
Theses and Dissertations

Fall 2018

On improving estimation of root cause distribution of volume diagnosis

Yue Tian
University of Iowa

Follow this and additional works at: <https://ir.uiowa.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

Copyright © 2018 Yue Tian

This dissertation is available at Iowa Research Online: <https://ir.uiowa.edu/etd/6653>

Recommended Citation

Tian, Yue. "On improving estimation of root cause distribution of volume diagnosis." PhD (Doctor of Philosophy) thesis, University of Iowa, 2018.

<https://doi.org/10.17077/etd.hi96-s8f5>

Follow this and additional works at: <https://ir.uiowa.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

ON IMPROVING ESTIMATION OF ROOT CAUSE DISTRIBUTION OF VOLUME
DIAGNOSIS

by
Yue Tian

A thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

December 2018

Thesis Supervisors: Professor Sudhakar M. Reddy
Dr. Wu-Tung Cheng

Copyright by

YUE TIAN

2018

All Rights Reserved

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Yue Tian

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree degree in Electrical and Computer Engineering at the December 2018 graduation.

Thesis Committee: _____
Sudhakar M. Reddy, Thesis Supervisor

Wu-Tung Cheng, Thesis Supervisor

Jon G Kuhl

Mona Garvin

Soura Dasgupta

Hantao Zhang

To My Family

ACKNOWLEDGMENTS

I would like to take this opportunity to express my appreciation to all people who have supported me to complete this journey.

First and foremost, I would like to express my sincere gratitude to my Thesis Supervisor, Professor Sudhakar M. Reddy. I truly appreciate the opportunity to work with him. His knowledge, encouragement, working ethic, and critical way of thinking in research have inspired me since the first day I worked with him. This work cannot be completed without his patience, guidance, and support. I also want to express my equal gratefulness to my Thesis Advisor Dr. Wu-Tung Cheng. I've learned a lot from all the guidance, discussion, and candid talks in the past few years. His patience, encouragement, and advice on both research and life help me go through the difficult time and reform me to be a better self.

I would also like to thank the members of my dissertation committee: Professors Jon G. Kuhl, Soura Dasgupta, Mona Garvin, and Hantao Zhang for serving on my committee and giving valuable suggestions. Many thanks also go to my colleagues Gaurav Veda, Huaxing Tang, and Manish Sharma at Mentor Graphics for all the discussions, support, and generous help throughout my research.

I would like to extend my thanks to my other friends and colleagues at the University of Iowa and Mentor Graphics. Thank you all for the support and kindness through the time. It is important to me.

Lastly, I want to thank my family, whose unconditional love is always the source of my strength. To my mother Ruishu Lin, father Suiliang Tian, and sister Xin Tian, at any difficult time in my life, I know you will never give up on me and always be proud of me. A special thanks to my husband Yushi Wang. Life is not always easy and might never be. From the first step of this journey till the future chapters of our life, your love and trust mean the world to me, more than you know.

ABSTRACT

Identifying common root causes of systematic defects in a short time is crucial for yield improvement. Diagnosis driven yield analysis (DDYA) such as Root cause deconvolution (RCD) is a method to estimate root cause distribution by applying statistical analysis on volume diagnosis. By fixing identified common root causes, yield can be improved.

With advanced technologies, smaller feature size and more complex fabrication processes for manufacturing VLSI semiconductor devices lead to more complicated failure mechanisms. Lack of domain knowledge of such failure mechanisms makes identifying the emerging root causes more and more difficult. These root causes include but are not limited to layout pattern (certain prone to fail layout shapes) and cell internal root causes. RCD has proved to have certain degree of success in previous work, however, these root causes are not included and pose a challenge for RCD. Furthermore, complex volume diagnosis brings difficulty in investigation on RCD. To overcome the above challenges to RCD, improvement based on better understanding of the method is desired.

The first part of this dissertation proposes a card game model to create controllable diagnosis data which can be used to evaluate the effectiveness of DDYA techniques. Generally, each DDYA technique could have its own potential issues, which need to be evaluated for future improvement. However, due to limitation of real diagnosis data, it is difficult to, 1. Obtain diagnosis data with sufficient diversity and 2. Isolate certain issues and evaluate them separately. With card game model given correct statistical model parameters, impact of different diagnosis scenarios on RCD are evaluated. Overfitting problem from limited sample size is alleviated by the proposed cross validation method.

In the second part of this dissertation, an enhanced RCD flow based on pre-extract layout patterns is proposed to identify layout pattern root causes. Prone to fail layout patterns are crucial factors for yield loss, but they normally have enormous number of types

which impact the effectiveness of RCD. Controlled experiment shows effectiveness of enhanced RCD on both layout pattern root causes and interconnect root causes after extending to layout pattern root causes. Test case from silicon data also validates the proposed flow.

The last part of this dissertation addresses RCD extension to cell internal root causes. Due to limitation of domain knowledge in both diagnosis process and defect behavior, parameters of RCD model are not perfectly accurate. As RCD moves to identify cell internal root causes, such limitation become an unescapable challenge for RCD. Due to inherent characteristics of cell internal root cause, RCD including cell internal root cause faces more difficulty due to less accurate model parameters. Rather than enhancing domain knowledge, supervised learning for more accurate parameters based on training data are proposed to improve accuracy of RCD. Both controlled experiments and real silicon data shows that with parameters learned from supervised learning, accuracy of RCD with cell internal root cause are greatly improved.

PUBLIC ABSTRACT

For manufacturing of VLSI semiconductor devices, acceptable and stable yield must be achieved in a short time before volume production. Yield learning is a procedure to identify root cause of defects which need to be fixed. Root cause deconvolution (RCD) is a yield learning method that estimates root cause distribution based on volume diagnosis data. RCD has proved to have certain degree of success in previous work.

At advanced technology node, layout pattern (certain prone to fail layout shapes) and cell internal root causes (root cause of defect inside a library cell) have been main causes of yield loss in many cases. Improvements to RCD are needed to adapt to such new types of root causes.

Improvements to RCD should be based on proper evaluation and understanding of its statistical model. However, volume diagnosis data is design dependent and difficult to have sufficient diversity, which brings difficulty in evaluation of the statistical model. We propose a card game model to create controllable diagnosis data with various scenarios. Using card game data, issues in RCD model can be separated and evaluated.

We then propose an enhanced RCD flow to identify layout pattern root causes effectively. Lastly, we propose a supervised learning method to improve RCD accuracy including cell internal root causes. Due to inherent characteristic of cell internal root causes, inaccurate RCD model parameters bring challenges to RCD accuracy. Proposed supervised learning method learns a set of more accurate model parameters and improves RCD accuracy.

TABLE OF CONTENTS

LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER I INTRODUCTION.....	1
1.1 Background and Motivation of DDYA.....	1
1.2 Challenges for RCD	2
1.3 Organization of Thesis	4
CHAPTER II REVIEW OF DIAGNOISIS AND DIAGNOSIS DRIVEN YIELD ANALYSIS.....	5
2.1 Defects and Physical Features.....	5
2.1.1 Overview of Defect.....	5
2.1.2 Types of Defect.....	6
2.1.2.1 Systematic Defects and Impact to Yield Loss	7
2.1.2.2 Random Defects and Their Impact on Yield Loss.....	8
2.1.3 Design Feature	9
2.1.3.1 Extraction of Prone-to-fail Design.....	10
2.2 Fault Diagnosis.....	11
2.2.1 Fault Model	11
2.2.2 Logic Diagnosis	13
2.2.3 Layout-aware Diagnosis.....	14
2.2.4 Cell Internal Defect Diagnosis.....	15
2.2.5 Diagnosis Ambiguity	17
2.3 Yield Learning Method (Traditional, Non-volume Diagnosis Based)	18
2.3.1 Memory Based Analysis	19
2.3.2 Analysis of Wafer History	20
2.3.3 Inline Defect Inspection.....	20
2.3.4 Test Chip.....	21
2.3.5 Test Structures.....	21
2.3.6 PFA	21
2.4 Yield Learning Method (Volume Diagnosis Data Based)	23
2.4.1 Volume Diagnosis Based Analysis	23
2.4.1.1 Overview.....	23
2.4.1.2 Advantages	24
2.4.1.3 Challenges.....	25
2.4.2 Previous Work.....	26
2.4.2.1 Feature Failure Rate Estimation	27
2.4.2.2 Feature/Root Cause Probability Estimation.....	32
2.4.2.3 Defect Probability Estimation.....	33
2.4.2.4 Layout Shape Feature Identification.....	33
2.4.3 Summary	35
CHAPTER III STATISTICAL MODEL OF RCD	36
3.1 Background and Objective	36
3.2 Motivation and Problem Description	37
3.3 Stages of RCD.....	41
3.4 Details of RCD Learning.....	42
3.4.1 Typical Diagnosis Report.....	42

3.4.2	Bayes Net of Volume Diagnosis	44
3.4.3	EM Procedure.....	46
3.5	RCD Flow and Bayesian Model Parameter	47
3.6	Possible Challenge for RCD	50
CHAPTER IV CARD GAME MODEL FOR DDYA INVESTIGATION.....		51
4.1	Introduction and Motivation.....	51
4.2	Simplified Volume Diagnosis Model.....	54
4.3	Card Game.....	55
4.4	Card Game Experiment.....	60
4.4.1	Single Picked Card Deck	60
4.1.2.1	Scenario 1	61
4.1.2.2	Scenario 2	61
4.1.2.3	Scenario 3	62
4.1.2.4	Scenario 4	63
4.1.2.5	Scenario 5	64
4.4.2	Multiple Picked Card Deck	64
4.5	Volume Diagnosis Practical Usage	66
4.5.1	Volume Diagnosis Model Parameter Issues	67
4.5.2	Data Bias by Limited Sample Size.....	69
4.6	Conclusion.....	76
CHAPTER V IDENTIFICATION OF YIELD LIMITING LAYOUT PATTERNS		77
5.1	Introduction and Previous Work	77
5.1.1	Previous Work.....	78
5.1.2	Handling Issues of Previous Work.....	80
5.2	Layout Pattern Analysis (LPA) Flow	81
5.2.1	Automatic Layout Pattern Extraction	83
5.2.2	Pattern Matching	85
5.2.3	RCD with Layout Patterns	86
5.2.3.1	Huge number of root causes	86
5.2.3.2	Equivalent Root Cause	88
5.3	Defect Injection Procedures	89
5.3.1	General Steps of Defect Injection Experiment.....	89
5.3.2	Pre-defined Root Cause List	90
5.3.3	Step 2 - Random Sampling	92
5.3.4	Step 3 - Defect Injection	93
5.4	Controlled Experiment	93
5.4.1	Objective	93
5.4.2	Experiment Setting.....	94
5.4.3	Result	95
5.4.4	Discussion	98
5.4.5	Conclusion	99
5.5	Silicon Data Validation	99
5.5.1	Setting	99
5.5.2	Silicon Data Validation Result.....	102
5.6	Conclusion.....	106
CHAPTER VI SUPERVISED LEARNING BASED RCD USING VOLUME CELL-AWARE DIAGNOSIS.....		107
6.1	Introduction	107
6.2	Background and Problem Motivation	109
6.3	Supervised Machine Learning Techniques	115
6.3.1	Feature Extraction	115

6.3.2	Type of Classifiers	116
6.4	Our Use of Supervised Learning.....	116
6.4.1	Training Data	117
6.4.2	Learning Algorithm.....	119
6.4.3	Objective Function.....	122
6.5	Experiment Setup	124
6.6	Experiment Result.....	127
6.7	Result on Silicon Data.....	139
6.8	Conclusion.....	144
CHAPTER VII CONCLUSION		145
7.1	Conclusion.....	145
7.2	Future Work	146
REFERENCES		149

LIST OF TABLES

Table 4-1 MLE success rate of single picked card deck.....	65
Table 4-2 MLE success rate of double picked card decks.....	66
Table 4-3 MLE success rate of single picked card deck with half sample size.....	66
Table 4-4 Single picked card deck with Cross Validation.....	75
Table 4-5 Double picked card decks with Cross Validation.....	75
Table 4-6 Single picked card deck with half sample size with Cross Validation.....	75
Table 5-1 Results of layout pattern injection experiment.....	95
Table 5-2 systematic defect identification resolution improvement.....	105
Table 5-3 PFA candidate searching area reduced.....	105
Table 6-1 Percentage of root causes experiencing high co-occurrence and domination	114
Table 6-2 Number of predefined root causes per design	125
Table 6-3 Success rates for cell internal root causes	127
Table 6-4 Success rates for interconnect root causes.....	127
Table 6-5 Comparison of RCD top root cause vs. PFA results. W id stands for Wafer id. Interconnect is abbreviated to Inter.	142

LIST OF FIGURES

Figure 2-1 Example of “spot defect” [1].....	6
Figure 2-2a Image of open [14] and bridge [48] defects	7
Figure 2-2b Image of open defects related to via [39], [41]	7
Figure 2-3 Image of prone-to-fail layout structure and open defect caused by such structure [43].....	10
Figure 3-1 Example of 500 points drawn from the mixture of 3 Gaussians [68]	39
Figure 3-2 schematics of a typical diagnosis report.....	43
Figure 3-3 Flow Chart of RCD	48
Figure 4-1 An example of a bridge fault.....	53
Figure 4-2 A simple example.....	57
Figure 4-3 (a) 4 drawn cards; (b) 3 drawn cards; (c) 2 drawn cards; (d) 1 drawn cards	59
Figure 4-4 Mapping between Card game and Diagnosis.....	67
Figure 4-5 Biased data from underlying probability.....	70
Figure 4-6 (a) Over-fitting distribution; (b) Under-fitting distribution	71
Figure 4-7 (a) Cross validation 2 fold data; (b) Under-fitting distribution on training data; (c) fitting distribution on training data; (d) Over-fitting distribution on training data; (e) Under-fitting distribution on test data; (f) Fitting distribution on test data; (g) Over-fitting distribution on test data.....	74
Figure 5-1 Shifted layout patterns.....	79
Figure 5-2 Proposed Layout pattern analysis flow	82
Figure 5-3 Layout pattern window formation.....	83
Figure 5-4 Inner points that can be open locations	84
Figure 5-5 Projections adds more POIs	85
Figure 5-6 most layout patterns occur in only a very few diagnosis reports	87
Figure 5-7 Non-distinguishable Root Cause.....	89
Figure 5-8 Example of layout pattern extracted from layout.....	92

Figure 5-9 RCD result as sample size increases: Critical area open injection excluding layout patterns as candidate root causes	96
Figure 5-10 RCD result as sample size increases: Critical area open injection including layout patterns as candidate root causes	97
Figure 5-11 RCD result as sample size increases: Critical area bridge injection excluding layout patterns as candidate root causes	97
Figure 5-12 RCD result as sample size increases: Critical area bridge injection including layout patterns as candidate root causes	98
Figure 5-13 Pareto for RCD root causes.....	101
Figure 5-14 Zoom-in view of 5 equivalent layout patterns from the top list.....	101
Figure 5-15 Searching area without layout-aware diagnosis	102
Figure 5-16 M4 OpenPattern 15980827 highlighted by POI.....	103
Figure 5-17 M2 OpenPattern highlighted by blue POI.....	103
Figure 5-18 PFA results for 3 dies	104
Figure 6-1 Flow chart of RCD using supervised learning for parameter estimation.....	126
Figure 6-2 (a) Accuracy of cell-internal root causes; (b) Accuracy of interconnect root causes.....	132
Figure 6-3 (a) Original $P(r/c)$ per report; (b) $P(r/c)$ per report after transformation by learnt A	133
Figure 6-4 Accuracy of cell-internal root causes with adjacent layer measurement	139
Figure 6-5 Root-cause distribution output by RCD for wafer C using the new method and baseline method.....	143
Figure 6-6 PFA result highlighting CELL_OPEN between layer 2 and layer 4	144

CHAPTER I

INTRODUCTION

1.1 Background and Motivation of DDYA

Yield is defined as the percentage of good dies among all dies manufactured. It is an important factor in product profits. In the early stages of manufacturing a new product, when a new manufacturing process is introduced for an existing design, or when a new design is introduced to an existing manufacturing process, the first lot of manufactured devices usually has a yield lower than expected. When lower-than-expected yield occurs, yield engineers need to identify the defects, understand their root cause, and modify the design or the manufacturing process to improve the yield. Acceptable yield needs to be achieved before volume production begins and needs to be maintained during volume production. Due to decreasing time-to-market and time-to-volume constraints [37], a high and stable yield has obviously become the key factor of product profit in the semiconductor industry.

To improve yield, firstly, the root causes of systematic yield loss need to be identified. The design or manufacture process then needs to be modified according to the nature of these root causes. With decreasing feature sizes and increasing complexity of fabrication processes for manufacturing VLSI semiconductor devices, more systematic defects occur at the advanced technology nodes. Product yield ramp up is mostly determined by how quickly systematic defects are identified and fixed. Identifying common root causes in a short time is crucial for yield improvement.

Traditional yield learning methods such as physical failure analysis (PFA), inline inspection and test structure have been used to identify systematic root causes. However, these methods appear to be less effective in identifying root causes due to the design-iteration process. For example, given the long times and expense of PFA, using PFA on a large number of failing devices to find systematic defects is becoming unfeasible.

For this reason, yield learning by appropriate statistical analysis on volume diagnosis reports (that is, a large amount of reports) has come into use as a way to automatically identify common physical defect features. This approach can reduce turnaround time and cost by speeding up the process of systematic defect identification. The identified root cause information can be used not only to improve yield analysis, but also to reduce PFA costs by focus on failing devices with systematic defects. Such an approach will be referred to below as diagnosis driven yield analysis (DDYA) in this thesis.

Inherent ambiguity from diagnosis data is one major challenge DDYA needs to address. Among the existing DDYA techniques, Root Cause Deconvolution (RCD) [48] is a statistical method that addresses such ambiguity directly by estimating the underlying root cause distribution base on unsupervised learning from volume diagnosis data.

1.2 Challenges for RCD

Emerging new defect features: According to Moore's law, the number of transistors that can be integrated on a chip of a given size will double approximately every 18 months. At advanced technology node, systematic defects caused by new design features become major yield limiting factors due to more complex manufacturing process, smaller feature size and more complicated failure mechanism. Those emerging defect features include, but are not limited to, layout pattern root cause and cell internal root causes.

Layout pattern root cause refers to a specific hard-to-manufacture layout structure that becomes prone to fail. Such prone to fail features occur more and more frequently as feature size of the layout shrinks dramatically till smaller than lithography wavelength. How to identify such layout features is an open question. Also, the enormously large number of potential layout patterns to be considered for statistical analysis poses a challenge for statistical analysis.

Cell internal root cause refers to cause of defect inside a standard library cell. At more advanced technology node, library cells require more process steps and more

complicated transistor structures. Also, to handle higher process variations, there is increasing use of custom designed cells. All these factors lead to a large number of manufacturing defects and systematic root cause inside library cells, which have more subtle defect behaviors. Due to the complicated structure of library cell and current limited domain knowledge of the defect behaviors, identifying cell internal root causes poses a challenge for RCD.

RCD has been proven effective and has had a degree of success in previous work [44], [48], [49]; this existing work, however, does not consider prone-to-fail layout structures and cell internal root causes as candidate root causes.

Limited and complex diagnosis data for DDYA: In reality, volume diagnosis data are not comprehensive, and are limited in size due to practical reasons. One of the reasons is that not all failing die are sent for diagnosis. For each die on a wafer, several tests are applied, and only those dies passing the previous test are then sent on for Automatic Test Pattern Generation (ATPG) scan testing and diagnosis. To investigate the effectiveness of not only RCD but also other DDYA techniques, we used the failure files of both silicon defects and simulated defects for experiments. However, due to complex nature of design and test patterns used, it was not easy to ensure that simulated defects and silicon defects can create diagnosis reports with sufficient diversity. Such constraints on both size and diversity of diagnosis data limited the thoroughness and potential usefulness of the present investigations on DDYA techniques. It is necessary to create controllable diagnosis reports with the desired diversity of scenarios in order to investigate and improve DDYA techniques, including RCD.

To sum up, improvement and adaptation to new root causes are needed for RCD to overcome the challenges from emerging new root causes. Also, a vehicle with controllable diagnosis data for better understanding of not only RCD but general DDYA method would be beneficial for yield learning research.

1.3 Organization of Thesis

The following chapters of this thesis are organized as follows.

Chapter 2 first reviews the background and basic concept of defect, root cause and fault diagnosis. Then traditional yield learning methods and yield learning approaches using volume test data are brief discussed. Lastly, several DDYA techniques, including RCD, are reviewed.

Chapter 3 describes the methodology of RCD. The details of RCD probability model are presented, and some challenges RCD might face are discussed.

In Chapter 4, a card game model is proposed to create various controllable diagnosis data that can be used to evaluate the effectiveness of DDYA techniques and conduct investigation for improve DDYA techniques. With correct statistical model parameters and limited samples, card game model is used to evaluate the impact of root cause numbers, instance counts and correlation among root causes on RCD result. The effectiveness of RCD with limited samples is evaluated. Overfitting problem due to limited sample is alleviated by the proposed cross validation method.

In Chapter 5, an enhanced RCD flow is proposed to handle layout shape related root causes. The advantages of the proposed RCD flow are discussed comparing to an existing method. Results of RCD from both controlled experiments and industrial silicon data validate the effectiveness of proposed flow.

Chapter 6 addresses RCD extension to cell internal root causes. A supervised learning technique is proposed to improve the parameter estimation of the RCD model, and thereby increase the accuracy of the final RCD result. Both controlled experiments and real silicon data show that with learned parameters, the accuracy of RCD with cell internal root causes are greatly improved.

Chapter 7 concludes the thesis and discusses future work.

CHAPTER II

REVIEW OF DIAGNOISIS AND DIAGNOSIS DRIVEN YIELD ANALYSIS

In this chapter, we provide an overview of existing diagnosis techniques and diagnosis based yield learning methods. Section 2.1 explains some basic terminology and concepts. Section 2.2 reviews techniques in fault diagnosis. In section 2.3, a review of the traditional yield learning methods is presented. Section 2.4 compares the traditional methods of yield learning with volume diagnosis based yield learning and discusses the challenges associated with learning from volume diagnosis, before finally discussing and evaluating previous work on volume diagnosis based yield analysis.

2.1 Defects and Physical Features

2.1.1 Overview of Defect

Defects: Physical defects of Integrated Circuit (IC) chips are the catastrophic deformation of physical structures on the layout of a manufactured die. Figure 2-1 [1] shows an example in which a “spot defect” caused the deformation. Possible catastrophic deformations include: break of a conducting path, an unintended short-circuit between active areas, and the unwanted appearance of an active or parasitic device[2]. Based on the source and the nature of their behavior, defects can be categorized into two types: random defects and systematic defects. In diagnosis context, a defect is also referred as a specific physical defect effect, as mentioned in Chapter 3.

Sources of defect: During each step of the manufacturing process, many types of incidents or changing factors can lead to a defect that fails the die and therefore causes yield loss. Sources that can cause a defect and impact yield can include but are not limited to human errors, technology maturity, equipment failure, process variation (instability of

process parameters and material inhomogeneities), wafer environmental conditions, and tool set [3], [37].

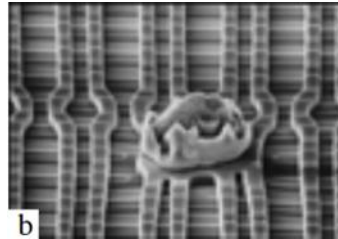


Figure 2-1 Example of “spot defect” [1]

2.1.2 Types of Defect

Defect can be categorized by type of malformation, landing locations or characteristic of occurrence. Open defects and bridge defects are two basic types of malformations that can occur during chip fabrication. Possible scenarios are bridging between two neighboring signal lines that are too close, open interconnect of a net that is too narrow, open or bridge defects inside a library cell, and single vias. Via connects mental layers and a missing via or deformed via can cause open defects on layers. Defects can also be classified into two categories based on the defect location: a defect in a library cell is called a cell internal defect, and a defect on interconnecting wires is called an interconnect defect.

Figure 2-2a shows images of a bridge defect [48] and open defect [14] at interconnect wire. Figure 2-2b [39], [41] shows different defect scenarios related to via.

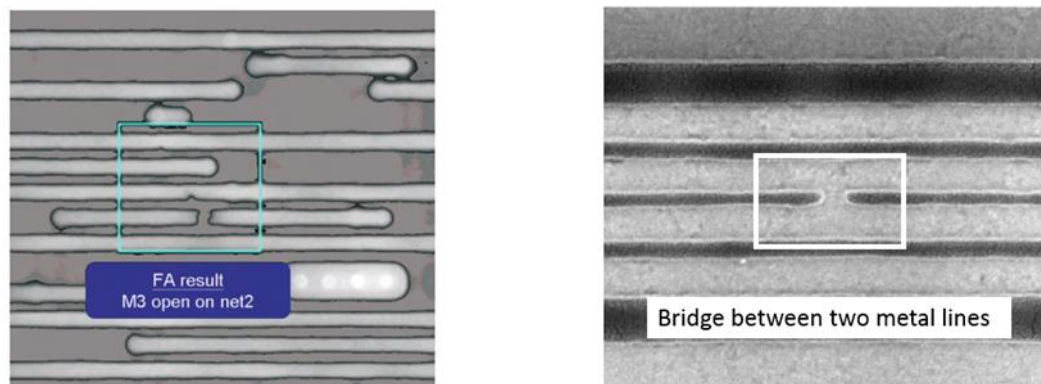


Figure 2-2a Image of open [14] and bridge [48] defects

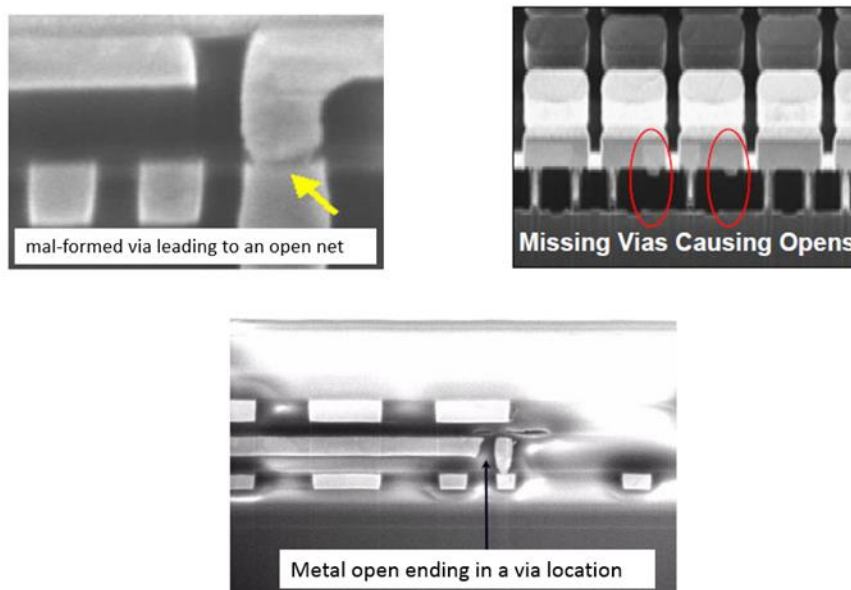


Figure 2-2b Image of open defects related to via [39], [41]

2.1.2.1 Systematic Defects and Impact to Yield Loss

Systematic defects are those that occur repeatedly at different locations of the same design and have a common cause. Such causes can include but are not limited to tool excursions, design marginalities, design-process interactions, test issues, mask errors, and

parametric variations [42]. Fixing the common cause will eliminate occurrence of such a group of defects and improve yield significantly.

In today's technologies, as the size of physical features of design become smaller and smaller, manufacturing processes become more complex and much less predictable [4],[5]. At the point that the minimal feature size of a layout becomes smaller than the lithographic wavelength, certain layout features become hard to fabricate correctly and are more likely to cause failures than other features. This kind of prone-to-fail feature is called critical feature in some literature [43]. Besides limitations of lithography, improperly validated Design for Manufacture (DFM) rules can also lead to defects during the manufacturing process of a critical feature. Other systematics, such as parameter variation and antenna effects, have also been studied [42].

Systematic defects caused by design-process iteration can be fixed by design changes, process changes, or multiple new masks [55]. For example, two neighboring nets that are too close to each other can cause a bridging defect. If there are 10 instances of this feature in the entire layout, any of them could be defective and fail a die. We need to change the minimal distance of two adjacent lines for only these 10 instances to avoid failing the die. Such a change requires much less effort than changing the width between every two adjacent lines. In this example, the distance between two adjacent lines of the 10 instances is considered the design feature of this systematic defect. In section 2.1.3, such system physical feature will be discussed in detail.

2.1.2.2 Random Defects and Their Impact on Yield Loss

Other than systematic defect, random defect is also a main reason a die fails. Random defects can be caused by different sources. One of them is random contamination particles. Random particles can cause bridge defect if they land between two nets, or open defect if they land on a net. Traditionally, this type of defect limited the yield and has been studied and modeled by critical area of the defect site. Historically it has been considered

the dominant yield loss mechanism [37] for high volume production. Critical area-based yield model could be based on either test structure data or production test data [6]. Distribution of random defects can be modeled by a distribution such as Poisson and Negative Binomial [7].

Eliminating random defects has a lower priority than eliminating systematic defects due to its cost and impact. For example, random defects due to contaminate particles can be controlled by cleaning the manufacturing environment and modifying the manufacturing setting or design. The former is done by ensuring regular cleaning and maintenance of the processing tools, chambers, and wafer containers [52]. It is difficult to remove all particles, but important to remove as many as possible. To eliminate random defects, every location of a layout need to be considered because each of them could be potentially affected. Comparing to eliminating systematic defects by fixing certain design features appearing in limited locations, it would be too expensive.

2.1.3 Design Feature

A design feature [40] is the characteristic of a cell, metal layer, via or layout shape. Library cells can be characterized by cell type, logic function, and drive strength. Features inside a library cell can also be characterized by their physical layout. Similarly, characteristics of interconnects include metal-via overlap, stacked/not, single/multiple/array, via layer, metal density, and metal length. A physical feature is the property of physical defects. In this thesis, we used “feature,” “physical feature” and “root cause” interchangeably.

Sharma *et al* [39] gives a detailed description of design features leading to open defects, especially different scenarios of prone-to-fail vias. Keim *et al* [40] identifies all single vias as the source of open defects, and identifies 5 known layout features causing bridge defects, such as side-to-side and corner-to-corner. Schuermyer *et al* [41] provides an overview of design features with additional description.

Recently, as manufacturing processes have advanced, specific layout patterns in a design that are hard to fabricate have been identified as key causes of systematic defects [42], [43], [51], [54], [55], [58]. Layout shape can be characterized by pixel image of the entire shape, polygons or just center lines [51], [57]. Figure 2-3 shows images of a prone-to-fail layout structure and defects caused by this structure [43].

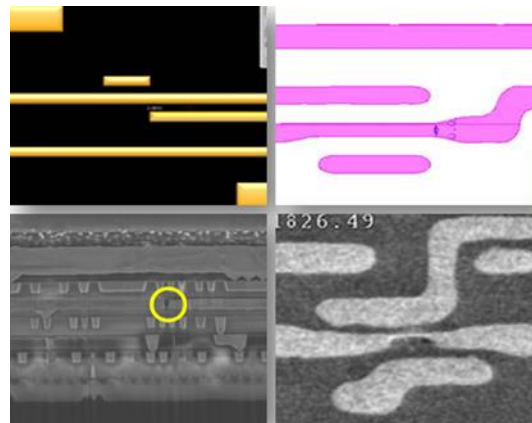


Figure 2-3 Image of prone-to-fail layout structure and open defect caused by such structure [43]

2.1.3.1 Extraction of Prone-to-fail Design

Critical design features can be defined in different ways depending on the amount of information extracted from a layout. For example, in [50], a skeleton of nets is extracted in each layout snippet, and the author claims that the skeleton can capture all critical features that could cause a defect.

Different tools and methods are applied to identify potential critical features. One method is lithographic simulation. In [40], lithographic simulation is performed on a layout, and potential shorts and opens in a standard cell are identified. In [57], hotspots with coordinates are identified by lithographic simulation as critical features. In the flow we proposed for layout shape related root causes in Chapter 5, features of layout shape are

defined in order to indicate certain layout physical structures that are prone to fail. These patterns can be exactly the same or similar, sharing some common features after transformation. The details of how such layout shape features are extracted are described in Chapter 5.2.

2.2 Fault Diagnosis

Fault diagnosis is a method for determining the cause of a failing die. During yield ramp up, physical failure analysis (PFA) relies on diagnosis results to provide possible locations of the defects. There are two main parts in fault diagnosis: scan chain diagnosis and logic diagnosis. Scan chain diagnosis aims to determine the causes of scan chain failure, while logic diagnosis, also called scan diagnosis [8], [9], [10], [11],[12],[13], deals with defects affecting the logic function of a design. Recently, diagnosis unitizing layout information [14], [15], [16] and cell internal defect diagnosis[17], [18], [19], [20], [21], [22], [23], [24], [25] have provided physical information and helped PFA to narrow down possible areas of defect, thus reducing cost and turnaround time. Furthermore, volume diagnosis, which uses diagnosis results from large amounts of failing dies, has shown increasing promise in yield learning to identify and quantify the possible root causes of failing dies. In this section, we will give an overview of fault model, logic diagnosis, layout-aware diagnosis, cell-internal diagnosis and cell-aware diagnosis. The end of this section briefly discusses inherent ambiguity in fault diagnosis.

2.2.1 Fault Model

A fault is the representation of a physical defect. Fault models are representations of failure behaviors of a defect. Fault models can be distinguished by their effect into two types [8]:

1. A *logic fault* affects the logic function of the circuit.
2. A *delay fault* affects the operation speed of the circuit.

In our defect injection experiment, we are interested in bridge and open defects. We focus on models of bridge and open defects that affect the logic function. A short fault indicates an unwanted connection between two points, while an open fault indicates an unintentional break in a connection [8]. Their behaviors can be modeled as follows:

1. Stuck-at-fault model

A signal line that remains at a fixed low/high voltage is modeled as stuck at fault, denoted by s-a-0/1. One example of such a fault is when a short occurs between a signal line and the ground/power.

2. Bridge fault model

Shorts occurring between two signal lines are modeled as bridge faults. This type of defect can often be seen in the layout level when two nets are close to each other. The short usually behaves as a new logic function that produces an unwanted value that changes the destination of the two bridging lines. A bridge fault can be modeled as a logic OR function or a logic AND function. This type of modeling is not always accurate.

Dominant bridge models the situation when the driver of one signal line dominates the logic value of both signal lines. For example, if the signal lines A and B are shorted and the fault type is that A dominates B, then the value of both A and B are the same as the value of A.

3. Open fault model

An open defect occurring on a line will also affect any fan-out branch of that line. If a single stuck-at-fault model is applied for each line, then an open fault will be modeled as stuck at faults that occur at the same time on each of the fan-out branches.

4. Cell aware fault model [26]:

Defects inside library cells, which locate between transistors and interconnect inside gates, can be modeled in a similar way as defects outside cells, which locate at input/output of gates on interconnect wires. Stuck-at, stuck-open, resistor open or bridge are some of the models proposed.

A delay fault [27] causes excessive delay along a path, such that the total propagation delay fault falls outside the specific limit. A transition fault model [28] models the delay of signal transition from an input of a gate to the output. When the signal 0/1 of input is not given enough time to transit to 1/0 at output, transition fault is modeled as slow-to-rise/slow-to-fall. A path-delay fault model [29] focus on modeling the cumulative propagation delay along a pre-defined critical signal path.

2.2.2 Logic Diagnosis

Logic diagnosis is a procedure to compare the test response of the logic part of a failing circuit with that of a good circuit, to analyze the faulty behavior based on a fault model, and finally to provide possible defect types and locations. There are two main approaches: cause-effect and effect-cause.

Cause-effect diagnosis: In this approach, a look-up table is built by recording all possible faults and their test responses. To build the look-up table, the first step is to first specify possible fault models and candidate faults. Then a fault simulation is performed, and the possible test responses of a given test and the faults on-site are stored in the table.

Having the observed test response of a failing circuit, one can find the matching recorded test response in a dictionary and look up its corresponding fault type and location. The most matched test response in the dictionary indicates that its corresponding fault type and location is most likely the actual defect.

This approach requires an accurate fault model and is of limited practical value for the diagnosis of defects whose fault models are not considered in the table.

Effect-cause diagnosis: This approach usually includes three steps: path tracing from failing output, fault simulation on potential fault sites, and ranking based on test response. Before the first step, a good machine fault simulation with a specific test is performed. The test responses of the simulation and the observation are compared. Mismatched outputs are identified.

Path tracing essentially partitions the circuit into different primary output fan-in cones by back-tracing from each mismatched output pin. Each cone where a mismatched output pin resides is considered as a potential fault cone. Therefore, after tracing back from the different failing outputs, the fault candidate should locate at the common area of all potential fault cones. Within the common area, there usually exists more than one possible fault site, each of which indicates the fault type and location. Another possible scenario is that there is no common region found after path tracing. Such an outcome implies that there could be multiple faults in the failing chip [27].

Fault simulation is then performed on these possible fault sites. Test responses for all fault sites are obtained and compared. Finally, all possible fault sites are ranked based on how well they match the observed test responses of the failing circuit.

This approach is more time-consuming and costly because it needs to be applied individually for each failing chip, while cause-effect diagnosis involves only a one-time overhead cost to build the dictionary for different failing chips of the same design.

2.2.3 Layout-aware Diagnosis

Layout-aware diagnosis [14], [15], [16] is a logic level diagnosis with physical information that provides not only a logical explanation, but also layout information for each possible defect site. For each design, the physical layout information of all logical suspect candidates is extracted. This information basically indicates the design features that might cause the target suspect defect. Such features include but are not limited to size and type of metal layers, vias, cells and various layout shapes. Those logical suspect candidates that cannot be mapped to the layout will not be included in the result of diagnosis because it is physically impossible for them to cause a defect. For each suspect reported [15], [37], layout-aware diagnosis identifies the layout polygons (or defect bounding box [14]) around the suspect. Any predefined design feature is then listed as a possible root cause of this suspect, along with its X-Y coordinates.

Layout-aware diagnosis improves resolution of diagnosis by filtering out the physically impossible suspects [14], thereby reducing the size of the suspect list. Also, directly reporting design features such as layer type, cell type, and layout shape supports analysis for the derivation of the root cause by enabling more convenient access to the diagnosis data. Yield learning methods such as PFA and volume diagnosis analysis would be greatly benefited by incorporating layout information.

2.2.4 Cell Internal Defect Diagnosis

Traditional logic diagnosis is based mainly on stuck-at fault model and usually reports the most likely defect location at or between interconnecting wires connecting to a cell. However, this information is not enough to determine whether the real defect occurs on wires or inside the cell that a wire connects to. As feature sizes decrease and fabrication processes become more complicated, finding systematic defects occurring inside cells become essential for yield ramp up. Cell internal defect diagnosis targets to locate defects inside a cell.

Previous work on cell internal defect diagnosis can be put into three categories: Gate-level based, excitation conditions extraction based, and simulation based cell-aware diagnosis.

Gate-level diagnosis based: This approach [23], [24], [25] starts with modeling cell internal defect in transistor level, and then maps the transistor level defect into gate-level defects. On the translated gate-level design, traditional logic diagnosis techniques on gate level then are used to locate the most likely faulty gate and pinpoint the faulty transistor. The success of this approach depends on how well the cell internal defect is modeled. In general, this method is not sufficient for transistor level physical defects [20].

Excitation conditions extraction based[17], [18], [19], [20]: This approach is based on the assumption that excitation of internal defect is highly related to the logic value of input pins [20]. There are two steps in this approach. In first step, conventional gate-level

diagnosis is applied to locate possible defective cells. In second step, excitation conditions of those possible defective cells are extracted based on the logic value on input pins of investigated cells. The failing excitation conditions are a combination of input logic value, which can activate and propagate the cell internal defects to output pins. Accordingly, passing excitation condition is a combination of input logic value that cannot activate or propagate the cell internal defects to output pins. With information of the determined excitation conditions, one can isolate a defective cell from interconnect defects by correlating the passing and failing condition [19], or even determine the actual defect inside the cell with SPICE or switch level simulation [18].

The accuracy of locating defective cells and accuracy of extracted excitation conditions have a direct impact on the effectiveness on this approach. Sharma *et al* [19] proposed active excitation condition to improve accuracy of extracted excitation condition. Fan *et al* [20] further improved excitation condition extraction by tracking fault effect propagation paths. However, since traditional logic diagnosis usually uses the stuck-at fault model, difficulties can arise when multiple cycle patterns are present. To deal with this challenge, [20] proposed X-based simulation and suspect validation by simulating extracted excitation conditions.

One disadvantage of work proposed in [20] for yield learning is that only the defective cell is located but not the defect location within the cell. When the real root cause is a certain layer or layout shape within cells among all cell types, it would be difficult to locate it.

Simulation based (cell-aware diagnosis) [21], [22]: Tang *et al* [22] proposed a methodology to avoid inaccurate extraction on excitation conditions and alleviate the impact of stuck-at fault model by creating an accurate fault model based on analog simulation. The user-define fault models (UDFM) are created as follows: 1. Extract all physical defects from library cell layout. 2. Create fault model based on extracted physical defects. 3. Save both physical information and test condition for each UDFM fault. The

diagnosis algorithm is built on top of existing layout-aware diagnosis with extra steps. Based on simulation results for stuck-at fault based suspects, certain cell instances will be picked, and UDFM faults corresponding to those instances will be simulated explicitly. Following UDFM simulation, the suspect lists from both stuck-at fault based and UDFM fault-based simulations will be merged. At the end, the suspects that most closely match layout information, among both interconnect suspects outside cells and cell internal suspects, are reported. One benefit of the simulation-based fault model compared to the stuck-at fault model is that when a cell-internal defect behaves differently in two cycles, one can inject different fault values of that fault in different cycles. Also, each UDFM faults contains the layout information inside cells; therefore, investigating cell internal root causes for the purpose of yield learning become feasible. This approach is termed cell-aware diagnosis. The proposed work in Chapter 6 is built upon the RCD framework using cell-aware diagnosis.

2.2.5 Diagnosis Ambiguity

Terminology of Diagnosis Report: Typical logic diagnosis provides a list of ranked defect suspects for each report. A report represents a defect, and a suspect represents a potential defect candidate that could explain the behavior of failing chip. The suspect is also referred as physical defect of diagnosis. For layout-aware diagnosis, the physical feature associated with a suspect could be termed a design feature, critical design feature, or root cause. The terminology used in the proposed volume diagnosis model will be summarized in detail in Chapter 3 and illustrated with a graph.

Metric of Diagnosis Performance: Resolution of diagnosis is a metric defined by the total number of defect candidates reported by a diagnosis tool [27]. Ideally the metric is 1, meaning only one defect site is reported. Another metric is accuracy, which indicates how often the target defect can be correctly called out by diagnosis. Although resolution and accuracy are both metrics for diagnostic performance, one doesn't necessary imply

another. The main factor affecting diagnosis [61] is the matching of the fault model and the real physical defect.

Diagnosis Ambiguity: Logic diagnosis is based on the logic description of a circuit and a fault, so it can only provide a logic-level description of a defect. Therefore, multiple faults that are logically equivalent cannot be distinguished. Each logically equivalent fault is possibly real, and there is not much that logic diagnosis can do at this point. This characteristic of logic diagnosis is called diagnostic ambiguity and is measured by resolution of diagnosis. Similarly, just as logic diagnosis cannot distinguish logically equivalent suspects, layout-aware diagnosis also cannot distinguish between physical suspects that have the same logic behavior, and thereby pin down the one true root cause among multiple design feature candidates associated with each physical defect. For example, it is not easy to distinguish all instances of potential physical bridges between a pair of nets [48].

Challenges in diagnosis performance for yield analysis: As we discussed earlier, the performance of logic diagnosis is evaluated in terms of accuracy and resolution. Layout-aware diagnosis is impacted by how design features are extracted, which impacts the root cause uncertainty. Improving accuracy and resolution is always a goal for diagnosis improvement. However, such ambiguity is still inevitable, and bring challenges in yield analysis based on both individual and volume diagnosis. Whichever method is applied should be able to address these issues.

2.3 Yield Learning Method (Traditional, Non-volume Diagnosis Based)

Historically, different yield learning methods have been applied to the industry yield ramp process. These methods, such as test chips, visual wafer maps, memory bitmaps, inline inspection, PFA, and special test structures for parameter evaluation [3], [30], [31], [32], [47], have been proven effective in different scenarios. Still, depending on defect

location distribution and root cause characteristics, these methods are facing challenges for the following reasons:

1. Limited ability to pin down all root cause types. Not all failure mechanisms can be identified by these methods. Some methods (for instance, wafer history analysis) can only detect incidents caused by equipment. Especially for defects related to subtle design layout features, these methods (such as optical in-line inspection) are becoming less applicable.
2. Limited ability to represent diverse layout geometry and all layers. Methods like memory bitmap and PFA can only be applied to a small portion of the layout.
3. Limited ability to discover unknown root causes. Test structures need a predefined set of features and are not able to identify new features.
4. Expensive and time-consuming. As more and more new defects cannot be identified by traditional methods, retest and PFA need to be applied. This process will take additional days or weeks, a requirement which is not acceptable due to short time-to-market cycles. Also, PFA is very expensive.
5. Relying on results from other yield learning methods. PFA can be used as an example. PFA is effective because it can observe the defect site on a layout directly. But this also means it is intrusive and requires the defect site to be accurately located. Other yield learning methods, such as volume diagnosis analysis, are needed to reduce the ambiguity of the diagnosis and aid in PFA.

A brief description of various yield learning methods, and their pros and cons compared to yield learning using volume diagnosis, will be given in the following subsections.

2.3.1 Memory Based Analysis

Memory based analysis has been used to identify defects in a memory part (Static Random-Access Memory, Dynamic Random-Access Memory). A memory bitmap [36] is

created to record the failure of a memory part after a series of read and write operations [3] are performed. By analyzing the bitmap, a failure analyst can easily pinpoint a physical defect [33]. This method is effective in memory parts of which the structure is regular, but cannot handle defect scenarios in more random and complex layout geometry across all layers.

2.3.2 Analysis of Wafer History

Wafer history analysis uses fabrication process data to find a common attribute for the affected failing die [39]. This method focuses on identifying the fabrication process or fabrication equipment that caused the failure. However, for failure mechanisms such as systematic critical layout features it become less effective.

2.3.3 Inline Defect Inspection

Inline inspection [3], [33] collects abnormal data during the wafer manufacturing process. These data are obtained by scanning one layer of the wafer. In the yield learning process, different tests will be applied sequentially. Usually after the failure of first test of inline inspection, a retest is needed for failure analyses. However, one has to wait until the IC is packaged and assembled before one can perform the retest. Days or weeks may pass before the retest can be performed, leading to a long turnaround time. Additionally, optical inspection is unable to detect subtle layout features. Therefore, this process alone is not suitable for the current product and market. However, combining inline inspection with layout analysis [34], [35] can shorten turnaround time and function as a supplement source in identifying defects when diagnosis resolution is low. Desineni *et al* [55], by overlaying defects from inline inspection and layout, identified defects located in a hard-to-diagnose Intellectual Property (IP) core.

2.3.4 Test Chip

Building a test chip is [40] highly effective, but only feasible on a small sample of defective parts. The cost of building a test chip can be prohibitive compared to using production IC for learning.

2.3.5 Test Structures

Test structures[4], [37], [51], [52] are special layout structures such as via chains, comb and serpentine structures, and densely populated lines. They are designed to extract information about specific critical design features during the manufacture process. Test structures are designed to be sensitive only to specific design features, and therefore they are required to have prior knowledge of the targeted root cause for each type of test structure. Once a wafer demonstrates low yield, information about target features, such as size, location and feature failure rate, can be collected immediately. One example [37] is a via chain used to stress a single via feature. By analyzing information regarding the set of potential root causes, action can be taken accordingly. Yield learning by test structure is simple, fast and accurate for specific features.

However, test structures are less effective for systematic defects caused by irregular prone-to-fail layout structures. Such features are design-dependent and can be randomly distributed in a layout. Therefore, for a new design, there could be many unknown features which cannot be tested with test structures. Also, test structures are relegated to the wafer scribe lines [52], or limited to a small number of wafer lots, and are small in size. Therefore, the sample resulting from such a test is not likely to represent the failure distribution of the whole layout. Lastly [37], using test structures assumes that the targeted single root cause triggers the failure each time and gives a theoretical upper bound of its failure rate.

2.3.6 PFA

Physical Failure Analysis (PFA) is applied to manufactured wafers with low yield. The procedure starts by selecting a die from the wafer suffering unexpected low yield. This

die is then sent to yield engineers for physical failure analysis. A potential defect location on the failing chip is exposed, and an image of the potential defect site is taken by one of various methods (e.g., optical microscopy, scanning electron microscopy (SEM), and transmission electron microscopy [TEM]). From the image, yield engineers are able to directly view possible physical layout deformations. If a physical layout deformation has occurred, the chip will be sent to experts. The result of PFA from different chips can differ, but if a similar result is obtained from multiple chips, a group of experts can then decide if this is the true systematic root cause.

PFA is an effective way to identify systematic critical features of design, but it is also very slow and expensive because of the reliance on failure analysis and a Subject Matter Expert (SME) [43].

During the procedure, several decisions need to be made: which typical die to pick in the wafer, which suspect is most likely truly defective, and which root cause is the true cause. Rather than randomly choosing one, engineers can use fault diagnosis and other yield-learning techniques to make the above decision.

The die selected should be typical and best represent the systematic failure mechanism. A spatial yield signature [39] on a wafer map of failing dies can help engineers to choose the target die. Huisman *et al* [62] helps pick the target chip by identifying a cluster of chips that represent a potential systematic root cause.

Fault diagnosis provides a list of potential defect locations and root causes. A PFA engineer can then bring the image of the defect site to a group of experts and let them decide which possible root cause really is the cause of the defect. If the location exposed happens not to be defective, another location will be selected for a new round of failure analysis. However, the increasing damage of deconstructive failure analysis processes requires a higher resolution of diagnosis and more accurate root cause identification methods. Layout-aware diagnosis [14], [15], [16] helps improve the resolution of diagnosis, and volume diagnosis has been used to provide information about systematic

root causes for a population of failing chips. The approach described by Benware *et al* [48] is one where root cause probability distributions are estimated by analyzing volume diagnosis data.

Lastly, due to budget and time-to-market requirements, PFA can analyze only a small portion of all failing dies. If two underlying systematic defects exist in the population of failing chips, it is possible that the dies submitted for failure analysis contain defects caused by only one of the two root causes. In such cases, the PFA result is misleading because of its limited sample size. A method that could identify unknown failure mechanisms in less time and based on the overall population of failing chips would benefit the field.

2.4 Yield Learning Method (Volume Diagnosis Data Based)

2.4.1 Volume Diagnosis Based Analysis

2.4.1.1 Overview

Volume diagnosis is a process in which diagnosis is performed on every individual failing die of a large numbers of failing dies, which are drawn typically from several tens of wafer lots [42]. Volume diagnosis analysis applies statistical analysis to the diagnosis results from large numbers of failing dies and then, through different statistical methods, derives an overview of the defect root cause. Diagnosis may refer to logic diagnosis, scan chain diagnosis, or any other diagnostic technique that attempts to localize the defects [42].

Volume diagnosis based yield learning can be applied to different stages of the yield ramp procedure. First, yield loss from potential failures caused by random spot defects is modeled by certain yield model-for example, a yield model based on critical area. Then the presence of systematic defects is revealed by the deviation of wafer final test (WFT) yields from those predicted by yield models based on IC critical area[2]. These

underlying systematic defects need to be discovered, identified, qualified, ranked, and then fixed. Next, in high volume production, continued yield monitoring is required in case of unexpected yield drop that might be caused by a drifting litho process [40] or new systematic defects caused by previous modifications.

In the following section, we will first review the advantages of volume diagnosis based yield learning over traditional yield learning methods. Difficulties inherent to volume diagnosis are then discussed. Lastly, several previous works are reviewed.

2.4.1.2 Advantages

As discussed above, significant differences exist between volume diagnosis based yield analysis and various other yield learning techniques described in the previous section. Overall, volume diagnosis based analysis is cost efficient, allows short turn-around time, is able to identify previously unknown yield loss mechanisms, and is naturally more representative of complex layouts over all layers. It can also work as a supplement to an existing method such as PFA.

Compared to analysis using a failure signature [42], [55]: Yield learning methods that analyze the signature of failing chips are based on the assumption that the same underlying root cause will produce repeating failure signatures. Volume diagnosis data, however, can handle a case in which one root cause could have resulted in different signatures.

Compared to memory based analysis: All of the process layers and the diverse layout geometry can be represented in the diagnosis of random logic [40].

Compared to analysis using a test structure[4], [51]: Statistical analysis of volume diagnosis is more cost-effective and less intrusive compared to using a test structure. Volume diagnosis data is able to represent the overall geometric diversity of a layout, while a test structure can be applied to only a small region of a die. Therefore, for identification of systematic defects caused by a prone-to-fail physical structure, using a test structure

could be less effective than volume diagnosis. Lastly, volume diagnosis can identify the unknown root cause of the failure of a new design, while a test structure, because it needs an existing set of known root causes, cannot be applied on such a feature. A study by Kruseman *et al* [4] shows an example where two systematic defects are not detected by test structures.

Compared to PFA based on individual diagnosis report:

1. Helping PFA to reduce diagnosis ambiguity and increase resolution [47], [48], [50]: Information provided to PFA by the diagnosis of an individual die suffers from inherent ambiguity and therefore affects the turnaround time of PFA. Statistical analysis on volume diagnosis helps to minimize the ambiguity by providing an overview of the underlying root cause to aid Failure Analysis (FA) experts identify the true root cause.
2. Independent of PFA: Besides aiding PFA, volume diagnosis can also act as an independent learning approach in identifying the root cause [48], [50], [51], [62]. PFA could then be used as an optional next step in validating the statistical analysis' results.

2.4.1.3 Challenges

In the following section, potential issues of volume diagnosis based analysis are discussed. These issues pose a challenge for yield learning.

1. Ambiguity and uncertainty inherent in diagnosis data: As mentioned in section 2.2.5, a defect could trigger multiple faults that are logic-equivalent to each other. Such ambiguity is inevitable with current diagnosis technology. This ambiguity is also the main reason [39] that naïve analysis methods such as simply adding up all the features in diagnosis results would result in misleading conclusions. Root cause uncertainty from layout-aware diagnosis brings out the issues of correlated root causes. In layout-aware diagnosis, a suspect signal line is mapped to a net in the

physical layout, which might be associated with several different design features/root causes. Presumably the design features should be considered independent of one another, and each of them can be considered an individual source of the defect. A pair or a group of correlated features would be called out together even when only one of them causes a defect. This is a challenge for the statistical analysis of volume diagnosis.

2. Limited diagnosis data: Volume diagnosis data is limited in size due to practical reason in reality. One of the reason is that not all failing die are sent for diagnosis. For each die on a wafer, several tests are applied, and only those dies passing the previous test are then sent on for Automatic Test Pattern Generation (ATPG) scan testing. Therefore, volume diagnosis data cannot represent the real overall failure distribution of whole wafer, and might lead to misleading conclusions. Volume diagnosis cannot represent the real failure distribution of wafers [44]. When sample size is limited, its distribution may not always accurately represent the original distributed data. This potential for biased data poses a challenge for statistical learning.

2.4.2 Previous Work

Volume diagnosis based analysis [37-56, 58-66] can be applied to different stages of yield ramp-up and serves different purposes, such as identification and quantification of an existing critical feature[37-42, 47, 48], identification of an unknown systematic feature [43], [51], [62], validation and calibration of DFM rules [43], [63], defect density and distribution estimation for a random defect [64], and yield monitoring [40], [61]. In this work, these approaches are all referred as diagnosis driven yield analysis (DDYA). In this section, we will focus on reviewing DDYA techniques that automatically identify common physical defect features such that yield can be improved by fixing such common physical defect features. These methods aim to obtain root cause information of defect, based on

which one can identify and quantify root cause impact on yield loss. The root cause information of defects thus obtained can be categorized as following: 1. Feature failure rate (FFR) estimation [37-46]; 2. Feature/root cause probability [47], [48]; 3. Defect distribution estimation [50]; and 4. Layout shape feature estimation [34], [42], [66], [51-60].

2.4.2.1 Feature Failure Rate Estimation

Feature failure rate estimation refers to a method of identifying and quantifying root causes by estimating feature failure rate (FFR). Feature Failure Rate (FFR) is defined as the number of defective features divided by the total number of manufactured features. It is a metric to estimate how often a feature fails for a given population, often measured in Parts Per Million (PPM) and Parts Per Billion (PPB). FFR is a frequently used metric in volume diagnosis analysis. Each feature is a possible defect root cause here. A possible defective feature is described in some literature as a “feature hit” [42], defined as a feature called out by diagnosis. For each such feature, the chance of failure depends on the instance count and the total feature hits in the population. Total feature manufactured is also referred to as occurrence of this feature [47]. Structure test fail data can also be used to calculate FFR [41].

General procedure of FFR estimation: For root cause identification, the general procedure usually starts with choosing the interesting features set or set of known factors limiting yield, then calculating FFR, performing statistical analysis by ranking or comparison between the observed result and the expected FFR, and identifying the outlier or statistically significant feature for further failure analysis. In the rest of this subsection, previous approaches for each step are reviewed.

Feature (root cause) choice: Many approaches to identification and qualification start with a pre-defined root cause list. Extracted physical layout features based on DFM rules are defined as candidate root causes in [38], [39]. Desineni *et al* [42] built an adaptive

table of possible root causes based on previous experience, and this table will be updated with new found features from feedback of FA. Different than [38], [39], [47], [51], design features such as geometric properties, the antenna characteristics of interconnect nets, and power level are all included in the adaptive table. In a case study, antenna technology ground rules, which are designed to avoid a charging-related defect that was not visible to inline wafer inspection, are refined.

FFR calculation (how to calculate FFR): FFR is traditionally estimated by using test structures, which give fast and accurate feedback but cannot handle unknown critical features. For volume diagnosis based analysis, FFR can be calculated using information extracted from layout-aware volume diagnosis, by which all suspect features can be estimated even if it is not validated by PFA beforehand. This form of analysis also can expand the set of candidate critical features and could possibly identify unknown systematic critical features. Malik *et al* [37], after comparing the FFR derived from volume diagnosis and test structures, concludes that volume diagnosis could be an alternative source for FFR estimation. It is also proposed that FFR and count of features together can quantify yield loss, which allows quantified prioritization of critical features on the design. The FFR estimation is based on scan diagnostics data, which was shown to be empirically comparable to those estimated from test structures. However, ambiguity issues from logic level diagnosis, as well as the uncertainty from layout-aware diagnosis, will need to be addressed in FFR calculation.

Other than direct calculation based on feature count, failure rate can be calculated based on some conceptual metric that reflects impacts on yield. Recent studies [38], [39] chose diagnosed net count (the number of a specific net being called out in diagnosis) as the metric for the impact from a design feature (open metal and via on layer), which is basically the same concept as FFR. The diagnosed net counts are calculated per net group, a procedure which is intended to reduce ambiguity. In [41], failure signatures, which are derived based on different assumptions of possible systematic failure mechanisms, are used

as the metric. In one example, the failure rate of the diagnosed cell type is used as the failure signature. In [43], DFM hits are used for FFR calculation. Work in [44] used probability results from RCD to calculate failure rate.

Analysis of FFR for yield learning (Ranking and Comparing with Expectation):

FFR is often used to identify root causes by quantifying the impact to yield loss from each feature and then ranking those features [37], [40], [41], [42]. Ranking the pre-extracted feature [40] is a straightforward measurement for identifying root cause, continual yield modeling, and validation and calibration of features. Ranking features by FFR also provides information on the selection of dies or identify outliers. The prediction model parameterized by FFR in [40] can point to new (previously unknown) systematic yield limiters when such cases result in a mismatch between the predicted behavior and the observed behavior. Based on the mismatch, outliers among all failing dies can be identified. Such dies are good representatives of the discrepancies between reality and what is currently known. They therefore become ideal candidates for PFA to locate unknown systematic root causes. In this situation, the observed FFR of a systematic defect is assumed to be much higher than the expectation. This outlier usually can be shown by visualization of the FFR, along with total feature count in a curve or pareto [40], [41].

With estimated FFR, another way to identify the abnormality is by comparing with observation and expectation. Generally, observed data is obtained directly from volume diagnosis results of failing ICs or the defect injection experiment [39], [47]. In [38], [39], the expected diagnosis noise is modeled as being distributed throughout each net group evenly [39] or proportionally to the number of total fan-outs of nets in that group [38]. In [41], failing chip populations are divided into two groups of approximately the same size by process parameters (time, etc.). One of the groups is considered “observation” and the other “expectation.” The failure behaviors of these two groups are compared, and it is concluded that if a systematic failure mechanism exists in the original population, then its failure signature related to that systematic failure mechanism would show a significant

difference compared to the other failure signature. This method of comparing failure behavior assumes that if there are only random defects in a population, then its subpopulations would be affected equally. Therefore, the existence of any systematic failure mechanisms could be identified by their failure signatures. One possible weakness in the assumption is that a layout-related systematic failure mechanism is distributed randomly all over a die, and therefore the failure signature of that mechanisms in multiple subpopulations could be similar to each other.

In [42], an adaptive table of critical features is built based on previous experience. These candidate features are then compared with features called out by diagnosis. If a feature is proven to negatively affect the yield, the FFR is calculated for such a feature to quantify its yield impact. If a feature in the diagnosis data does not appear in the adaptive table, PFA is needed for validation, and the table will then be updated with the new-found feature.

Statistical analysis in FFR estimation: Different statistical analysis techniques are practiced in the FFR estimation.

In [37], FFR is calculated with feature count information based on yield model assuming Poisson distribution of defects. In [38], [39], expectation of how the failing chips will behave with existence of one root cause is generated by null hypothesis. Comparing with actual “observed” data, a one-sided chi-squared test is applied to disprove each one of the candidate root causes. When a sole root cause cannot be disproved, it is considered as the dominant root cause. In [43], Pearson’s cumulative chi-squared test statistic is used to calculate the error between the “expected” populations (whole wafer) and “observed” populations (different wafer zones). This error is an indicator of how close estimated FFR is to true FFR. In [40], a probability model is built based on the assumption that there is no correlation between any two features and that each feature causes only one defect. Initially, the failure rate of each feature is assumed to be equal. An iterative method is used to

estimate fail rates by finding a convergence between expected and calculated number of failure for each type.

Assumption for ambiguity: To handle diagnosis ambiguity, several assumptions are set for the above methods.

In [37], [42], FFR is estimated based on the yield model assumption that if a single feature is called out by diagnosis, this feature must be the root cause of the defect and this defect must cause the die failure. Due to diagnosis ambiguity, a feature could be called out even if it does not cause the failure. Thus, this estimation of yield loss per critical feature based on this assumption is actually an upper bound. Work presented in [38], [39] are based on the assumption that there is only one dominant root cause. This assumption is not applicable for all cases, because in real life, defects could be caused by multiple root causes. Also, the impact from the sole root cause cannot be properly quantified. Another assumption in [38], [39] is that features with low instance counts will not be considered as potential root cause. The reason for that is these rarely occurring features might lead to an unreliable result in statistical analysis [39]. Excluding features with low instance counts could be considered as a filtering step for an outlier in the data set.

Correlated features are considered as outcome of diagnosis ambiguity. Sharma *et al* [38], [39] uses a net group to amplify the difference of correlated features, and [40] proposes the possible solution of combining two correlated feature into one.

Challenge and Limitation: The works discussed above have shown a certain degree of success which is validated by their experiment results. However, diagnosis ambiguity and root cause uncertainty of DDYA are all handled in a way that relies on certain assumptions which place limitations on possible diagnosis behavior and defect scenarios. With such limitations these methods could be insufficient to handle real scenarios in silicon data at advanced technology nodes. Other than heuristics applied under different assumptions, a more systematic approach is needed.

2.4.2.2 Feature/Root Cause Probability Estimation

As mentioned above, feature failure rate defined by instance occurrence and total count suffers from diagnosis ambiguity and root cause uncertainty. The resulting estimates provide an upper bound of the failure rate, which could mislead the identification and quantification. Feature failure probability [47], together with an iterative algorithm, is proposed to address the ambiguity problem. Estimation of feature failure probability is proposed to provide more information on which features are more likely to fail. This measurement straightforwardly filters out unreal diagnosis callout.

In [47], feature failure probability is estimated based on volume diagnosis, with a general assumption that different features fail independently. For each fail die, two events are defined: A: One feature instance is the only defect of the die. B: At least one feature instance causing the defective die. By enumerating all possibilities, conditional probability of one feature being the real defect root cause given observed diagnosis reports is computed and failure probability of each feature is computed based on the conditional probability. This approach inherently avoids the presumption of FFR that each feature instance is treated the same, instead giving a probability interpretation for each instance. Based on the feature failure probability, further analysis such as identifying systematic root cause and ranking suspects can be applied.

Following [47], RCD [48] is proposed to also find the probability distribution of feature failure/root cause. It is under a constraint that the sum of all features' probabilities needs to be 1. This constraint comes from the understanding that each feature/root cause contributes to a failing population by certain probability and triggering the population all together. Rather than using relative frequency to calculate conditional probability [47], this approach proposed a Bayesian belief network for volume diagnosis. Based on this Bayesian net, each feature, and each instance of the same feature, are modeled individually based on diagnosis reports and feature characteristics extracted from the design. We do not need make assumptions about the failing population, such as the existence of only one

dominating root cause existing. Most importantly, diagnosis ambiguity and root cause uncertainty are handled using a probability perspective, so statistically related techniques can be applied to this framework for the purpose of future improvement. Our work in this thesis is built upon RCD, and details of the method will be discussed in Chapter 3.

2.4.2.3 Defect Probability Estimation

Similarly, rather than finding feature probability distribution, some researchers assume that the probability distribution of defect types is also useful. This distribution essentially describes the possibility of each defect type occurring in the whole population given information about the defect behavior for each defect type. In [50], three types of defects are discussed: open, bridge and cell. The defect behavior of different defect types is modeled by a subset of signal lines that activate the possible defect sites. Since one defect behavior could possibly result in more than one defect type, which leads to undesired ambiguity, an iterative algorithm is also applied to reduce such ambiguity. Prior knowledge of defect behavior is needed for the expectation step and needs to be collected from the volume diagnosis results. One drawback of this approach is that the probability of a defect behavior given a certain defect type is obtained from the PFA finding. The accuracy of results relies on the availability of defect behavior distribution from PFA, which could be limited when PFA information is limited. Also, defect type is logic level description of behavior of real defect. To order to fix the cause of yield loss, we might still need to further investigate which root cause is the yield limiting factor, even given a defect probability distribution.

2.4.2.4 Layout Shape Feature Identification

Layout shape related features [51-60, 34, 42, and 43] have been an emerging source for yield loss as the feature size of design gets close to the lithographic wavelength. Investigating such features could also help validation and updating of DFM rules. Depending on design and process, such layout shape features could be unknown for PFA.

Layout shape feature identification usually need solve two issues: 1. How to extract feature of layout shape; 2. How to reduce diagnosis ambiguity to identify the “defective” feature. Several works [34], [57], [58], [59], [60] have been conducted to extract the layout feature all-over design and apply different analysis to identify layout shape features.

A clustering based method [51], [52], [53], [54] is proposed to automatically identify prone-to-fail layout features. Firstly, layout features are extracted from volume diagnosis with layout information. A layout snippet is defined as an area containing the features of a suspect net and the nets in its vicinity. Those features of a certain layout shape are represented as a skeleton of the layout snippet image. Then clustering of layout snippet is performed in two stage on an extracted snippet images or their representatives. It is expected that each cluster will have common features that are prone to fail. Those extracted features can be validated by lithographic simulation or PFA.

There are several issues with this approach: First, this method could help for SEM review, but still requires a failure analysis expert’s judgment to pick a snippet from among all others in a cluster. Also, this work does not address the ambiguity of diagnosis. All the layout shapes called out by diagnosis will be included in the clustering process. Furthermore, this method cannot quantify the impact of each feature. Lastly, though it is designed to identify unknown layout features, the extracted layout features are still limited by how the layout information is extracted. The feature extracted for clustering is based on the center line of the layout shape, so it presumably assumes that the width of the net will not be a critical feature, an assumption which might not always be true. Also, for two identical shape with some shifting, this method considered them as having different features, which is not correct. In Chapter 5, an enhanced RCD flow with automatic layout pattern extraction is proposed to address both extraction issues and ambiguity issues.

2.4.3 Summary

Volume diagnosis based statistical analysis methods have increasingly contributed to yield learning. All the previous work discussed above have had a certain degree of success in finding ways to identify error causes. However, the inherent ambiguity and uncertainty issues have not been properly addressed in a systematic fashion in previous studies.

In this thesis, our proposed work is based on a DDYA method called Root Cause Deconvolution (RCD), which reduces ambiguity and uncertainty by learning a root cause probability distribution. RCD is a statistical learning method on volume diagnosis data from scan diagnosis reports, and is proposed to describe the dependence between a defect root cause and a real defect using Bayes probability model. In Chapter 3, the RCD model [48], [77] will be explained in detail. An enhanced RCD flow [78] for layout pattern identification is also presented in Chapter 5; the proposed work is compared to existing work [51], [53], [54], and its advantages are discussed. Lastly, in Chapter 6, improvement on RCD for better model parameter is presented.

CHAPTER III

STATISTICAL MODEL OF RCD

In this chapter, the DDYA method we investigate, Root Cause Deconvolution (RCD) [48], [77], is described and explained. The work discussed in the following chapters of this thesis is built around the RCD model described below.

3.1 Background and Objective

RCD [48] is a statistical method used to determine the probability distribution of failure root causes by mining volume diagnosis data.

Diagnosis ambiguity and root cause uncertainty: Volume diagnosis is a set of diagnosis reports, each of which contains the analyzed information based on failure behavior of a failing die observed on the tester. Based on defective behavior observed, there could be more than one logic fault reported for causing observed failure behavior, more than one physical defect triggering the logic fault, and more than one root cause associated with that physical defect. Therefore, for one observed defect behavior, multiple root causes may be reported as responsible.

Probability distribution of root cause: A *root cause distribution* is a probability distribution of all possible root causes responsible for generating the population of failing dies. Each root cause has a certain probability of being the real root cause, given the diagnosis report. Likewise, a group of diagnosis reports can be produced due to a number of root causes, with an associated probability for each root cause. In our work, a root cause distribution refers to a vector of probabilities, where the n^{th} entry is equal to the probability of the n^{th} root cause. The sum of all entries in this vector is one. The probability of one root cause in a population indicates how many dies in the population failed due to defects caused by this root cause. Root causes with higher probability are responsible for generating more diagnosis reports. Root cause distribution of whole pollution throughout a whole

population shows which candidate root causes may fail the die and how many failing dies each root cause is responsible for.

Root cause distribution can be used to identify the root causes of failing dies and quantify the impact of each root cause on the yield loss. Furthermore, it can also help FPA to pick dies and narrow the scope of possible defect sites. In the following sections, we will first describe the problem RCD solves in general terms, and then present the procedures for deriving an optimal probability distribution of root causes given the information from volume diagnosis in RCD.

3.2 Motivation and Problem Description

Ideally, if a diagnosis result can pinpoint only one potential defect location and only one root cause of a defective die, the probability that it is the real root cause and real defect location given the observed failure behavior is 100%. In this scenario, the root cause distribution of a large number of manufactured dies is unique and can be obtained by simply summing up the occurrences of each root cause reported by the diagnosis and normalizing that by the number of failing dies. If each root cause can be considered a source of the observed defects, this distribution obtained by normalized sum is the only one that can generate the observed diagnosis results.

In reality, each diagnosis report is usually associated with more than one root cause. So possibly more than one possible root cause distributions can trigger the given diagnosis results. One extreme example is that if each of the failing dies in population is reported to have two possible root causes (root cause A and root cause B), possible root cause distributions generating this population and each failing die could be 100% root cause A, 100% root cause B, or any probability distributions including both root cause A and root cause B. The problem then is how to identify the underlying root cause distribution among all possibilities for the whole population. On the other hand, without further information, we cannot tell whether a failing die is caused by root cause A or root cause B by only

looking at the die itself. It is expected that information about every other individual failing die obtained from rest of the population would help to distinguish root cause A and B. Root cause distribution is expected to extract such diagnosis information from each failing die and provide guidance to identify the contribution from each root cause to either the whole population or an individual die.

The problem of root cause probability distribution can be categorized as an existing problem: Parameter estimation of Mixture model [67]. A typical finite-dimensional mixture model is a hierarchical model composed of several individual components of distribution. Each component is associated with a component weight, which is a probability. All component weights sum to 1. Observed data points are distributed according to the mixture of all component distributions. For each observed data point, in mixture distribution, we want to define assignment of data points to specific components of the mixture model [68]. The assignment here refers to component weight or mixture coefficients, which indicate how much each component is responsible for observed random variables. In [68], mixture of Gaussian distributions is used as an example to explain estimation of the mixture coefficients. Starting with mixture of Gaussian as expressed below,

$$p(x) = \sum_{k=1}^k \pi_k \varphi(x|\mu_k, \Sigma_k)$$

π_k is the mixture coefficient indicating responsibility from one of the component to explain observation points. φ is a Gaussian distribution component. As illustrated in Figure 3-1 below, random data points x are generated by a mixture model of Gaussian distribution in Figure 3-1(a). Different colors denote different Gaussian distribution components. Therefore, for each data point, there would be an underlying label (Figure 3-1(a)) indicating the component (Gaussian distribution) generating this data point. After randomly sampling from marginal distribution $P(x)$, we can only observe the set of pink

points in Figure 3-1(b), with no information of which component the data are from. Our goal is to make the estimation on such a label on each point (Figure 3-1(c)) to be as close as possible to its real label (Figure 3-1(a)). However, as the example shows, points generated by two different Gaussian models could be located next to or even overlapping each other, and it is very difficult to label them correctly if we assign the point a hard assignment to one distribution. In such a situation, soft assignment would be a better choice. Other than assigning it to one component, we assign the contribution from all components to this point by probability value. So, the point in between two clusters can be assigned as contributed; for example, by each distribution of 50%, as shown as the purple points in Figure 3-1(c). One goal of mixture model parameter estimation is to find the soft assignment π_k of each component in the mixture model to these points; in other words, the responsibility each component holds for generating these points.

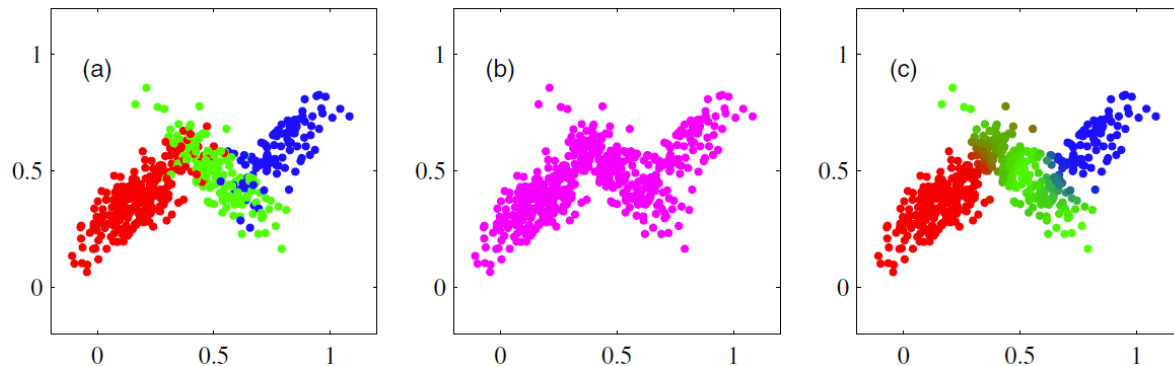


Figure 3-1 Example of 500 points drawn from the mixture of 3 Gaussians [68]

In our root cause probability estimation problem, we observed a set of diagnosis reports and wanted to know how much each root cause contributed to generating those reports. The contribution π_k is probability value and sum to 1 over all candidate root causes. It is the underlying root cause distribution that we want to estimate.

Each diagnosis report is an observed data point. Diagnosis reports triggered by each root cause follow an underlying distribution. We call this *diagnosis reports distribution* of the given root cause. All the possible diagnosis reports triggered by one root cause should follow diagnosis reports distribution of that root cause. The parameters or underlying variables of diagnosis reports distribution could include, but are not limited to, factors such as layout location of its instance, logic effect of its failure, design dependent test patterns, root cause characteristic and defect behavior.

Similar to mixture model of Gaussian distributions, which is a superposition of individual Gaussian distribution. Diagnosis reports distribution of each root cause is a component of the mixture model of diagnosis reports distributions. Combining diagnosis reports distribution of all candidate root causes, we establish the mixture model of diagnosis reports distributions as a superposition of diagnosis reports distribution for each candidate root cause, with a mixture coefficient of π_k accordingly. All observed data points follow the mixture model of diagnosis report distribution. Similar to the example of mixture of Gaussian, though each point has an underlying root cause label, other than providing a hard assignment for points from one component of the mixture model, we do a soft assignment on each diagnosis report in which the responsibility of each component is estimated. Such assignment should be based on an optimal estimation on parameters of mixture model of diagnosis report distribution. The parameters could include π_k and a set of parameters for diagnosis report distribution.

After estimation of the root cause distribution π_k for the overall diagnosis report population, we can estimate how many of the failing dies fail due to each root cause. For diagnosis report of each failing die, we can estimate the probability of root cause given that die, indicating for each of the root causes how much chance there is that it is responsible for the defective die. To summarize, our goal is to find the root cause “responsibility” to the observed diagnosis reports. Such “responsibility” is based on the estimated root cause distribution π_k of an entire population.

To achieve above goal, first of all, we need to construct mixture of diagnosis report distribution, which is a probability model needed to describe the dependent relationship between all root causes and the observed diagnosis reports. There are established probability distributions such as Gaussian distribution that are often used to model random data. In RCD, *diagnosis report distribution* is modeled based on domain knowledge of the diagnosis process on failing dies by a Bayesian net [69]. In the following discussion, the mixture model of diagnosis report distributions is termed *volume diagnosis model*, *Bayesian model*, or *probability model of RCD*. Based on the mixture model of diagnosis report distributions, there are more than one possible root cause distributions that might generate the observed diagnosis results. Since the number of such possible root cause distributions can be extremely large, an exhaustive search is not practical. So, the next step is to find an optimization technique to identify the optimal root cause probability distribution. In the next section, we will explain in detail the stages of RCD on modeling diagnosis data and finding the optimal solution.

3.3 Stages of RCD

RCD includes two Stages: 1. Construct a probability model of volume diagnosis data and calculate likelihood of observed data 2. Apply unsupervised learning on volume diagnosis to obtain the most likely root cause distribution.

Volume diagnosis Model: To capture the causal relationship between unknown variables of a root cause and an observations report, a probabilistic model is constructed to model the situation in which a die fails due to a specific root cause. This modeling process will involve a thorough domain knowledge of the diagnosis process, layout feature extraction and defect behavior, from how a design feature causes a defect to the observed volume diagnosis data. The causal relationship is modeled by a Bayesian belief network. The mixture of diagnosis report distribution is denoted as $P(report)$.

$$\begin{aligned}
P(\text{report}) &= \sum_{\substack{\text{All candidate} \\ \text{root cause}}} \pi_C * P(\text{report} | \text{domain knowledge of root cause } C) \\
&= \sum_{\substack{\text{All candidate} \\ \text{root cause}}} P(\text{root cause } C) \\
&\quad * P(\text{report} | \text{domain knowledge of root cause } C)
\end{aligned}$$

π_C is the mixture coefficients needed to be estimated and is essentially the root cause probability termed as $P(\text{root cause } C)$. This parameter is referred as π_k above in mixture of Gaussian example. For each root cause, a diagnosis report distribution $P(\text{report} | \text{domain knowledge of root cause } C)$, of which the parameters are estimated based on domain knowledge, is modeled as a component of the mixture model. Based on this mixture model, the likelihood of the observed diagnosis reports given a certain root cause probability distribution can be calculated.

Search for an optimal root cause distribution by unsupervised learning: The Expectation Maximization (EM) algorithm [70] is applied to find the most likely root cause distribution given a population of failing dies. Our approach to finding the underlying root cause distribution is essentially unsupervised learning to find the probability distribution with maximal likelihood.

3.4 Details of RCD Learning

In this section, the details of the statistical model of RCD and the unsupervised learning process are described. We will first explain terminology used in this model and diagnosis, then present the Bayes net of volume diagnosis, and lastly describe steps in the EM algorithm for unsupervised learning.

3.4.1 Typical Diagnosis Report

A typical diagnosis report: Terminology used in diagnosis reports in RCD model is defined. Subsequent discussion will use the following terms.

Fault: A specific logic failure model at a specific logic location in a netlist. Different faults can be either related to different logic failure model or they are at different logic locations. It is also referred as logic fault or logic suspect in diagnosis report.

Defect: A specific physical defect effect at a specific physical location in a layout. Different defects can be either related to different physical effects or they are at different physical locations. It is also referred as physical suspect or suspect in diagnosis report.

Physical feature: Properties of physical defects. They are the root causes why a defect happens. In the following work, the terms root cause and physical feature are used interchangeably. A physical feature is also referred as a feature or design feature.

A schema represents a typical diagnosis report with layout information, shown in Figure 3-2. Without loss of generality, all failing devices are assumed to be caused by one root cause only. If a failing die has multiple defects, then we assume they are independent, and we can break the diagnosis report into independent diagnosis reports.

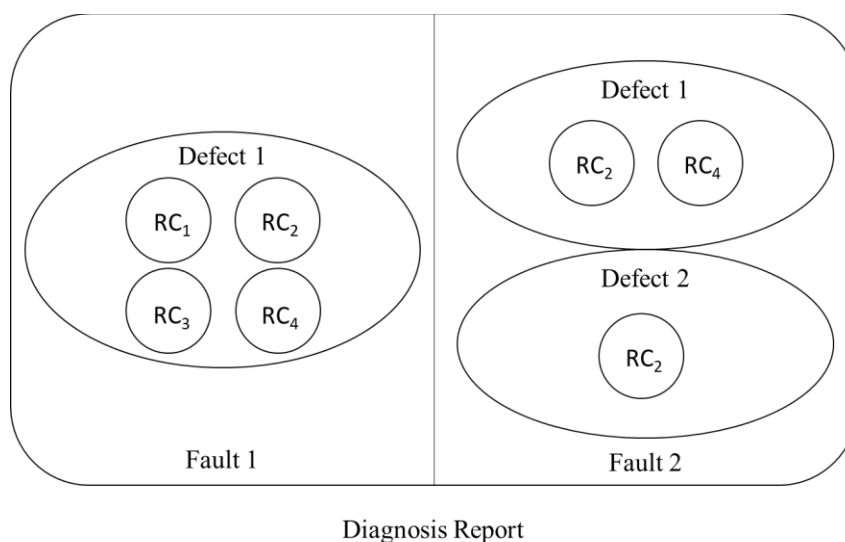


Figure 3-2 Schematics of a typical diagnosis report

Typically, in one diagnosis report, there can be more than one logic fault. Figure 3-2 shows the schematic representation of a diagnosis report with two logic faults. Each logic fault can cause the failures observed at the tester. For each logic fault, all physical defects which can cause the logic fault will be reported. For each defect, a score is calculated based on the failing bits. In Figure 3-2, the first fault has only one physical defect, while the second fault has two physical defects. Further, for each physical defect, the diagnosis report contains a list of physical features/root causes, each of which can cause the physical defect. In Figure 3-2, the physical defects have 4, 2, and 1 root causes, respectively.

3.4.2 Bayes Net of Volume Diagnosis

In this Bayes net, the conditional probability among defect, fault and physical features (root causes) are constructed. As mentioned above, in each diagnosis report, there can be several faults, each of which matches the failures observed at testers. There are several defects, each of which can cause one specific fault. There are several physical features, each of which can be responsible for triggering one specific defect. It is possible for one physical feature to trigger two different defects at two different physical locations. In the following, P stands for probability. The following terms are used in the Bayes net of volume diagnosis.

$P(v)$: the probability of sampled volume diagnosis reports.

$P(r)$: the probability of one diagnosis report.

$P(f)$: the probability of one fault.

$P(d)$: the probability of one defect.

$P(c)$: the probability of one physical feature (root cause).

$P(v)$ is calculated based on the assumption that all diagnosis reports are independent. r_n is the n^{th} report.

$$P(v) = \prod_{\text{all report } r_n} P(r_n)$$

$P(r)$: If all faults are mutually exclusive and $P(f)$ sum to 1 over all faults in one report, $P(r)$ can be calculated as indicated below. $P(r|f)$ is a conditional probability of report r if a specific fault f is true. f_i is the i^{th} fault in the n^{th} report.

$$P(r_n) = \sum_{\substack{\text{all fault } f_i \\ \text{in} \\ \text{report } r_n}} P(r_n | f_i) * P(f_i)$$

$P(f)$: If all defects are mutually exclusive and $P(d)$ sum to 1 over all defects that can cause this fault, $P(f)$ can be calculated as below. $P(f|d)$ is a conditional probability of fault f if a specific defect d is true. d_j is the j^{th} defect for i^{th} fault.

$$P(f_i) = \sum_{\substack{\text{all defect } d_j \\ \text{for} \\ \text{fault } f_i}} P(f_i | d_j) * P(d_j)$$

$P(d)$: If all root causes are mutually exclusive and $P(c)$ sum to 1 over all candidate root cause, $P(d)$ can be calculated as below. $P(d|c)$ is a conditional probability of defect d if a specific root cause c is true. c_k is the k^{th} candidate root cause.

$$P(d_j) = \sum_{\substack{\text{All} \\ \text{candidate} \\ \text{root cause } c_k}} P(d_j | c_k) * P(c_k)$$

Combining all these equations, we get update $P(v)$, as below:

$$P(v) = \prod_{\substack{\text{all} \\ \text{reports } r_n}} \sum_{\substack{\text{all faults } f_i \\ \text{in} \\ \text{a report } r_n}} P(r_n | f_i) * \left(\sum_{\substack{\text{all defect } d_j \\ \text{for} \\ \text{a fault } f_i}} P(f_i | d_j) * \left(\sum_{\substack{\text{All} \\ \text{candidate} \\ \text{root cause } c_k}} P(d_j | c_k) * P(c_k) \right) \right)$$

$P(v)$ can be also expressed as following:

$$\begin{aligned}
P(v) &= \prod_{\substack{\text{all} \\ \text{reports } r_n}} \sum_{\substack{\text{All} \\ \text{candidate} \\ \text{root cause } c_k}} P(r_n, c_k) \\
&= \prod_{\substack{\text{all} \\ \text{reports } r_n}} \sum_{\substack{\text{All} \\ \text{candidate} \\ \text{root cause } c_k}} P(r_n|c_k) * P(c_k)
\end{aligned}$$

The first equality holds because the root causes (c) are assumed to be mutually exclusive, and their probabilities ($P(c)$) add up to one [74]. The second equality follows from the chain rule of probability [75]. $P(r/c)$ is a conditional probability of report r if a specific root cause c is true.

Conditional probability $P(r_n|f_i)$, $P(f_i|d_j)$ and $P(d_j|c_k)$ are three model parameters estimated based on domain knowledge. They are fixed and known values in the model for all root causes c_k , defects d_j , faults f_i and reports r_n . In EM unsupervised learning, $P(c_k)$ is the unknown variable to learn.

3.4.3 EM Procedure

The EM algorithm is an effective unsupervised learning method to find the underlying probability distribution of complete data by learning from incomplete data [71]. The EM algorithm iteratively applies an expectation step (E-step) and a maximization step (M-step) till the value of the likelihood value reaches an optimal point. $P(v)$ is the likelihood function. The log likelihood function $Ln(v)$ is expressed as below:

$$Ln(v) = \log(P(v)) = \sum_{\substack{\text{all} \\ \text{reports } r_n}} \log \left(\sum_{\substack{\text{All} \\ \text{candidate} \\ \text{root cause } c_k}} P(r_n|c_k) * P(c_k) \right)$$

We use maximum likelihood estimator (MLE) [72] to maximize $P(v)$ with respect to $P(c)$ of all root causes in the following steps:

1. **Parameter initialization:** The only unknown parameter in RCD model is $P(c_k)$. $P(c_k)$ is initialized with random value, and we call $P(c_k)'$ the current value of estimated $P(c_k)$. Log likelihood value is evaluated using $\mathbf{P}(c_k)'$.
2. **E step:** Evaluate expected value of $\mathbf{P}(c_k|r_n)'$ using current $P(c_k)'$.

$$\mathbf{P}(c_k|r_n)' = \frac{P(c_k)' * P(r_n|c_k)}{\sum_{all\ root\ cause\ c_k} (c_k)' * P(r_n|c_k)}$$

$\mathbf{P}(c_k|r_n)'$ is the updated posterior probability given an observed report r_n . It indicates how much responsibility a root cause c_k takes for generating report r_n .

3. **M step:** Update current $\mathbf{P}(c_k)'$ value by maximizing $Ln(v)$. $\mathbf{P}(c_k)'$ is obtained by setting the derivatives of $Ln(v)$ with respect to $P(c_k)'$ and a constraint that $\sum_{all\ root\ cause\ c_k} P(c_k) = 1$.

$$\mathbf{P}(c_k)' = \frac{\sum_{reports\ r_n} P(c_k|r_n)'}{N}$$

N is the number of all observed reports. $\mathbf{P}(c_k)'$ is essentially the normalized responsibility that root cause c_k takes for generating all N reports.

After initialization, the E-step and M-step are performed consecutively during each iteration. The value of the likelihood function $Ln(v)$ is evaluated in each interaction and compared with its previous value from previous iterations. The EM iterations continue until $P(c_k)'$ converges to a certain point where $Ln(v)$ stops increasing, and thus likelihood function is maximized. This maximization is guaranteed because changes to $P(c_k)'$ will always increase the value of likelihood [70].

3.5 RCD Flow and Bayesian Model Parameter

Figure 3-3 shows the general stages of the RCD procedure. The input information of RCD learning has two main parts: Layout-aware diagnosis of defective dies and information of candidate root causes of the whole design.

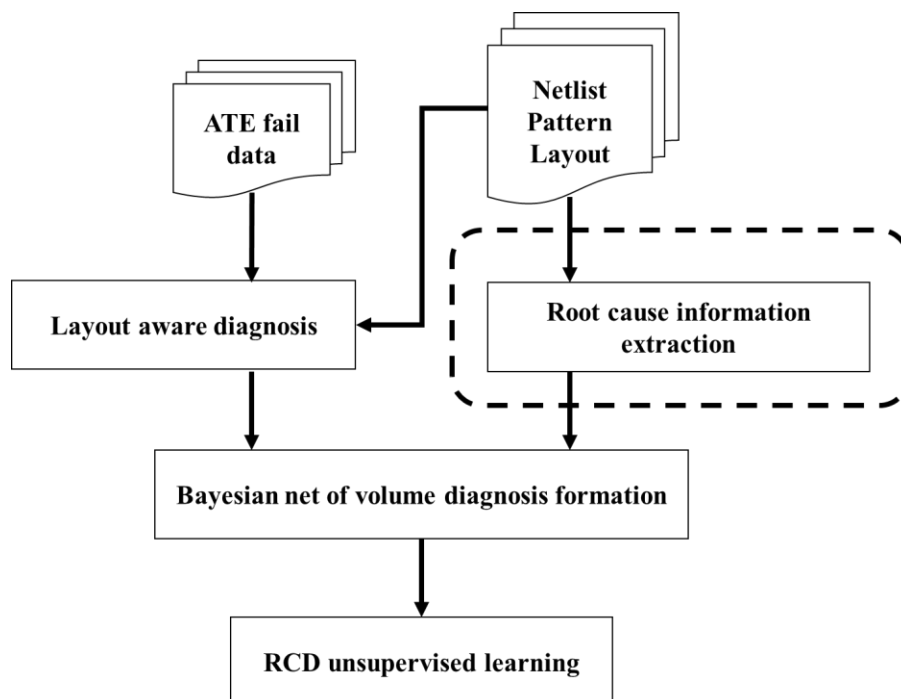


Figure 3-3 Flow Chart of RCD

Layout-aware diagnosis: With failing files of defective dies, layout file, netlist of design and test patterns generated by ATPG, layout-aware diagnosis is performed, and a set of diagnosis reports are generated with information about root causes, defects and faults.

Root cause information from design: This information will be used to compute the parameters of the Bayesian net, as shown in the block inside the dashed line.

Root cause instance: A layout part, of which some extracted characteristics are defined as a root cause. Such an instance is associated with a physical defect and is called out by diagnosis along with that defect. For the same defect location, different root cause instances could be associated with it. The properties of the same physical defect associated with different root causes are different.

Root cause instance weight/feature weight: Indicates the possibility of an instance of this feature being defective. This number is estimated based on how the defect caused by this root cause is modeled. For example, for a random spot defect model based on critical

area, an instance with a larger critical area will more likely be defective, so the feature weight of an instance of a random spot defect is the size of its critical area. For root causes such as layout physical structure, each instance will have the same chance of being defective, so the weight of each instance is the same.

Defect distribution/feature weight distribution: For each candidate root cause, root cause instance associated with every candidate physical defect is extracted from the layout, along with the weight of each root cause instance. This distribution shows how often a defect would occur if a certain root cause presents. Each candidate defect is a diagnosable defect which can be called out by a diagnosis tool. Which root causes are associated with a given defect depends on the definition of root causes. The above two factors, which are limited by diagnosis techniques and prior knowledge of failure mechanism, will affect accuracy of estimated parameter of Bayesian model. Total feature weight of one root cause is the sum of the feature weights of all diagnosable instances of that root cause. Total feature weight is also referred to as RCD constants.

Bayesian model parameter estimation [48]:

$P(r_n|f_i)$ Estimation: Assumed to be 1.

$P(f_i|d_j)$ Estimation: Due to logical equivalence, the same failure behavior we obtain from test pattern failing bits could be explained as different individual defects. Such an explanation might vary by setting of diagnosis, and can include the score of each defect, root causes for defect, number of defects, etc. For one report, each possible defect site is a defect, and is scored by how well the observed behavior of the defective die matches the behavior explained by the fault model we applied in the diagnosis. For now, this probability is calculated based on diagnosis score.

$P(d_j|c_k)$ Estimation: If a reported physical defect occurred because of a given root cause, there would then be at least one instance of that root cause associated with that suspect. The probability of such an instance is calculated by taking the value of the feature weight of that root cause instance divided by the total feature weight of a given root cause.

3.6 Possible Challenge for RCD

Previous work on RCD [44], [48], [49] has proven its effectiveness. However, as moving to new technology node, the failure mechanisms of defects have become more complicated and bring new challenges to RCD. The following are potential issues might affect RCD.

Limited Sample Data: In unsupervised learning of RCD, we used MLE to obtain the optimal root cause distribution. It has been proven [73] that MLE are consistency, normality and efficiency when the sample size approaches infinity. Researchers should also be aware that the behavior of the ML estimator working with a small sample size is largely unknown. However, in reality we usually do not have enough data to completely fulfill the conditions of optimal ML estimation. In volume diagnosis, it is not practical to obtain an infinite number of data from real silicon data, as discussed in Chapter 2.

Bayesian Model parameter: The estimation of the Bayesian parameter is based on domain knowledge of diagnosis techniques and failure mechanisms. It is also built open certain assumptions we make that might not exactly reflect real scenarios of defect. As new design features appear, such domain knowledge might also become insufficient to provide a precise estimation of model parameters.

CHAPTER IV

CARD GAME MODEL FOR DDYA INVESTIGATION

In this chapter, a card game model is presented to create controllable diagnosis data which can be used to evaluate the effectiveness of volume diagnosis data mining techniques. The effectiveness of RCD statistical models given limited sampling data is discussed using the card game model [77].

4.1 Introduction and Motivation

Scan diagnosis, also called logic diagnosis [8-13], is used to determine the defect locations and defect mechanisms for a given failing device and the scan test patterns used. Scan diagnosis results have been successfully used to guide physical failure analysis (PFA) to focus on a small area, and thus improve PFA success rate with reduced turnaround time and cost.

Recent advancements in scan diagnosis technologies include use of more physical information, such as layout-aware diagnosis [14-16], cell-internal diagnosis [17-20], and cell-aware diagnosis [21-22]. This extra information not only improves diagnosis resolution to smaller and smaller defect locations, but also more precisely identifies physical features associated with each defect. These physical defect features include, but are not limited to, defect type (open, short), defect layer, via macro type, cell type, critical area, defect shape, specific layout pattern and specific DFM rule.

With physical defect features reported by diagnosis tools, numerous papers [38], [40], [43], [47], [48], [49], [51], [53], [66], have proposed to use volume (large amount of) diagnosis reports with appropriate statistical analysis to automatically identify a common physical defect feature. Fixing such a common physical defect feature results in improved yield. Every DDYA work among [38], [40], [43], [47], [48], [49], [51], [53], [66] has had a certain degree of success, as evidenced by their experimental results. In this work, we will focus on investigating RCD, one type of DDYA technique, as discussed in Chapter 3.

Typically, in one diagnosis report, several logic faults may be identified, each of which can cause the failures observed at testers. Also reported are several physical defects, each of which can cause one fault. Also reported are several physical features, each of which is responsible for causing one defect. In other words, a diagnosis report can be caused by several possible physical features, with various probabilities of each. A group of diagnosis reports can be caused by their combined possible physical features, with various probabilities of each. The sum of these probabilities should be 1. Probability distribution or distribution refers to the probabilities of all physical features in this paper. Physical features with higher probabilities are responsible for more diagnosis reports. The goal of RCD is to identify these high probability physical features within a volume of diagnosis reports. In general, the number of high priority physical features is very small. PFA should focus on the defects affected by these high probability physical features, and thus further improve diagnosis resolution and PFA success rate.

A simple example is shown in Figure 4-1. This figure illustrates a bridge fault involving two nets in one diagnosis report. These two nets are neighbors in 5 different defect locations. All defects have different physical features: layout layers as shown in 4 blue ovals and 1 red oval. There are 5 possible physical features (metal bridges at certain layers) for this bridge fault. For this diagnosis report, if the most likely physical features cause the defect at the red oval (metal bridge at certain layer) and not at the other 4 blue ovals, DDYA can recognize that the most likely defect is at the red oval. This information improves diagnosis resolution and PFA success rate.



Figure 4-1 An example of a bridge fault

Because of design complexity and limited test patterns used, diagnosis reports often identify multiple faults, multiple defects and multiple physical features; thus, simply summing up is not sufficient to obtain correct distribution. With volume diagnosis reports, RCD uses statistical methods to identify the correct distribution of these diagnosis reports. In statistical analyses, MLE [72] is a method of estimating the most likely distribution for observed data. It is known that the distribution identified by MLE is correct if the likelihood model of volume diagnosis reports is correct and the number of diagnosis reports is unlimited. So, the RCD problem has two parts: how to obtain correct volume diagnosis likelihood models, and how to ensure that the distribution identified by MLE is correct with limited diagnosis reports. As discussed in Chapter 3, the two parts of RCD have two potential issues: limited diagnosis sample data and inaccurate model parameters.

To verify the effectiveness of these DDYA techniques, previous research [38], [40], [43], [47], [48], [49], [51], [53], [66] have obtained failure files for the experiments using a handful of silicon defects and many simulated defects. However, diagnosis reports depend on design behavior and test patterns used. It is difficult to ensure that simulated defects can create diagnosis reports with sufficient diversity to verify the effectiveness of the investigated procedures.

In this chapter, a simple card game is introduced and used to create controllable diagnosis reports with the desired diversity of scenarios for detailed analysis of

effectiveness of a given DDYA. The card game model is designed to map the volume diagnosis model to card game model of which we have all precious information of data, so that the obtained card game model parameter can be accurate. This procedure allows flexibility in both data and model, making it an effective and convenient way to evaluate DDYA. In Section 4.2 below, a simplified Bayesian network of RCD used to model volume diagnosis reports will be briefly presented.

Assuming that the model is correct, to explain clearly the issues encountered with limited data, a simpler and similar data model based on a card game is introduced in Section 4.3. In Section 4.4, various data parameters are used for the card game to further illustrate the issues encountered when data are limited. The solutions to overcome these issues and to tolerate the industrial scan failure data will be discussed in Section 4.5. Section 4.6 concludes this Chapter.

4.2 Simplified Volume Diagnosis Model

As mentioned above, each diagnosis report can identify several faults, each of which matches the failures observed at testers. Several defects may exist, each of which can cause one specific fault. There are several physical features, each of which can be responsible for triggering one specific defect. It is possible for one physical feature to trigger two different defects at two different physical locations.

The Bayesian model built for volume diagnosis is described in detail in Chapter 3. Below, we will revisit the simple version of equations, mapping them to card game model equations.

The likelihood function is $P(v)=\prod(P(r))$. That is the probability of all sampled volume diagnosis reports. It is equal to the product of the probability of each diagnosis report if all diagnosis reports are independent.

$P(r)=\sum(P(r|f)*P(f))$ if all faults are mutually exclusive and $P(f)$ sum to 1. $P(r|f)$ is a conditional probability of report r if a specific fault f is true.

$P(f) = \sum(P(f|d) * P(d))$ if all defects are mutually exclusive and $P(d)$ sum to 1. $P(f|d)$ is a conditional probability of fault f if a specific defect d is true.

$P(d) = \sum(P(d|c) * P(c))$ if all root causes are mutually exclusive and $P(c)$ sum to 1. $P(d|c)$ is a conditional probability of defect d if a specific root cause c is true. Combining all these equations, we get $P(v) = \prod(\sum(P(r|f) * (\sum(P(f|d) * (\sum(P(d|c) * P(c))))))$). MLE finds the distribution of $P(c)$ of all root causes such that $P(v)$ has maximum value.

As mentioned above, MLE is accurate if the model used is correct and the sampled data are infinite. The accuracy of this Bayesian network depends on the accuracy of $P(r|f)$, $P(f|d)$ and $P(d|c)$. How to assess their values accurately will be discussed in Section 6.5. With an accurate Bayesian network, the impact of limited sample data is discussed first. To clarify limited data issues, a card game is introduced in the next section. This card game data has statistical characteristics similar to those of volume diagnosis data.

4.3 Card Game

Setting: In this card game, there are several card decks. Each card deck has several cards, with each card marked with a number. It is not necessary that all cards have unique numbers. The information of what numbers are included in each card deck is known. The sizes of different card decks can differ.

Game: Assume there is a genie who will select a card deck and randomly draw a card from that deck. Each time, the number on the drawn card is recorded before the card is returned to its deck. The genie draws N cards independently, resulting in N numbers reported. One card deck can be selected multiple times.

Problem to solve: Based on the N reported numbers, determine the probability distribution of decks selected by the genie. Additionally, determine the probability distribution of decks for each card. The Bayesian network of the card game can be established as:

- $P(t) = \prod(P(n))$

- $P(n) = \sum(P(n|d) * P(d))$

$P(t)$: probability of all reported numbers

$P(n)$: probability that a specific number n is drawn

$P(n|d)$: probability that number n is drawn if a specific deck d is picked

$P(d)$: probability that a specific deck d is picked

Combining all equations together, we get $P(total) = \prod(\sum(P(n|d) * P(d)))$.

As mentioned above, the information about numbers on all cards included in each card deck is known. For example, a card deck with size S has c cards with the number 17, $P(n|d) = c/S$ for the number 17 in this card deck. Hence, $P(n|d)$ of any specific number in any specific card deck can be easily and accurately calculated. In other words, the Bayesian network $P(t) = \prod(\sum(P(n|d) * P(d)))$ models the card game accurately.

The card game provides the flexibility to create all sorts of scenarios that can be used to investigate the limitation of MLE with limited sampled data on an accurate Bayesian network. Without loss of generality, in this card game all numbers in one card deck are unique, such that $P(n|d) = 1/S$. Basically, the Bayesian network shows that all numbers within one deck have equal probability of being selected, but numbers on different decks have different probabilities.

An example of the card game is shown in Figure 4-2. There are 3 card decks. Deck A has card numbers 1, 2, 3, 4, 5, 6. Deck B has card numbers 1, 3, 5. Deck C has card numbers 1, 2, 3. Four cards are drawn by the genie. Assume that the numbers on these 4 cards are 1, 2, 4, and 5. Using MLE, the problem is to identify the probability distribution of card deck selection by the genie such that the probability of drawing these 4 numbers has the maximum value.

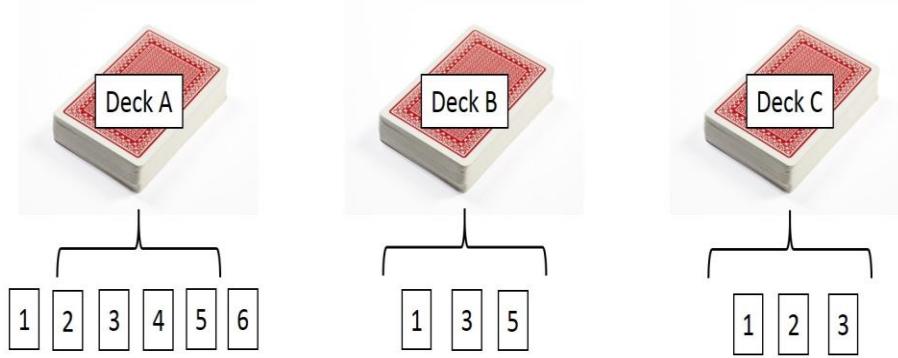


Figure 4-2 A simple example

For these 3 decks, $P(n/d)$ is $1/6$ for all numbers in deck A, since there are 6 numbers, all numbers are unique, and with the assumption that the genie picks a card from the card deck randomly with equal probability. Similarly, $P(n/d)$ is $1/3$ for (1, 2, 3) in deck B and $1/3$ for (1, 3, 5) in deck C. There are 4 probability equations for these 4 drawn cards.

$$P(1) = 1/6 * P(A) + 1/3 * P(B) + 1/3 * P(C)$$

$$P(2) = 1/6 * P(A) + 1/3 * P(C)$$

$$P(4) = 1/6 * P(A)$$

$$P(5) = 1/6 * P(A) + 1/3 * P(B)$$

Since all 4 cards are drawn independently, $P(\text{total})$ is

$$\begin{aligned} P(1) * P(2) * P(4) * P(5) \\ = (1/6 * P(A) + 1/3 * P(B) + 1/3 * P(C)) * (1/6 * P(A) + 1/3 * P(C)) * (1/6 * P(A)) * (1/6 * P(A) + 1/3 * P(B)) \end{aligned}$$

$P(\text{total})$ has 3 variables: $P(A)$, $P(B)$ and $P(C)$. Since there are only 3 possible card decks, $P(A)+P(B)+P(C)=1$. Using $P(C)=1-P(A)-P(B)$, the total probability equation can be reduced to 2 variables, $P(A)$ and $P(B)$. This 2-variables total probability equation

$$\begin{aligned} P(\text{total}) &= P(1,2,4,5) \\ &= (2 - P(A)) * (2 - P(A) - 2 * P(B)) * P(A) * (P(A) + 2 * P(B)) \\ &\quad * 1/1296 \end{aligned}$$

can be plotted as shown in Figure 4-3(a). The coordinates of $(P(A), P(B))$ are marked inside the figures. The maximal-value point is plotted as color yellow. This two-dimensional plot has the highest value at $P(A)=1, P(B)=0$ ($P(C) = 0$). $P(total)=1/1296$. The distribution with maximal likelihood is (100% deck A, 0% deck B and 0% deck C), meaning that all reported numbers (1, 2, 4, 5) are from deck A.

If only three cards (1, 2, 5) are drawn,

$$P(total) = P(1,2,5) \\ = (2 - P(A)) * (2 - P(A) - 2 * P(B)) * (P(A) + 2 * P(B)) * 1/216$$

and is plotted as shown in Figure 4-3(b). This two-dimensional plot has the highest value at $P(A)=0, P(B)=0.5$ ($P(C)=0.5$). $P(total)=1/108$. The distribution with maximal likelihood is 0% deck A, 50% deck B and 50% Deck C. This means that the number 1 has 50% probability of being from deck B and 50% probability of being from deck C. The number 2 has 100% probability of being from deck C, and the number 5 has 100% probability of being from deck B.

If there are only two reported numbers (1, 2),

$$P(total) = P(1,2) = (2 - P(A)) * (2 - P(A) - 2 * P(B)) * 1/36$$

and is plotted as shown in Figure 4-3(c). This two-dimensional plot has the highest value at $P(A)=0, P(B)=0$ ($P(C)=1$). $P(total)=1/9$.

This means that both number 1 and 2 have 100% probability of being from deck C. If there is only one reported number (1), the $P(total)$ equation becomes $P(1)=(2-P(A))*1/6$, and is plotted as shown in Figure 4-3(d). This two-dimensional plot has the highest value at $P(A)=0, P(B)+P(C)=1$. $P(total)=1/3$. The answer is not really a point, but a line segment. Any point on this line segment has the same highest value, $1/3$. This means that number 1 is from deck B or deck C, with a combined probability equaling 100%.

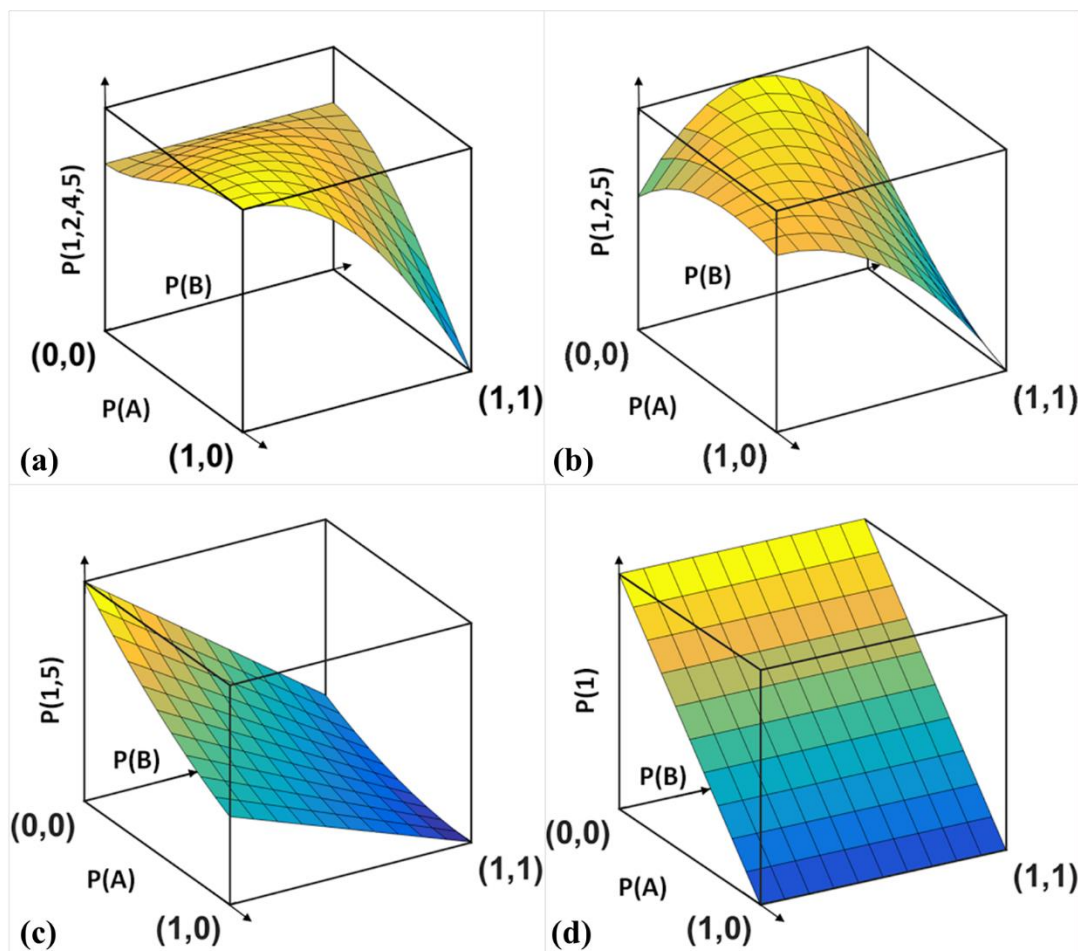


Figure 4-3 (a) 4 drawn cards; (b) 3 drawn cards; (c) 2 drawn cards; (d) 1 drawn cards

It should be noted that MLE is not able to find the correct answer unless the number of reported numbers is large enough. For example, if the genie picks the number (1) from deck A, MLE is incorrect in determining that this number (1) is from either deck B or deck C (not from deck A), as illustrated in Figure 4-3(d). MLE is not correct until more cards are picked from deck A. As shown in Figure 4-3(a), MLE is correct after numbers (1, 2, 4, 5) are picked from deck A. This issue will be discussed in detail in Section 4.5.

4.4 Card Game Experiment

The most likely deck distribution estimated by MLE is correct if the sampled data size is large enough. In practice, though, diagnosis sample size is limited even in volume diagnosis. We applied the above-mentioned Bayesian network on various card game scenarios generated by changing several parameters. These scenarios have various success rates using MLE on the simple Bayesian network of the card game. In the card game, we refer to the decks picked by the genie as picked card decks. The other decks not picked by the genie are unpicked card decks. To account for the correlation among different decks, all the numbers in the card decks are chosen from a pre-defined number pool. Below, we discuss several scenarios we examined to bring out the issues related to limited sample data. We also conducted experiments with the goal of identifying ways to improve the results using a popular machine-learning technique.

4.4.1 Single Picked Card Deck

In this experiment, we used the simplest scenario, in which all cards are drawn from one deck only. This scenario mimics one in which there is a single dominant root causes in volume diagnosis, with a single suspect called out. There are 5 different scenarios. In each scenario, 520 cases are randomly created using 4 parameters. In each case, 100 cards are drawn by the genie from one picked card deck. The results of these scenarios are shown in Table 4-1. In Table 4-1, the first column lists scenario names. The other columns show the success rate of identifying the deck from which each card originated. For example, a 90% success rate means the card decks of 90 cards out of 100 are correctly identified by MLE, and 10 cards are identified as selected from some unpicked card decks. That is an approximate based on estimated card deck distribution but not by probability of root cause in each individual card.

4.1.2.1 Scenario 1

This scenario is created with the following parameters.

- a) The number pool used is integers in the range [1-1M].
- b) The size of the single picked deck is randomly assigned a value between 1K and 5K cards in each of the 520 cases.
- c) There are 9K unpicked card decks in each case.
- d) The sizes of the 9K unpicked card decks are randomly assigned a value between 1 and 100 in each of the 520 cases.

With this scenario, in 28 cases out of 520 cases MLE achieves a 100% success rate; that is, MLE identifies the deck correctly for all 100 cards. In 176 out of 520 cases, MLE achieves a 99% success rate. This distribution allocates 99% of the cards to the picked card deck and 1% to an unpicked card deck.

In Scenario 1, there are 9K unpicked card decks. Each card deck has 50 cards on average. In other words, there are $50 \times 9K = 450K$ unpicked cards. With a number pool size of 1M, for each number drawn from the picked card deck, on average, there are a total of 0.45 unpicked card decks that could also have this drawn number. In diagnosis scenario, this means each diagnosis report identifies an average of 0.45 extra root causes besides the single real root cause. From Table 4-1, we see that MLE is not always correct even with such a small number of unpicked card decks per drawn number.

Scenario 2 is created to have even more unpicked card decks per drawn number by using a smaller number pool of [1-100K]. This mimics a situation in which, for one failure file with the applied scan patterns, a significant number of faults have identical failure behavior to that of the failure file.

4.1.2.2 Scenario 2

This scenario is modified from Scenario 1 by changing parameter a) from [1-1M] to [1-100K], resulting in more unpicked card decks per drawn number.

- a) The number pool used is integers in the range [1-100K].
- b) The size of the single picked deck is randomly chosen to be between 1K and 5K cards in each of the 520 cases.
- c) There are 9K unpicked card decks in each case.
- d) The sizes of the 9K unpicked card decks are randomly assigned a value between 1 and 100 in each of the 520 cases.

In Scenario 2, on average, for each number drawn from the picked card deck there are 4.5 unpicked card decks that have the same number. As expected, the success rate of MLE in this scenario is much worse compared to that seen in Scenario 1, as can be noted from the row for Scenario 2 in Table 4-1.

Another way to change the number of unpicked card decks per drawn number is to change the number of unpicked card decks instead of changing the size of the number pool used for the card game. Scenario 3 is created to have fewer unpicked card decks by using a smaller number of unpicked card decks.

4.1.2.3 Scenario 3

This scenario is modified from Scenario 1 by changing parameter c) from 9K to 90 unpicked card decks, resulting in fewer unpicked card decks per drawn number.

This scenario is created with the following parameters.

- a) The number pool used is integers in the range [1-1M].
- b) The size of the single picked deck is randomly chosen to be between 1K and 5K cards in each of the 520 cases.
- c) There are 90 unpicked card decks in each case.
- d) The sizes of the 90 unpicked card decks are randomly assigned a value between 1 and 100 in each of the 520 cases.

In Scenario 3, the number of unpicked card decks is reduced from 9K to 90. On average, for each number drawn there are 0.0045 unpicked card decks that have the same

number. The success rate of MLE in this scenario is better than in Scenario 1, as can be seen from the corresponding row in Table 4-1. However, even with this substantially fewer number of unpicked card decks per drawn number, we still see that MLE is not always correct.

Another way to change the number of unpicked card decks per drawn number is to change the size of all unpicked card decks. Scenario 4 is created to have more unpicked card decks per drawn number by increasing the size of all unpicked card decks 10-fold.

4.1.2.4 Scenario 4

This scenario is modified from Scenario 1 by changing parameter d) from [1-100] to [1-1K] to have more unpicked card decks per drawn number.

This scenario is created with the following parameters.

- a) The number pool used is integers in the range [1-1M].
- b) The size of the single picked deck is randomly chosen to be between 1K and 5K cards in each of the 520 cases.
- c) There are 9K unpicked card decks in each case.
- d) The sizes of the 9K unpicked card decks are randomly assigned a value between 1 and 1K in each of the 520 cases.

In this scenario, on average there are 5K cards in each unpicked card deck, which means that for each drawn card, there are 4.5 unpicked card decks that have this drawn number. This is the same situation as for Scenario 2. Surprisingly, the success rate of MLE in this scenario is actually much better than for Scenario 2 and better even than for Scenario 1 also. After further analysis, it is found that there is another factor that affects success rate. As mentioned previously, $P(n) = \sum(P(n/d) * P(d))$. It means that if $P(n/d)$ of a picked card deck is higher than the $P(n/d)$ of unpicked card decks, MLE will choose the picked card deck to get higher $P(n)$. In this scenario, $P(n/d)$ is randomly distributed from [1 to 1/1K] for all unpicked card decks, which is 10 times smaller than that in Scenarios 1 and 2. In

Table 4-1, we see that the success rate is affected more by $P(n/d)$ than by the count of unpicked card decks per drawn number.

Another way to change the relative $P(n/d)$ between the picked card deck and the unpicked card decks is to change the $P(n/d)$ of the picked card deck. Scenario 5 was created to reduce $P(n/d)$ of the picked card deck by increasing its size.

This scenario is modified from Scenario 1 by changing parameter b) from [1K-5K] to [5K-10K] to have lower $P(n/d)$ of the picked card deck.

4.1.2.5 Scenario 5

This scenario is created with the following parameters.

- a) The number pool used is integers in the range [1-1M].
- b) The size of the single picked deck is randomly chosen to be between 5K and 10K cards in each of the 520 cases.
- c) There are 9K unpicked card decks in each case.
- d) The sizes of the 9K unpicked card decks are randomly assigned a value between 1 and 100 in each of the 520 cases.

Since the $P(n/d)$ of the picked card deck decreases as its size is increased in Scenario 5, the success rate of MLE in Scenario 5 is worse than in Scenario 1, as shown in the corresponding row of Table 4-1.

4.4.2 Multiple Picked Card Deck

It is harder to identify the picked card decks of drawn cards if multiple card decks are picked by the genie. This difficulty mimics the fact that it is harder to identify multiple root causes in volume diagnosis. We consider this case next, using two picked card decks. The same 5 different scenarios used in the case of single picked card deck are repeated with two picked card decks. In each scenario, 520 cases with sample size (total number of cards in both picked card decks) of 100 are created with two picked card decks. Fifty cards are drawn from each picked card deck. The results of these 5 scenarios with two picked

card decks are shown in Table 4-2. It can be seen that the MLE success rates are now worse than those in Table 4-1 for a single picked card deck. When two card decks are picked and each one has the same sampled data (same number of cards from each card deck), this can be seen as each picked card deck having a sample size half as large as that for a single picked card deck. In Table 4-2, each picked card deck has a sample size of 50, which is smaller than the sample size of 100 used in Table 4-1. To validate this view, the experiment of Table 4-1 was repeated with sample size reduced from 100 to 50, and the results are presented in Table 4-3. The results in Table 4-3 can be seen to be consistently worse than the results of Table 4-1. This validates the idea that smaller sample size reduces the success rate of MLE. The results in Table 4-3 and Table 4-2 are about the same on average but not consistent in all scenarios. Thus, one cannot say that MLE success rate reduction due to more picked card decks is the same as the reduction due to smaller sample size.

Table 4-1 MLE success rate of single picked card deck

Single Root Cause Sample size 100	0%	0% ~ 50%	50% ~ 60%	60% ~ 70%	70% ~ 80%	80% ~ 90%	90% ~ 95%	95% ~ 99%	99% ~ 100%	100%
Scenario 1	0	0	0	0	0	0	18	298	176	28
Scenario 2	59	91	23	44	56	99	80	65	3	0
Scenario 3	0	0	0	0	0	0	0	1	35	494
Scenario 4	0	0	0	0	0	0	0	5	233	282
Scenario 5	0	0	0	0	13	246	188	73	0	0

Table 4-2 MLE success rate of double picked card decks

Multiple Root Causes Sample size 100	0%	0% ~ 50%	50% ~ 60%	60% ~ 70%	70% ~ 80%	80% ~ 90%	90% ~ 95%	95% ~ 99%	99% ~ 100%	100%
Scenario 1	0	0	0	0	11	163	129	174	43	0
Scenario 2	141	270	2	18	21	52	16	0	0	0
Scenario 3	0	0	0	0	0	0	0	17	247	256
Scenario 4	0	0	0	0	0	0	0	144	376	0
Scenario 5	0	0	0	87	338	95	0	0	0	0

Table 4-3 MLE success rate of single picked card deck with half sample size

Single Root Causes Sample size 50	0%	0% ~ 50%	50% ~ 60%	60% ~ 70%	70% ~ 80%	80% ~ 90%	90% ~ 95%	95% ~ 99%	99% ~ 100%	100%
Scenario 1	0	66	51	51	57	97	95	87	15	1
Scenario 2	447	2	11	13	18	25	4	0	0	0
Scenario 3	0	0	0	0	0	0	0	33	84	403
Scenario 4	0	0	0	0	0	0	5	201	192	122
Scenario 5	4	414	81	18	2	0	0	0	0	0

4.5 Volume Diagnosis Practical Usage

The card game experiment conducted in section 4.4 mimics the diagnosis scenario where there is single suspect/physical defect called out per diagnosis report. The mapping between card game and diagnosis is illustrated in Figure 4-4 below. Drawing one card, with card number 1 recorded, maps the scenario of a diagnosis report with only one defect 1 reported.

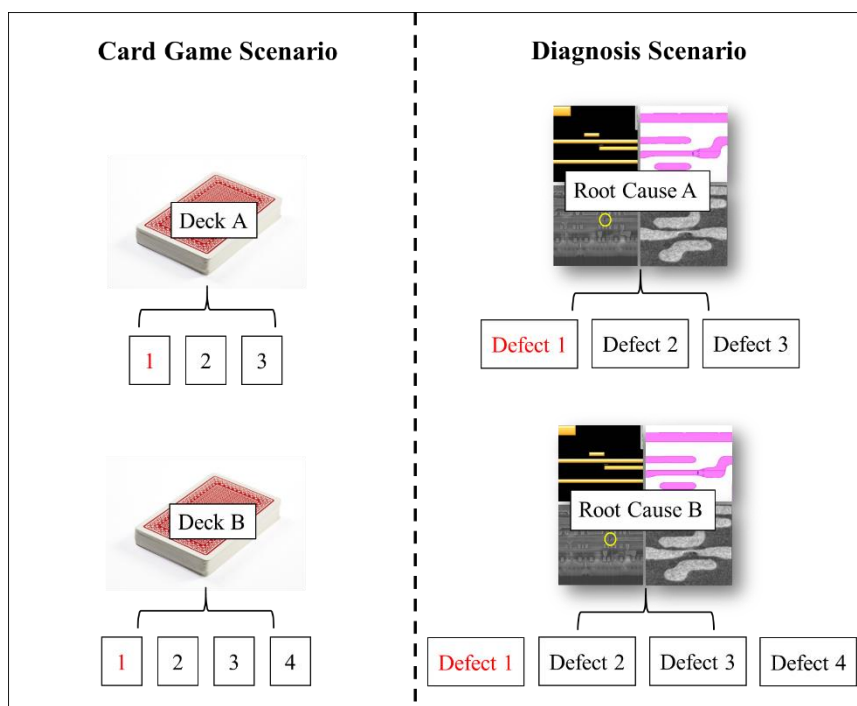


Figure 4-4 Mapping between Card game and Diagnosis

By controlling the number of unpicked card decks per drawn number and $P(n/d)$, we created scenarios that mimic real cases in volume diagnosis, such as, large number of candidate root causes, root causes with sparse existence, and highly correlated root causes. These are also possible scenarios in volume diagnosis after including layout pattern and cell internal root causes, which will be discussed and investigated in Chapters 5 and 6. In the rest of this section, we will discuss general issues in volume diagnosis, using the card game model as reference.

4.5.1 Volume Diagnosis Model Parameter Issues

In Section 6.2, the Bayesian network of volume diagnosis is defined as below:

$$P(v) = \Pi(\Sigma(P(r|f) * (\Sigma(P(f|d) * (\Sigma(P(d|c) * P(c)))))))$$

The accuracy of $P(v)$ depends on the accuracies of $P(r/f)$, $P(f/d)$ and $P(d/c)$.

$P(r/f)$ is a conditional probability. It is the probability of one report when a specific fault in this report is true. Ideally, one fault should cause one failure file, such that it should have one specific diagnosis report. By definition, $P(r/f)$ should be 1. However, due to the complexity of design and tests used, it is possible that some failure files do not match any fault. $P(r/f)$ cannot be 0, which means the diagnosis report does not exist. To resolve this, one can remove such reports from the volume diagnosis reports. Due to complex design and un-modeled fault behavior, this will lead to quite a few diagnosis reports cannot be used. Another alternative is to choose $P(r/f)$ to be less than 1 for faults included in the diagnosis report to indicate that they are close but not exact. To obtain correct $P(r/f)$ requires the understanding of the score systems used in diagnosis tools. Some diagnosis tools even have different score systems for different fault models.

$P(f/d)$ is the probability of a fault if a specific defect is true. Again, this parameter requires the understanding of the relationship among logic faults and physical defects used in diagnosis tools. Typically, one defect causes only one fault, and one defect's physical location is within one fault's logic location. In other words, $P(f/d)=1$. However, some defects may cause un-modeled faulty behavior. Similar to $P(r/f)$, adjustment is needed to obtain correct $P(f/d)$.

$P(d/c)$ is the probability of a defect if one of its root causes is true. It can be calculated based on how many defects can result from this root cause. This information can be derived from layout and defect behavior of each root cause. Defects include opens and shorts in interconnects and inside cells. Their root causes can be process layers, layout pattern shapes, DFM rules and others. The correlation among these root causes and the defects should be used to obtain accurate $P(d/c)$.

Without good domain knowledge, an alternative is to use supervised machine learning techniques to derive these parameters based on good training data. With more aggressive deep learning techniques it is possible that a new model can be created to replace

the Bayesian network. Some domain knowledge is still needed to get proper training data. In Chapter 6, a supervised learning method is proposed to derive parameters.

4.5.2 Data Bias by Limited Sample Size

In Section 6.4, even with a correct Bayesian network for the card game, the success rate of MLE on this card game is still not always 100%. We performed detailed analysis of the cases with less than 100% MLE success rate in the experiments described above. This analysis showed that distribution chosen by MLE always included the picked card decks, but also with some unpicked card decks.

Also, although all experimental data are generated based on a random number generator, the occurrence of each card in the sample does not exactly follow $P(n/d)$ parameter. For example, using deck A in Figure 4-2, its $P(n/d)$ is 1/6. However, drawing a card from this deck repeatedly 100 times, one can obtain the following sampled numbers:

3 2 2 2 4 1 1 1 1 4 6 6 2 1 1 4 5 6 5 6 5 3 2 3 2 1 1 4 4 3 5 6 6 4 5 3 2 4 6 1 4 4 4 4
6 1 4 5 6 3 5 6 3 5 6 5 3 5 6 1 4 2 5 6 4 1 1 4 2 1 1 4 4 6 3 2 2 4 1 1 1 4 3 5 3 2 6 3 2 3 5 6
5 6 4 5 6 5 6 4

In the sequence above, there are 18 occurrences of 1, 13 occurrences of 2, 13 occurrences of 3, 21 occurrences of 4, 16 occurrences of 5, and 19 occurrences of 6. Clearly the occurrences are not exactly 1/6 for each number. Some numbers occur more than others. Statistically, the probability of all numbers will converge to 1/6 if the drawing continues indefinitely. $P(n/d)$ is based on the assumption that all numbers in each card deck have equal probability of being drawn from the deck even if the drawing is limited. As in volume diagnosis, $P(r/f)$, $P(f/d)$ and $P(d/c)$ can be correct based on unlimited diagnosis reports. As with $P(n/d)$, they may not be true for each fault, each defect and each root cause when only limited diagnosis reports are used. The problem caused by limited sample data is referred to in this thesis as data bias. The biased data do not exactly follow assumed underlying probabilities, and can be distorted, as shown in Figure 4-5. Beside biased data, diagnosis

results can be incorrect due to design complexity and un-modeled defect behavior. These incorrect diagnosis reports are called outliers, since the majority of diagnosis reports should be correct. The outliers are not many, but they have bigger distortions, as shown at the right top corner of Figure 4-5. In this paper, to make it simple, biased data include outliers as well.

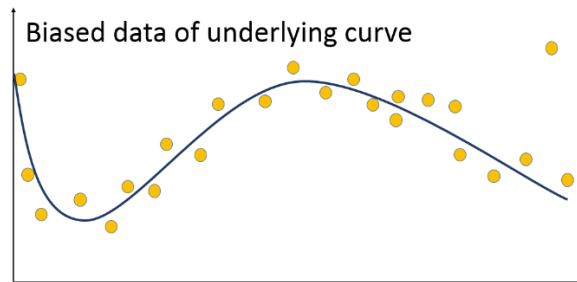


Figure 4-5 Biased data from underlying probability

As mentioned above with biased data in the card game, MLE distribution always includes the picked card decks along with extra unpicked card decks in the experiments of Section 4.4. As shown in Figure 4-6(a), MLE distribution needs to include more card decks to obtain the maximum value on the biased data in Figure 4-5. This problem is also referred to as the over-fitting problem [79]. In volume diagnosis, the over-fitting problem generally adds extra root causes in the MLE distribution. Sometimes these extra root causes can overtake the correct root causes and reduce the success rate to 0%.

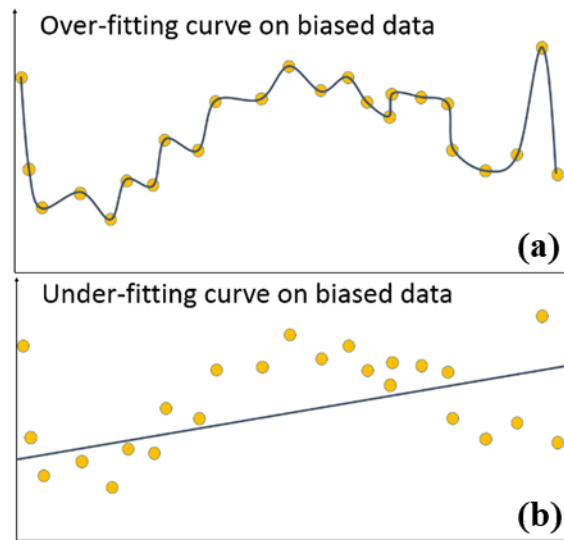


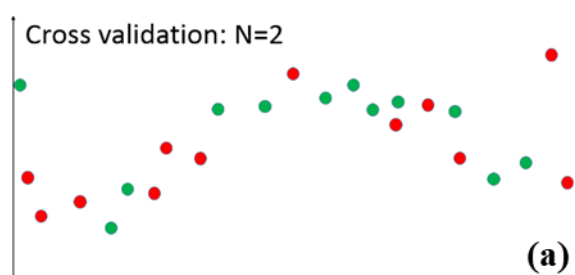
Figure 4-6 (a) Over-fitting distribution; (b) Under-fitting distribution

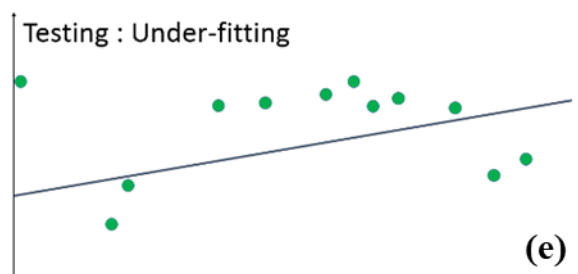
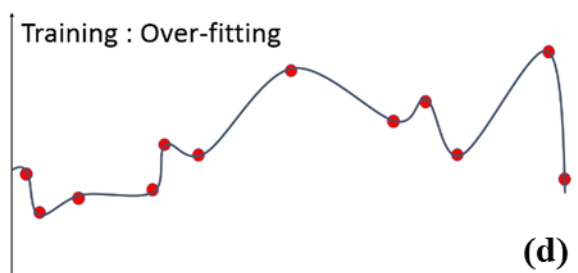
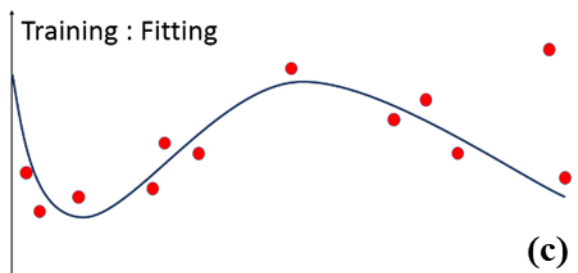
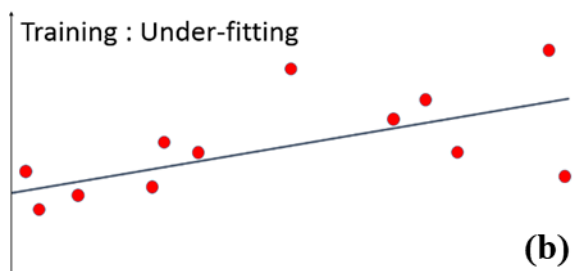
Over-fitting can be alleviated by increasing sample data size, as shown in Table 4-1 and Table 4-3. In volume diagnosis, the sample size is typically not large enough to avoid the over-fitting problem. Therefore, solutions are needed to resolve the over-fitting problem. On the other hand, to avoid over-fitting, if more root causes than necessary are removed, under-fitting could happen, as shown in Figure 4-6(b).

Both over-fitting and under-fitting problems reduce success rates. In most applications, the correct distribution tends to be less complex, such that it is unlikely to have under-fitting problem. When yield is low and quite a few systematic defects exist, the correct distribution can have more root causes and under-fitting can happen. For practical purposes, the under-fitting problem is not a concern since multiple systematic defects are normally fixed one by one during the yield ramp process.

There are several popular machine learning techniques to deal with the over-fitting problem [80]. In this paper, cross validation was used in RCD. The cross-validation process is to divide the total sampled data set randomly into N parts. $N-1$ parts are used as training data, and the remaining 1 part is used as a test data. MLE finds the most likely distribution

of training data and applied this distribution on testing data to measure its fitness. Repeating this process N times with different $N-1$ combinations getting total fitness and then chooses the best distribution based on the total fitness. N can be any integer larger than 1. For example, using $N=2$, sample data in Figure 4-5 are randomly divided into two parts as in Figure 4-7(a) with data shown in red and green. Here the red data are used as the training data, and the green data are used as the test data. Shown in Figure 4-7(b), (c), (d), different variations on the original Bayesian model are used such that their respective MLE finds different distributions. In general, these variations should reduce the number of root causes used to reduce over-fitting. The fitness of these distributions are obtained based on the green test data, as shown in Figure 4-7(e), (f), (g). In this example with 3 variations, it is clearly shown that Figure 4-7(d) and Figure 4-7(g) with the most complex curve fits training data perfectly, but fits test data poorly (over-fitting). Figure 4-7(b) and Figure 4-7(e), with the least complex curve, does not fit the training data or the test data (under-fitting). The fitting distribution that is most similar to underlying distribution fits best in test data, as shown in Figure 4-7(c) and Figure 4-7(f). Therefore, over-fitting and under-fitting problems are avoided based on cross-validation.





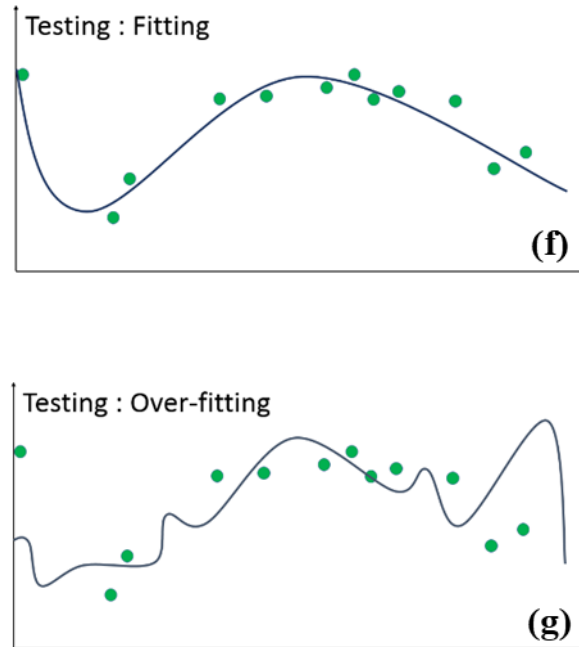


Figure 4-7 (a) Cross validation 2 fold data; (b) Under-fitting distribution on training data; (c) fitting distribution on training data; (d) Over-fitting distribution on training data; (e) Under-fitting distribution on test data; (f) Fitting distribution on test data; (g) Over-fitting distribution on test data

The experiments of single picked card deck cases in Table 4-1 and Table 4-3 were enhanced to include cross-validation. The results are shown in Table 4-4 and Table 4-6. These data show that MLE success rates are improved significantly in all scenarios. The result shows the effectiveness of cross-validation for resolving over-fitting problem. The experiments of double picked card deck cases in Table 4-2 were enhanced to include cross-validation. The results are shown in Table 4-5. MLE success rates of all scenarios for two picked card decks are also improved significantly, but not as dramatically as the improvement for single picked card deck cases. This result shows that increasing the number of card decks picked by the genie not only reduces MLE success rate, but also reduce the effectiveness of cross-validation.

Table 4-4 Single picked card deck with Cross Validation

Single Root Cause Sample size 100	0%	0% ~ 50%	50% ~ 60%	60% ~ 70%	70% ~ 80%	80% ~ 90%	90% ~ 95%	95% ~ 99%	99% ~ 100%	100%
Scenario 1	0	0	0	0	0	0	0	15	0	505
Scenario 2	0	0	0	0	0	0	1	24	0	495
Scenario 3	0	0	0	0	0	0	0	1	0	519
Scenario 4	0	0	0	0	0	0	0	0	0	520
Scenario 5	0	0	0	0	0	0	0	38	0	482

Table 4-5 Double picked card decks with Cross Validation

Multiple Root Causes Sample size 100	0%	0% ~ 50%	50% ~ 60%	60% ~ 70%	70% ~ 80%	80% ~ 90%	90% ~ 95%	95% ~ 99%	99% ~ 100%	100%
Scenario 1	0	0	0	0	0	0	0	31	76	413
Scenario 2	0	0	0	0	0	2	2	182	206	128
Scenario 3	0	0	0	0	0	0	0	12	81	427
Scenario 4	0	0	0	0	0	0	0	12	80	428
Scenario 5	0	0	0	0	0	0	0	132	148	240

Table 4-6 Single picked card deck with half sample size with Cross Validation

Single Root Causes Sample size 50	0%	0% ~ 50%	50% ~ 60%	60% ~ 70%	70% ~ 80%	80% ~ 90%	90% ~ 95%	95% ~ 99%	99% ~ 100%	100%
Scenario 1	0	0	0	0	0	0	5	27	0	488
Scenario 2	0	0	0	0	6	14	9	11	0	480
Scenario 3	0	0	0	0	0	0	0	3	0	517
Scenario 4	0	0	0	0	0	0	0	2	0	518
Scenario 5	0	0	0	0	0	1	20	89	0	410

4.6 Conclusion

In this chapter, a card game model and an investigation of RCD based on card game data is discussed. This method of analysis separates the issues of two parts of RCD, creates controllable volume diagnosis data and enables a feasible investigation of these two parts. The first issue is related to MLE statistical method on a Bayesian network. A simple card game is used to produce flexible data with controllable scenarios to investigate over-fitting issues, even with a clean Bayesian network, given a limited amount of sample data. Cross-validation, a machine learning technique, is investigated to mitigate over-fitting issues. The second part of the problem concerns how to obtain parameters for the Bayesian network of diagnosis data. With an accurate card game model, this issue is avoided in an investigation on limited samples. In Chapter 6, a supervised learning method is proposed to solve the second problem.

CHAPTER V

IDENTIFICATION OF YIELD LIMITING LAYOUT PATTERNS

In this chapter, based on pre-extract layout pattern, an enhanced RCD flow is proposed and described to identify layout patterns root causes [78].

5.1 Introduction and Previous Work

Achieving quick product yield ramp is highly dependent on quickly identifying sources of systematic defects. With advanced manufacturing processes, more systematic defects are being caused by specific layout patterns in designs which are hard to fabricate [42], [43], [51], [55], [56]. Such layout patterns can cause open or short defects in neighboring wires, thus resulting in yield loss. Since each new design may have new layout configurations, conventional methods of identifying low yielding layout patterns like test chips, SRAMs, etc. are becoming less effective. The most straightforward but time-consuming way to confirm that a layout pattern is the root of the problem is through physical failure analysis (PFA). To do this, the PFA images from various samples have to be compared, in order to find any potential commonality in the design associated with the defects. This is manual work which requires both experience and good understanding of the fabrication process. It also requires good PFA result documentation methodology so that design commonalities can be detected across large amounts of samples from a long period of time. This is an expensive process.

More recently, a diagnosis drive yield analysis (DDYA) [38], [40], [43], [47], [48], [49], [51], [53], [66] have approach has been proposed to identify layout patterns causing systematic yield loss [42], [51], [54], [56]. This method relies on using a large volume of scan pattern diagnosis results and analyzing them to identify systematic yield-loss root causes. Layout pattern related root cause has be found be more and more crucial to yield loss as feature size shrinks ruthlessly at advanced technology node. Layout pattern root cause usually has huge amount of types across the design which presents a challenge to

statistical analysis because of the possibility of over-fitting. How to address this issue is an important test of a DDYA technique. In the following sections, we will review a recent work on layout pattern identification and compare it with the solution provided by current RCD.

5.1.1 Previous Work

Recently, Tam et al. [51], [54] proposed to include layout patterns around each diagnosis suspect defect, such that DDYA can be used to identify systematic layout patterns as well. Four steps are used: (1) volume diagnosis, (2) layout pattern extraction, (3) layout pattern clustering, and (4) layout pattern validation. This work has shown a certain degree of success, as demonstrated by its experiment results. More controlled simulation experiments were done in [53]. This research shows that with single layout pattern systematic defect, the layout pattern will be ranked in the top 40 even when the population consists of 80% random defects. To recognize which of these top-ranked layout patterns are real, any simulators for yield-loss mechanism, such as lithography simulation or CMP simulation, can be used to identify which layout patterns can cause Pattern Dependent Systematic (PDS) defects.

In Step 2 of [51], all layout patterns associated with all diagnosis suspect defects are extracted with the following sequences.

1. Each diagnosis suspect defect polygon is shrunk to center lines based on a straight-skeleton algorithm [81].
2. Each center line is broken into multiple segments. No overlapping among these segments.
3. One layout pattern is extracted from each segment with the segment at the center.

In Step 3 of [51], a k-means algorithm is used to do the clustering. The algorithm used bitmap cosine function to define the distance between two layout patterns. As pointed out in [51], for two patterns to be compared correctly with the bitmap cosine function,

pattern alignment is needed. That is the reason for putting defect line segments at the center of layout patterns. However, this step is not sufficient. For example, Figure 5-1 shows two layout patterns from two different diagnosis suspect of OPEN defect. The defects are at the red color nets of (a) and (b). Both nets are quite long, but not exactly the same length. Therefore, when they are broken into segments, two patterns can be extracted, as shown in (a) and (b), with both red net segments in the center. Using the bitmap cosine function, these two patterns can be evaluated as different. In reality, these two layout patterns are identical, except one is a shifted version of the other.

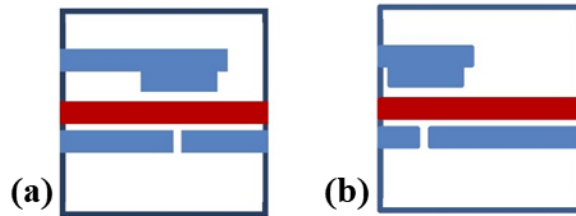


Figure 5-1 Shifted layout patterns

To fix this problem, either more alignments should be done during layout pattern extraction, or the pattern comparison function must be able to accommodate shifted versions.

In Step 3 of [51], all layout patterns associated with all diagnosis suspect defect locations are inputs to its k-means algorithm. The results of this algorithm will produce k clusters of layout patterns by similarity, and each cluster is ranked by its size. However, ranking by size may not always be correct since the majority of these layout patterns are not really responsible for these diagnosis suspect defects. In our experience, the addition of layout patterns in physical properties increases the number of suspect defects in typical diagnosis reports of real silicon failures. However, only one suspect is generally a true

positive. Therefore, false positive suspects need to be filtered out, and, for this the signal to noise ratio should be as high as possible. This chance of false positives is the diagnosis ambiguity we discussed earlier. Work in [51] does not address such issues directly, an oversight that could lead to overfitting on fake layout patterns.

5.1.2 Handling Issues of Previous Work

With the huge amount of layout pattern root causes and existing diagnosis ambiguity, overfitting is a serious concern, and is not adequately handled in [51], [53], [54]. Below, a better solution provide by RCD is discussed.

In order to filter out fake layout patterns in diagnosis reports, we propose RCD as an advanced statistical technique in [48]. RCD uses a Bayesian net to model the probability relationship between all possible defect properties (root causes) over all diagnosis reports. It then uses the EM algorithm [70] to find the best-fit root cause distribution that can obtain the maximum likelihood estimation of all diagnosis reports used. The experimental results show that RCD is able to identify real root causes for 93% simulated cases with single systematic defects. With one experiment including a diagnosis population from 3 different systematic defects, RCD can identify all 3 root causes correctly. To be successful with RCD, it is very important to make use of a correct Bayesian net model with precise probability parameters, which can be derived from the design, layout, and scan test patterns used, as explained in [48] and discussed in Chapter 4. These probability parameters are called RCD constants, defined in Chapter 3. In [49] RCD was put into practical use with proper RCD constants in a Bayesian net to successfully identify systematic defects.

Given the previous success of RCD in dealing with more conventional defect root causes, such as opens/shorts on certain metal layers, it is natural to ask whether it can be extended to handle layout pattern root causes as well. As mentioned before, the main challenge is the huge increase in the number of root causes to consider; they significantly increase the chance of overfitting in the Bayesian model used in RCD. The main

contribution of the proposed work is to show how this problem can be overcome, and demonstrate the application on a real industrial case. The main advantage of our flow as compared to the work in [51], [54] is the use of RCD to filter out the majority of fake suspect layout patterns in all diagnosis reports. By performing such filtering, RCD gives more accurate results. It thus removes the need for the additional Step 4 of the method in [51], which involves process simulation to validate layout patterns. In general, after RCD, the number of possible layout patterns are minimized such that Step 3 of [51] may not be needed.

The remainder of this chapter is organized as follows. Section 5.2 explains our proposed Layout Pattern Analysis (LPA) flow, including details of automatic layout patterns extraction and RCD operation. A control experiment is conducted to show the impact from including layout pattern root cause on non-layout pattern root causes. Section 5.3 describe the general procedure of defect injection for the control experiment; the result of the control experiment in section 5.4 shows that RCD can effectively identify both layout pattern and non-layout pattern root cause. Section 5.5 uses one real silicon case with a test chip data to validate the flow and present the analysis results, including PFA findings. Section 4 5.6 draws conclusions.

5.2 Layout Pattern Analysis (LPA) Flow

In this section, we proposed a layout pattern analysis (LPA) flow that builds upon existing RCD flow, with extra handling of layout pattern root cause. Section 5.2.1 and 5.2.2 below provide details of extracting pattern from layout. Section 5.2.3 discusses how RCD deals with the impact on statistical modeling from introducing layout pattern root causes. The challenge of a huge number of root cause types, mentioned above, will be addressed.

As shown in Figure 5-2, the LPA flow consists of three major steps: layout-aware volume diagnosis, layout pattern extraction with layout pattern matching as an option, and RCD based statistical analysis. Compared with regular DDYA flow, described in Chapter

3, LPA uses the same input data, which include design, layout, scan test patterns and ATE failure files. Layout database (LDB) is the database containing a matching table between the design logics and its layout. Unlike in [51], all layout pattern extraction work is done before volume diagnosis. This is done to avoid performing layout-pattern extraction repeatedly for each diagnosis suspect, and to reduce the run time for volume diagnosis. For LPA, LDB is expanded to store the results of automatic layout pattern extraction before volume diagnosis. In other words, layout patterns are extracted based on layout only, and are independent of diagnosis suspect defect locations. The goal is to extract all unique layout patterns around any location that could be the site of a physical defect. Currently, only interconnect layout patterns are included. Cell internal layout patterns will not be discussed in this work. With this pre-extraction step, diagnosis can be performed by looking up only the layout patterns associated with suspects in the diagnosis results.

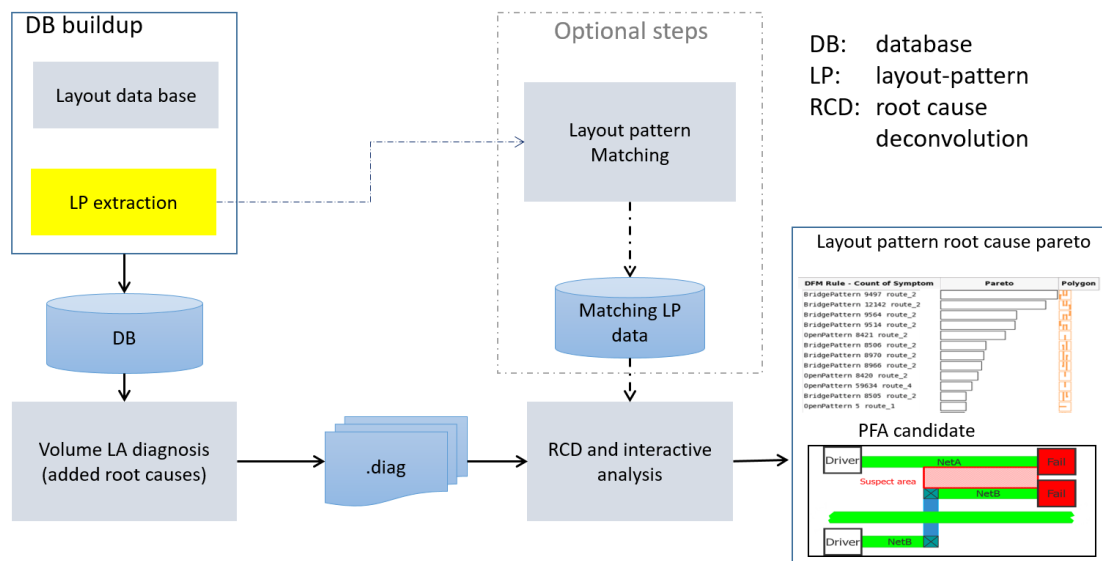


Figure 5-2 Proposed layout pattern analysis flow

5.2.1 Automatic Layout Pattern Extraction

The basic concept here is to first find all possible points that can have open defects or bridge defects. Each of the identified points is called a point of interest (POI). Once all the POIs have been discovered, a layout pattern window is projected around each point, and the contained polygons are extracted to form the layout pattern of each POI, as shown in Figure 5-3. The window size is based on a pre-defined area of influence. All layout patterns are stored in LDB with RTree [82] information, such that later diagnosis suspect location can efficiently match these layout patterns.

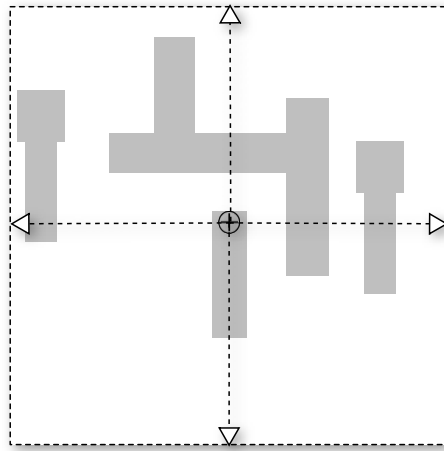


Figure 5-3 Layout pattern window formation

It should be noted that layout patterns extracted in [51] do not allow overlapping with each other. In our approach here, two POIs can be closer than their window size such that their layout patterns can overlap. This avoids the problem mentioned in [51] that some layout patterns at the boundary can be missing in their approach.

The POI identification algorithm proceeds through all metal layers in the layout. On each layer, all layout rectangles associated with a net are merged to form a single polygon.

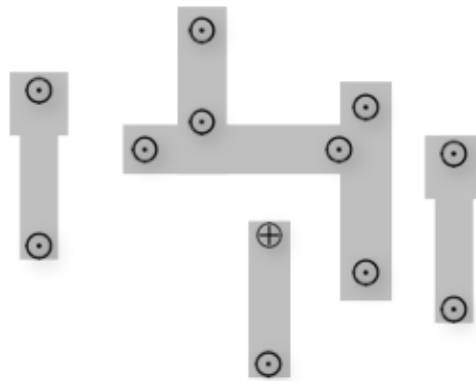


Figure 5-4 Inner points that can be open locations

Step 1 (Open POI definition): Determining an initial set of potential open defects that may occur within the polygon. For example, in Figure 5-4, we first identify inner points that represent potential OPEN defect locations when there is shape disruption in the net itself.

Step 2 (Define Bridge POI definition): Each of these inner points identified at step 1 is projected in 4 orthogonal directions to discover intersections with other polygons, as shown in Figure 5-5. The projection distance is determined by the pitch and track or mental width layout information. Intersected polygons will have a new projected disruption point, defined as shown with '*' in Figure 5-5. The spanning distance between the source and destination polygons adds an additional BRIDGE disruption point, shown as 'b' in Figure 5-5.

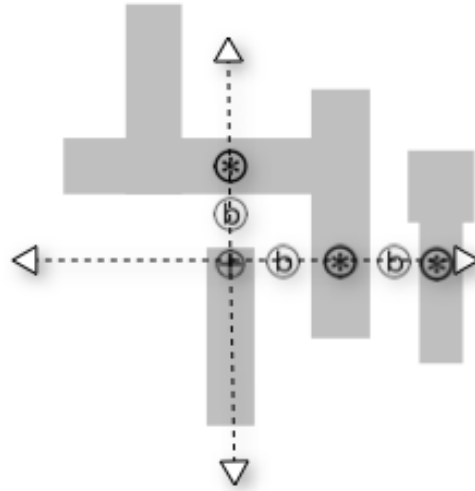


Figure 5-5 Projections adds more POIs

Since POIs are based on their layout environment with disruption, layout patterns from these POIs generally do not have the alignment problem found in [51]. Nevertheless, optional layout pattern matching allows users to match non-identical layout patterns that they believe have the same manufacture process effects. It should be noted that, similar to layout pattern extraction, pattern matching here is independent of diagnosis results, such that it can be done up-front before volume diagnosis. The pattern clustering in [51], by contrast, is based on diagnosis reports and can be contaminated due to a large number of fake layout patterns in diagnosis reports.

5.2.2 Pattern Matching

Here, basic pattern matching allows two patterns to be recognized as identical if they are shifted, rotated, scaled, or mirrored versions of each other. To simplify the process, we transform all patterns into their canonical form first. It is important to make sure that there is one and only one canonical form for all patterns that can be identical after any number of shift, rotate, mirror, and scale operations. The details of calculating canonical form of a given pattern are left out due to brevity.

5.2.3 RCD with Layout Patterns

Each extracted unique layout pattern stands for an instance of canonical form and is assigned a unique ID, which represents a possible systematic defect root cause. For all unique layout patterns, of which there could be tens of millions for one design, their IDs and associated layout polygons are stored in the LDB. To reduce RCD run time, all layout related RCD constants of each root cause are pre-calculated and stored in LDB as well. Each root cause could be a layout pattern, via type, cell type and critical area. The layout-aware diagnosis finds physical suspects for each failure file and correlates them with all physical properties, including layout patterns. If optional layout pattern matching is used to group several individual unique layout patterns, each group is treated as one root cause in RCD. All RCD constants of each group need to be adjusted based on the constants of its members.

RCD run should be the same with and without layout patterns, except that the following special handlings are needed to maintain RCD performance.

5.2.3.1 Huge number of root causes

This is one of challenges that could impact accuracy of RCD. With limited sample size, more root causes will increase the chance of seeing more fake root causes for each report, as we discussed in our analysis of the card game model.

It should be noted that without layout patterns, typically the number of root causes is less than 100. With layout patterns, the number of root causes can be around one billion for a 30 million-gate digital design, due to the presence of so many unique layout patterns in current-day designs. Even when one uses layout pattern matching to group some patterns, the number of root causes is still quite large, while a majority of root causes are not associated with any suspects of all diagnosis reports. However, even if only 1% of the extracted layout patterns are associated with diagnosis results, the number of root causes is still quite large. This large number of causes significantly affects the run time and

memory usage of RCD. To overcome this issue, we exploit the fact that even when the number of root causes is large, a majority appear only in a small number of diagnosis reports. This fact is illustrated in Figure 5-6 below, which plots the occurrence rate of layout patterns in 200 diagnosis reports from a population of dies with a single layout pattern yield limiter. As can be seen, all but one layout patterns occur in fewer than 75 reports, and most layout patterns (>5000) occur only in a single report.

More root causes also create a higher chance of over-fitting. Again, RCD takes advantage of the fact that most of these root causes have a sparse existence, and they are most likely can be filtered out. With that being said, a large number of root causes might still impact the RCD result in the context of limited sample size, according to the conclusion of the card game investigation discussed in Chapter 4. Therefore, a controlled experiment is a good way to evaluate such impact and validate proposed LPA flow.

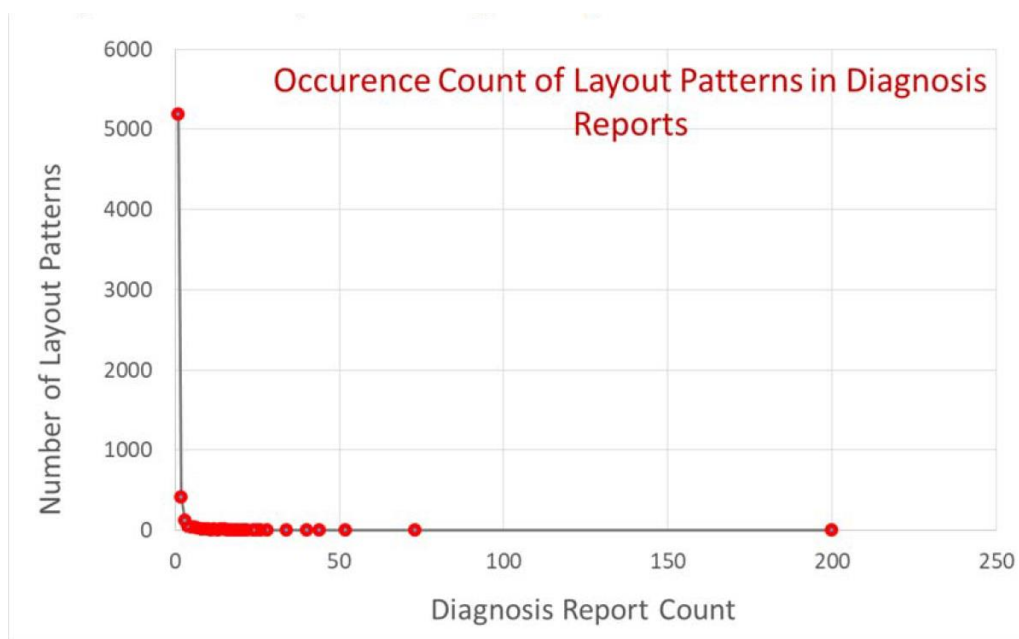


Figure 5-6 most layout patterns occur in only a very few diagnosis reports

5.2.3.2 Equivalent Root Cause

To ensure that all layout patterns are included in the RCD process, all layout disruption points are considered in POI identification. For example, in Figure 5-7, the polygon in red has 7 layout patterns extracted. In RCD, that means that this suspect can be explained by these 7 root causes. If this red-colored net segment is a systematic defect, all 7 layout patterns could be the root causes. If these 7 patterns are used only used in this red-colored net segment systematic defect, these 7 layout patterns cannot be distinguished statistically among themselves. Here we call them (statistically) equivalent layout patterns. It should be noted that they are not the same as pattern groups identified from pattern matching. Unlike in the use of logic diagnosis to report all equivalent faults with same high score, the best fit root cause distribution found by RCD will share probability among these statistically equivalent root causes. For example, suppose that 70% of systematic defects are caused by one of these 7 layout patterns and 30% of random defects are caused by others. Since RCD cannot distinguish these 7 layout patterns, it is possible that each one will be assigned only a 10% probability and may not rank higher than random defect layout patterns. These non-distinguishable root causes are called equivalent root causes in RCD. To fix this problem, all equivalent root causes should be identified and only one representative root cause is kept in RCD operation. Thus all 70% probability is credited to this representative root cause. In the final report, non-representative layout patterns root causes can be identified by their representative layout pattern. After RCD is done, in general, the number of systematic defects left should be very small. To further narrow down any remaining similar patterns, pattern clustering of [51] can be used.

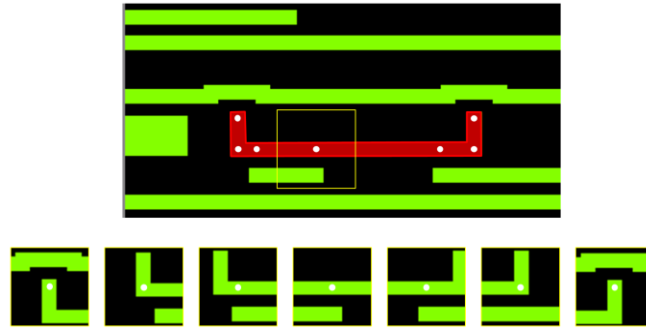


Figure 5-7 Non-distinguishable Root Cause

5.3 Defect Injection Procedures

Section 5.4 describes a controlled experiment that was conducted on LPA flow. The experiment was intended to illustrate the effectiveness of proposed LPA flow in identifying layout pattern root causes, and to evaluate the impact of including a large number of layout-pattern root causes on LPA's ability to distinguish them from non-layout-pattern root causes. In this section, we will first present the general step of the defect injection, and explain each step in detail.

5.3.1 General Steps of Defect Injection Experiment

The general steps of defect injection and the details of each step are presented below. The training data used in the supervised learning method in Chapter 6 also are created following these general steps, with some changes to the fault model and root cause used.

1. Pick a design, define a candidate root cause list and calculate *feature weight distribution/defect distribution*, as mentioned in Chapter 3. This is a one-time-cost step for each design. In the layout pattern experiment, layout pattern extraction is done in this step.
2. Specify a *root cause distribution* and sample size (population size) N based on step 1.

3. Randomly select *instances/defects* of specified root causes following defect distribution calculated in step 1. This is random sampling with replacement.
4. Inject the sampled defects.
5. Perform *fault simulation* on the circuit with injected defects.
6. Repeat steps 3-5 N times.
7. Perform *layout aware volume diagnosis* [14], [16] based on N fail logs.
8. Statistically analyze the volume diagnosis results using RCD to obtain the *estimated root cause distribution*.
9. Compare the estimated root cause distribution and the specified injected root cause distribution. Based on that, *accuracy* of RCD is collected.

5.3.2 Pre-defined Root Cause List

The predefined candidate root cause list represents possible causes of defects that might occur in a die. This root cause list need not be an exhaustive list of all possible root causes. Depending on the information we have on the failing population and purpose of experiment, the root cause list consists of root causes we are interested in. The root causes include open or bridge root causes on interconnect metal layers, open root cause on interconnect via layers, and cell related defects. Count-based and critical-area-based root cause models are used to model defects caused by random particles on certain layers, prone-to-fail physical structures, and library cell types. For a count based modeled root cause, the probability of each root cause instance being defective is the same. For critical area based modeled root cause, the probability of each root cause instance being defective depends on the size of a defect's critical area. The following are 4 types of root cause model used in experiments.

1. Critical area model (bridge/open defects on metal layer): Model of a defect caused by random particles on a specific layer. Critical area of each defect is estimated based on particle size and layout structure around defect sites by a

certain distribution model. The critical area of a defect is used as the feature weight of root causes associated with that suspect.

2. Count model (open via macro and cell type): Model of a systematic defect inside the library cell or on via. Count is used as feature weight. A cell is a collection of connected transistors providing the logic function in a circuit. Open and bridge defects happen inside the cell, causing a defective cell. In controlled experiments discussed in this chapter, cell internal root causes are not considered. Each type of cell is considered an individual root case. This means even two cells of the same type might be defective due to different cell internal root causes; in such a case we would consider them to have failed due to the same root cause. This is an approximation for the root cause of a defective cell, and further investigation on dealing with cell internal root causes will be presented in Chapter 6.
3. One cell area model: Model of a defective cell regardless of which cell internal defect and cell type cause it. The physical area of the cell is used as feature weight to measure the probability.
4. Layout pattern model (open/bridge defect): Model of a defect caused by prone-to-fail physical structures. As discussed in Section 5.2.2, a layout pattern is a group of layout shapes sharing common features. Each layout pattern is a group of layout shapes that are exactly the same after rotation, shifting and/or flipping. Based on proposed layout pattern extraction and pattern matching algorithm, each shape has the exact same size as others. Each shape in this group is referred to as an instance of this layout pattern. Figure 5-8 shows two extracted layout patterns that are used in our experiments. The red point indicates the defect location.

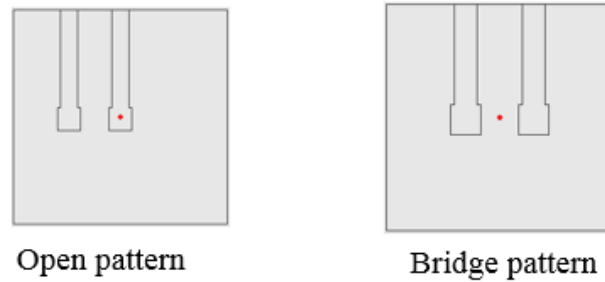


Figure 5-8 Example of layout pattern extracted from layout

In the following discussion, critical area open/bridge are referred to as root causes modeled by critical area as 1st model in the above list and layout pattern open/bridge are referred to as layout pattern root causes as 4th model in the four root cause models listed above.

5.3.3 Step 2 - Random Sampling

A root cause instance is picked randomly based on its feature weight. Each instance of that root cause is a possible defect site triggered by this root cause, and is associate with a physical defect. In the injection experiment, we also refer to it as a defect of that root cause. Random sampling is done based on this feature weight distribution for each root cause defined in Chapter 3.

The feature weight of all layout pattern instances is the same. Each instance of a layout pattern represents a physical structure that would have the same chance of being defective under the same condition. Each layout pattern instance will have the same chance of being sampled. Therefore, for layout pattern injection, all instances are injected.

The feature weight of a defect modeled by critical area is calculated based on the size of its critical area. So, a root cause instance/defect with a larger critical area will be more likely to be sampled. The probability that a region on a layout is defective depends

on the size of its critical area, since contamination particles are more likely to cause defect on a region with a larger critical area.

5.3.4 Step 3 - Defect Injection

For experiment done in this chapter, defect injection is done by modifying the flat model of the circuit in logic simulation. Additionally, for the training data created in Chapter 6, defect of cell internal root cause is injected by SPICE simulation.

Bridge defect injection: A bridge defect is a short occurring between two neighboring nets. In our experiment, we use a wired-AND gate and a dominant bridge model. First, we identify the neighboring nets of sampled instances. Then a wired-AND gate is placed between the nets. If one of the nets is a ground/power line, it is considered the aggressor of dominant fault, and a stuck-at 0/1 fault is injected on the other net as victim being dominated by ground/power line.

Open defect injection: An open defect happens when there is a break on the layout net. An open defect on one segment of a net will affect all branch pins driven by this segment. First, we locate the defect segment and then trace down all branch pins of this segment. Lastly, we inject a stuck-at-1 fault on the defective segment and all pins driven by the defective segment.

5.4 Controlled Experiment

5.4.1 Objective

Layout pattern root causes have an enormous number of types; this wide variety can lead to overfitting in analysis results. The proposed enhanced RCD flow with LPA is expected to extract and group layout shape features, filter out fake layout pattern root causes, and identify the underlying layout pattern root cause. To validate such expectation of RCD, we inject defects of layout pattern root causes and validate whether they can be successful identified. Furthermore, bringing such huge number of candidate root causes

could also have a negative impact on accuracy of RCD for non-layout pattern root causes. To evaluate such impact, we also injected defects of non-layout pattern root causes and compare the accuracy of RCD including and excluding the huge number of layout pattern root causes.

5.4.2 Experiment Setting

In this experiment, we use an industry design. The candidate root cause set includes open and bridge critical area root causes on interconnect layer, open root cause on via layer, cell-type root cause and layout pattern root cause.

Layout pattern root cause injection: This experiment simulates root cause distribution estimation of defects caused by prone-to-fail layout structure. A 100% single layout-pattern root cause is specified as injected root cause distribution. A total of 345 layout patterns, including bridge pattern and open pattern, each of which has more than 50 instances, are injected. For each layout pattern, all instances are injected and each instance is injected only once. The sample size of each layout pattern is its diagnosable instance counts.

Non-layout pattern root cause injection: This experiment simulates root cause distribution estimation of defects caused by random particle contamination on metal layer. A 100% single critical area root cause on metal layer is specified as injected root cause distribution. Open and bridge defects modeled by critical area are injected on all metal layers (layer 1, 3, 5, 7, 9) in this design. In the following discussion, critical area open and critical area bridge are used as root cause types, referring to root causes of open and bridge defects in interconnect layers modeled by critical area. Injected defects/root cause instances are randomly selected with replacement based on the size of their critical area. The one with a larger critical area is more likely to be selected. Two sets of candidate root causes are specified: including and excluding layout patterns. For each injected root cause, 20

cases with different sample size are created. The samples are created by incrementally increasing from 50 to 1000, adding 50 more data points randomly each time.

Success criteria: The accuracy of RCD result is defined as the value of probability of injected root cause in distribution. For each case of injected root cause, RCD is considered to successfully identify the root cause as the dominant one if the probability of the injected root cause in the RCD result is at least 50%. If that probability is less than 50%, but the root cause is ranked at the top in estimated probability distribution, we considered RCD to have successfully identified the injected root cause. Ranking is done by sorting each root cause by its probability in descending order.

5.4.3 Result

Layout patten injection: As shown in Table 5-1, for bridge layout patterns, 100% injected root causes are successfully identified. For open layout patterns, 99.5% of injected root cause are successfully identified. The effectiveness of enhanced RCD in identifying layout pattern root cause are validated.

Table 5-1 Results of layout pattern injection experiment

All instances of root cause injected	Bridge layout pattern injection	Open layout pattern injection
Successful injected patterns	139	206
> 66.7 % accuracy	138 (99.3%)	205 (99.5%)
> 50 % accuracy	139 (100%)	205 (99.5%)
< 50 % accuracy	0 (%)	1 (0.5%)

Critical area injection excluding layout pattern root causes: Accuracy of RCD of injected critical area open and bridge excluding layout pattern root causes are shown in Figure 5-9 and Figure 5-11. All injected root causes are successfully identified with an accuracy higher than 90% for bridge and 66% for open.

Critical area injection including layout pattern root causes: Accuracy of RCD of injected critical area open and bridge excluding layout pattern root causes are shown in Figure 5-10 and Figure 5-12. We saw some negative impact from including layout patterns, but overall, RCD can still effectively identify non-layout pattern root causes in contexts where there are many layout-pattern root causes. In 31 out of 200 cases, accuracy dropped lower than 50%, but in 219 out of 200 cases, the injected root causes was ranked at #1.

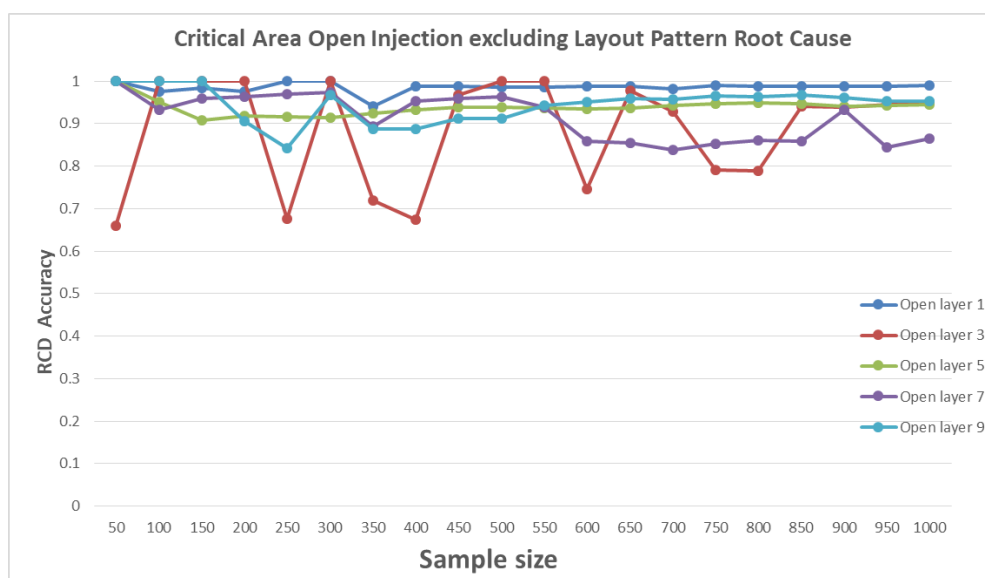


Figure 5-9 RCD result as sample size increases: Critical area open injection excluding layout patterns as candidate root causes

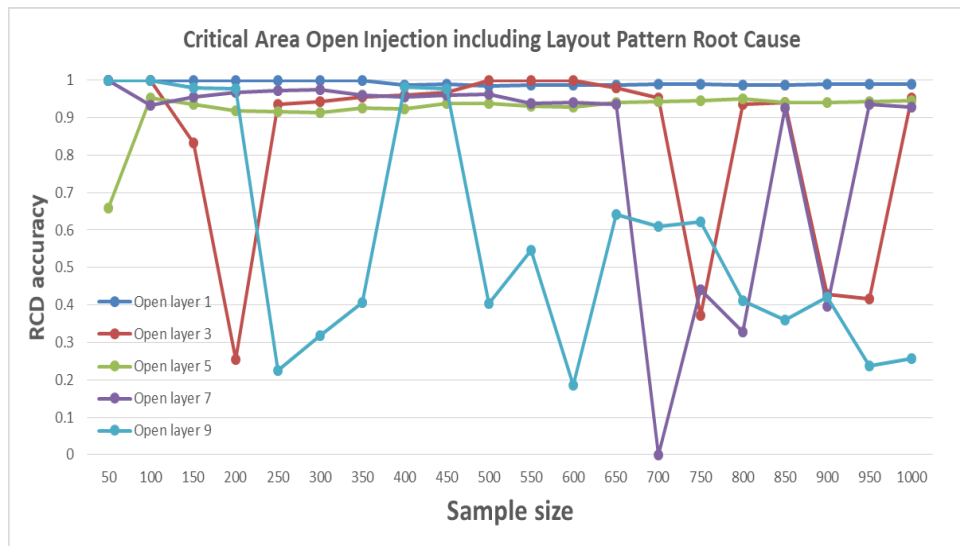


Figure 5-10 RCD result as sample size increases: Critical area open injection including layout patterns as candidate root causes

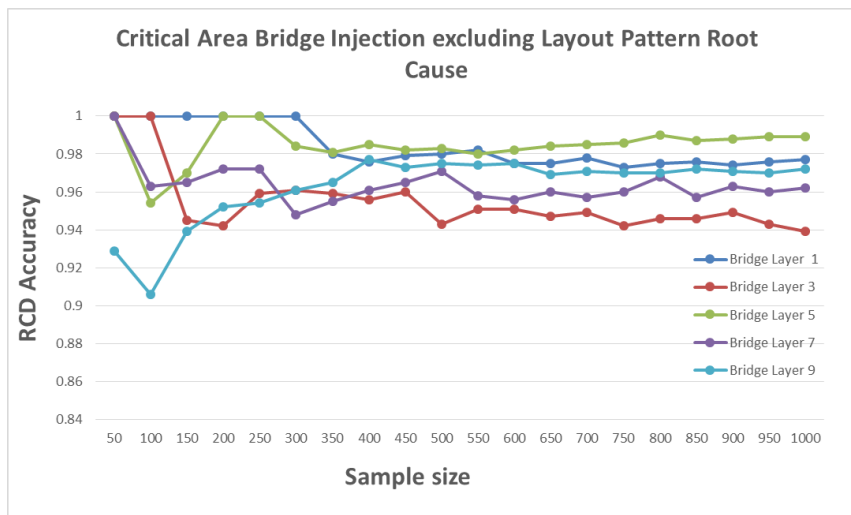


Figure 5-11 RCD result as sample size increases: Critical area bridge injection excluding layout patterns as candidate root causes

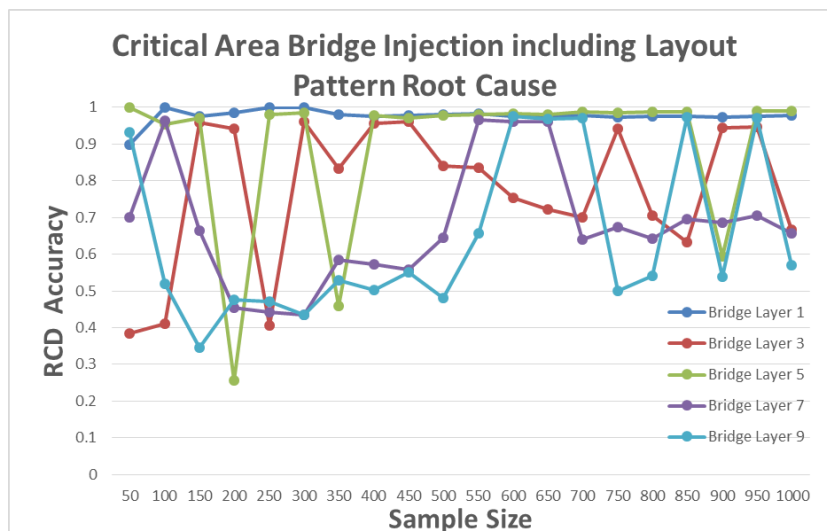


Figure 5-12 RCD result as sample size increases: Critical area bridge injection including layout patterns as candidate root causes

5.4.4 Discussion

As discussed in Chapter 4, limited sample size, larger numbers of fake root causes per report, higher probability of seeing a defect given a fake root cause ($P(d|fake\ root\ cause)$), and more sample outliers would all have a negative impact on the accuracy of MLE. Below, we discuss three crucial factors in critical area injections, including layout pattern root causes.

Number of fake root causes per report and smaller $P(d|fake\ root\ cause)$: As more layout pattern root causes are introduced, the number of candidate fake root causes per report likely increases. As an example, take open layer 9, with a sample size of 1000. In this case, when layout-pattern root causes are included, the called-out candidate root causes increase in number by 75189%. $P(d|fake\ root\ cause)$ for layout pattern root cause is usually much higher than for critical area root cause because the possible defects' number is much smaller. Accordingly, layout pattern root causes usually have lower occurrence than critical area root cause. Occurrence of each root cause is expected to follow the underlying distribution $P(d|root\ cause)$. When samples are limited and biased, occurrence of layout

pattern root causes could be higher than it is expected to. Therefore, MLE might be overfitting to some layout pattern root causes by including them as possible root causes. The controlled results show that even with negative impact from these two factors, RCD still successfully identifies injected non-layout pattern root causes, with some sacrifice on accuracy.

Outlier impacts: The diagnosis accuracy was not perfect in our injections experiment. There are wrong diagnosis reports, which we call outliers in diagnosis data. Outliers bring distortion into samples. For all 21 cases of layer 7 and layer 9 where RCD accuracy is less than 50%, including layout pattern root causes, after removing the outliers (wrong reports) accuracy of 13 cases rose to higher than 50%. Meanwhile, the accuracy of these 21 cases is always higher than 50% if we exclude layout pattern root causes and keeps outliers, as shown in Figure 5-9 and Figure 5-11. This result shows that the low accuracy is from negative impact of outliers. After including layout pattern root causes, such impact from outliers is more severe than excluding layout pattern root causes.

5.4.5 Conclusion

Controlled experiments show that the enhanced RCD flow with automatically extracted layout pattern correctly identify injected layout pattern root causes. With a large number of layout pattern root causes included, the effectiveness of RCD for non-layout pattern root causes is also validated, with some tradeoff on accuracy.

5.5 Silicon Data Validation

5.5.1 Setting

A leading-edge technology test chip is selected to validate the methodology and flow. This test-chip design has 12 cores, and each core design has roughly 1 million gates. To perform RCD statistical analysis, it is recommended to have at least 100 failure files per root cause. As it is hard to know up front how many root causes exist, a good starting

point is to have several hundreds of failure files. We collected 425 failure files from two wafers of the same lot. There are 4 dies with PFA results based on layout-aware diagnosis, and they are all from core3.

Running layout pattern extraction: The LPA algorithm is implemented in a Mentor Graphics tool, and we ran through the layout pattern extraction on the full design for each interconnect layer stored in the existing standard LDB. By using the default project distance, which is 2X of pitch multiplier for both bridge and open layout pattern extraction, we successfully extracted 14.8 million unique layout patterns in 24 hours. This number could be different when a smaller or greater pitch multiplier is used. Theoretically, the greater the pitch multiplier, the more unique the extracted layout patterns, which may impact the RCD analysis resolution on layout patterns.

Once the LDB is updated with extracted layout patterns, the RCD constants were generated for each core and were updated into the LDB. As explained in our prior work [48], RCD root cause constants consider scan test patterns used to make sure that defects triggered by these root causes can be detected during testing on ATE. All the above flow usually involves a one-time setup for existing design and its test pattern sets.

Running layout aware diagnosis: With the new LDB, the layout-aware diagnosis was performed based on each core, and the 425 failure file diagnosis took a couple of hours. The diagnosis engine checked each suspect bounding box against layout patterns stored in the LDB and listed them in the diagnosis report. RCD constants were saved in diagnosis reports as well.

Running RCD statistical analysis: Diagnosis of 425 failure files results in a total of 859 diagnosis symptoms, and each symptom represents a potential defect location. When the diagnosis reports were loaded into RCD, we identified a total of 4,9060 unique layout patterns associated with all diagnosis suspect defects. The final RCD analysis result is presented in a “RCD-sum of probability” pareto, shown in Figure 5-13.





RCD - Sum of Probability	Max Error	Total	Polygon
Count OpenPattern 15980827 M4, OpenPattern 1368083 M2, OpenPattern 1368084 M2, OpenPattern 1368086 M2, OpenPattern 1368087 M2	3.43334	268.653	
Count ViaMacro VIA03	1.42613	49.0233	
Count ViaMacro VIA01	0.014076	21.1157	
Count ViaMacro VIA02	0.391352	13.9746	
Count OpenPattern 15659143 M4	0.642569	13.4902	
Count OpenPattern 18260215 M5	0.432195	12.0677	
Count OpenPattern 15659110 M4	0.201207	11.7834	
Count CELL nor3_1	1.34117	11.4934	
Count OpenPattern 988186 M2	0.516379	11.2588	
Count OpenPattern 18259373 M5	1.01451	10.2554	

Figure 5-13 Pareto for RCD root causes

There are 108 total root causes listed, compared with 49,060 layout patterns that hit the diagnosis suspects before RCD analysis. This list also includes some via and cell type related root causes. M4 OpenPattern 15980827 and its four equivalent layout patterns (M2 OpenPatterns 1368083, 1368084, 1368086 and 1368087) dominate the root cause pareto and others mitigate quickly, a result which suggests that one or more of the 5 equivalent layout patterns could be the root cause of a systematic defect. The zoomed-in view of these layout patterns is shown in Figure 5-14. Please note that in the figure below, two pairs of layout patterns are shifted versions of each other. These layout patterns will be matched by the basic matching process.

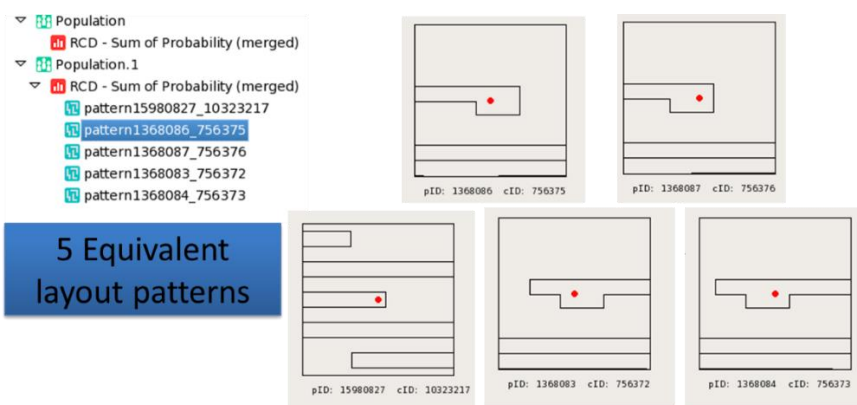


Figure 5-14 Zoom-in view of 5 equivalent layout patterns from the top list

5.5.2 Silicon Data Validation Result

Correlating flow results with PFA data: From the above step, the RCD gives a list of dies containing these 5 equivalent layout patterns, and the die is sorted based on the most suitable suspect for PFA. We checked the 4 dies with available PFA results and found 3 of them (X19Y3, X11Y2 and X19Y4) were on the list. Here, we use die X19Y3 as an example to illustrate our findings, shown in Figure 5-15, Figure 5-16, and Figure 5-17.

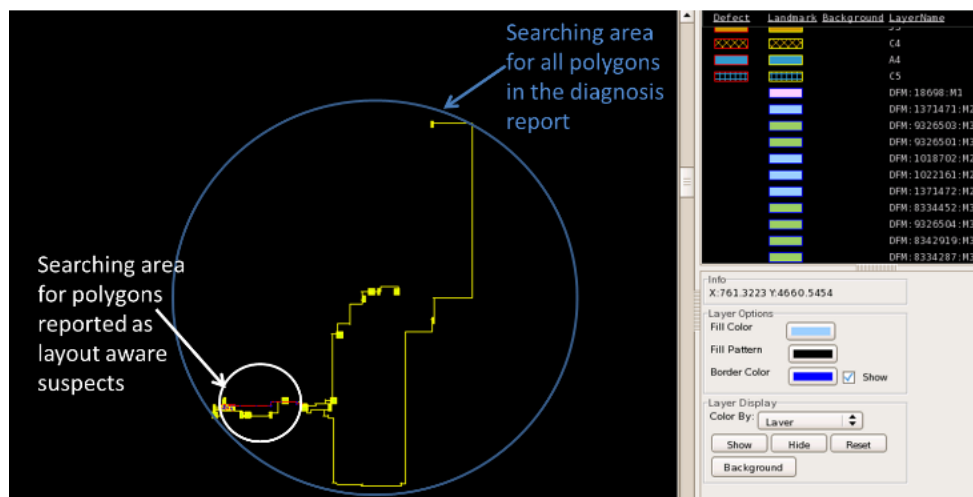


Figure 5-15 Searching area without layout-aware diagnosis

The diagnosis result of die X19Y3 shows 1 symptom and 4 suspects with 37 layout pattern hits, including the 5 equivalent ones. Without layout-aware diagnosis and LPA, the suspects from logic diagnosis run across a large area (blue circle in Figure 5-15), a result which provides little information for root cause and PFA. With layout-aware diagnosis, the suspect polygons represent a much smaller searching area (white cycle in Figure 5-15).

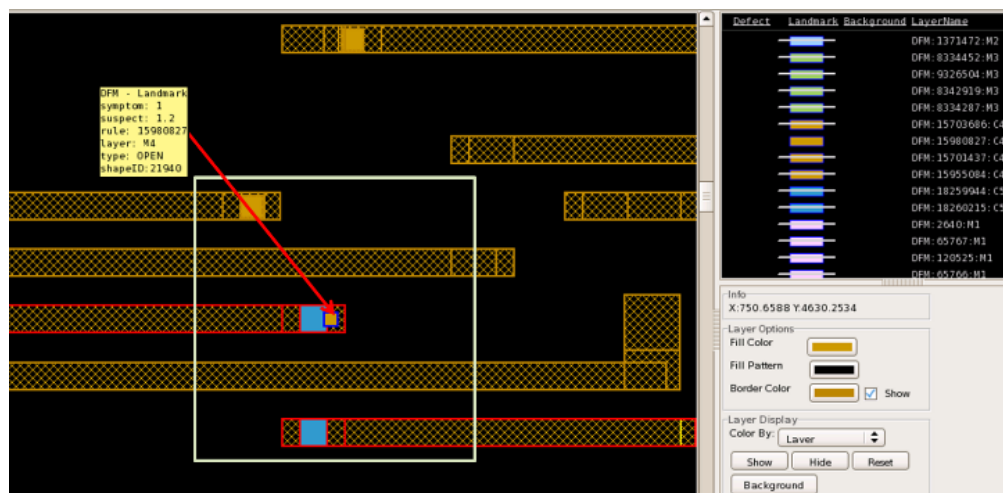


Figure 5-16 M4 OpenPattern 15980827 highlighted by POI

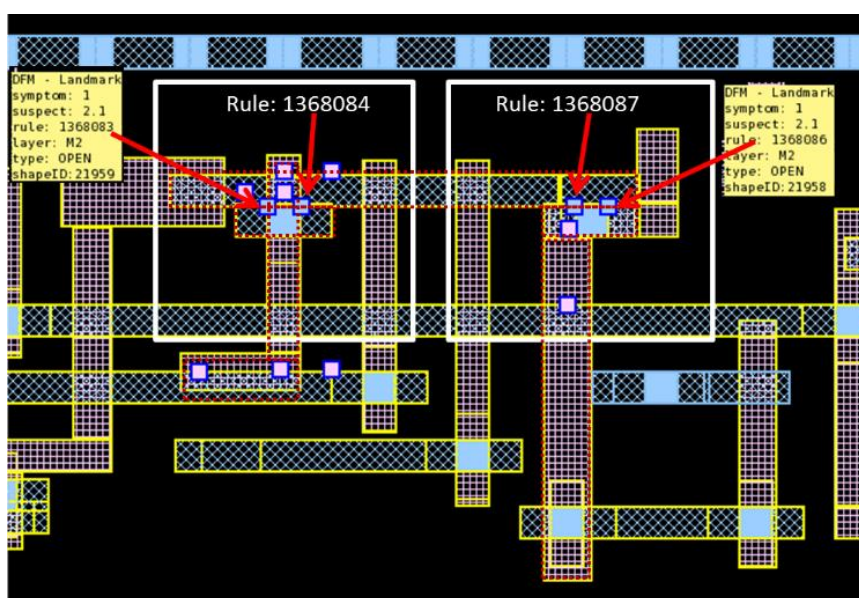


Figure 5-17 M2 OpenPattern highlighted by blue POI

With layout pattern extraction, the analysis can be focused on all the layout patterns hit by the diagnosis (37 of them), but fails to find information on what the root causes are. With LPA, the top root cause of 5 equivalent patterns is identified first by the RCD (Figure 5-13). The following analysis and PFA validation focuses on the RCD selected dies with

the hit of 5 equivalent layout patterns and their surrounding areas. In this case, they are M4 OpenPattern 15980827, shown in Figure 5-16 , M2 OpenPattern 1368086/1368087, highlighted on the right in Figure 5-17, and M2 1368083/1368084, shown at left in Figure 5-17. The center of each individual layout pattern is flagged by a small square or POI. In Figure 5-17, other layout patterns are represented by the pink POIs; analysis of these is skipped since they are not considered as root causes by RCD.

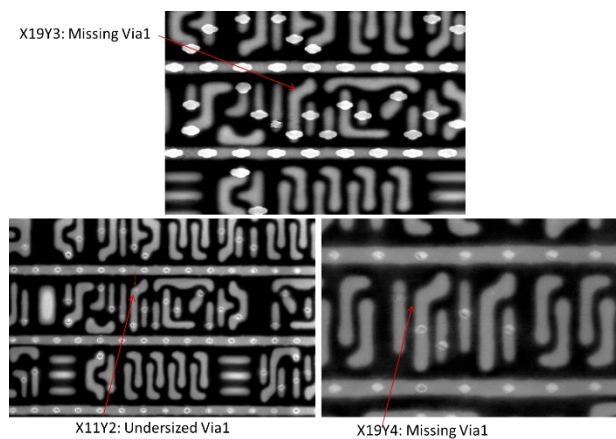


Figure 5-18 PFA results for 3 dies

The PFA results of the three dies are in Figure 5-18, showing that the real defects are missing via, with its location correlated to M2 OpenPattern 1368086/1368087. Though the PFA images in Figure 5-18 show only the M1 and via, which seems to be a via issue on the jog M1, it is very interesting to find out later on that the missing or under-sized via was indeed triggered by the identified M2 open pattern. For the specific wafers tested, the M2 shapes identified by M2 OpenPattern 1368086/1368087 had abnormal shrinkage, which blocked the proper formation of the via underneath.

Silicon result summary: In this presented test case, LPA flow automates the layout pattern extraction process based on self-contained data used for layout-aware diagnosis.

RCD conducts a thorough analysis on the layout patterns and greatly improves the resolution of root cause identification for systematic defects (Table 5-2).

Table 5-2 systematic defect identification resolution improvement

LPA steps	number of layout patterns	resolution improved	comment
initial layout pattern extracted	14849441	N/A	full chip layout pattern extraction
Layout patterns hit with LA diagnosis	49060	303x	425 LA diagnosis
RCD layout pattern root cause pareto	108	454x	425 LA diagnosis
PFA candidate	5	22x	based on 3 dice
Total improvement		2969888x	

LPA flow can also be used for die/suspect picking since it helps narrow down the PFA searching area as well. Table 5-3 shows the comparison. The total time spent on from going through the LPA flow to PFA validation of identified root causes can be completed within one week.

Table 5-3 PFA candidate searching area reduced

methods	Suspect searching area	resolution improved searching area reduced	comment
Logic diagnosis	Fig 15 blue circle	N/A	
LA diagnosis	Fig 15 white circle	19x	compared with logic diagnosis
LA diagnosis with RCD/LPA	Fig 16 &17 (white squares)	20x	compared with logic diagnosis
Total improvement		380x	compared with logic diagnosis

5.6 Conclusion

In this chapter we present enhancements to the RCD flow that enable it to overcome potentially weaknesses and identify, correctly and automatically, the layout patterns causing systematic yield loss. We discussed several innovative enhancements on top of layout pattern systematic defect analysis, with volume diagnosis proposed in [51]. A controlled experiment was conducted to validate the effect on identifying both layout patterns root cause and non-layout pattern root cause when including layout pattern root causes as candidate root causes. With real silicon data, we have demonstrated the practicality of this work flow, with its huge advantage in efficiency and turnaround time in layout-related systematic defect identification. A layout pattern is identified by LPA and was validated by PFA to be the dominant yield loss mechanism.

CHAPTER VI

SUPERVISED LEARNING BASED RCD USING VOLUME CELL-AWARE DIAGNOSIS

In this chapter, to address the aggravated parameter issue brought by including cell internal root cause in RCD, a supervised machine-learning algorithm is proposed to accurately learn RCD model parameters from labeled training data.

6.1 Introduction

Logic diagnosis [8-13] has been used to determine the most likely defect locations and defect mechanisms for a defective device with scan-test failure data. Recent advances in scan diagnosis technologies include use of more physical information, such as layout-aware diagnosis [14-16], cell-internal diagnosis [17-20], and cell-aware diagnosis [21-22]. This additional information helps improve diagnosis resolution. Such diagnosis results have been used successfully to guide physical failure analysis (PFA) to focus on a much smaller area and identify a given physical feature. Such identified features include, but are not limited to, defect type (open, short), defect layer, via macro type, cell type, critical area, defect shape, specific layout pattern and specific DFM (design for manufacturability) rule.

Diagnosis tools typically consider each failing die separately and produce a diagnosis report per die. Several papers [38], [40], [43], [47], [48], [49], [51], [53], [66] have demonstrated that we can apply statistical analysis on volume diagnosis data (i.e. on a collection of diagnosis reports), to identify a set of common physical defect features or root causes. Yield can be substantially improved by fixing the common root cause(s). RCD [48], [77] is one such statistical analysis technique that uses an unsupervised Bayesian model, and has proven very effective for identifying interconnect, cell-type and layout-pattern root causes using volume diagnosis. At a high level, RCD's input consists of, for each diagnosis report, a certain set of values, which parameterize the Bayesian model. RCD

combines all this information from individual diagnosis reports and endeavors to give a root-cause distribution that best explains all these reports.

For advanced technology nodes, cells require multiple process steps and have more complicated transistor/routing structures. This added complexity leads to more root causes and more subtle defect behaviors for cell-internal defects. Due to increased complexity and smaller feature size, it is becoming increasingly difficult to accurately identify cell-internal root causes. At the same time, cell-internal root causes are becoming more prevalent as yield limiting factors. Due to several reasons discussed in Section 6.2 below, in order to extend RCD to support cell-internal root causes, we need a better way to compute the input parameters of RCD, on a per-diagnosis-report basis. One way to do this is to gain more expertise with and domain knowledge of cell-internal root causes. In this chapter, we propose an alternate approach for learning these input parameters. We show that supervised machine learning techniques can be used effectively to learn these model parameters from training data. Once we have a better estimate of these parameters, we plug these back into RCD and obtain accurate results, as described in the results section later in the chapter. The use of supervised learning methods at an individual diagnosis-report level, followed by using unsupervised learning methods (e.g. RCD) at the volume level, is a novel idea and is one of the key contributions of this work.

This rest of the chapter is organized as follows: In section 6.2, a high-level description of RCD and the targeted RCD model parameters are given. Additionally, the problems that arise when trying to identify cell-internal root causes are explained. Section 6.3 gives an overview of supervised machine learning. Section 6.4 contains the main technical contributions of this chapter. We describe our use of supervised learning, including how we generate training data, and our learning algorithm. A flowchart is provided to illustrate changes made to RCD flow. Experimental setup and experiments conducted to validate the proposed method are presented in Sections 6.5 and 6.6. In Section

6.7 we demonstrate the effectiveness of our method on real silicon data. Section 6.8 concludes the chapter.

6.2 Background and Problem Motivation

Due to design complexity and the use of a limited number of test patterns, diagnosis reports often have multiple faults, defects, and root causes. As a result of this diagnosis ambiguity, we cannot get simply sum up the root causes in each report to get the correct root-cause distribution. In [48], [77], a new method called root cause deconvolution (RCD) was presented to address this problem. RCD constructs a Bayesian network model and uses the information in the diagnosis reports to do inference to figure out the underlying root-cause distribution. RCD and the Bayesian network model for volume diagnosis have been discussed and described in detail in Chapter 3. In the discussion below, we will present a brief review of how we obtained the information we used in the RCD model from diagnosis reports. We will then re-visit the parameters of the Bayesian network model.

Possible Defect locations: Layout-aware diagnosis extracts all possible defect locations or physical suspects based on layout. For an open defect, the location is a segment of a net (set of polygons). It is expected that the faulty behavior will be the same if a defect occurs on any part of the segment. For a bridge defect, the location is a pair of nets. Similarly, the faulty behavior is expected to be the same if a defect occurs at any location between this pair of nets.

Diagnosis procedure: Logic diagnosis takes in the response of test patterns applied to defective die (injected or real defect), traces back to potential fault sites, performs fault simulation on the faulty sites, and ranks the potential fault sites based on how well each modeled fault simulation response fits the observed output response. Top-ranked fault sites will be reported with ranking scores. Each faulty site is termed a fault in the diagnosis report. Layout-aware diagnosis then matched possible defect sites extracted from layout with the logic faults. During the matching, many logic faults will be excluded. For example,

a pair of bridging nets, which have the same failure behavior as observed, will not be reported. It is because they are too far away to cause a short between signal lines. For an open defect, it can locate which segment best matched the defect behavior. These physical defect sites (segment or net pair) are physical suspects or defects associated with each fault in the diagnosis report. Lastly, for each physical defect reported, the possible root causes of that defect that might occur on each layer at that location are reported. For each root cause, the exact location of the actual defect can differ. Using an open defect as an example, a defective segment (reported as open defect) could consist of a set of polygons of metal layer one and a different set polygons of metal layer two. Accordingly, polygons of metal layer one/two would be defined as a root cause instance of open root cause on layer one/two.

Probability of defect occurring: In the RCD probability model, we calculated the probability of a defect occurring for each possible defect that is caused by a given root cause. First, we list all instances of that root cause. As explained above, each root-cause instance is associated with a physical defect. For one root cause, probability of a root-cause instance occurring is the same as the probability of its associated defect occurring. We assume that each instance of a given root cause occurs with certain probability. This probability is proportional to an assigned weight of each instance of the given root cause. The sum of weights of all instances of a root cause is defined as the RCD constant of that root cause. The probability of a specific defect occurring given a root cause is calculated as the weight of an instance of given root cause divided by RCD constants of the given root cause. That particular root-cause instance need be associated with the specified defect. Given different root causes, the probability of the same defect occurring differs. In the discussion following, an instance of given root cause will also be called a defect instance or a defect of that root cause.

Defect occurring assumption: The weight of each defect instance given a root cause is calculated based on assumptions about how defects occur. These assumptions are the following:

1. Defects on metal layers are caused by random particles. A random particle causes an open defect if it lands on a net, and a bridge defect if it lands between two neighboring nets.
2. Open defects on via are caused by malformation or missing via.
3. Open and short defects are caused by certain hard-to-fabricate layout shapes.

Based on the above assumptions, the weight of an instance can be calculated as critical-area-based and count-based:

1. Critical-area-based: Critical area is calculated for each polygon in a segment (for open defects) and for each net pair (for bridge defects). The critical area of a segment is the sum of critical areas of all polygons in that segment. The weight of each defect caused by open or bridge root cause on a specific layer is proportional to the critical area of that segment, or the net pair of that layer.
2. Count-based: For root causes such as layout pattern and via type, with the same manufacturing process, each instance of a root cause is considered to have the same chance of being defective. The weights of instances of these two types of root cause are all the same. We used count as weight. For example, if there are two via instance of the same via type located at the same segment of net, the weight will be two.

Below, we will explain how parameters in RCD models are estimated based on the above information and assumptions.

Let $P(v)$ be the probability of observing the given set of volume diagnosis reports. The RCD model tries to find a root-cause distribution that will maximize $P(v)$. It expresses $P(v)$ as the product of the probability of observing the individual diagnosis reports ($P(r)$). It further breaks down $P(r)$ into several parameters, $P(r/f)$, $P(f/d)$ and $P(d/c)$. We consider

a re-parameterization of the RCD model, and express the probability of observing each report as follows:

$$\begin{aligned}
 P(r) &= \sum_{n=1}^N P(r, c_n) \\
 &= \sum_{n=1}^N P(r|c_n)P(c_n) \\
 &= \sum_{i=1}^I P(r|f_i) * \left(\sum_{j=1}^J P(f_i|d_j) * \left(\sum_{n=1}^N P(d_j|c_n) * P(c_n) \right) \right)
 \end{aligned}$$

where c_n is the n^{th} root cause, and we sum over all possible root causes. $P(c_n)$ is the probability of c_n being the real root cause. $P(r|c_n)$ is the conditional probability of observing the report r , when the real root cause is c_n . f_i is the i^{th} fault. d_j is the j^{th} defect. The first equality holds because the root causes (c) are assumed to be mutually exclusive, and their probabilities ($P(c)$) add up to 1 [74]. The second equality follows from the chain rule of probability [75]. In RCD, we are interested in finding the root-cause distribution, i.e. the value of $P(c)$ for every root cause c . With the above re-parameterization, we have captured all the parameters $P(r/f)$, $P(f/d)$ and $P(d/c)$ in RCD in a single parameter $P(r/c)$. The performance of RCD hinges on the accuracy of the estimate for $P(r/c)$.

As discussed in Chapters 3 and 4, $P(r/f)$, $P(f/d)$ and $P(d/c)$ are estimated based on assumptions and domain knowledge of diagnosis scenarios and defect behavior of each root cause.

$P(r/f)$ is the probability of a given report when a specific fault in this report is true. Ideally, one fault should cause one failure file, such that the fault should have one specific diagnosis report. However, due to the complexity of designs and tests used, it is possible that some failure files do not match any fault. $P(r/f)$ cannot be 0, which means the diagnosis report does not exist. To resolve this problem, one can remove such reports from the volume diagnosis reports. Due to complex design and un-modeled fault behavior, quite a

few diagnosis reports may have to be removed. Another alternative is to designate $P(r/f)$ as less than 1 for faults included in the diagnosis report to indicate that they are close but not exact. To obtain the correct $P(r/f)$ requires an understanding of the scoring systems used in diagnosis tools. Some diagnosis tools even have different score systems for different fault models.

$P(f/d)$ is the probability of a fault if a specific defect is true. This parameter requires an understanding of the relationship among logic faults and physical defects used in diagnosis tools. Due to logical equivalence, the same failure behavior we obtain from test pattern response could be explained as resulting from different individual defects. Such an explanation might vary by setting of diagnosis, and can include the score of each defect, root causes of defect, number of defects, etc. For one report, each possible defect site is considered a defect, and is scored by how well the observed behavior of the defective die matches the behavior explained by the fault model we applied in the diagnosis. For now, this probability is calculated based on that diagnosis score.

$P(d/c)$ is the probability of a defect if one of its root causes is true, and can be calculated based on how many defects are triggered by this root cause. This information can be derived from layout and defect behavior of each root cause. If a reported physical defect occurred because of a given root cause, there would then be at least one instance of that root cause associated with that physical defect. This is the defect occurring probability we described above.

These three parameters are not perfectly accurate if defects have un-modeled faulty behaviors. And discrepancies thus exist between assumptions and real silicon data. One of discrepancies is between simulated defects and real silicon data. $P(d/c)$ of both simulated defects and the original Bayesian model are based on critical area and count-based model. However, in reality, other factors might also contribute to occurrence of a defect. For example, the critical area of short defect between two long neighboring nets is large, but such a defect is less likely to occur if the manufacturing process at the new technology

node is improved. Based on domain knowledge of defect mechanisms, adaptation to new defect occurrence assumptions for $P(d/c)$ can be achieved. However, this approach is not the focus of our work.

The other discrepancy is between $P(r/f)$ and $P(f/d)$ of the real diagnosis process, and of Bayesian model assumption. Whenever an unmolded fault presents itself, problems can arise. However, current RCD parameters have shown effective robustness through previous successful use of RCD to identify interconnect, cell and layout-pattern root causes.

Cells have a small number of input and output pins. Due to this feature, many defects within the cell corresponding to different root causes produce the same failing bits. This dynamic results in several root-cause candidates for a single diagnosis report, making it hard to distinguish them from each other. This is a much bigger issue for cell-internal root causes, as compared to interconnect root causes. Moreover, candidates corresponding to cell-internal root causes have lesser variability between reports in terms of shape/length/critical area, as compared to interconnect root causes. This exacerbates the problem of identifying cell-internal root causes. We illustrate this using Table 6-1.

Table 6-1 Percentage of root causes experiencing high co-occurrence and domination

	Root cause type	Design B	Design C	Design D
High co-occurrence & domination	Interconnect	0	0	0
	Cell-internal	92%	46%	68%

Suppose we have a collection of diagnosis reports, where we know the real root cause (C_{real}). If, in more than 85% of the reports in the above population, candidates

belonging to a different root cause (c_{fake}) also appear, we say that c_{fake} has high co-occurrence with c_{real} . Moreover, if in at least 50% of the reports we find that $P(r/c_{fake})$ is larger than $P(r/c_{real})$, we can then say that c_{fake} dominates c_{real} . In Table 6-1, for several industrial designs, we show the percentage of interconnect and cell-internal root causes that become dominated by, and have high co-occurrence with, some other root cause.

Thus, we need a more accurate estimate of $P(r/c)$ in order to distinguish cell-internal root causes using the RCD Bayesian model. Such improvement can be done by using domain knowledge, but that is not the focus of the work proposed in this chapter. In this work, we use supervised machine learning to accurately estimate $P(r/c)$ for each report and root-cause combination.

6.3 Supervised Machine Learning Techniques

Supervised learning refers to the class of machine-learning algorithms that allow us to learn the mapping from inputs to outputs, based on a labeled training data set. A labeled dataset comprises example input and output pairs, where the “label” refers to the known output values. Over the last few years, machine learning in general, and supervised learning in particular, has been successfully applied in various new fields and has proved to be transformative [83]. The algorithm’s performance typically keeps improving as it sees more training data. Machine learning techniques often work very well in practice—even with relatively small datasets and simple models, we see very good results.

6.3.1 Feature Extraction

One of the key steps that plays a big role in determining the success of a supervised learning algorithm is the choice of features. With a good choice of features, even simple algorithms may perform very well, whereas if the features are not very informative or discriminative, then even a complex learning algorithm will not be able to do a good job.

Supervised learning can be used for various tasks, such as regression and classification. In classification, the output can be one of several different categories or classes, and the task is to correctly predict the class for the given input value.

6.3.2 Type of Classifiers

Various types of classifiers are commonly used in machine learning. Some of these classifiers output only the predicted class. Popular classifiers such as Support Vector Machines (SVMs) and random forests fall in this category [85]. Although researchers have attempted to associate a probability with the output using these classifiers, the latter do not include a natural notion of a probability associated with each class. Another type of classifiers, such as logistic regression and naïve Bayes, naturally output a probability for each class. If we wanted to choose a single class, we would choose the class with the highest probability. However, we can also use the probability for other applications. In this paper, we are interested in the second type of classifiers, which output a probability associated with each class. We are using a linear transformation function which outputs the probability of each class based on the value of a linear combination of input features. Other techniques, such as logistic regression or native Bayes, can also be used as classifiers in our supervised learning framework, as long as the output of the classifier is assigning probability to each class. Investigating this possibility could be a direction for our future work.

6.4 Our Use of Supervised Learning

One approach to identify root causes is to go directly to the root-cause distribution using supervised learning. In such an approach, a single training data point would consist of an entire population of volume diagnosis reports as input, and the output would be a root-cause distribution. We would need a number of training data points for each root cause, where each data point itself consists of a population of diagnosis reports. This

method would require a large number of diagnosis reports for training. For example, suppose we need at least 200 diagnosis reports to obtain a root-cause distribution. These reports comprise a single training data point for the supervised learning approach described above. Further, suppose that we need at least 200 training data points per class (i.e. root cause) to learn a classifier. This requirement implies that we need at least $200 \times 200 = 40,000$ diagnosis reports per root cause for training. If we have 50 root causes, we would need more than $50 \times 40,000 = 2,000,000$ diagnosis reports for training. Therefore, we consider this approach impractical. In the approach that we describe below, each data point consists of just a single diagnosis report. Continuing to use the assumptions stated here, we would need 200 diagnosis reports per root cause for training. Therefore, if there were 50 root causes present, we would need only $50 \times 200 = 10,000$ diagnosis reports for training.

Our goal with supervised learning is to obtain an accurate prediction for $P(r/c)$ —the probability of a report, given a root cause. We use a custom classifier that takes a report as input and outputs an accurate estimate for $P(r/c)$. Details of the classifier, i.e. the learning algorithm, are described in the following subsection.

6.4.1 Training Data

In supervised learning, one key assumption is that the distribution and characteristics of data in the test set are similar to those of the data in the training set. In other words, the training data should be a good proxy for the test data set. We generate training data using the following procedure:

1. *Compute the list of root causes in which we are interested.* On each layer in a design, a short between two neighboring nets causes a bridge defect. A disconnected net causes an open defect. A missing via causes an open defect. The number of interconnect layers, via types and layers inside cells depends on the technology node of the manufacturing process. A list of candidate root causes should theoretically include all defined possible root causes of defects

on all layers in case a real root cause is missed. This list would typically consist of open and bridge root cause for each interconnect metal layer, open root cause for each interconnect via layer, open and bridge root cause for each layer inside the cell (e.g. contact to poly, contact to diffusion, metal layers, etc.), inter-layer bridge root causes within the cell. In proposed supervised learning, root causes of which the occurrence is quite small are excluded.

2. For each of the root causes, we then compute the *list of possible defects* corresponding to that root cause.
3. For each root cause, we *compute the probability of each defect occurring*. Depending on the root cause, we assume either that the probability of each defect is proportional to its critical area, or that each instance is equally likely.
4. We now *sample (with replacement) from the defect distribution* for each root cause. Ideally, we want a data sample that captures most of the variability in terms of root-cause behavior and diagnostic tool behavior for the root cause. In general, we expect the results to be better as we increase the number of training data samples. For critical-area-based root causes, defect location is randomly chosen based on critical area of each defect instance. Defects with larger critical areas are more likely to be injected. For counted-based root causes, each location has some chance to be defective. The location of defect thereby is purely random.
5. We *simulate the behavior for each selected defect* for the set of scan test patterns. All defects are injected on the logic-level netlist by changing the logic value on impacted signal lines and pins. Logic simulation is then applied on the faulty circuit for each defect. As a result of our simulation, we obtain the failure file for each defect. How to change the logic value depends on the fault model used. For interconnect bridge defects, AND/OR bridge fault and dominate bridge fault are chosen randomly in each injection. For interconnect open

defects, stuck at 0 and 1 fault are placed in a defective line and its downstream branches randomly. For cell-internal defects, the faulty logic values of cell pins are obtained by SPICE simulation on parasitic elements extracted from the layout inside a cell. To summarize, for injection of both types of defect, we use logic simulation based on defined fault models. Fault models of cell-internal defects are defined by SPICE simulation [21], [22] on defects inside cell. Defects inside cells are obtained by changing the value of resistor and capacitor in an extracted transistor netlist of one cell. The process variations are not considered.

6. We run a state-of-the-art cell-aware diagnosis tool [22] to generate a diagnosis report for each failure file. Each diagnosis report is labeled by its injected root cause.
7. For each diagnosis report, we calculate the probability of the report being generated by the root cause. We use the same calculation as in [77] and discussed in Chapters 3 and 4. We thus obtain a vector-per-diagnosis report with length equal to the number of root causes. This is the input for our learning algorithm.

Our goal with supervised learning is to obtain an accurate prediction for $P(r/c)$ —the probability of a report, given a root cause. We use a custom classifier that takes a report as input and outputs an accurate estimate for $P(r/c)$. Details of the classifier i.e., the learning algorithm, are described in the next subsection.

6.4.2 Learning Algorithm

The goal of our learning algorithm is to output an accurate estimate of $P(r/c)$, i.e. the probability of a report, given a root cause. Consider two root causes, $c1$ and $c2$. The high-level idea behind our approach is that when the real root cause is $c1$, the diagnosis reports that are produced have certain common characteristics. These characteristics differ

from those of the reports that are produced when c_2 is the real root cause. In the training phase, we obtain sample diagnosis reports for each root cause and try to determine characteristics that differentiate them from reports belonging to other root causes.

In order to achieve the above goal, we propose learning a linear transformation, i.e. a matrix A , that transforms the input probabilities (inaccurate $P(r/c)$) to accurate probabilities. Let the original (inaccurate) $P(r/c)$ vector be denoted by P_O and the accurate $P(r/c)$ vector be denoted by P_A . The transformation is done as follows:

$$P_A = A * P_O$$

We denote the total number of root causes by N . In the above equation, P_A and P_O are vectors of dimension N . P_O is equal to $[P_O(r|c_1), P_O(r|c_2), \dots, P_O(r|c_N)]^T$. P_A is equal to $[P_A(r|c_1), P_A(r|c_2), \dots, P_A(r|c_N)]^T$. The n^{th} entry in P_O is equal to $P(r/c_n)$ where c_n is the n^{th} root cause.

A is an $N*N$ matrix. We further require that all entries in A should be non-negative. This requirement ensures that each entry in P_A will be non-negative, since we know that each entry in P_O represents a probability and is therefore non-negative.

Intuitively, we try to learn the best *representative* for diagnosis reports generated by each root cause c . We believe that a *representative* of all diagnosis reports of one root cause c is more similar to diagnosis reports generated by same root cause c than the diagnosis reports generated by the other root causes. This belief is based on an assumption that defect behavior and diagnosis behavior of the same root causes under the same diagnosis procedure should hold some underlying characteristics. The learned *representative* captures the characteristics of diagnosis behavior from training data. The representative vector of the n^{th} root cause is denoted as A_n , the n^{th} row of the A matrix. Each representative vector provides a “typical” diagnosis report triggered by the n^{th} root cause. We define the accurate $P(r/c)$ based on how *similar* the report in question is to the

representative for root cause c . Each report in question is an input data point represented by its probability $P(r/c)$ vector P_o .

We measure similarity as the dot product or inner product [76] of two vectors (P_o and A_n). This similarity shows the likelihood that a given report is generated by the n^{th} root cause.

$$P_o \cdot A_n = \sum_{i=1}^N P_o(r|c_i) * A_{ni} = P_o(r|c_1) * A_{n1} + P_o(r|c_2) * A_{n2} + \dots + P_o(r|c_N) * A_{nN}$$

The higher the dot product, the more similar that report is to n^{th} root-cause representative A_n , and the more likely its real root cause is the n^{th} root cause c_n . So, we define this value as $P_A(c_n|r)$, the probability that root cause c_n is the true root cause given a report. By definition of conditional probability, $P_A(c_n|r)$ can be expressed as follows:

$$P_A(r|c_n) = \frac{P_A(c_n|r) * P(r)}{P_A(c_n)}$$

In the current RCD model, we do not have an assigned prior probability of seeing each root cause $P_A(c_n)$. We also assumed that the probability of seeing each report $P(r)$ is the same. Therefore, the following approximation is established:

$$P_A(r|c) \propto P_A(c|r)$$

Strictly speaking, each entry $P_A(r|c)$, in the vector P_A is a proxy for the accurate value of $P(r/c)$. The entry might be different from the true value of $P(r/c)$; e.g., it might be larger than 1. However, each entry in P_A will be proportional to the true $P(r/c)$. For the purpose of the RCD algorithm, we need to know the probability $P(r/c)$ up to a constant factor only. For the rest of the paper, to make the exposition simple, we will assume that $P_A(r|c)$ is equal to $P(r/c)$.

An alternative interpretation exists of A matrix. Each row of A, A_n is a set of weights applied on feature vectors of input points. For each i^{th} feature, weight A_{ni} indicates how important that feature is to the estimated $P(r/c)$ of n^{th} root cause. In our current

approach, original $P(r/c)$ is used as a feature. We know the value of original $P(r/c)$ is not accurate. One example of inaccuracy is that $P(r|interconnect\ open)$ is biased toward being higher than true when the real root cause is cell-internal. This bias results in interconnect open root causes dominating cell-internal root causes. By applying appropriate weights learned from training data on both $P(r|interconnect\ open)$ and $P(r|cell\ internal)$, we can boost probability of cell-internal while reducing the biased probability of interconnect open.

For the training data, we know the injected root cause. Therefore, in order to achieve our goal of learning an accurate estimate of $P(r/c)$, we want to learn a matrix A , such that for each report: i) the probability for the injected root cause, $P_A(r|c_{injected})$, is as high as possible; ii) the probability, $P_A(r|c)$, for each of the other root causes is as low as possible. This second condition is very important. Without this condition, every entry in the matrix A will be infinity. The second condition is what constrains the matrix A , and leads to meaningful learning. In our approach, we formulate an objective function that expresses these conditions, and we try to find a matrix that maximizes the objective function. We describe our objective function in detail in the next section.

6.4.3 Objective Function

Let us denote $P_O(r|c)$ for the i^{th} diagnosis report, and the n^{th} root cause by $P_O(r_i|c_n)$. For the i^{th} diagnosis report, the vector P_O defined above is therefore equal to $[P_O(r_i|c_1), P_O(r_i|c_2), \dots, P_O(r_i|c_N)]^T$.

In the training data set, each diagnosis report is labeled by the root cause of its injected defect. For each diagnosis report r , this root cause is denoted as $c_{injected}$. As stated in the previous section, our goal is to make, for each report, $P_A(r|c_{injected})$ as high as possible, and $P_A(r|c)$ as low as possible for all the other root causes. We can achieve both goals by maximizing the ratio of $P_A(r|c_{injected})$ and the sum of $P_A(r|c)$ for all root causes c . Note that maximizing this ratio for reports belonging to one root cause can make this

ratio bad for reports belonging to some other root cause. Therefore, we need to consider all the diagnosis reports belonging to the various injected root-cause populations simultaneously.

One possible objective function is to simply sum up this ratio for all diagnosis reports and maximize it. However, this is not a good choice. Consider two reports whose ratio is $R1$ and $R2$. $R1=1$ and $R2=0$ is better in terms of objective function value (since it has a higher sum) than $R1=0.5$ and $R2=0.49$. For our application, we want the right root cause to be dominant for as many reports as possible. Therefore, a much better choice is the product of the ratio for all reports, as expressed by $f(A')$ below.

$$\begin{aligned} f(A') &= \prod_{n=1}^N \prod_{i=1}^{S_n} \frac{P_A(r_{ni}|c_n)}{\sum_{m=1}^N P_A(r_{ni}|c_m)} \\ &= \prod_{n=1}^N \prod_{i=1}^{S_n} \frac{\sum_{j=1}^N A_{nj} \cdot P_O(r_{ni}|c_j)}{\sum_{m=1}^N \sum_{j=1}^N A_{mj} \cdot P_O(r_{ni}|c_j)} \end{aligned}$$

However, another factor that needs to be considered is training data size. If the numbers of diagnosis reports for different labeled root causes differ too much, it is possible that the matrix we learn is biased towards the root cause with a larger number of diagnosis reports. One way to solve this is to make sure that the number of diagnosis reports for each root cause are exactly the same. However, it is possible that for some root causes we have more training data. Having this restriction would mean that we need to discard some reports from the training data set. To alleviate this concern, and to give each root cause an equal emphasis, we assign a weight to each diagnosis report. The weight is inversely proportional to the number of reports for a given root cause. Note that all the diagnosis reports belonging to the same injected root cause have the same weight. We denote this as w_n for reports belonging to the n^{th} root cause.

$$w_n = \frac{1}{S_n}$$

where S_n is the number of diagnosis reports in the training data set for the n^{th} root cause.

Putting this all together, our objective function $f(A)$ takes the following form:

$$\begin{aligned} f(A) &= \prod_{n=1}^N \prod_{i=1}^{S_n} \left(\frac{P_A(r_{ni}|c_n)}{\sum_{m=1}^N P_A(r_{ni}|c_m)} \right)^{w_n} \\ &= \prod_{n=1}^N \prod_{i=1}^{S_n} \left(\frac{\sum_{j=1}^N A_{nj} \cdot P_O(r_{ni}|c_j)}{\sum_{m=1}^N \sum_{j=1}^N A_{mj} \cdot P_O(r_{ni}|c_j)} \right)^{w_n} \end{aligned}$$

where the first product goes over all the root causes, and the second product goes over all the diagnosis reports present in population S_n . r_{ni} is the i^{th} diagnosis report in population S_n . A_{nj} is the entry in the j^{th} column of the n^{th} row of matrix A .

$f(A)$ denotes the value of the objective function evaluated at the matrix A . In the training phase, our goal is to find the matrix A that maximizes $f(A)$.

With the learnt A matrix, the transformed output P_A of our supervised learning approach is a more accurate estimate of $P(r/c)$, and this feeds into the RCD algorithm. The objective function in RCD unsupervised learning is the likelihood of diagnosis reports, which is the product of the probability of each report. Thus, our choice of objective function also bears resemblance to the objective function for RCD.

To maximize the objective function, we use a very popular off-the-shelf optimizer using the Limited-memory BFGS (L-BFGS-B) algorithm [84].

6.5 Experiment Setup

Four industrial designs are used to validate our methodology. Designs B, C, and D are for advanced technology nodes using FinFETs. For each design, the number of predefined root causes varies (Table 6-2). In this paper, we sample 200 defects for each root cause for the training data set, and another 200 defects for the test data set. We choose the number 200. Ideally, we want to have as many training data as possible. However, creating a large amount of training data involves expensive simulation. Therefore, it is necessary to find a data size that balances the tradeoff between practical cost and effectiveness. The effectiveness of using 200 injected defects as training data is validated

by our experiment results. It shows a sample of this size captures most of the variability in terms of root-cause behavior and diagnostic tool behavior for the root cause.

Table 6-2 Number of predefined root causes per design

Design	A	B	C	D
number of interconnect root causes	13	34	23	17
number of cell-internal root causes	9	14	24	19

For each design, training data and testing data are created as described in Section 6.4. Recall that we create a population of diagnosis reports for each root cause for training, and a separate population for testing. For training, we combine the populations across different root causes, since we are learning a single matrix and all the diagnosis reports are part of the objective function. For testing, we keep each population of reports corresponding to a separate injected root cause. Thus, we have as many test cases as the number of root causes. The root-cause distribution for each testing population is, by construction 100%, for the injected root cause. During testing, we first apply the learnt A matrix on the vector P_O . Recall that vector P_O consists of $P(r/c)$ derived from the diagnosis reports. We then obtain the vector P_A , which is a more accurate estimate for $P(r/c)$. The collection of vectors P_A , corresponding to each report in the population, is then fed to the RCD algorithm to determine the root-cause distribution.

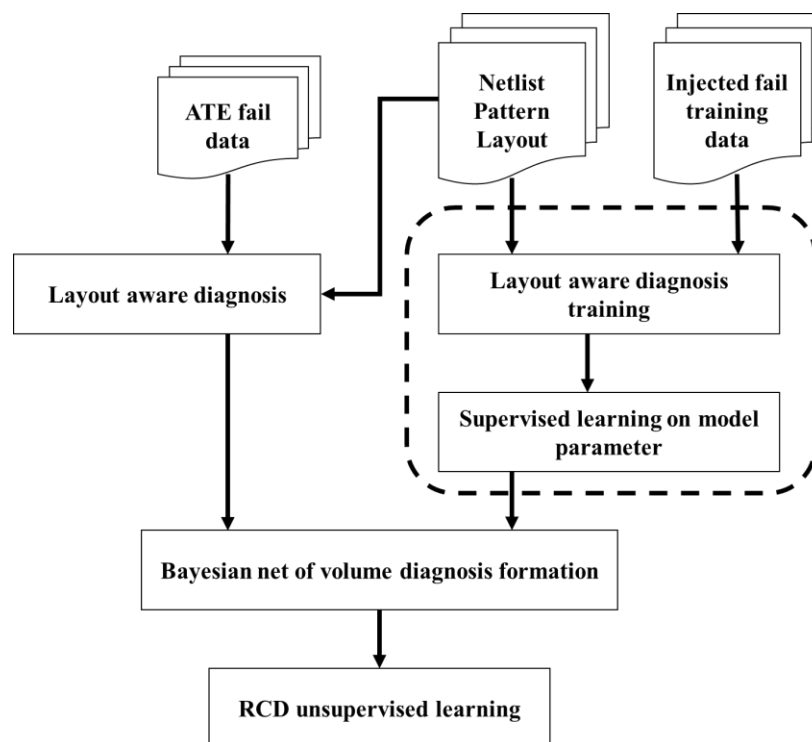


Figure 6-1 Flow chart of RCD using supervised learning for parameter estimation

The flow chart of RCD using model parameter estimated by supervised learning is shown in Figure 6-1. The only difference is the dash block. We replace the root-cause information extraction step in Figure 3-3 with a supervised learning step. In the supervised learning step, training data are created separately. The output of supervised learning block will be feed into Bayesian model-building as an input, and unsupervised RCD learning will then be applied based on that.

In the following discussion, the root-cause distribution obtained based on the original $P(r/c)$ from the diagnosis reports is tagged as baseline. The root-cause distribution obtained by using the corrected values of $P(r/c)$ based on our approach is denoted as the new method. Root-cause distribution obtained by RCD is termed the RCD result. We have a test case for each injected root cause in the design. For each given test case, we check the

probability of the injected root cause in the RCD result and define it as the accuracy of that test case.

6.6 Experiment Result

Success rate of proposed method and baseline method for cell internal cases and interconnect root causes are shown in Table 6-3 and Table 6-4. Success rate are calculated by averaging accuracy of all test cases in that design. For example in Table 6-3, for baseline method, average accuracy of all 13 test cases for cell internal root causes is 46%. For the baseline method, the success rate for cell-internal root causes is much lower than that for interconnect root causes. Our method substantially improves the success rate for cell-internal root causes for all four designs, and the success rate for interconnect root causes are comparable with baseline method.

Table 6-3 Success rates for cell internal root causes

Design	A	B	C	D
Baseline	46%	20%	54%	29%
New method	87%	78%	77%	89%

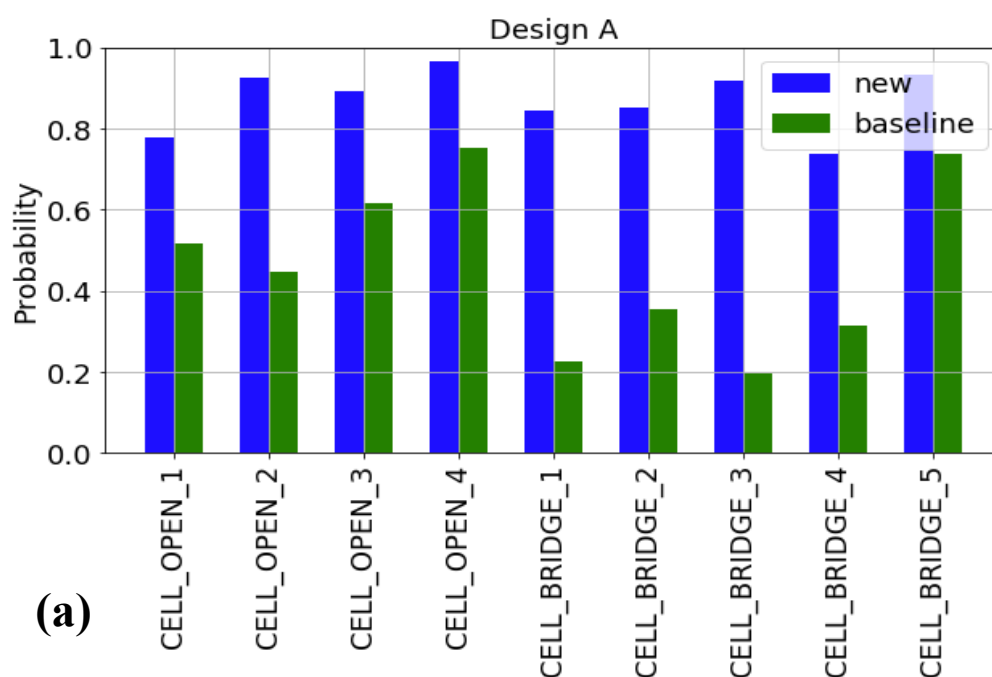
Table 6-4 Success rates for interconnect root causes

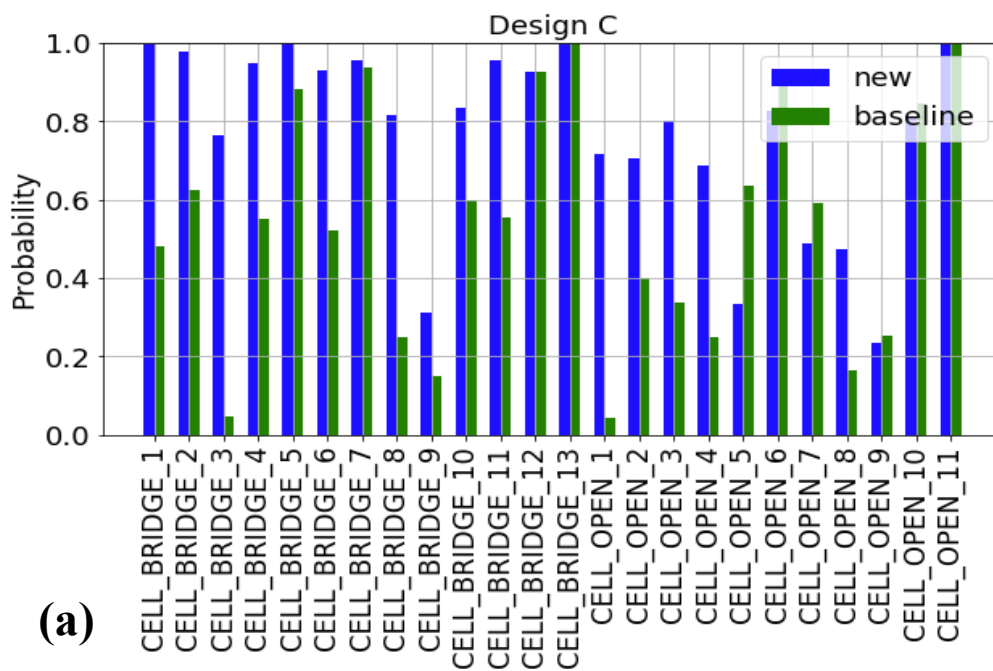
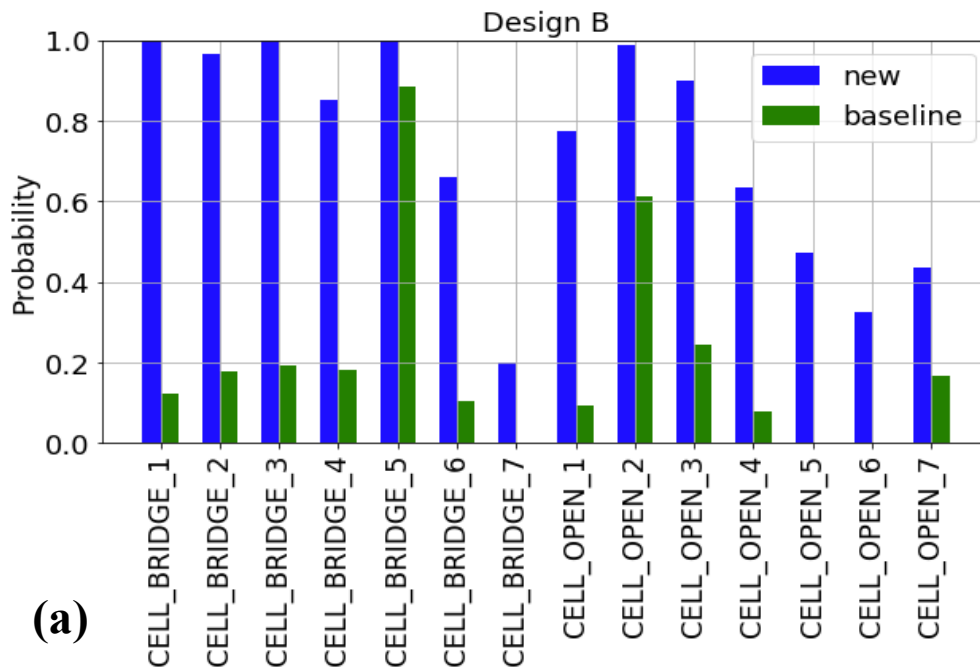
Design	A	B	C	D
Baseline	80%	97%	92%	86%
New method	83%	97%	90%	84%

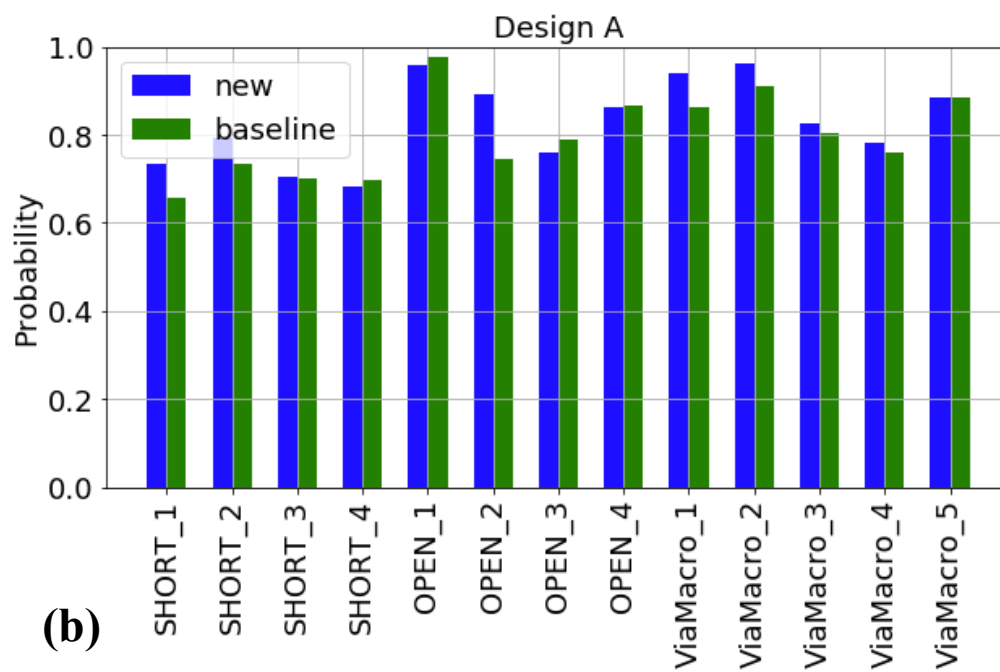
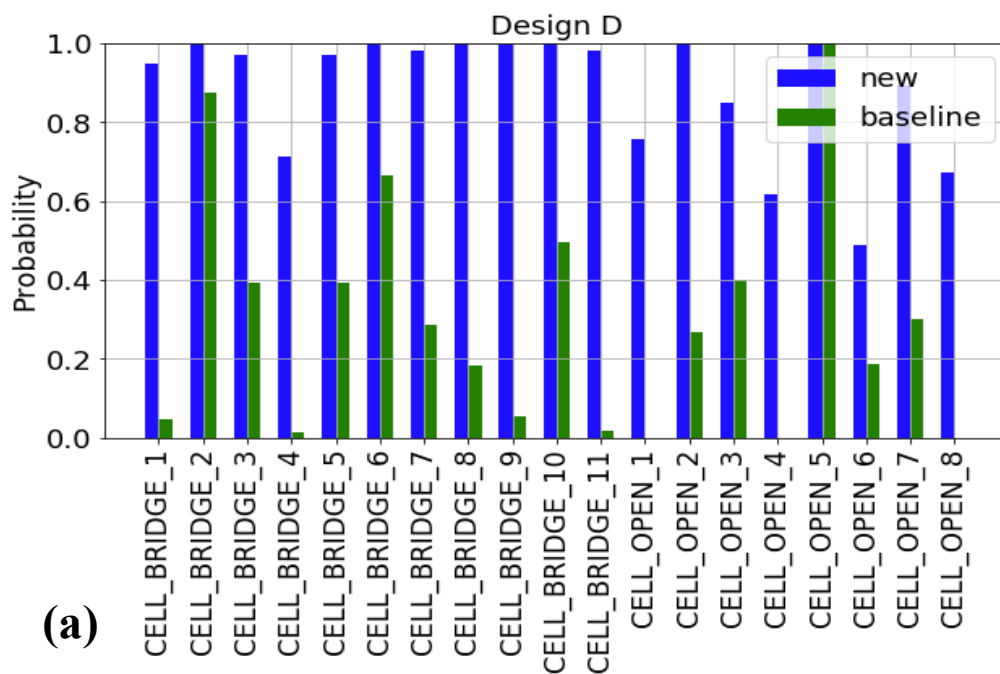
We present detailed results regarding cell-internal root causes in Figure 6-2a. Each set of bars represents a test case created by injecting defects for the root cause shown at the base of the bar. The height of the bar represents the probability of seeing the injected root

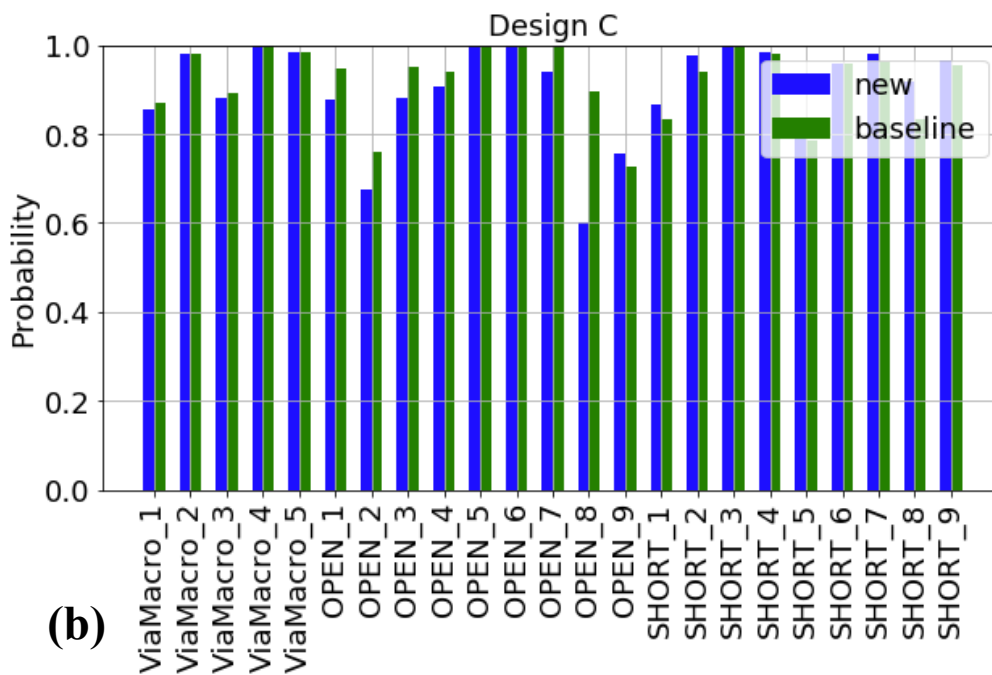
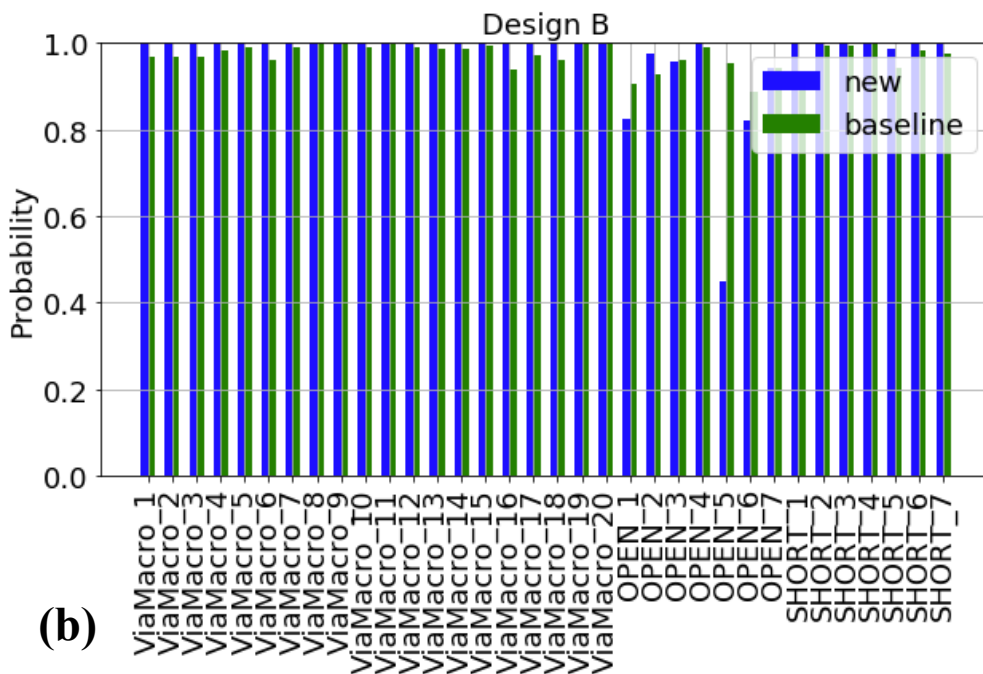
cause in the root-cause distribution output by RCD (RCD pareto). For example, let us consider the test case where we inject the root cause Cell_Bridge_1 in Design B. If we use the baseline method, in the RCD pareto Cell_Bridge_1 accounts for only about 10% of the probability. If we use the new method that we propose, Cell_Bridge_1 is the only root cause in the RCD pareto, and accounts for 100% of the probability. For these injected cases, the higher the bars are, the better it is. As shown, accuracy of cell-internal root causes have been greatly improved, except some cases in design C.

Detailed results for interconnect root causes are presented in Figure 6-2b. Even given the already high accuracy of RCD on interconnect root causes using the baseline method, improvements can still be achieved by the proposed method in some cases.









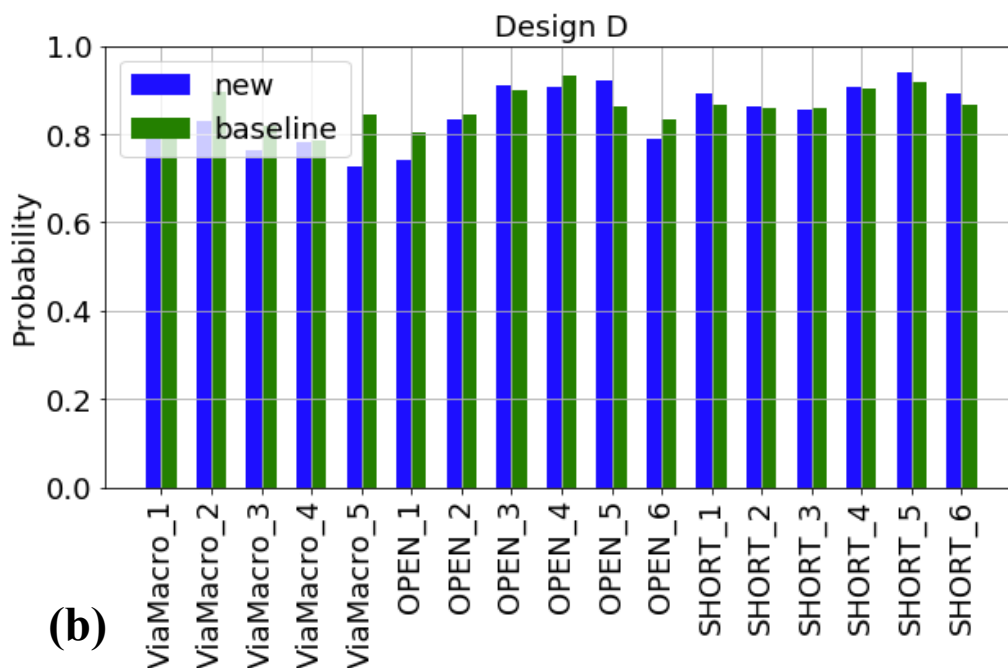


Figure 6-2 (a) Accuracy of cell-internal root causes; (b) Accuracy of interconnect root causes

In the following discussion, we will explain why the proposed method works well, and the reason its accuracy is lower for some cases.

Why does the approach work so well for cell internal root causes? We injected a cell-internal open root cause and generated a number of diagnosis reports for it using the approach detailed in the previous section. Figure 6-3a plots the $P(r/c)$ for the injected cell-internal root cause (in blue), the lowest interconnect metal layer open (in green), and the lowest interconnect via layer (in red). The diagnosis reports are sorted by $P(r/c)$ for the injected root cause. Ideally, we would like the $P(r/c)$ for the injected root cause to be very high, and the $P(r/c)$ to be very low for all other root causes. However, we see that with the original $P(r/c)$, for almost all reports, the $P(r/c)$ for the interconnect open and via layer is higher than the $P(r/c)$ for the real root cause. Given this behavior, it is no surprise that RCD results show a bias towards interconnect root causes, even when the injected root cause is cell-internal. One reason for such bias is that interconnect open defects are more

complicated to model than bridge defects. Diagnosis algorithm has looser criteria to call out an open fault. Therefore, $P(r/f)$ of a report produced by a non-interconnect-open defect will be more inaccurate because a fake interconnect open fault is more likely to be called out by diagnosis. This observation leads to the conclusion that when a cell-internal root cause is injected, $P(r/c)$ of interconnect open root cause is also very strong. In Figure 6-3b we show the value of $P(r/c)$ for the same root causes, after transformation using our learnt A matrix. We see that now $P(r/c)$ of injected root cause is higher for almost all diagnosis reports. As a result, RCD is able to easily identify the correct root cause.

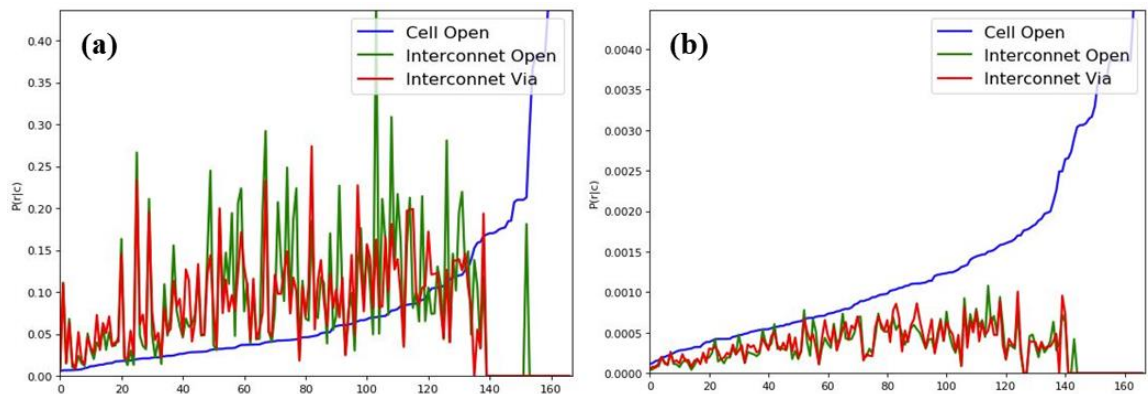


Figure 6-3 (a) Original $P(r/c)$ per report; (b) $P(r/c)$ per report after transformation by learnt A

Why does the approach work poorly for some cell-internal cases? As shown in Figure 6-2a, the proposed method produces worse results than the baseline for Cell_Open_5, Cell_Open_6 and Cell_Open_7 in design C. We will use Cell open 5 as example to explain the reason.

For Cell open-5, though the baseline result is good, it is biased by inaccurate $P(r/c)$. As mentioned in Table 6-1, a cell-internal root cause could have high co-occurrence with another root cause or root cause group and be dominated by other root causes. We can call

them a pair or group of co-occurrence and dominating root causes. RCD results for such a group would always be biased to the dominating root cause. For example, Cell_Open_1 and Cell_Open_5 always occur together in the population when either of them is the true root cause. The original $P(r/c)$ of Cell_Open_5 is slightly higher than the original $P(r/c)$ of Cell_Open_1 in every report of either Cell_Open_1 or Cell_Open_5. Therefore, the baseline result of Cell_Open_5 has high accuracy. The result of Cell_Open_1 is bad because it is dominated by not only Cell_Open_5 but also by other root causes. Based on original $P(r/c)$, these two root causes cannot be distinguished with a limited sample size. However, assigning high probability to a dominant root cause such as Cell_Open_5 is misleading for understanding the underlying distribution. Instead of doing that, the proposed method tries to distinguish such root causes by using the $P(r/c)$ of all the other root causes. It learns a representative of Cell_Open_1 that can have high similarity with reports of Cell_Open_1. The proposed method boosts Cell_Open_1. Since Cell_Open_1 and Cell_Open_5 have high co-occurrence, a representative of Cell_Open_5 would also have some similarity with report of Cell_Open_1. Therefore, the proposed $P(r/c)$ of Cell_Open_1 for Cell_Open_5 report is boosted. The accuracy of the RCD result for Cell_Open_5 decreases accordingly.

Generally, the objective function of our approach is to optimize a matrix that is a set of reports representative of all root causes. The goal of optimization is make all $P(r/c)$ for correct root causes high, and those for wrong root causes low. If the features (inaccurate original $P(r/c)$) do not provide enough distinguishable information, input reports from two different root causes may be quite similar. Accordingly, representatives of one of the two similar root causes would have some similarity to the other root cause. To achieve the goal of objective function, the learned representative needs to consider the trade-off between getting higher $P(r/c)$ when it is the true root cause and lower $P(r/c)$ when it is not. Ultimately, it is possible that the value of learned $P(r/c)$ of these two root cause will be close, and the probability of true root causes in RCD results will be shared among those

two. Furthermore, such similar behaviors of different root causes do not necessarily occur only between two root causes, but might occur in a group of root causes.

Overall, there is a need to improve our techniques for distinguishing such similar root-cause group. Such improvement can be achieved by increasing features, using a non-linear classifier and increasing sample size.

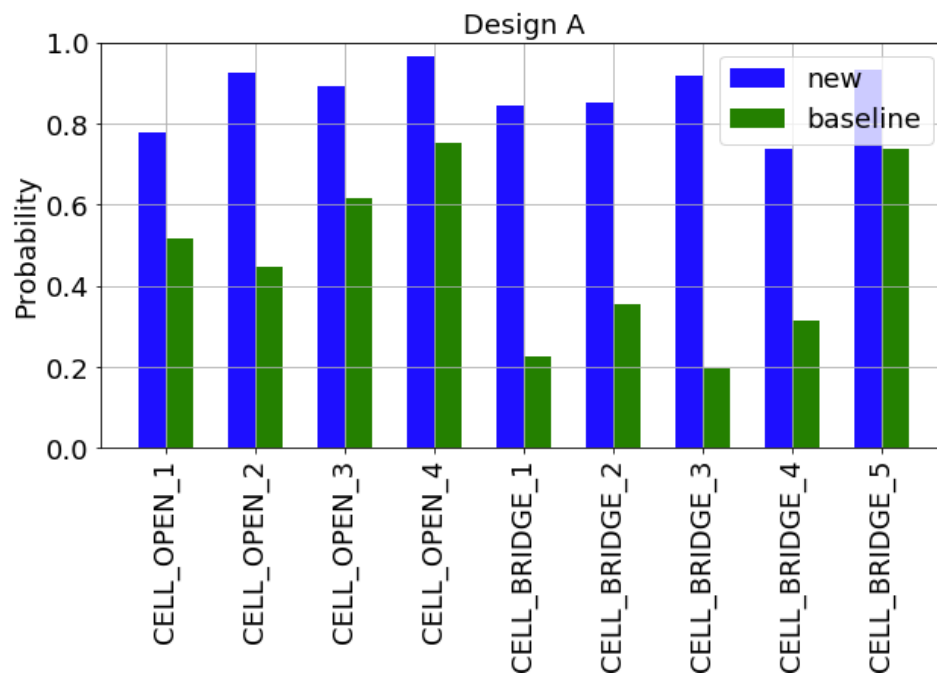
Why the results are for some interconnect open cases worse using the proposed method? As shown in Figure 6-2b, results of interconnect open and via on root cause are worsen in some case with proposed method. One example is Open_8 in design C. As shown in Figure 6-2b, the accuracy of the new method (blue bar) drops 33% compared to the accuracy of the baseline method (green bar). In the distribution obtained by the new method, 31% probability is assigned to 5 different cell-internal root causes. By contrast, distribution obtained by baseline assigns 0% probability to any cell-internal root cause. Among those cell-internal root causes, Cell_Bridge_3 takes 14% probability. Looking at Figure 6-2a, one can see that the accuracy of Cell_Bridge_3 increases from 5% to 77% when we use the proposed method. In baseline distribution of Cell_Bridge_3, Open_8 is assigned 46% probability. This result shows how inaccurate the original $P(r/c)$ of Open_8 in the population of Cell_Bridge_3. This also fits the analysis earlier in Section 6.2 as to why $P(r/c)$ is not accurate. Open defects are harder to model compared to bridge defects. Therefore, the current diagnosis algorithm has looser constraints on including open faults in reports. This leads to a less accurate $P(r/f)$ when the real root cause is not open on interconnect. A learned representative of open root cause might show more similarity to reports of non-open root cause, compared to the other way around. The estimated $P(r/c)$ of open root cause is downgraded, and the accuracy of RCD results for interconnect open cases decreases accordingly.

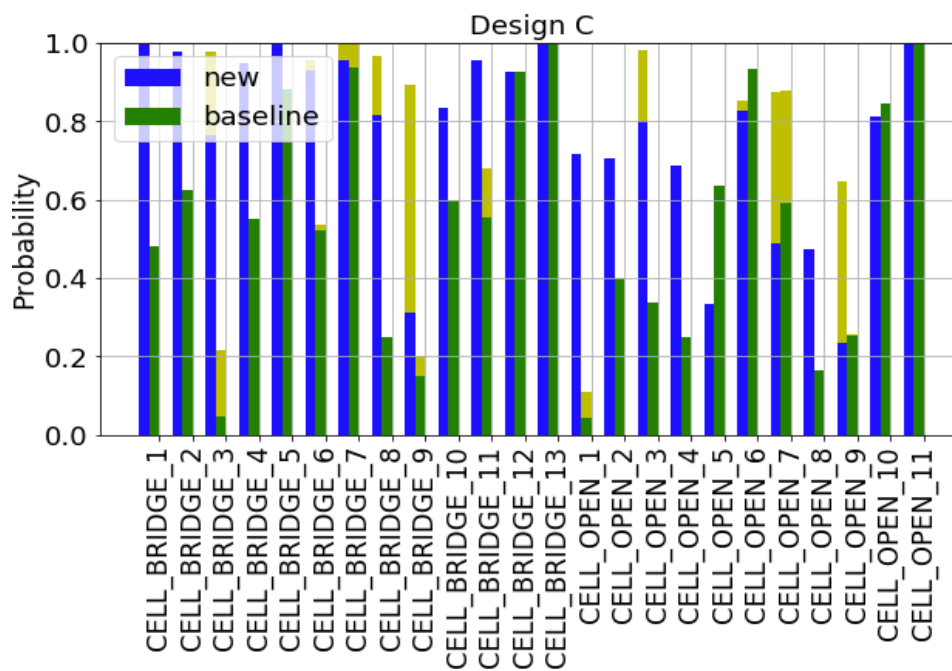
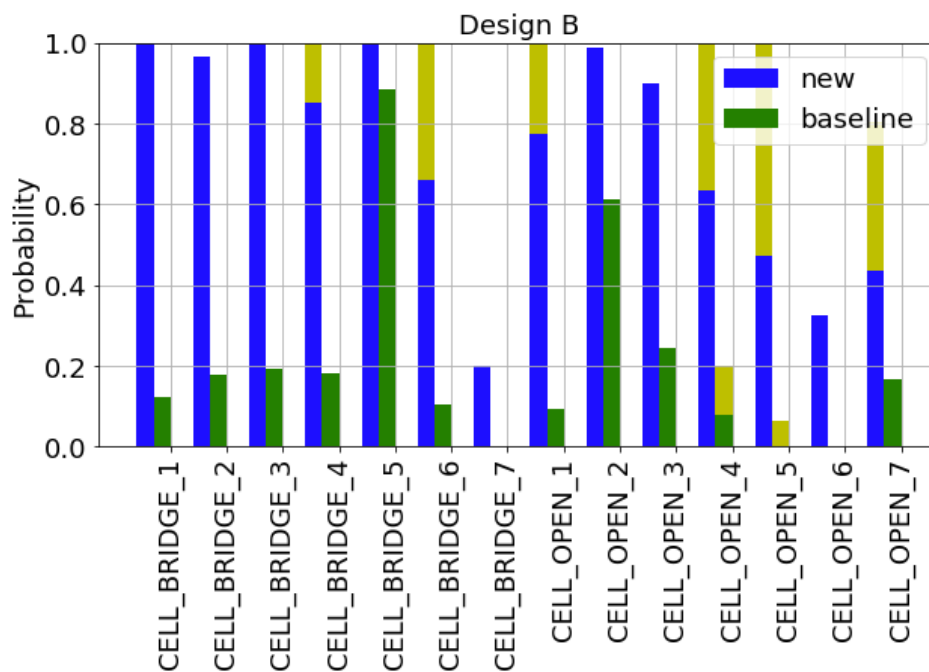
An Optional measurement: To improve yield, yield engineers need to identify the root cause of systematic defects in a short turnaround time. RCD results from volume diagnosis assign probability to each possible root cause. The probability provides

information as to how much each root cause contributes to fail the population. Such information gives guidance to failure-analysis experts in selecting the most likely candidate root causes for further analysis. Then, based on the picked candidate root cause, the probability of each die being defective due to that root cause is calculated. A most likely die will be selected for PFA. Therefore, a root cause with higher probability will provide more confidence and clearer information in this decision process.

For cell-internal root cause, root causes on adjacent layers usually have high co-occurrence. This means the diagnosis report will always call out layers next to the layer where real defect occurs. Statistically, such pairs or group of root causes behave quite similarly and are difficult to be distinguished. Rather than assigning probability only to real root cause, RCD gives probability to other root causes that are highly correlated to real one. For the above reason, we saw that probability of true root cause might not be dominant in the distribution. One example is a distribution of 30% layer 1, 30% layer 2 and 40% layer 3. The correct root cause is layer 1. Layer 2 is its adjacent layer. If we look at probability of only a single layer, layer 3 is the mostly likely root cause. If we go directly to layer 3, the defect will not be found. Since we know adjacent layers of cell-internal root causes are frequently called out together and share the probability in distribution of RCD, we can combine the probability for layer 1 and layer 2 and get 60% probability for layer 1 and layer 2. This means it is more likely that either layer 1 or layer 2 are defective than that only layer 3 is defective. This information gives yield experts the option of considering that the real root cause could be either layer 1 or layer 2. With that being said, the goal of RCD is still to accurately pinpoint each single true layer by assigning a probability reflecting the underlying distribution. The effectiveness of our proposed method needs to be and has been validated by the accuracy of RCD result considering only a single layer. However, adjacent-layer measurement is an optional loose measurement that could be useful for PFA with lower resolution.

For a few root causes, such as Cell_Bridge_9 in design C, we see that although the supervised learning approach performs better than baseline method, it is still have a relative low accuracy ($< 40\%$). This means in the RCD result, it might not be the dominating root cause. It turns out that in addition to Cell_Bridge_9, we call out another layer, while preserving the behavior (i.e. we correctly identify it as a bridge). Both layers are next to each other in the process stack and almost always occur together in the diagnosis report. As long as we identify the right fail mode (or failing behavior) on the layer close to the real defect, such a result is still a meaningful signal that can guide failure analysis and yield improvement. Therefore, we use “adjacent-layer success rate” as a less precise but more tolerant measurement showing the improvement of proposed method over baseline. We show the contribution of adjacent layers in Figure 6-4 by stacking them in yellow on top of the bars for the injected root cause. Results show that with less precise measurement, the proposed method is still much better than the baseline method.





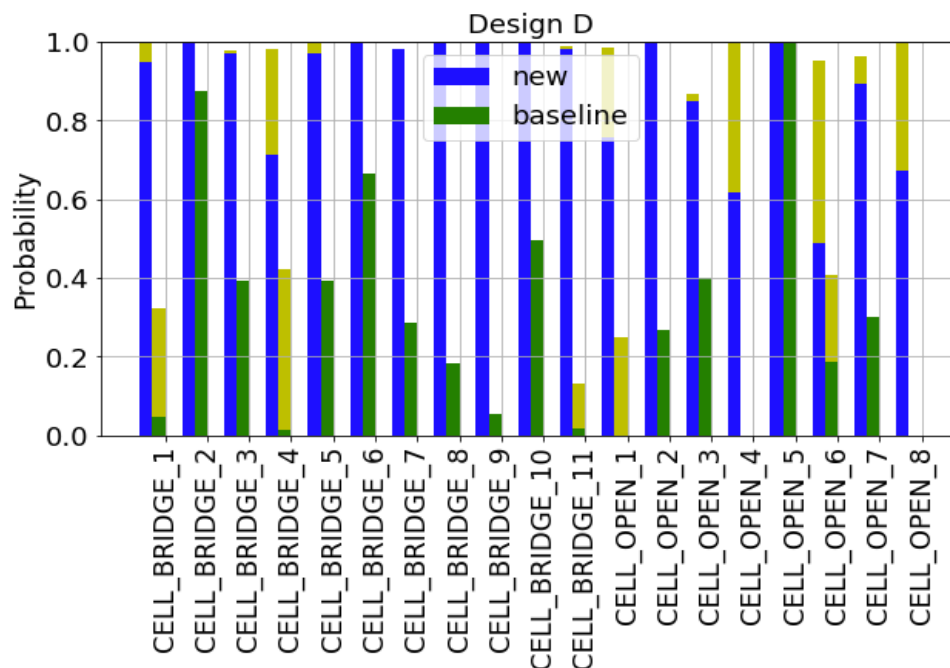


Figure 6-4 Accuracy of cell-internal root causes with adjacent layer measurement

6.7 Result on Silicon Data

In this section, we used the silicon data of Design D to validate the proposed method. We applied the method on silicon fail data and validated result of RCD analysis by PFA results. Overall, we achieved a 71% hit rate among 21 PFA cases using the new method. Using the baseline method (RCD without learning), a 42% hit rate is obtained.

Before describing the validation experiment on silicon data, we discuss how RCD is used in volume diagnosis based root cause identification procedure. Given a set of volume diagnosis reports of failing dies, yield experts choose a target root cause they are most interested in to investigate. RCD estimates a root cause probability distribution of given volume diagnosis reports. This distribution assigns probability to each possible root cause. The probability provides information as to how much each root cause contributes to failing population. Such information gives guidance to failure-analysis experts in selecting the most likely candidate root causes for further analysis. Other than selecting the top root

cause in the probability distribution, sometimes yield expert will instead select second highest root cause if the top one has been confirmed as a known root cause. Sometimes, they have information of a specific root cause from other tests and select it as a candidate root cause. After the candidate root cause is selected, the probability of each die being defective due to that root cause is calculated based on RCD result. Dies with highest root cause probability for the target root cause will be selected for PFA.

In our validation experiment, we used 21 dies which have confirmed root cause information from PFA. These dies are from 3 different wafers of design D. They are selected following the above general criteria. To compare the effectiveness of baseline method and the new method, we use the hit rates as a measurement.

The definition of a hit is given next. Firstly, root cause distribution of a wafer is estimated by RCD using diagnosis reports of failing dies of a wafer. In our experiment, the number of failing dies for volume diagnosis were 226, 222 and 1805 for wafers A, B and C, respectively. Then, based on the root cause distribution and $P(r/c)$ of each die, probability of a die being defective due to each root cause is calculated. For each die, a root cause is defined as RCD-predicted root cause if it has the top probability to cause the defect on that die. If the RCD-predicted root cause causes the failure confirmed by PFA, we call it a hit. Hit rate is the ratio of hit over all 21 dies used in the experiment. Thus, the larger the hit rate of a method, the more effective that method is.

In the above calculation of hit rates, baseline method uses original $P(r/c)$ and the proposed method uses the new $P(r/c)$. As presented in Section 6.6, an **A** matrix learned is from 200 training data for design D. For each die, new $P(r/c)$ of 17 interconnect root causes and 19 cell internal root causes for each of the defective die are calculated using learned **A** matrix.

As shown in Table 6-5, for wafer B, 10 defective dies are provided. Root cause of die 9 (Die id =9) confirmed by PFA is Cell Short contact. Root cause predicted by new

method is Cell short contact but by baseline method it is interconnect Open 1. Therefore, it is counted as a hit for the new method and a miss for the baseline method.

Table 6-5 Comparison of RCD top root cause vs. PFA results. W id stands for Wafer id. Interconnect is abbreviated to Inter.

Die id	W id	RCD root cause (New)	RCD root cause (Baseline)	PFA root cause	Result (New)	Result (baseline)
1	A	Cell Open 1	Cell Open 1	Cell Open 1	Hit	Hit
2	A	Cell Open 1	Cell Open 1	Cell Open 1	Hit	Hit
3	A	Inter. Short 2	Inter. Short 2	Inter. Short 2	Hit	Hit
4	A	Cell Open 3	Inter. Via 1	Cell Open 1	Miss	Miss
5	B	Inter. Short 1	Inter. Short 1	Cell Short contact	Miss	Miss
6	B	Inter. Open 1	Inter. Open 1	Cell Short contact	Miss	Miss
7	B	Cell Short 3	Inter. Open 1	Low level defect	Miss	Miss
8	B	Inter. Open 4	Inter. Via 3	Inter. Open 4/Via3	Hit	Hit
9	B	Cell Short contact	Inter. Open 1	Cell Short contact	Hit	Miss
10	B	Inter. Open 1	Inter. Open 1	Cell Short contact	Miss	Miss
11	B	Cell/Short contact	Cell Short contact	Cell Short contact	Hit	Hit
12	B	Inter. Open 1	Inter. Open 1	Cell Short contact	Miss	Miss
13	B	Cell Short contact	Cell Short contact	Cell Short contact	Hit	Hit
14	B	Inter. Open 3	Inter. Open 3	Inter. Open 3	Hit	Hit
15	C	Cell Open 2	Inter. Via 1	Cell Open 4	Hit	Miss
16	C	Cell Open 2	Inter. Via 1	Cell Open 4	Hit	Miss
17	C	Inter. Short 2	Inter. Short 2	Inter. Short 2	Hit	Hit
18	C	Cell Open 2	Inter. Via 1	Cell Open 4	Hit	Miss
19	C	Cell Open 2	Inter. Via 2	Cell Open 4	Hit	Miss
20	C	Cell Open 2	Inter. Via 1	Cell Open 4	Hit	Miss
21	C	Inter. Short 2	Inter. Short 2	Inter. Short 2	Hit	Hit

Figure 6-5 describes the RCD pareto for Wafer C. Blue bar represents the results of new method. Red bar represents the result of baseline result.

For new method, OPEN_1 and CELL_OPEN_2 are the top root causes in distribution. OPEN_1 had prior silicon confirmation and did not require further PFA. The CELL_OPEN_2 samples were identified by PFA as CELL_OPEN_4. On further investigation, cell-internal layer 2 and layer 4 are directly connected to each other and inseparable. Also, the failure is prominent only at the interface of layer 2 and layer 4; hence we considered them a match. The PFA finding for one CELL_OPEN_2 sample is shown in Figure 6-6 below.

For baseline method, VIA 1, OPEN 1 and VIA 2 are the top three root causes in distribution. Baseline RCD-predicted root causes of 5 out of 7 dies from wafer C are either VIA1 or VIA2. All 5 PFA cases are considered miss. This shows the improvement from the new method.

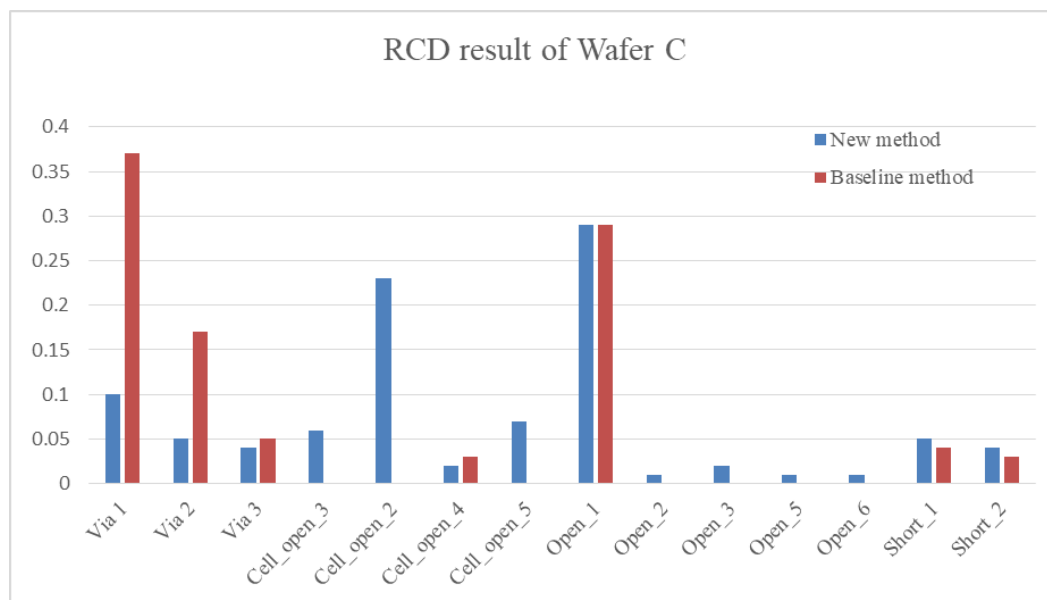


Figure 6-5 Root-cause distribution output by RCD for wafer C using the new method and baseline method

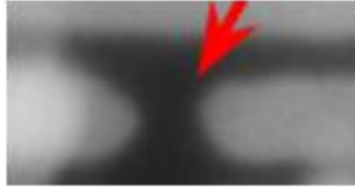


Figure 6-6 PFA result highlighting CELL_OPEN between layer 2 and layer 4

These silicon data results boost our confidence in the proposed supervised RCD methodology for yield learning and fast ramping up of advanced technology nodes.

6.8 Conclusion

In this chapter, we present a novel approach to root-cause identification using volume diagnosis. Our method relies on supervised machine learning to more accurately estimate the parameters for the Bayesian model used in RCD [48], [77]. We show that using our approach we are able to accurately identify both cell-internal and interconnect root causes. We present results on several industrial designs for advanced technology nodes, and also present results on real silicon that validate our approach and prove its effectiveness for root-cause identification.

CHAPTER VII

CONCLUSION

7.1 Conclusion

For yield improvement, identifying root causes of systematic defects in a short time is crucial. DDYA techniques such as RCD identify the common root cause by analyzing a large amount of volume diagnosis data, using much less time and resource than PFA. RCD estimates root cause probability distribution by learning from volume diagnosis data. Root cause probability distribution can also help reduce diagnosis ambiguity for single dies, and narrow the scope of the candidate defect sites for PFA. RCD consists of two parts: 1. Building a Bayesian net modeling the distribution of defects given each candidate root cause. The parameter of Bayesian model is estimated based on domain knowledge. 2. Unsupervised learning for most likely root cause distribution on volume diagnosis. MLE is used with EM to obtain the optimal distribution.

RCD has been proved to be effective for interconnect root causes [44], [48], [49]. In the context of advanced technologies, as the feature size of layout shrinks aggressively and process procedure of the library cell gets more complex, identifying the emerging new root causes with limited and complex diagnosis data becomes challenging. Those new root causes include, but are not limited to, layout pattern root cause (certain prone to fail layout shapes) and cell internal root causes. The relevant challenges are as follows: Firstly, how to model a new root cause and feed it into existing model and flow. Secondly, how to handle characteristics of new root causes, such a large number of types or high co-occurrence with each other. Lastly, to overcome these challenges to RCD, improvement based on better understanding of the RCD is desired.

To better understand and evaluate possible issues with the RCD model, Chapter 4 proposed a card game model to create controllable diagnosis data with various diversities. With complete information of diagnosis data created by the card game model, issues of

inaccurate Bayesian model parameters and limited sample size can be separated and evaluated individually. Various scenarios of diagnosis data are created. Given an accurate model parameter, the impact from limited sample size on RCD is evaluated. The proposed cross validation is shown to effectively alleviate overfitting brought about by limited sample size under various scenarios. Issues in volume diagnosis are further discussed using card game data as a reference.

Chapter 5 addresses the challenge of identifying layout pattern root cause in RCD by enhancing RCD flow with a step for automatic layout pattern extraction. Comparing to existing methods [51], [53], [54], RCD can filter out layout patterns that are unlikely to be defective, group shifting layout patterns, and handle equivalent layout patterns. All of above help reduce the chance of overfitting in results given the huge number of layout pattern types. The control experiment shows that RCD identifies injected layout pattern root cause successfully with high accuracy. For injected non-layout pattern root cause, RCD can identify them correctly, though some accuracy is sacrificed. Ultimately, the effectiveness of an enhanced RCD flow that identifies layout pattern is validated by PFA results of silicon data.

Chapter 6 shows the impact on a Bayesian model of RCD of including cell internal root causes. The estimated parameter of the Bayesian model need to be fixed to obtain an accurate RCD result. A supervised learning method is proposed to find a more accurate RCD result. With the new estimated parameter, the RCD result for injected cell internal root causes is improved greatly, while the result for injected interconnected root causes still retains high accuracy. Silicon PFA results from an industrial design confirm the effectiveness of the proposed method.

7.2 Future Work

The research work proposed in this thesis successfully addresses the challenge from layout pattern root cause and cell-internal root causes. These research points toward the

effectiveness of supervised learning for model parameter estimation, and provides a vehicle to generate diagnosis reports for future investigation. Further research on card game model and supervised learning flow can be extended with more capabilities.

Card game model: Currently we assume a simplified diagnosis scenario, in which there is single suspect and multiple root causes for each diagnosis report and all reports are correct. To mimic a more complex diagnosis scenario, one could make several changes to the card game setting. Multiple suspects per diagnosis report can be mapped to a card game by recording more than one card numbers. Wrong diagnosis reports can be mimicked by writing down a wrong card number. Other than that, certain assumptions can be made when creating candidate card decks for mapping the correlated root cause set and dominant root cause set. Furthermore, the impact of different fault models on diagnosis can be mapped as setting different probabilities of seeing a specific card drawn from a given deck for different types of card decks. Depending on which problem we want to investigate, the card game model can be quite flexible in creating diagnosis data, with few limitations on resources.

Supervised learning on parameter estimation: While we obtained good results for both cell internal and interconnect root causes, there are several avenues to explore to further improve our accuracy.

In the current work, we use a matrix to transform the input features and obtain a more accurate estimate of $P(r/c)$. This is a simple linear classifier. We could explore the benefit of using a non-linear classifier, or obtaining a non-linear classification boundary using techniques such as the kernel trick [85]. We could also consider popular classifiers used in machine learning, such as SVM, random forests etc., and explore how they can be adapted for our problem. One challenge is that classification is only the first step of our approach. The output of the classification step serves as the input to RCD, which results in a root cause distribution, which is our ultimate goal. So, any classifier that we employ must take the RCD objective function into account and produce estimates that are meaningful as inputs for RCD.

Currently, we only use the probability of a report given a root cause as features. We can easily extend our approach to consider other features in addition to the existing features. Using more features should result in a better classifier.

One of the drawbacks of our approach is that it requires training for each root cause. This makes it prohibitively expensive when we consider root causes such as layout patterns, which might number in the millions. One way to alleviate this problem is by considering a hierarchical classifier. In the first step, we could broadly classify root causes as layout patterns vs non-layout patterns, or into other groups that we wish to treat differently. With root causes classified in this way, after the first step, we would know the probability that a given die is failing due to layout patterns being a root cause and vice versa. In the second step, we could have different classifiers for each category. We could use an unsupervised learning algorithm for layout patterns, considering layout patterns only as root causes. For interconnect and cell internal root causes we could use the approach outlined in this thesis.

REFERENCES

- [1] W. Maly, A. Gattiker, T. Zanon, T. Vogels, R. D. Blanton, and T. Storey, "Deformations of IC structure in test and yield learning", in Proceedings of International Test Conference, pp. 856-856, 2003.
- [2] W. Maly and J. Deszczka, "Yield Estimation Model for VLSI Artwork Evaluation", Electronics Letters, vol. 19, no. 6, pp. 226-227, March 1983.
- [3] C. Hora, R. Segers, S. Eichenberger, and M. Lousberg, "On a statistical fault diagnosis approach enabling fast yield ramp-up", Journal of Electronic Testing, vol. 19, no. 4, pp. 369-376, 2003.
- [4] B. Kruseman, A. Majhi, C. Hora, S. Eichenberger, and J. Meirlevede, "Systematic Defects in Deep Sub-Micron Technologies", in Proceedings of International Test Conference, pp. 290-299, 2004.
- [5] B. Koenemann, "Design/Process Learning from Electrical Test", in Proceedings of International Conference on Computer Aided Design, pp. 687-694, 2004.
- [6] L. Milor, "A Survey of Yield Modeling and Yield Enhancement Methods", IEEE Transactions on Semiconductor Manufacturing, vol. 26, no. 2, pp. 196-213, 2013.
- [7] L. T. Wang, C. E. Stroud, and N. A. Toubia, "System-on-chip test architectures: nanometer design for testability", Morgan Kaufmann, 2010.
- [8] M. Abramovici, M. A. Breuer, and A. D. Friedman, "Digital systems testing and testable design (Vol. 2)", New York: Computer science press, 1990.
- [9] T. Bartenstein, D. Heaberlin, L. Huisman, and D. Sliwinski, "Diagnosing combinational logic designs using the single location at-a-time (SLAT) paradigm", in Proceedings of International Test Conference, pp. 287-296, 2001.
- [10] J. A. Waicukauski and E. Lindbloom, "Failure diagnosis of structured VLSI", IEEE Design & Test of computers, vol. 6, no. 4, pp. 49-60, 1989.
- [11] S. Venkataraman and S. Drummonds, "POIROT: A logic fault diagnosis tool and its applications", in Proceedings of International Test Conference, pp. 253-262, 2000.
- [12] Y. Huang, W. T. Cheng, and F. M. Kuo, "Scan Chain Diagnosis by Adaptive Signal Profiling with Manufacturing ATPG Patterns", in Proceedings of Asian Test Symposium, pp. 35-40, 2009.
- [13] W. T. Cheng and F. M. Kuo, "Embedded Tutorial Summary: Diagnosis for Accelerating Yield and Failure Analysis", in Proceedings of Asian Test Symposium, pp. 271, 2012.
- [14] Y. J. Chang, M. T. Pang, M. Brennan, A. Man, M. Keim, G. Eide, and T. P. Tai, "Experiences with Layout-Aware Diagnosis—A Case Study", Electronic Device Failure Analysis, vol. 12, no. 12, pp. 12-18, 2010.

- [15] R. Desineni, O. Poku, and R. D. Blanton, "A Logic Diagnosis Methodology for Improved Localization and Extraction of Accurate Defect Behavior", in Proceedings of International Test Conference, pp. 1-10, 2006.
- [16] M. Sharma, S. Schwarz, J. Schmerberg, K. Yang, T. P. Tai, Y. S. Chen, C. Y. Chuang, and F. M. Kuo, "Layout-aware Diagnosis Leads to Efficient and Effective Physical Failure Analysis", In Proceedings of the 37th International Symposium for Testing and Failure Analysis, p. 86-90, 2011.
- [17] R. D. Blanton, J. T. Chen, R. Desineni, K. N. Dwarakanath, W. Maly, and T. J. Vogels, "Fault tuples in diagnosis of deep-submicron circuits", in Proceedings of International Test Conference, pp. 233-241, 2002.
- [18] M. E. Amyeen, D. Nayak, and S. Venkataraman, "Improving precision using mixed-level fault diagnosis", in Proceedings of International Test Conference, pp. 1-10, 2006.
- [19] M. Sharma, W. T. Cheng, T. P. Tai, Y. S. Cheng, W. Hsu, C. Liu, S. M. Reddy, and A. Mann, "Faster defect localization in nanometer technology based on defective cell diagnosis", in Proceedings of International Test Conference, pp. 1-10, 2007.
- [20] X. Fan, M. Sharma, W. T. Cheng, and S. M. Reddy, "Diagnosis of cell internal defects with multi-cycle test patterns", in Proceedings of Asian Test Symposium, pp. 7-12, 2012.
- [21] F. Hapke, M. Reese, J. Rivers, A. Over, V. Ravikumar, W. Redemund, A. Glowatz, J. Schloeffel, and J. Rajski, "Cell-aware Production test results from a 32-nm notebook processor", in Proceedings of International Test Conference, pp. 1-9, 2012.
- [22] H. Tang, B. Benware, M. Reese, J. Caroselli, T. Herrmann, F. Hapke, R. Tao, W. T. Cheng, and M. Sharma, "Diagnosing Cell Internal Defects Using Analog Simulation-based Fault Models" in Proceedings of Asian Test Symposium, pp. 318-323, 2014.
- [23] X. Fan, W. Moore, C. Hora, and G. Gronthoud, "A novel stuck-at based method for transistor stuck-open fault diagnosis", in Proceedings of International Test Conference, pp. 253-262, 2005.
- [24] X. Fan, W. Moore, C. Hora, M. Konijnenburg and G. Gronthoud, "A Gate-Level Method for Transistor-Level Bridging Fault Diagnosis", in Proceedings of VLSI Test Symposium, pp. 266-271, 2006.
- [25] X. Fan, W. R. Moore, C. Hora, and G. Gronthoud, "Extending gate-level diagnosis tools to CMOS intra-gate faults", IET Computers & Digital Techniques, vol. 1, no. 6, pp. 685-693, 2007.
- [26] F. Hapke, J. Schloeffel, W. Redemund, A. Glowatz, J. Rajski, M. Reese, J. Rearick, and J. Rivers, "Cell-aware analysis for small-delay effects and production test results from different fault models", in Proceedings of International Test Conference, pp. 1-8, 2011.
- [27] L. T. Wang, C. W. Wu, and X. Wen, "VLSI test principles and architectures: design for testability", Academic Press, 2006.
- [28] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyengar, "Transition fault simulation", IEEE Design & Test of Computers, vol. 4, no. 2, pp. 32-38, 1987.

- [29] G. L. Smith, "Model for Delay Faults Based Upon Paths", in Proceedings of International Test Conference, pp. 342-351, 1985.
- [30] J. Hammond and G. Sery, "Knowledge-based electrical monitor approach using very large array yield structures to delineate defects during process development and production yield improvement", in Proceedings of International Workshop on Defect and Fault Tolerance on VLSI Systems, pp. 67-80, 1991.
- [31] J. B. Khare, W. Maly, S. Griep, and D. Schmitt-Landsiedel, "Yield-oriented computer-aided defect diagnosis", IEEE Transactions on Semiconductor Manufacturing, vol. 8, no. 2, pp. 195-206, 1995.
- [32] W. Maly, "Yield diagnosis through interpretation of tester data", in Proceedings of International Test Conference, pp. 10-20, 1987.
- [33] H. Balachandran, J. Parker, D. Shupp, S. Butler, K. M. Butler, C. Force, and J. Smith, "Correlation of logical failures to a suspect process step", in Proceedings of International Test Conference, pp. 458-466, 1999.
- [34] S. Jansen, G. Florence, A. Perry, and S. Fox, "Utilizing design layout information to improve efficiency of SEM defect review sampling", in Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp. 69-71, 2008.
- [35] J. H. Yeh and A. Park, "Novel technique to identify systematic and random defects during 65 nm and 45nm process development for faster yield learning", in Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp. 54-57, 2007.
- [36] C. H. Gim, "A novel bitmap analysis technique-test sensitivity intensity bitmap", in Proceedings of the 9th International Symposium on the Physical and Failure Analysis of Integrated Circuits, pp. 105-109, 2002.
- [37] S. Malik, T. Herrmann, S. Madhavan, R. Desineni, C. Schuermyer, and G. Eide, "Deriving Feature Fail Rate from Silicon Volume Diagnostics Data", IEEE Design & Test, vol. 30, no. 4, pp. 26-34, 2013.
- [38] M. Sharma, C. Schuermyer, and B. Benware, "Determination of dominant-yield-loss mechanism with volume diagnosis", IEEE Design & Test of Computers, vol. 27, no. 3, pp. 54-61, 2010.
- [39] M. Sharma, B. Benware, L. Ling, D. Abercrombie, L. Lee, M. Keim, and A. Man, "Efficiently performing yield enhancements by identifying dominant physical root cause from test fail data", in Proceedings of International Test Conference, pp. 1-9, 2008.
- [40] M. Keim, N. Tamarapalli, H. Tang, M. Sharma, J. Rajski, C. Schuermyer, and B. Benware, "A rapid yield learning flow based on production integrated layout-aware diagnosis", in Proceedings of International Test Conference, pp. 1-10, 2006.
- [41] C. Schuermyer, K. Cota, R. Madge, and B. Benware, "Identification of systematic yield limiters in complex ASICs through volume structural test fail data visualization and analysis", in Proceedings of International Test Conference, pp. 137-145, 2005.

- [42] R. Desineni, L. Pastel, M. Kassab, M. F. Fayaz, and J. Lee, "Identifying design systematics using learning based diagnostic analysis", in Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp. 317-321, 2010.
- [43] C. Schuermyer, S. Malik, and T. Herrmann, "Identifying systematic critical features using silicon diagnosis data", in Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp. 1-6, 2012.
- [44] Y. Pan, A. Chittora, K. Sekar, S. Malik, and L. S. Keat, "DiagBridge: Analyzing scan diagnosis data in a yield perspective", in Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp. 15-20, 2014.
- [45] R. Turakhia, M. Ward, S. K. Goel, and B. Benware, "Bridging DFM analysis and volume diagnostics for yield learning-A case study", in Proceedings of VLSI Test Symposium, pp. 167-172, 2009.
- [46] B. Seshadri, P. Gupta, Y. T. Lin, and B. Cory, "Systematic defect screening in controlled experiments using volume diagnosis", in Proceedings of International Test Conference, pp. 1-7, 2012.
- [47] H. Tang, S. Manish, J. Rajski, M. Keim, and B. Benware, "Analyzing volume diagnosis results with statistical learning for yield improvement", in Proceedings of the 12th IEEE European Test Symposium, pp. 145-150, 2007.
- [48] B. Benware, C. Schuermyer, M. Sharma, and T. Herrmann, "Determining a failure root cause distribution from a population of layout-aware scan diagnosis results", IEEE Design & Test of Computers, vol. 29, no. 1, pp. 8-18, 2012.
- [49] Y. Pan, A. Chittora, K. Sekar, G. S. Huat, Y. G. Feng, A. Viswanatha, and J. Lam, "Leveraging root cause deconvolution analysis for logic yield ramping", in Proceedings of International Symposium of Testing and Failure Analysis, pp. 602-607, 2013.
- [50] X. Yu and R. D. Blanton, "Estimating defect-type distributions through volume diagnosis and defect behavior attribution", in Proceedings of International Test Conference, pp. 1-10, 2010.
- [51] W. C. Tam, O. Poku, and R. D. Blanton, "Systematic defect identification through layout snippet clustering", in Proceedings of International Test Conference, pp. 1-10, 2010.
- [52] R. D. Blanton, W. C. Tam, X. Yu, J. E. Nelson, and O. Poku, "Yield Learning Through Physically Aware Diagnosis of IC-Failure Populations", IEEE Design & Test of Computers, vol. 29, no. 1, pp. 36-47, 2012.
- [53] W. C. Tam and R. D. Blanton, "Physically-aware analysis of systematic defects in integrated circuits", in Proceedings of International Test Conference, pp. 1-10, 2011.
- [54] W. C. J. Tam and R. D. S. Blanton, "LASIC: Layout analysis for systematic IC-defect identification using clustering", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 8, pp. 1278-1290, 2015.

- [55] R. Desineni, L. Pastel, M. Kassab, and R. Redburn, "Hard to find, easy to find systematics; just find them", in Proceedings of International Test Conference, pp. 1-10, 2010.
- [56] D. Carder, S. Palosh, and R. Raina, "High-Volume Scan Analysis: Practical challenges and applications for industrial IC development", in Proceedings of International Test Conference, pp. 1-7, 2010.
- [57] N. Ma, "Automatic IC hotspot classification and detection using pattern-based clustering", Ph.D. dissertation, Dept. Mech. Eng., University of California, Berkeley, 2009.
- [58] H. Yao, S. Sinha, C. Chiang, X. Hong, and Y. Cai, "Efficient process-hotspot detection using range pattern matching", in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, pp. 625-632, 2006.
- [59] J. Ghan, N. Ma, S. Mishra, C. Spanos, K. Poolla, N. Rodriguez, and L. Capodiec, "Clustering and pattern matching for an automatic hotspot classification and detection system", In Proc. of SPIE vol. 7275, Design for Manufacturability through Design-Process Integration III, pp. 727516, 2009.
- [60] C. Huang, C. Young, H. Liu, S. F. Tzou, D. Tsui, A. Tsai, and E. Chang, "Using design based binning to improve defect excursion control for 45nm production", in Proceedings of International Symposium on Semiconductor Manufacturing, pp. 1-3, 2007.
- [61] H. P. Erb, C. Burmer, and A. Leininger, "Yield enhancement through fast statistical scan test analysis for digital logic", in Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp. 250-255, 2005.
- [62] L. M. Huisman, M. Kassab, and L. Pastel, "Data mining integrated circuit fails with fail commonalities", in Proceedings of International Test Conference, pp. 661-668, 2004.
- [63] W. C. Tam and S. Blanton, "To DFM or not to DFM?", in Proceedings of the 48th Design Automation Conference, pp. 65-70, 2011.
- [64] J. E. Nelson, T. Zanon, J. G. Brown, O. Poku, R. D. Blanton, W. Maly, and C. Schuermyer, "Extracting defect density and size distributions from product ICs", IEEE Design & Test of Computers, vol. 23, no. 5, pp. 390-400, 2006.
- [65] U. Hessinger, W. Chan, and B. Schafman, "Data mining for significance in yield-defect correlation analysis", IEEE Transactions on Semiconductor Manufacturing, vol. 27, no. 3, pp. 347-356, 2014.
- [66] R. D. Blanton, F. Wang, C. Xue, P. K. Nag, Y. Xue, and X. Li, "DREAMS: DFM rule Evaluation using manufactured silicon", in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, pp. 99-106, 2013.
- [67] B. G. Lindsay, "Mixture models: theory, geometry and applications", In NSF-CBMS regional conference series in probability and statistics. Institute of Mathematical Statistics and the American Statistical Association, 1995.

- [68] M. B. Christopher, "Pattern Recognition and Machine Learning", Springer-Verlag New York, 2016.
- [69] T. Koski and J. Noble, "Bayesian networks: an introduction (Vol. 924)", John Wiley & Sons, 2011.
- [70] G. McLachlan and T. Krishnan, "The EM algorithm and extensions (Vol. 382)", John Wiley & Sons, 2007.
- [71] D. Prescher, "A short tutorial on the expectation-maximization algorithm", in course Notes on CD of the 15th European Summer School in Logic, Language and Information, 2003.
- [72] R. B. Millar, "Maximum likelihood estimation and inference: with examples in R, SAS and ADMB (Vol. 111)", John Wiley & Sons, 2011.
- [73] J. V. Psutka and J. Psutka, (2015). "Sample Size for Maximum Likelihood Estimates of Gaussian Model", in Proceedings of International Conference on Computer Analysis of Images and Patterns, pp. 462-469, 2015.
- [74] G. Casella and R. L. Berger, "Statistical inference (Vol. 2)", Pacific Grove, CA: Duxbury, 2002
- [75] D. A. Schum, "The evidential foundations of probabilistic reasoning", Northwestern University Press, 2001.
- [76] S. Lipschutz and M. Lipson, "Schaum's Outline of Linear Algebra Fourth Edition", McGraw Hill Professional, 2008.
- [77] W. T. Cheng, Y. Tian, and S. M. Reddy, "Volume diagnosis data mining", in Proceedings of the 22nd IEEE European Test Symposium, pp. 1-10, 2017.
- [78] W. T. Cheng, R. Klingenberg, B. Benware, W. Yang, M. Sharma, G. Eide, Y. Tian, S. M. Reddy, Y. Pan, S. Fernandes, and A. Chittora, "Automatic Identification of Yield Limiting Layout Patterns Using Root Cause Deconvolution on Volume Scan Diagnosis Data", in Proceedings of Asian Test Symposium, pp. 219-224, 2017.
- [79] N. R. Draper and H. Smith, "Applied regression analysis (Vol. 326)", John Wiley & Sons, 2014.
- [80] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection", in Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 1137-1145, 1995.
- [81] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner, "A novel type of skeleton for polygons", Journal of Universal Computer Science, vol. 1, no. 12, pp. 752-761, 1995.
- [82] C. Du Mouza, W. Litwin, and P. Rigaux, "Sd-rtree: A scalable distributed rtree", in Proceedings of the IEEE 23rd International Conference on Data Engineering, pp. 296-305, 2007.

- [83] B. Marr, "The Top 10 AI And Machine Learning Use Cases Everyone Should Know About", Retrieved from <https://www.forbes.com/sites/bernardmarr/2016/09/30/what-are-the-top-10-use-cases-for-machine-learning-and-ai/>. [accessed June 27, 2018].
- [84] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization", SIAM Journal on Scientific Computing, vol. 16, no. 5, pp. 1190-1208, 1995.
- [85] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of machine learning. MIT press, 2012.