. In *Proc. DAC*, pages 315–320, 2005.

[3] C. Zhuo, $et.al$. Design time body bias selection for parametric yield improvement. In *Proc. ASPDAC*, pages 681-688, 2010.

[4] B. Stine, $et.al$. Analysis and decomposition of spatial variation in integrated circuit processes and devices. *IEEE TSM*, vol. 10, no. 1:24–41, 1997.

[5] J. Xiong, $et.al$. Robust extraction of spatial correlation. In *Proc. ISPD*, pages 2–9, 2006.

[6] S. Reda and S. Nassif. Analyzing the impact of process variations on parametric measurements: novel models and applications. In *Proc. DATE*, pages 375–380, 2009.

[7] L. Cheng, $et.al$. Physically justifiable die-level modeling of spatial variation in view of systematic across wafer variability. In *Proc. DAC*, pages 104–109, 2009.

[8] K. Qian and C. J. Spanos. A comprehensive model of process variability for statistical timing optimization. In *Proc. SPIE*, pages 1–11, 2008.

[9] H. Chang and S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single pert-like traversal. In *Proc. ICCAD*, pages 621–625, 2003.

[10] C. Visweswariah, $et.al$. First-order incremental block-based statistical timing analysis. In *Proc. DAC*, pages 331–336, 2004.

[11] D. Beece, $et.al$. Transistor sizing of custom high-performance digital circuits with parametric yield considerations. In *Proc. DAC*, pages 781–786, 2010.

[12] M. Ketchen and M. Bhushan. Product-representative at-speed test structures for cmos characterization. *IBM JRD*, vol.50 no.4/5:451–468, 2006.

[13] M. Mani, A. Singh, and M. Orshansky. Joint design-time and postsilicon minimization of parametric yield loss using adjustable robust optimization. In *Proc. ICCAD*, pages 19–26, 2006.

[14] Q. Liu and S. Sapatnekar. Synthesizing a representative critical path for post-silicon delay prediction. In *Proc. ISPD*, pages 183–190, 2009.

[15] M. Bushnell and V. Agrawal. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Kluwer Academic Publishers, 2000

[16] C. Zhuo, $et.al$. Post-fabrication measurement-driven oxide breakdown prediction and management. In *Proc. ICCAD*, pages 441–448, 2009.

[17] A. Gattiker, $et.al$. Data analysis techniques for CMOS technology characterization and product impact assessment. In *Proc. ITC*, pages 1–10, 2006.

[18] A. Gattiker. Using test data to improve ic quality and yield. In *Proc. ICCAD*, pages 771–777, 2008.

[19] X. Li, $et.al$. Virtual probe: a statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits. In *Proc. ICCAD*, pages 433–440, 2009.

[20] M. Tipping. Sparse bayesian learning and the relevance vector machine. *JMLR*, vol. 1:211–244, 2001.

[21] R. Warner. *Applied Statistics: From Bivariate Through Multivariate Techniques*. Sage Publication, 2008.

[22] M. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In *Proc. AIS*, pages 3–6, 2003.

[23] S. Ji, $et.al$. *Bayesian compressive sensing*. *IEEE TSP*, vol. 56, no. 6:2346-2356, 2008.