Test Data Analytics — Exploring Spatial and Test-Item Correlations in Production Test Data

Chun-Kai Hsu, Fan Lin, Kwang-Ting Cheng, Wangyang Zhang*, Xin Li*, John M. Carulli Jr.† and Kenneth M. Butler†
Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106
*Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213
†Texas Instruments, Dallas, TX 75243

***Abstract*—The discovery of patterns and correlations hidden in the test data could help reduce test time and cost. In this paper, we propose a methodology and supporting statistical regression tools that can exploit and utilize both spatial and inter-test-item correlations in the test data for test time and cost reduction. We first describe a statistical regression method, called *group lasso*, which can identify inter-test-item correlations from test data. After learning such correlations, some test items can be identified for removal from the test program without compromising test quality. An extended version of this method, *weighted group lasso*, allows taking into account the distinct test time/cost of each individual test item in the formulation as a weighted optimization problem. As a result, its solution would favor more costly test items for removal from the test program.**

We further integrate weighted group lasso with another statistical regression technique, virtual probe, which can learn spatial correlations of test data across a wafer. The integrated method could then utilize both spatial and inter-test-item correlations to maximize the number of test items whose values can be predicted without measurement. Experimental results of a high-volume industrial device show that utilizing both spatial and inter-test-item correlations can help reduce test time by up to 55%.

I. Introduction

As systematic and variation-induced failures become increasingly common, more types of tests are needed in testing, and each type of the tests must be applied multiple times to cover different process and design corners for product quality assurance. As a result, the number of test items has grown significantly, resulting in excessive test time and a huge amount of test data.

On the other hand, it is often observed that there exist meaningful patterns in test data. By exploring the hidden patterns and correlations in the test data, test data analytics has a wide range of applications such as reducing test time, predicting test quality, identifying outliers for diagnosis, discovering weak links in the manufacturing process, and improving the robustness of the design.

Within the scope of test data analytics for test cost reduction and test quality improvement, several methods have been proposed. In [1], the authors proposed an adaptive approach for multi-site testing which exploits device-to-device correlations and neighborhood statistics. A spatial correlation model was proposed in [2], [3] for modeling spatially distributed characteristics of variability. In [4], [5], the authors introduced statistical and machine learning based methods which predict the pass/fail of a set of test items for test compaction. The *alternate test* framework was proposed in [6]–[11] and was used to predict circuit performance values based on a set of strongly correlated signature values captured from simpler test setups.

In [12], [13], adaptive schemes were developed for effective test item ordering, with which re-ordered tests can detect failures earlier, benefiting stop-on-fail test programs. In [14], [15], test data were analyzed to detect outliers. An adaptive test flow was proposed in [16] that adjusts the test process to devote more test resources to marginal devices and less resources to passing devices with large margins. In [17], an *adaptive test scheme* was proposed to dynamically control the test flows and test contents on-tester in real-time through continuous per-die updates of test fail rate.

It is well known that, for some test items, there exist spatial correlations among dies on the same wafer. There also exist correlations among multiple measurements taken from the same chip (i.e. inter-test-item correlations). Identifying these two types of correlations from the test data has several known applications.

For spatial interpolation, a scheme called *virtual probe* (VP), based on recent breakthroughs in compressive sensing, is proposed in [18]–[21]. Another method using *Gaussian process* models was introduced to extrapolate a function over the Gaussian random field based on a small set of data observations [22]. Targeting the same problem, the two schemes use fundamentally distinct statistical algorithms. Both of them can capture the spatial correlation across the wafer for a test item under consideration by making measurements for only a randomly sampled fraction of dies on a wafer.

Known applications in production testing that utilize the inter-test-item correlations include: an early work that used a Monte Carlo based approach to analyze the joint probability distributions of test items for constructing a regression model of the untested performances [23], reordering the test items to screen out faulty dies earlier [13], and predicting passed test items to optimize the test set accordingly [16], [17], [24].

Test data analytics can help reduce both test time and test cost. In general, test time and test cost are highly correlated due to the high capital cost and operating expenses of test equipment. For simplicity and consistency, in the rest of this paper we refer to such methods which can reduce both test time and test cost as *test time reduction* (TTR) methods.

In this paper, we propose a method based on a statistical regression technique called *group lasso* (GL) [25], [26] to capture the correlations among test items using the test data of training chips. Not only identifying correlated parametric test items in any given test program, this method could also find correlations between test items in different test phases, such as wafer probe tests and package tests. The correlated test items can be removed from explicit testing and their values can be predicted by the measured values of other test items of the same chip. If the die IDs are traceable, such identified correlations can be used to reduce test time by removing those nearly redundant package test items.

Different test items often incur different amounts of test time and cost. For test cost reduction, it is preferable to predict more costly test items if such options exist. We then extend GL and reformulate the optimization problem to allow factoring in the distinct test times/costs of individual test items. As a result, the extended method, named *weighted group lasso* (WGL), tends to find a solution where more expensive test items are more favored than less expensive ones as candidates for removal from the test program.

Finally, we integrate both VP and WGL techniques to enable

978-1-4799-0859-2/13/$31.00 ©2013 IEEE

utilization of both spatial and inter-test-item correlations for test time/cost reduction. For the integrated method, we can run VP first to identify items that can be predicted without measurement (referred to as VP-predictable items) based on spatial correlations. It is then followed by running WGL for which those VP-predictable items are assigned a small weight and the other test items (i.e. spatially unpredictable items) are assigned a large weight. With such assignments, WGL, which identifies additional predictable items based on inter-test-item correlations, will find predicable items mainly from the pool of spatially unpredictable test items, thus maximizing the union of the predicable items derived from the spatial and inter-test-item correlations.

The proposed method offers the flexibility of exploring the trade-off between the number of removed test items and the prediction accuracy. We propose to use two predictability criteria, the bound of relative prediction error and the margin from the specification limits, to control the training process. We conducted experiments on a high volume industrial device and identified 47% of the test items to be candidates for sampling or elimination from the test list, out of 338 parametric test items, with a potential test time savings of 55% given our test time assumptions.

The rest of the paper is organized as the following. Section II gives an overview of the VP technique. Section III describes the GL method for learning inter-test-item correlations. Section IV shows WGL which allows assigning different weights to different test items in the formulations, and illustrates how to use WGL for integrating the VP method. Section V discusses the criteria for classifying test items as predictable or not and the flow of our proposed methodology. Section VI provides experimental results, and Section VII concludes the paper.

II. Background: Virtual Probe

This section describes a statistical regression method, virtual probe, in more detail as it is integrated into our proposed methodology. The essence of VP is to test only a subset of dies at selected locations on a wafer, transform the measurements into spatial frequency domain, and use a statistical algorithm to accurately recover the test values of the remaining dies [18]–[21]. Fig. 1 shows the concept of applying VP to a test item of an industrial product. In this example, the spatial model constructed from 10% randomly sampled dies accurately predict the test values of the remaining 90% dies on the same wafer.

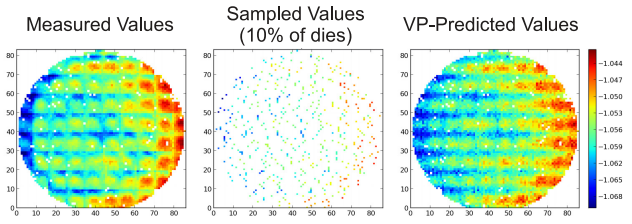


Fig. 1: Measured values, sampled values, and VP-predicted values of a test item from an industrial product. The different colors represent different test values.

The mathematical background of VP is briefly introduced as the following. Let $\{g(x,y);\ x = 1,2,\ldots,P,\ y = 1,2,\ldots,Q\}$ be a performance metric of the die at coordinate (x,y) on a size of $P \times Q$ wafer. The spatial variations of $g(x,y)$ can be represented by a two-dimensional linear transform in the frequency domain. In VP, the discrete cosine transform (DCT) is chosen for the transform. Let $\{G(u,v);\ u = 1,2,\ldots,P,\ \ v = 1,2,\ldots,Q\}$ be the DCT coefficients after the transform, i.e., the coefficients of different frequencies in the spatial pattern.

The purpose of VP is to accurately recover $g(x,y)$ from a small number, M, of dies at the locations $\{(x_m, y_m;\ m = 1,2,\ldots,M\}$, where $M \ll PQ$. Toward this goal, the following linear equation is formulated:

$$A \cdot \eta = B \tag{1}$$

where

$$A = \begin{bmatrix} A_{1,1,1} & A_{1,1,2} & \cdots & A_{1,P,Q} \\ A_{2,1,1} & A_{2,1,2} & \cdots & A_{2,P,Q} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M,1,1} & A_{M,1,2} & \cdots & A_{M,P,Q} \end{bmatrix} \tag{2}$$

$$A_{m,u,v} = \alpha_u \cdot \beta_v \cdot \cos\frac{\pi(2x_m - 1)(u-1)}{2P} \cdot \cos\frac{\pi(2y_m - 1)(v-1)}{2Q} \tag{3}$$

$$\eta = [G(1,1)\ \cdots\ G(P,Q)]^T \tag{4}$$

$$B = [g(x_1,y_1)\ \cdots\ g(x_M,y_M)]^T \tag{5}$$

Once η is determined by solving (1), the metric values $g(x,y)$ can be recovered by the inverse discrete cosine transform (IDCT).

It is, however, not trivial to solve (1). Since $M \ll PQ$, i.e., the number of equations is significantly less than the number of unknowns, (1) is profoundly underdetermined. The solution of η is therefore not unique and additional constraints are required. To obtain a unique solution of η, VP assumes η to be sparse [18]. That is, most of the DCT coefficients are close to zero, though the locations of the zeros are unknown. We can use maximum posterior estimation (MAP) to statistically solve (1) by reformulating it to

$$\begin{aligned} \underset{\eta}{\text{minimize}} \quad & \|\eta\|_1 \\ \text{subject to} \quad & A \cdot \eta = B \end{aligned} \tag{6}$$

where $\|\eta\|_1$ stands for the L_1-norm of η. Equation (6) can be solved efficiently with linear programming [18].

The generated sparse solution finds the sparsest set of coefficients in the frequency domain that accurately picture the spatial pattern of the sampled dies. The sampled dies, however, are only a very small portion of all the dies on a wafer. Therefore the spatial pattern reconstructed from the sampled dies may not be sufficient if the measurement data exhibit a more random distribution. In other words, if the assumption of sparsity is not valid for a certain test item, finding the sparse solution is not sufficient to recover the spatial pattern of the test item. In [19], a test item was categorized as highly-predictable, predictable, and unpredictable in a pre-test analysis phase based on the number of samples required by VP for the test item to reconstruct the spatial pattern within a certain error bound.

To improve prediction accuracy, the random sampling scheme in VP was modified to iteratively sample the optimal location in Bayesian virtual probe [20]. The correlations among different wafers within the same lot were utilized to further reduce the number of sampled dies on each wafer without compromising the prediction accuracy in multi-wafer virtual probe [21].

III. Inter-Test-Item Correlations

There exist correlations among the measurement data for different test items taken from the same chip. One goal of our methodology is to learn such inter-test-item correlations from the test data of a set of training chips. Specifically, the objective is to identify the test

items whose values can be predicted as a linear combination of the measured values of other test items of the same chip.

In this paper, we focus on parametric test item only, for which the test value of the chip-under-test is a real number. In the first subsection, we define the inter-test-item correlation and how to model it. In the second subsection, we discuss the use of the correlation for identifying *predictable* test items whose values can be predicted by the values of other test items of the same chip. Finally, we show the statistical regression method that can efficiently find such predictable test items.

A. Inter-Test-Item Correlation Model

A first-order linear correlation may exist among test items. If such a correlation exists, we can predict the values of some test items, without actual measurement, using linear combinations of the measured values of other test items. If no such correlation exists, all test items must be physically measured. We use the following equation to define the inter-test-item correlation for one test item:

$$\hat{\mathbf{f}}_k = \sum_{i=1}^{n} \alpha_{ki}\mathbf{f}_i + C_k \tag{7}$$

where $\hat{\mathbf{f}}_k$, a vector, denotes the predicted values of the target (the kth test item) for a set of chips, $\mathbf{f}_i$, a vector too, denotes the measured values of the ith test item of the same set of chips, n is the number of test items, and C_k is an offset constant. An element in $\mathbf{f}_i$ and $\hat{\mathbf{f}}_k$ represents the predicted or measured value of a chip and the dimension of these vectors is the number of chips in the set. We assume that the statistical characteristics, such as the correlation, are stationary (i.e., not varying) over all chips.

The vector of measured values of the kth test item, $\mathbf{f}_k$, is also included in the right hand side of (7). If $\hat{\mathbf{f}}_k$ is predictable based on the measured values of other test items, there exists an appropriate value for every α_{ki}, $i \neq k$, to form the model in (7) where α_{kk} is equal to zero. On the other hand, if $\hat{\mathbf{f}}_k$ is unpredictable, one trivial solution is that all α's except α_{kk} are zero and α_{kk} is equal to one.

If n test items are considered at the same time, the correlations are represented by a set of linear equations, i.e., n equations of (7) for $k = 1, 2, \cdots, n$. Without loss of generality, we normalize each test item to be zero mean and unit variance. As a result, the correlation model can be represented as:

$$\begin{cases} \hat{\mathbf{F}}_1 = \sum_{i=1}^{n} \alpha_{1i}\mathbf{F}_i \\ \hat{\mathbf{F}}_2 = \sum_{i=1}^{n} \alpha_{2i}\mathbf{F}_i \\ \quad\vdots \\ \hat{\mathbf{F}}_n = \sum_{i=1}^{n} \alpha_{ni}\mathbf{F}_i \end{cases} \tag{8}$$

where $\hat{\mathbf{F}}_i$ and $\mathbf{F}_i$ denote the normalized predicted values and the normalized measured values for the ith test items, respectively. Note that we no longer need C for this normalized version. Assuming that we derive this correlation based on d chips, these n vector equations correspond to a total of nd scalar equations.

The inter-test-item correlation model can therefore be encoded by a matrix formed by all α's in (8) as shown below.

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn} \end{bmatrix} \tag{9}$$

This model uses n^2 variables (α_{ij} for $i, j = 1, 2, \cdots, n$) to represent the correlations. Note that (9), representing the correlations derived from (8), is different from the traditional correlation matrix defined in statistics for the test items in which the (i, j) element represents $\mathrm{corr}(\mathbf{F}_i, \mathbf{F}_j)$. Given the measured test data of n test items of d chips, we can use several regression and learning methods to derive the matrix in (9). We will introduce an efficient way of solving this problem in the following subsections. An exemplar solution is the identity matrix where each test item is correlated to itself only.

B. Candidate Test Items for Removal From Test Program

Based on (7), a zero coefficient indicates that measurement of the corresponding test item is not needed for deriving the target item's value. For example, if $\alpha_{k1} = 0$ and $k = 1$ in (7), we can predict the first test item ($\hat{\mathbf{f}}_1$) without relying on the actual measurement of the same item ($\mathbf{f}_1$). Considering all prediction equations simultaneously, if all coefficients corresponding to a test item are all zero, we can eliminate the test item from the test program for actual measurement. For example, referring to (8), if the following condition is true,

$$\forall i \ni \mathbb{N} \wedge 1 \leq i \leq n, \alpha_{i1} = 0 \tag{10}$$

we can conclude that every test item, including the first test item itself, can be derived without relying on actual measurement of the first test item ($\mathbf{f}_1$ or $\mathbf{F}_1$). Test item one is then a candidate for removal from the test program. In the following, we refer to such a test item as a *candidate* test item. The general condition for a candidate test item is:

$$\forall i \ni \mathbb{N} \wedge 1 \leq i \leq n, \alpha_{ik} = 0 \Rightarrow \text{item } k \text{ is a candidate.} \tag{11}$$

In the correlation matrix (9), a column of zeros indicates that the corresponding test item is a candidate. Consider the following example:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0.5 & 0 & 0.1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

The second test item is a candidate because the second column is all zeros. Besides, no other candidate test item exists and we need to explicitly test each of them.

C. The Group Lasso Regression Problem

According to the model definition described in the previous subsection, we can formulate the problem of finding the correlation matrix (9) as a minimization problem which attempts to minimize the difference between predicted values and measured values. The minimization problem for finding a correlation model is formulated as:

$$\arg\min_{\alpha} \sum_{i=1}^{n} \left\| \mathbf{F}_i - \hat{\mathbf{F}}_i \right\|_2^2 \tag{13}$$

which is equivalent to

$$\arg\min_{\alpha} \sum_{i=1}^{n} \left\| \mathbf{F}_i - \sum_{j=1}^{n} \alpha_{ij}\mathbf{F}_j \right\|_2^2 \tag{14}$$

where $\|\cdot\|_2$ denotes the L_2 norm.

For many regression applications, it is often desirable to find a sparse solution of (14) that has as many zero coefficients as possible. The lasso (least absolute shrinkage and selection operator) method [25] was designed to find sparse solutions by adding an L_1 norm penalty to (13), resulting in a revised minimization problem as follows:

$$\arg\min_{\alpha} \sum_{i=1}^{n} \left\| \mathbf{F}_i - \hat{\mathbf{F}}_i \right\|_2^2 + \lambda^* \sum_{i=1}^{n} \sum_{j=1}^{n} |\alpha_{ij}| \tag{15}$$

where λ^* is a penalty parameter to control the trade-off between prediction error and the sum of all absolute values of alphas. In lasso, increasing λ^* forces more coefficients in correlation matrix to approach zero.

However, minimizing the number of nonzero coefficients in the correlation matrix does not address our goal of maximizing the number of candidates, i.e., maximizing the number of test items meeting Condition (11). We therefore introduce *group lasso* (GL) [26] which attempts to find a sparse solution which maximizes the number of columns with all near-zero entries in the solution matrix. The main idea of GL is to group coefficients corresponding to the same test item together and revise the minimization problem as follows:

$$\begin{aligned} \arg\min_{\alpha} \quad & \sum_{i=1}^{n} \left\| \mathbf{F}_i - \hat{\mathbf{F}}_i \right\|_2^2 \\ \text{subject to} \quad & \lambda \geq \sum_{g=1}^{n} \sqrt{\sum_{i=1}^{n} \alpha_{ig}^2} \end{aligned} \tag{16}$$

The term $\sqrt{\sum_{i=1}^{n} \alpha_{ig}^2}$ combines all coefficients in the gth column of the correlation model (9) together to form a group. As λ decreases, GL attempts to find a solution with nonzero groups instead of just nonzero coefficients.

D. The SOCP Problem

Because of the quadratic terms in the optimization problem (16), we reformulate it as a second-order cone programming (SOCP) problem that can be solved efficiently by interior point methods [27]. The reformulated problem becomes:

$$\begin{aligned} \underset{\alpha,u}{\text{minimize}} \quad & T \\ \text{subject to} \quad & T \geq \sqrt{\|\mathbf{u}_1\|_2^2 + \cdots + \|\mathbf{u}_n\|_2^2} \\ & g_i \geq \sqrt{\alpha_{1i}^2 + \alpha_{2i}^2 + \cdots + \alpha_{ni}^2}\,, \quad 1 \leq i \leq n \\ & \mathbf{u}_i = \mathbf{F}_i - \hat{\mathbf{F}}_i\,, \quad 1 \leq i \leq n \\ & \lambda = g_1 + g_2 + \cdots + g_n \end{aligned} \tag{17}$$

where $\mathbf{u}_i$ denotes the difference between predicted and measured values. All $\mathbf{F}_i$, $\hat{\mathbf{F}}_i$, and $\mathbf{u}_i$ are vectors with d dimensions. In general, a smaller λ would more likely result in a sparser solution, i.e., more near-all-zero columns in the correlation matrix (equivalent to having more candidate test items).

IV. Further Optimization for Test Time Reduction

In the previous two sections, we described two TTR methods, VP and GL, which target spatial and inter-test-item correlations, respectively. However, neither method can address the following issues:

1) Taking into account the distinct time of each individual test item for overall test time reduction. Different test items incur different test times. In finding predictable test items by TTR methods, it is preferred that the more costly test items are predicted because they contribute more to the total test time. Reflecting different test times of test items requires a scheme to assign different significance to differentiate test items. In GL, the measurements of all test items are included in the minimization equation (16), and the relative significance does influence the solution. Hence GL could in some way address the different test times of test items in (16). This problem also includes how to map the practical test times to reasonable parameters so the generated result is most improved in terms of TTR.
2) Considering both spatial and inter-test-item correlations in test data for overall test time reduction. As spatial patterns and inter-test-item correlations are two independent approaches, many TTR methods have been proposed targeting either of the two. It is natural to ask if there is a way to utilize both correlations simultaneously and expand the dimensions of test data analytics. For instance, having different sets of predictable test items from VP and GL, we want to maximize the union of the two sets so that we find the largest number of predictable test items. If a test item has been identified as predictable in one method, the other method should tend to predict the other test items instead of the one already predicted by the first method.

In the following, we propose weighted group lasso (WGL), an extension of GL, which could successfully address both issues mentioned above.

A. Weighted Group Lasso

In solving the optimization problem (16), GL treats every item equally and tends to find a solution with a maximum number of predictable test items. However, as different test items incur different test times, maximizing the number of predictable test items does not necessarily maximize the reduction of test time.

WGL, whose basic formulation is similar to that of GL, is expressed as follows:

$$\begin{aligned} \arg\min_{\alpha} \quad & \sum_{i=1}^{n} \left\| \mathbf{F}_i - \hat{\mathbf{F}}_i \right\|_2^2 \\ \text{subject to} \quad & \lambda \geq \sum_{g=1}^{n} w_g \sqrt{\sum_{i=1}^{n} \alpha_{ig}^2} \end{aligned} \tag{18}$$

where w_g denotes the weight of the gth test item (the gth group). A weight for the corresponding test item is therefore incorporated to reflect its actual test time. In the SOCP form, WGL is formulated as follows:

$$\begin{aligned} \underset{\alpha,u}{\text{minimize}} \quad & T \\ \text{subject to} \quad & T \geq \sqrt{\|\mathbf{u}_1\|_2^2 + \cdots + \|\mathbf{u}_n\|_2^2} \\ & g_i \geq \sqrt{\alpha_{1i}^2 + \alpha_{2i}^2 + \cdots + \alpha_{ni}^2}\,, \quad 1 \leq i \leq n \\ & \mathbf{u}_i = \mathbf{F}_i - \hat{\mathbf{F}}_i\,, \quad 1 \leq i \leq n \\ & \lambda = w_1 g_1 + w_2 g_2 + \cdots + w_n g_n \end{aligned} \tag{19}$$

Groups of α's with a larger weight will be more dominant in the constraint in (18) than the groups with a smaller weight. Therefore, WGL tends to find a solution that minimize the values of α's for heavier-weight groups. This results in a higher probability that a group with larger weight would have more near-zero α's, i.e., the corresponding item has a higher probability to be a candidate test item for removal from the test program.

B. Determining The Weight of A Test Item

In the following we discuss issues of assigning an appropriate weight for a test item in the WGL problem.

1) Reflecting Item's Test Time: Directly using the test time of the ith item as the weight w_i in (18) or (19) might lead to impractical solutions. For example, having groups with a very large weight in the constraint in (18) might cause undesired dominance of the constraint (which reflects the sparsity of the correlation matrix) over the cost function for minimization (which reflects the prediction error) as the optimization target. As a result, the solution might have unacceptably high prediction error or find very few predictable test items.

In our methodology, we use the normalized test times as the weights in WGL. For instance, assuming that we have n test items and their test times are $t_1, t_2, \ldots, t_n$, respectively, we normalize all t's so that their normalized values are in the range of 0.5 to 1.5 and the mean is equal to 1. These normalized t's are then used as the weights, w, in (18) and (19).

2) Weight Assignment for TTR Utilizing Both Spatial and Inter-Test-Item Correlations: A straightforward TTR strategy to utilize both spatial and inter-test-item correlations is to run both VP and GL on all test items independently and the aggregate their results for TTR. Assume T_{VP} and T_{GL} are the sets of predictable test items identified by VP and GL, respectively. Then their union $T_I = T_{VP} \cup T_{GL}$ would be the set of total predictable test items that can be removed from measurement. However, such a strategy often produces sub-optimal results. It is desirable to minimize the overlap (i.e. intersection) of T_{VP} and T_{GL} and maximize their union, which in turn maximizes the test time reduction. This optimized strategy can be implemented by running VP first, followed by running WGL. After identifying T_{VP}, we set a lower weight, w_l, for every test item in T_{VP} and a higher weight, w_h for test items not in T_{VP} before running WGL. Because WGL tends to minimize the α's for groups with a higher weight, the resulted predictable test items by WGL, T_{WGL}, would have minimum overlap with T_{VP}. As a result, the final set of predictable test items, $T_J = T_{VP} \cup T_{WGL}$, under this strategy would most likely be larger than the set, T_I, produced by the straightforward strategy.

V. Test Methodology Based on GL and WGL

In this section, we describe in detail the application of the inter-test-item correlation model of GL/WGL. The first issue to be addressed is to evaluate if the prediction accuracy of a candidate test item is sufficiently high and if the item can indeed be safely eliminated from the test program without compromising the test quality. Those candidate test items meeting a desired level of prediction accuracy (i.e. predictability) are referred to as *predictable test items* in the following.

We then discuss some practical issues of applying the proposed methods in production. Specifically, we discuss the issues of handling random defects, normalized prediction error, and the need of continuous cross-validation to monitor if the manufacturing process is sufficiently stationary.

Then we describe the two-stage test methodology of GL or WGL in detail: the pre-test analysis for learning the inter-test-item correlation model and the test application stage which utilizes the learned model for TTR.

A. Criteria for Classifying Predictability

Even if the correlations among the test items are weak, GL or WGL will still produce a correlation model. However, the prediction accuracy based on the model might not be accurate.

According to Section III-B, a useful correlation model for TTR has one or multiple near-zero columns. The values of the corresponding candidate test items can be predicted by a combination of other test items. Since we use a penalty parameter, λ, in the minimization problem (17) to control the sparse level of α, each $\mathbf{u}$ of the solution found may not be minimal. That is, there is no guarantee on the prediction accuracy for the candidate test items derived from the correlation model. Hence we need to apply one more filtering step to the set of candidate test items: only a subset of them that meet some criteria and achieve a desired level of prediction accuracy will be selected as *predictable* test items.

We evaluate the predictability of a test item based on two criteria:

1) The maximum relative prediction error among a total of d chips in the training set:
$$e = \max(\{|(\hat{g}_i - g_i)/g_i| : i = 1, \cdots, d\}) \qquad (20)$$
where $\hat{g}_i$ and g_i denote the predicted and the measured values of chip i in the training set.
2) The margin between specification limits of a test item and the range covering *most* of the training chips' predicted values. If we denote the 25%, 50%, and 75% points of the cumulative distribution function (CDF) of the predicted values of all training chips as Q_1, Q_2, and Q_3 respectively, the IQR, defined as the range of the middle fifty, would be equal to $Q_3 - Q_1$. The interquartile range method defines the range X from $Q_1 - 1.5IQR$ to $Q_3 + 1.5IQR$ as the range covering most of the data points for an arbitrary distribution (conceptually similar to the 3σ range for a normal distribution).
If we denote L_l and L_h as the low and high limits of a test item's specification range respectively and M as the desired margin between the specification limits and the range X defined above (i.e. $[Q_1 - 1.5IQR, Q_3 + 1.5IQR]$), the following is the second criterion used for classifying a test item as predictable:
$$L_l + M < X(\hat{g}_i, i = 1, \cdots, d) < L_h - M \qquad (21)$$

We define three levels of predictability, *high*, *medium*, and *low*, for each of the two criteria in Table I. Based on the predictability levels, we classify each test item as either predictable or unpredictable, as shown in Fig 2. That is, a test item is considered predictable only if it has a high predictability level for at least one criterion and does not have a low predictability level for any criterion.

TABLE I: The Predictability Levels

Predictability level	Relative error	Margin from spec limits, percentage of $(L_h - L_l)$
High	$0\% \sim 5\%$	$> 35\%$
Medium	$5\% \sim 25\%$	$15\% \sim 35\%$
Low	$25\% \sim 100\%$	$< 15\%$

B. Random Defects, Normalized Error, and Process Stationarity

1) Selecting Test Items Targeting Random Defects: VP and GL are effective only for test items and chips that are affected by process and systematic variations. For chips with random defects, their values of some test items might not follow the correlations captured from the training chips. Therefore, even for predictable items, the values of such defective chips predicted by VP and GL/WGL might be inaccurate.

However, a defective chip with a random defect is usually more catastrophic (than systematic and variation-induced failures) and can often be detected by multiple test items. In addition, it has been

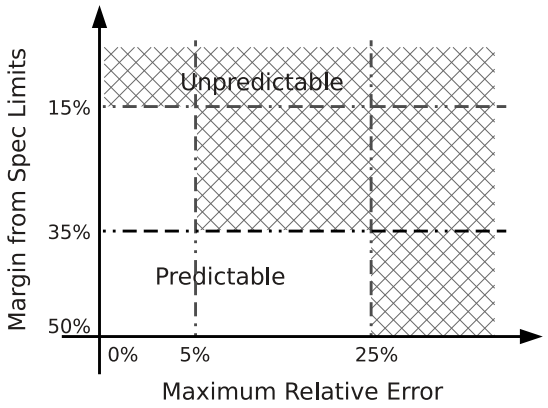


Fig. 2: We classify each test item into one of two possible categories, predictable or unpredictable, based on joint consideration of two criteria. Predictability is evaluated through relative prediction error and the distribution of the prediction error.

observed that chips with random defects can often be detected by a small subset of test items, carefully selected from a test program consisting of a large number of test items. As an example, Table II shows the number of test items of a high-volume production chip that are required to cover all failed chips in the training set whose test items cannot be accurately predicted by VP and GL (i.e., most likely the suspects with random defects). Out of 338 test items in total, while the number of test items required to cover all random defect suspects increases as more wafers are considered, the test item count required is still relatively small in comparison with the total number of test items.

Based on this observation, we address random defects by selecting additional test items for explicit measurement, among those items classified as predictable based on the criteria discussed in Section V-A. Specifically, in the model validation phase, if the number of chips escaping from a predictable test item is greater than a threshold (i.e., the prediction error is abnormally large for too non-trivial number of chips in the training set), we disqualify it as a predictable item and, instead, classify it as unpredictable and thus requiring explicit measurement.

TABLE II: Number of Test Items Required to Cover All Random Defect Suspects

Number of training wafers	1	2	3	5	10	25
Number of test items to cover all random defect suspects	9	12	18	26	38	57

2) Screening Test Items by Normalized Error: While using the relative prediction error in (20) and setting an upper bound on e as one of the criteria for classifying the test items ensures the test quality will not be compromised, the use of the relative prediction error, however, is biased by the test item's mean and variation when evaluating the prediction quality. It is possible that a model in which a test item's distribution is not accurately captured, but still has a small relative prediction error, if the test item's mean is large and its variation is small. For such a case, though the distribution of the test values are not accurately captured, the errors, divided by their large mean, are sufficiently small to pass the error bound e.

To address this problem, the test items that pass the two criteria in Section V-A are further examined using their normalized prediction error. The normalized prediction error, without the bias of the test item's mean and variation, better reflects the accuracy of capturing the test item's distribution. Specifically, those test items with a normalized prediction error greater than a threshold will be screened out and excluded from the final set of selected predictable test items.

3) Stationarity: There exist wafer-to-wafer and lot-to-lot variations. Therefore, in applying the model trained based on the test data of one wafer for testing of another wafer, it is necessary to perform additional validation to assure the correlation patterns of the chip/wafer under test are sufficiently close to the patterns exhibited in the training data.

The validation can be easily done by taking additional measurement for a small number of the chips for the predictable test items. If the statistics of the differences between the predicted values and measurements are significantly greater than those estimated from the training set, explicit measurements for all test items should be made for all dies in the wafer. The complete test data of the wafer will then go through further outlier analysis. If the analysis concludes that the wafer is an outlier, the original model will continue to be used. Otherwise, retraining based on the target wafer's new data is triggered and the retrained model will be used for further testing of other wafers. Through this continuous validation, the methodology can be adapted to address significant wafer-to-wafer and lot-to-lot variations.

C. Test Procedure

In this subsection, we summarize the procedures of both pre-analysis and test application stages.

1) Flow of Pre-Test Analysis: The input to the pre-test analysis procedure includes *a*) complete test data of a set of chips as the training set (including die locations and specification limits, measured values, and test time for each individual test item), *b*) criteria for predictability classification (as described in Section V-A), and *c*) the preferred statistical regression methods (VP and/or WGL with a choice of weight assignment as discussed in Section IV-B).

We run VP first, if VP is chosen as a preferred method. Based on the options discussed Section IV-B, we use either the test times or test items' predictability classified by VP to determine the weights before running WGL. Next, we build an inter-test-item correlation model by solving the WGL problem defined in (19). After that, we determine the predictable items based on the criteria illustrated in Table I and Fig. 2. In addition, extra test items are selected for measurement, based on the discussion in Section V-B1, to detect failed chips caused by random defects. Finally, we estimate test time saving, the yield loss, and the escape rate by comparing the predicted values with measured values of training chips.

2) Test Application Flow: In the test application stage, we skip predictable test items from measurement. Based on the partial measurement results and the correlation model, the values of those predictable test items are calculated. We can apply this procedure, summarized in Fig. 3, to any wafer or any collection of chips for testing. We first perform all tests for those predetermined sample dies/chips in the target wafer or collection of chips. The measured values of these samples are used for three purposes: *a*) used by VP to calculate the predicted values of the other dies based on the spatial correlations, *b*) used by GL/WGL to calculate each test item's mean and standard deviation for chips on the target wafer which are needed for converting the normalized predicted values, defined in (7), to the predicted values, defined in (8), and *c*) used for checking the stationarity for the validity of inter-test-item correlation model on the target wafer (or a target collection of chips) based on the discussion in Section V-B3.

Then we perform the following two processes concurrently for the remaining chips that are not those predetermined samples: *a*) testing the unpredictable test items for chips, and *b*) using the statistical methods (VP and/or WGL) to predict the values of those predictable test items. After gathering all predicted values and measured values, we can apply the validation process described in Section V-B3. If the wafer (or the collection of chips for testing) passes the validation for stationarity, the test results are considered valid. Otherwise, we cannot trust the predicted values and thus have to test all test items for the measured values of all chips on the wafer for the outlier analysis. Several proposed methods, such as [24], [28], can be used to help identify outlier wafers. If the wafer is an outlier, we do not change the correlation model and continue the above procedure to the next wafer. Otherwise, we rerun the pre-test analysis to build a new correlation model before applying the procedure to the next wafer.

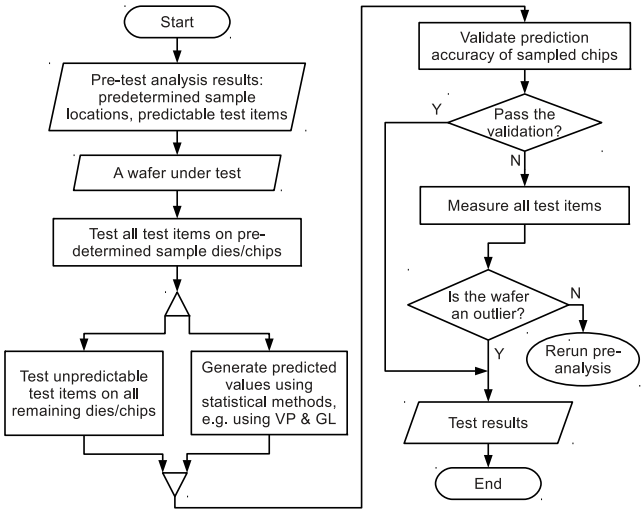


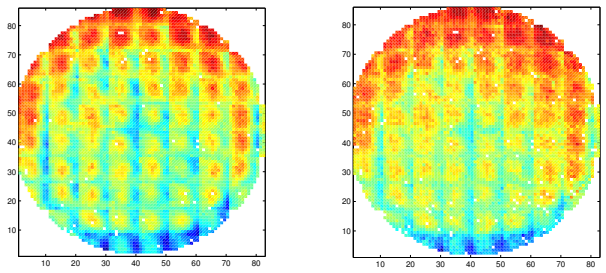
Fig. 3: The flow of test application.

VI. Experimental Results

We applied GL and WGL to the wafer sort data of a high-volume industrial device. There were 25 wafers per lot and 5500+ dies per wafer. For each lot, 500 randomly sampled dies (by Latin hypercube sampling method) on the first wafer were used for training, and the other 5000+ dies on the same wafer were used for validating the trained model, as discussed in Section V-C1. The test program we analyzed consists of hundreds of test items, approximately 70% of which are parametric in nature, and those tests are the ones to which we applied our proposed method.

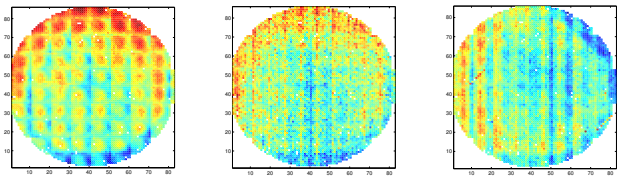
A. Inter-Test-Item Correlations Analysis

The correlation matrix, which is the solution to the GL regression problem of (16), reveals the correlations among test items, as formulated in (8). Coefficient α_{ij} reflects the significance of $\mathbf{F}_j$ in predicting $\mathbf{F}_i$. A larger α_{ij} means that $\mathbf{F}_j$ contributes more in predicting $\mathbf{F}_i$; therefore, test item j has a stronger correlation with test item i. One example of the prediction is shown in Fig. 4 where the values of the test item are color-coded. The prediction result of the 87th test item shows very high consistency with the measurement data. Fig. 5 shows the three test items with the largest coefficients α's in predicting test item 87 in Fig. 4. It can be observed from the wafer maps that test item 35, with the largest α, has the strongest correlation with test item 87.



(a) Measurement data (b) Prediction result

Fig. 4: Die map of measurement and prediction data of test item 87.



(a) Test item 35 ($\alpha = 0.2565$) (b) Test item 266 ($\alpha = 0.0731$) (c) Test item 97 ($\alpha = 0.0540$)

Fig. 5: Test items with the largest coefficient α in predicting test item 87.

Fig. 6 shows the number of predictable test items versus the penalty parameter λ for different relative error bounds e, defined in (20). With a fixed error bound, different λ values result in different numbers of predictable test items. For a given λ, the larger the error bound, the more the predictable test items. Furthermore, the λ value that maximizes the number of predictable test items varies for different e's. The optimal value of λ depends on the test data, including the number of test items, their measured values, and the number of dies for training.

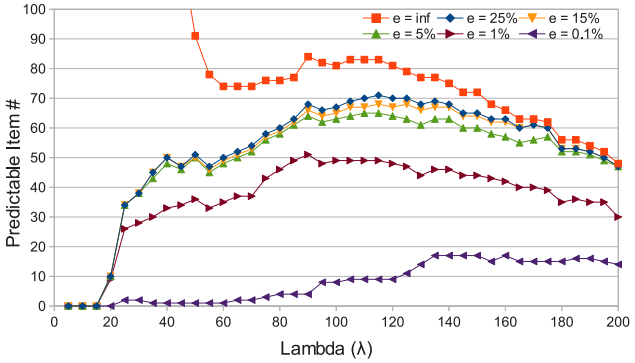


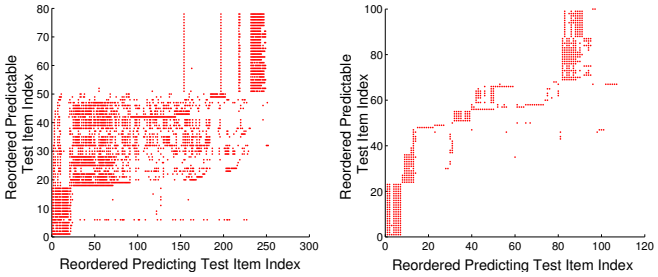
Fig. 6: Number of predictable test items versus penalty parameter λ for six different error bounds e's.

For e being infinity, which completely ignores the constraint imposed by the bound of prediction error, all candidate test items will be classified as predictable test items. The curve of $e = \inf$, thus showing the number of candidate test items, rises up rapidly when λ becomes smaller than 50. It approaches very close to the total number of test items when $\lambda \to 0$ because sparsity of the correlation matrix becomes the dominant emphasis while accuracy

is ignored. As a general trend, those candidate test items found at a low λ value, however, more likely have a larger prediction error and thus will more likely be screened out by the predictability criteria and considered unpredictable in our methodology. For e at 5%, the maximum number of test items identified as predictable is around 19% among the 338 test items we analyzed.

In addition to identifying predictable test items, the correlation matrix produced by GL also reveals the strength of correlations among test items. Fig. 7 shows the relations between the predictable test items and the test items used to predict others, i.e., the "predicting" test items. In the scatter graphs, a point at coordinate (j, i) means that test item $\mathbf{f}_j$ is in the set of test items that are used for predicting test item $\mathbf{f}_i$. The orders of the predictable and predicting test items were rearranged so the figures show clusters of points. Showing points with $\alpha_{ij} \geq 0.01$, Fig. 7a contains more points, some of which may not represent significant relations. Fig. 7b, on the other hand, shows only relations with $\alpha_{ij} \geq 0.05$ and reveals only relations that are sufficiently strong.

If we have a cluster of points with x-coordinates in the range of $\mathbf{u}$ and y-coordinates in the range of $\mathbf{v}$, we can conclude that $\mathbf{f_u}$ predict $\mathbf{f_v}$, and therefore $\mathbf{f_u}$ and $\mathbf{f_v}$ have strong correlations. For example, in Fig. 7b the first 23 reordered predictable test items are predicted by 7 of the first 8 predicting test items, implying that the 23 predictable test items and the 7 predicting test items are highly correlated.



(a) $\alpha_{ij} \geq 0.01$ (b) $\alpha_{ij} \geq 0.05$

Fig. 7: The relations between predictable test items and the "predicting" test items.

We then applied the model for testing the other 24 wafers. The prediction errors of using the model produced by setting $e = 5\%$, the prediction errors of an exemplar are illustrated by a boxplot in Fig. 8. The y-axis indicates the relative prediction error of dies corresponding to each predictable test item (the x-axis). The ends of a whisker represent the lowest datum still within the lower quartile minus $1.5IQR$ (where IRQ is the interquartile range, the difference between the upper and lower quartiles), and the highest datum still within $1.5IQR$ plus the upper quartile. Any data not included between the whiskers are plotted as a cross. As illustrated in Fig. 8, all prediction errors are within the 5% error bound e when applying the model obtained from the training wafer to another.

In our methodology, the margin from the spec limits is the other criteria for classifying test item predictability. Increasing the desired margin will reduce the number of test items classified as predictable as shown in Table III. If the manufacturing process is relatively stable, we can set a smaller margin and the number of test items classified as predictable would be larger. For example, when the margin is set to 5%, 71 test items (i.e., 21% of total items) are classified as predictable.

Fig. 9 shows an example of GL's training result of one wafer for which each test item's prediction error and its margin from the spec

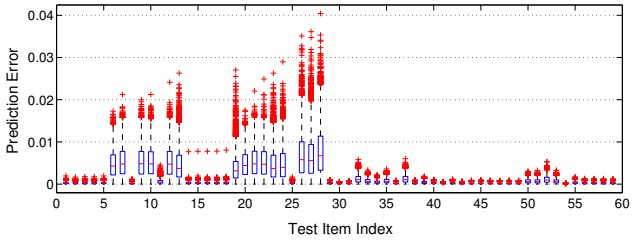


Fig. 8: Error boxplot of each predictable test item.

TABLE III: # of Predictable Item Versus Margin from Spec Limits

Margin from spec limits	5%	15%	25%	35%	45%
# of items as predictable	71	64	49	20	11

limits are illustrated. Each dot in the figure denotes a candidate test item, in which the x-coordinate is the item's prediction error and the y-coordinate is one minus the margin from the spec limits (so the smaller the y value, the larger the margin and the more predictable the test item). We classify the test items in the unhatched area as predictable test items, as discussed in Section V-A.

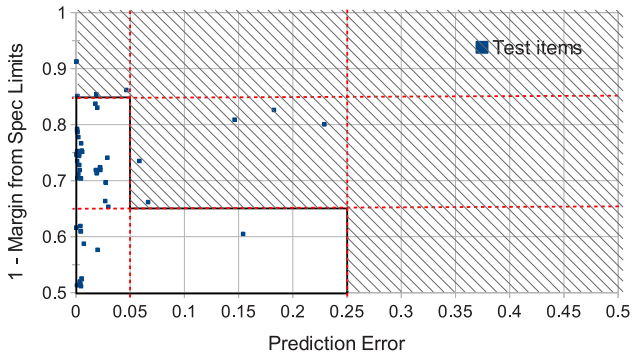


Fig. 9: The predictability of test items based on test data of one wafer. Each dot denotes a candidate test item and the test items in unhatched/hatched area are defined as predictable/unpredictable test items.

For the evaluation of the test application procedure described in Section V-C2, we used 500 dies in wafer No. 1 to train a model, and used all the remaining dies in the wafer for selection of predictable test items. When using the trained model for test application, we are concerned about test escapes — escaping faulty chips whose predicted values are mistakenly within the specification limits.

The test escapes of using various trained models are summarized in Table IV. We trained five different models. The first two models were generated by setting an error bound only, at $e = 5\%$ and $e = 25\%$ respectively. The third and the fourth models were generated by setting a lower bound on the margin from the spec limits only, at 15% and 35% respectively. The fifth model was generated using both criteria as showed in the unhatched area in Fig. 9. Assuming that the test time of all test items is identical, the percentages of test time savings are 19.2, 21.0, 18.9, 5.9, and 17.5% respectively while the number of escaped dies, among the 130,950 dies tested, are 3, 14, 12, 1, and 1, respectively. Setting large error bound derives more predictable test items with, however, more escaped dies. On the other hand, setting tight margin results in fewer escaped dies with lower test time savings. The fifth model have a balance between test time savings and prediction quality.

TABLE IV: Number of Escaped Dies for Various GL Models

Error bound	Spec margin	# of escaped dies [a]	Predictable item #	Test time savings
5%	–	3	65	19.2%
25%	–	14	71	21.0%
–	15%	12	64	18.9%
–	35%	1	20	5.9%
$\sim 25\%$ [b]	$\sim 35\%$ [b]	1	59	17.5%

[a] 130,950 dies in total
[b] The unhatched area in Fig. 9

B. TTR With Test Time of Individual Item

Enhanced from GL, WGL enables two applications for further test time reduction. First, it can take into account the distinct test times of individual test items. As the information of the actual test time and cost of each individual test item in the production test program is not available to us, we randomly generated two different sets of test times for our experiments to illustrate WGL's functionality and capability. In the first experiment, we used ten distinct test times, ranging from 0.1 to 3.5 unit, and randomly assigned 10% of the test items for each test time. In the second experiment, we also used ten distinct test times but with a larger range, ranging from 0.25 to 25 unit, and assigned them evenly to all test items.

The experimental results are shown in Table V, where T_{GL} and T_{WGL} denote the predictable test items found by GL and WGL, respectively. For both cases, WGL can achieve further test time reduction by taking into account the test time of individual test item. In comparison with GL, WGL achieves additional 11% and 18% test time savings than GL does for these two cases. Because the spread of test times for different test items in case 2 is larger than that in case 1, WGL achieves a greater test time saving in case 2 even though it classifies one fewer test item as predictable than the first case.

TABLE V: Test Time Improvement by WGL

Case	# of T_{GL}	Escaped dies #	Time savings	# of T_{WGL}	Escaped dies #	Time savings
1	59	1	17%	80	8	28%
2	59	1	17%	79	3	35%

C. Integrating Both Spatial and Inter-Test-Item Correlations

The second feature of WGL is the ability to exploit and integrate both spatial and inter-test-item correlations in test data. As described in Section IV-B2, we can merge the results of VP (targeting spatial correlations) and GL (targeting inter-test-item correlations) in a straightforward way by taking the union of the predictable test items classified by each methods. Rather than directly taking the union of predictable test items, WGL can be used to optimize the number of total predictable test items. This is achieved by setting different weights to VP-predictable and VP-unpredictable test items before running WGL to explore the inter-test-item correlations.

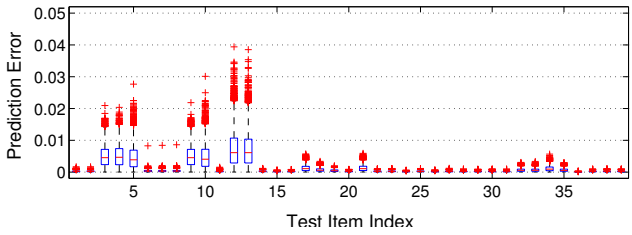
The experimental results of comparing these two strategies are shown in Table VI, where T_{VP} denotes the predictable test items derived by VP and T_{GL} denotes either the predictable test items derived by GL for the first strategy (in the row indicated as "Straightforward") or by WGL for the second strategy (in the row indicated as "Weighted"). The values shown in the table are counts of test items in different sets. In this experiment, we assume all test items have an identical test time. The weighted strategy can save an additional 5.7% test time over the straightforward strategy for using both spatial and inter-test-item correlations. WGL successfully reduces the overlap between the predictable test items produced by VP and GL — the number of test items in their intersection reduces from 21 to 8.

TABLE VI: Intuitive Merge vs. Weighted Merge

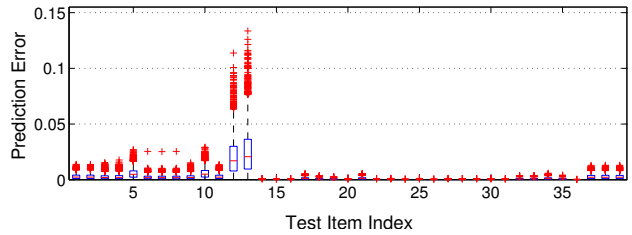
Method	$T_{VP} \cup T_{GL}$	T_{VP}	T_{GL}	$T_{VP} \cap T_{GL}$	Time savings
Straight-forward	134	96	59	21 (36%)	39.6%
Weighted	153	96	65	8 (12%)	45.3%

Fig. 10 attempts to compare the prediction errors of the same set of test items for using inter-test-item correlations only versus using both spatial and inter-test-item correlations. Fig. 10a shows the prediction errors by using only inter-test-item correlation for prediction and Fig. 10b shows the errors by using both spatial and inter-test-item predictions for prediction. When utilizing both spatial and inter-test-item correlations, VP was first run, whose results are then used as the input to WGL.

The fact that, under the VP+WGL strategy, some test items which are used for predicting other items in WGL were not actually tested but, instead, predicted using VP increase the overall prediction error. Test items 12 and 13 incurred a noticeable drop in accuracy. The reason for this is because these two items classified as predicable based on their relatively large margin from the spec limits, while they have a relatively large prediction error. As shown in Fig. 10b, the prediction error of test items 12 and 13 are still within the 25% error bound used for classification.



(a) Prediction errors by only inter-test-item correlations



(b) Prediction errors by both spatial and inter-test-item correlations

Fig. 10: Compare the prediction errors of the same set of test items which are the intersection of predictable test items derived by GL and by VP+WGL.

Table VII compares the results of using different statistical methods and strategies. In this comparison, the test time of each test item is based on the assumption of case 2 in Table V. Applying VP or GL only can achieve 31.5% and 15.7% test time savings, respectively. If we integrate VP and GL, which is equivalent to using VP and WGL with an assignment of the same weight to all test items, the test time saving increases to 41.8%. Integrating VP and WGL by assigning different weights to test items properly reflecting their test times can

further improve the test time saving to 46.3%. If we integrate VP with WGL whose weights are based on VP's classification results, the improvement reaches to 47.9%. Finally, the last case shows that the integration of VP and WGL whose weights are jointly determined by both test items' test times and VP's classification results can achieve 55.0% improvement. However, the number of escaped dies might slightly increase in order to gain some of such improvements.

TABLE VII: Summary of Test Time Saving for Various Strategies

Method	# of escaped dies [a]	# of predictable items	Test time savings [b]
VP only	4	96	31.5%
GL only	1	59	15.7%
VP+WGL with the same weight	5	136	41.8%
VP+WGL weighted by time	7	132	46.3%
VP+WGL weighted by VP	12	151	47.9%
VP+WGL weighted by time & VP	8	160	55.0%

[a] $130{,}950$ dies in total

[b] Based on the test time assumption of case 2 in Table V

VII. Conclusions

In this paper, we propose a methodology to utilize inter-test-item correlations for test time reduction. We further improve the methodology to take into account the test time of each individual test item for further reduction of production test time. Through integration with VP which can capture spatial correlations in test data, the methodology also allows exploiting and utilizing both spatial and inter-test-item correlations simultaneously for test time reduction. A case study of a high-volume industrial device show that the proposed methodology can reduce the test time by 55.0% with only 8 escaped dies out of $130{,}950$ tested dies in total.

Acknowledgments

This work was partially supported by the Gigascale Systems Research Center (GSRC) under the Focus Center Research Program, a Semiconductor Research Corporation program. The first three authors would also like to acknowledge the support by the Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan and the valuable input provided by Ka-Yi Yeh of ITRI.

References

[1] E. Yilmaz and S. Ozev, "Adaptive multi-site testing for analog/mixed-signal circuits incorporating neighborhood information," in *IEEE European Test Symp. (ETS)*, May 2012.

[2] F. Liu, "A general framework for spatial correlation modeling in VLSI design," in *Proc. IEEE/ACM Design Automation Conf. (DAC)*, Jun. 2007.

[3] B. Lee, L.-C. Wang, and M. Abadir, "Refined statistical static timing analysis through learning spatial delay correlations," in *Proc. IEEE/ACM Design Automation Conf. (DAC)*, Jul. 2006.

[4] H.-G. Stratigopoulos, P. Drineas, M. Slamani, and Y. Makris, "RF specification test compaction using learning machines," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, pp. 998–1002, Jun. 2010.

[5] S. Biswas and R. D. S. Blanton, "Statistical test compaction using binary decision trees," *IEEE Design & Test of Computers*, vol. 23, no. 6, pp. 452–462, Jun. 2006.

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

[7] R. Voorakaranam, S. S. Akbay, S. Bhattacharya, S. Cherubal, and A. Chatterjee, "Signature testing of analog and RF circuits: Algorithms and methodology," *IEEE Trans. on Circuits and Systems*, vol. 54, no. 5, pp. 1018–1031, May 2007.

[8] S. S. Akbay and A. Chatterjee, "Fault-based alternate test of RF components," in *Proc. IEEE Int'l Conf. on Computer Design (ICCD)*, Oct. 2007.

[9] D. Mannath, D. Webster, V. Montano-Martinez, D. Cohen, S. Kush, T. Ganesan, and A. Sontakke, "Structural approach for built-in tests in RF devices," in *Proc. Int'l Test Conf. (ITC)*, Nov. 2010.

[10] H. Ayari, F. Azais, S. Bernard, M. Comte, M. Renovell, V. Kerzerho, O. Potin, and C. Kelma, "Smart selection of indirect parameters for DC-based alternate RF IC testing," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2012.

[11] H. Ayari, F. Azais, S. Bernard, M. Comte, V. Kerzerho, O. Potin, and M. Renovell, "Making predictive analog/RF alternate test strategy independent of training set size," in *Proc. Int'l Test Conf. (ITC)*, Nov. 2012.

[12] N. Akkouche, S. Mir, E. Simeu, and M. Slamani, "Analog/RF test ordering in the early stages of production testing," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2012.

[13] M. Chen and A. Orailoglu, "Test cost minimization through adaptive test development," in *Proc. IEEE Int'l Conf. on Computer Design (ICCD)*, Oct. 2008.

[14] S. Krishnan and H. G. Kerkhoff, "A robust metric for screening outliers from analogue product manufacturing tests responses," in *IEEE European Test Symp. (ETS)*, May 2012.

[15] N. Sumikawa, J. Tikkanen, L.-C. Wang, L. Winemberg, and M. S. Abadir, "Screening customer returns with multivariate test analysis," in *Proc. Int'l Test Conf. (ITC)*, Nov. 2012.

[16] E. Yilmaz, S. Ozev, and K. M. Butler, "Per-device adaptive test for analog/RF circuits using entropy-based process monitoring," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 6, pp. 1116–1128, Jun. 2013.

[17] K. R. Gotkhindikar, W. Daasch, K. M. Butler, J. M. Carulli Jr., and A. Nahar, "Die-level adaptive test: Real-time test reordering and elimination," in *Proc. Int'l Test Conf. (ITC)*, Sep. 2011.

[18] X. Li, R. Rutenbar, and R. Blanton, "Virtual probe: A statistically optimal framework for minimum-cost silicon characterization of nanoscale integrated circuits," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design (ICCAD)*, Oct. 2009.

[19] H.-M. Chang, K.-T. Cheng, W. Zhang, X. Li, and K. Butler, "Test cost reduction through performance prediction using virtual probe," in *Proc. Int'l Test Conf. (ITC)*, Sep. 2011.

[20] W. Zhang, X. Li, and R. A. Rutenbar, "Bayesian virtual probe: Minimizing variation characterization cost for nanoscale ic technologies via bayesian inference," in *Proc. IEEE/ACM Design Automation Conf. (DAC)*, Jun. 2010.

[21] W. Zhang, X. Li, E. Acar, F. Liu, and R. A. Rutenbar, "Multi-wafer virtual probe: Minimum-cost variation characterization by exploring wafer-to-wafer correlation," in *Proc. IEEE/ACM Int'l Conf. on Computer-Aided Design (ICCAD)*, Nov. 2009.

[22] N. Kupp, K. Huang, J. Carulli, and Y. Makris, "Spatial estimation of wafer measurement parameters using gaussian process models," in *Proc. Int'l Test Conf. (ITC)*, Nov. 2012.

[23] J. B. Brockman and S. W. Director, "Predictive subset testing: optimizing IC parametric performance testing for quality, cost, and yield," *IEEE Trans. on Semiconductor Manufacturing*, vol. 2, no. 3, pp. 140–113, Aug. 1989.

[24] W. R. Daasch and R. Madge, "Variance reduction and outliers: Statistical analysis of semiconductor test data," in *Proc. Int'l Test Conf. (ITC)*, Nov. 2005.

[25] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Jour. of Royal Statistical Society*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[26] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Jour. of Royal Statistical Society*, vol. 68, no. 1, pp. 49–67, Feb. 2006.

[27] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and Its Applications*, vol. 284, pp. 193–228, Nov. 1998.

[28] E. J. Marinissen, A. Singh, D. Glotter, M. Esposito, J. M. Carulli, A. Nahar, K. M. Butler, D. Appello, and C. Portelli, "Adapting to adaptive testing," in *Proc. Conf. Design, Automation, and Test in Europe (DATE)*, Mar. 2010.