The Dissertation Committee for Wei Ye
certifies that this is the approved version of the following dissertation:

Design for Manufacturability and Reliability through Learning and Optimization

Committee:

David Z. Pan, Supervisor

Nan Sun

Nur A. Touba

Qiang Liu

Zhiyu (Albert) Zeng

Design for Manufacturability and Reliability through Learning and Optimization

by

Wei Ye

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2020

Acknowledgments

I would like to heartily thank my advisor, Professor David Z. Pan, for his persistent understanding, encouragement, and guidance over the years. He is a great mentor who encourages me to pursue essential and intriguing research problems and guides me to become an independent researcher with my own thoughts. I have learned more than I could have ever imagined from him: a broad perspective of research and industry, valuable skills for a future career, and also positive attitudes toward life. His warm advice and continuous support will always have positive impacts throughout my life in the future.

Also, I would like to express my appreciation to my committee members. I would like to thank Professor Nan Sun, Professor Nur A. Touba, and Professor Qiang Liu for their kindness and support to this dissertation. My sincere thanks also go to Dr. Albert Zeng for the great guidance and discussions during my internship at Cadence.

Besides, I deeply appreciate the help from other industrial mentors and collaborators: Osamu Yamane, Yuki Watanabe, and Shigeki Nojima at Kioxia Corporation, Dr. Dwight Hill, Dr. Rui Shi, Dr. Zac Levinson, Dr. Peter Brooker, and Dr. Kevin Lucas at Synopsys, Inc. This work would not be possible without their technical suggestions.

In addition to my advisor, committee members, and colleagues, I would also like to recognize the help and effort that I received from many other people: Taiki Kimura, Dr. Tetsuaki Matsunawa, and Dr. Chikaaki Kodama at Kioxia Corporation, Ahmed Omran and Marco Guajardo at Synopsys, Inc, Dr. Kai Zhong graduated from the Computer Science Department of UT Austin. It has been a great learning experience to work with other UTDA members and alumni: Dr. Bei Yu, Dr. Xiaoqing Xu, Dr. Yibo Lin, Dr. Jiaojiao Ou, Dr. Meng Li, Dr. Derong Liu, Dr. Biying Xu, Dr. Wuxi Li, Dr. Shounak Dhar, Zheng Zhao, Mohamed Baker Alawieh, Keren Zhu, Mingjie Liu, Rachel S. Rajarathnam, Zixuan Jiang, Jiaqi Gu, Hao Chen, Che-Lun Hsu, and Jingyi Zhou. Their continuous encouragement and inspiring discussion throughout my years of study helped develop and polish this dissertation.

I appreciate the help and support of all my friends over the past five years. Last but most importantly, I am deeply thankful to my family for their love and unwavering support. I dedicate this dissertation to my loving parents Pingwen Ye and Defeng Li, and my brother Zhi Ye. They encourage me to pursue my dream and are always there for me through this long journey. No words can express my thankfulness.

Design for Manufacturability and Reliability through Learning and Optimization

Publication No. ____________

Wei Ye, Ph.D.
The University of Texas at Austin, 2020

Supervisor: David Z. Pan

Modern society relies on technologies with integrated circuits (ICs) at their heart. In the last several decades, as the performance and complexity of ICs keep escalating, the semiconductor industry has demonstrated an ability to develop new process techniques and product designs that are both manufacturable and reliable. However, as the transistor feature size is further shrunk into extreme scaling (e.g., 10 nm and beyond), large scale integration of transistors and interconnects brings ever-increasing challenges revolving around manufacturability and reliability.

The major issues in manufacturability and reliability for modern ICs come from three aspects: (1) layout-dependent manufacturability (e.g., manufacturing yield sensitive to design patterns); (2) time-consuming process modeling (e.g., complex lithography systems); (3) design-sensitive reliability (e.g.,

lifetime related to layout designs). In order to close the gap between design and manufacturing and enhance design reliability, automated layout generation requires cross-layer information feed-forward and feedback, such as accurate process modeling and reliability-guided design optimization.

This dissertation attempts to address the aforementioned challenges in manufacturing closure and reliability signoff through efficient machine learning techniques for lithography hotspot detection and lithography modeling, and synergistic design optimization for electromigration (EM). Our research includes efficient lithography hotspot detection, learning-based lithography modeling, and EM-aware physical design to achieve efficient manufacturing closure and reliability signoff.

For lithography hotspot detection, due to the increasingly complicated design patterns, early and quick feedback for lithography hotspots is desired to guide design closure in early stages. Machine learning approaches have been successfully applied to hotspot detection while demonstrating a remarkable capability of generalization to unseen hotspot patterns. However, most of the proposed machine learning approaches are not yet able to answer two critical questions: model confidence and model efficiency. This study develops a lithography hotspot detection framework capable of providing modeling confidence with fewer training data and fewer expensive lithography simulations needed, and also provides a holistic measure for the intrinsic class imbalance in lithography hotspot detection.

For lithography modeling, one of the major limitations in process mod-

eling is considered: the trade-off between modeling efficiency and accuracy. The steady decrease of the feature sizes, along with the growing complexity and variation of the manufacturing process, has tremendously increased the lithography modeling complexity and prolonged the already-slow simulation procedure. Different modeling frameworks are proposed in this study, leveraging recent advancements in machine learning, particularly generative adversarial learning, to generate virtually simulated silicon image efficiently without running detailed optical simulations. With our proposed deep learning techniques, a significant improvement in modeling efficiency is achieved while maintaining high modeling accuracy.

For EM-aware physical design, we demonstrate the limitation of conventional design and EM signoff flow when faced with the ever-growing EM violations in advanced technology nodes. Two essential directions are explored with practical algorithms and new design flows: (1) Power grid EM detection and optimization with several detailed placement techniques; (2) Learning-based signal EM prediction and mitigation at different physical design stages.

The effectiveness of proposed design optimization and machine learning techniques is demonstrated with extensive experiments on industrial-strength benchmarks. Our approaches are capable of reducing turn-around time, saving modeling costs, and enabling fast manufacturing closure and reliability signoff.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In the last several decades, the semiconductor industry's ability to follow Moore's law has been the core engine of a virtuous cycle. Through transistor scaling, new products with better performance are obtained. Each new technology generation produces smaller and faster transistors that can switch faster than those produced with the previous technology generation while simultaneously became cheaper to manufacture, which induces an exponential growth of the semiconductor market. This in turn allows further investments in semiconductor technologies which will fuel further scaling [91]. Such a continuous drive towards scaling integrated circuits (ICs) technologies has been accompanied by a trend of the increasing complexity of chip functionalities. Especially when the transistor feature size is shrunk into extreme scaling (e.g., 10 nm and beyond), such large scale integration of transistors and interconnects comes with significant challenges revolving around manufacturability and reliability.

Very large-scale integration (VLSI) manufacturing with advanced lithography has been a holy grail for the semiconductor industry to march at the pace of Moore's Law. With continuous shrinking of semiconductor process technology nodes, the minimum feature size of modern IC is much smaller than the lithographic wavelength [3]. The 193-nm wavelength lithography is still the mainstream for pitch scaling in advanced technology nodes, with help from various innovative technologies to print features much smaller than the wavelength, including restrictive design rules, resolution enhancement techniques (RETs), and advanced source-mask optimization (SMO). Nevertheless, IC designs are increasingly challenged to achieve manufacturing closure, i.e., being fabricated with high product yield due to feature miniaturizations and process variations.

One fundamental limitation for sub-wavelength lithography is that what one sees at the physical layout stage is not necessarily what one will get after the chip is fabricated. These corresponding printability challenges not only cause possible open/shorts, but also lead to parametric yield loss. Even with complex source/mask optimizations for a single exposure, lithography hotspots still exist after the patterning processes. Figure 1.1 shows the printed images of two local regions from certain 32-nm design after applying RETs. We can see that there are various types of process hotspots, featuring complex patterns related to line-ends, jogs, corners, contacts, etc. Therefore, in physical verification stage, detecting and fixing these lithography hotspots beforehand play an important role in improving production yield.

Lithography is a patterning process where patterns of desired designs were transferred on to a base substrate, mostly using masks, and has rapidly

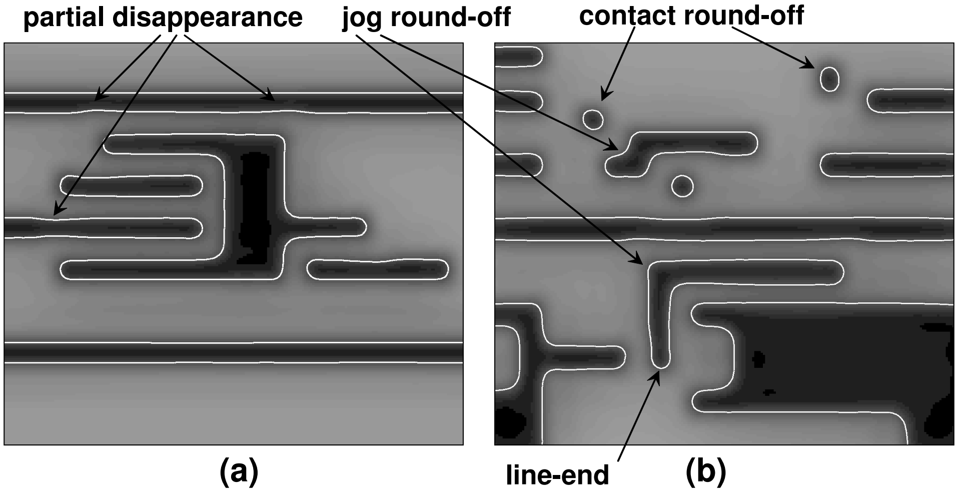
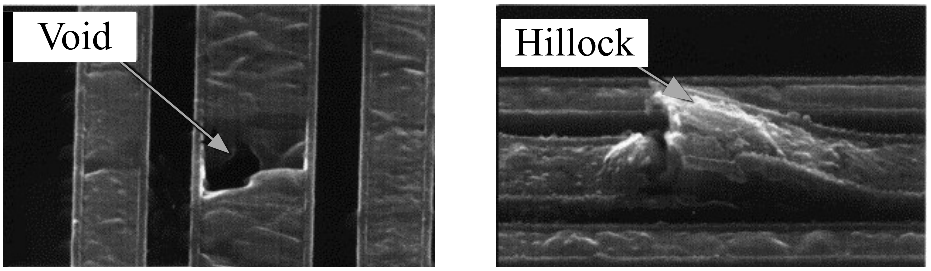


Figure 1.1: Examples of lithography hotspots [30].

become an extremely complex process step [80]. It covers optical lithography and its variations, including immersion, multiple patterning lithography, extreme ultraviolet (EUV), and electron beam (e-beam) lithography. For advanced process development and optimization, it is crucial to thoroughly analyze process-limiting effects within the imaging system of an exposure tool, taking the impact of mask and substrate topography on photoresist patterning into account. Moreover, when pushing the limits of resolution to achieve extreme scaling, more physical phenomena happening during the process need to be fully understood. Given the exorbitant cost of the lithography process, lithography modeling and simulation have become increasingly indispensable to bypass the cost-intensive and time-consuming experimentation stages during process development and performance verification [78,82]. For the development of new processes, simulation allows investigating the influence of process variables like illumination, exposure dose, and proximity gap on the resist pattern, before running experiments. Accordingly, critical processes could be optimized, process windows and yield could be improved. In advanced technology nodes, the complexities of lithography models have increased significantly. Accurate and efficient simulation can reduce experimental engineering efforts and short-loop experiments, bringing expedited process development, considerable cost savings, and a faster time-to-market.

Reliability is another major issue in modern ICs, which usually refers to how robust a chip is after manufacturing. Device aging and interconnect electromigration (EM) effects are likely to cause unexpected performance degradation and even malfunctions at the end of circuit life cycles. Hence, the reliability of designs needs to be verified and validated at all levels of the design phase. As IC technologies continue to scale, complex chip functionalities have been made possible by virtue of increasing transistor densities and aggressive scaling of interconnects. Besides, interconnects are getting thinner and running longer. These factors bring along higher current densities in metal wires, a phenomenon that further exacerbates EM.

EM is the gradual displacement of atoms in metal under the influence of an applied electric field and is considered the primary failure mechanism for metal interconnects. After the migration of atoms with electrons in a metal line for a certain period, a void grows on one side, which increases the resistance of the metal line and may eventually lead to open circuits. Hillock is formed on the other side and may cause short circuits. Figure 1.2 shows the scanning electron microscopy (SEM) images of void and hillock. In advanced technology nodes, the failure time from EM is worsened even further by the local temperature increase caused by self-heating of the underlying FinFETs. It is worth mentioning that, while the interconnects degrade due to various effects including electromigration (EM) and stress-migration, transistors also degrade temporally due to aging caused by effects like negative bias temperature instability (NBTI), time-dependent dielectric breakdown (TDDB), and hot carrier injection (HCI).



(a)

(b)

Figure 1.2: A void and a hillock generated by electromigration [15].

1.2 Challenges in Manufacturing Closure and Reliability Signoff

Design for manufacturing (DFM) techniques address the questions related to the exchange of information between design and manufacturing, and the use of this information for better printability and enhanced yield. Although various resolution enhancement techniques have been developed for extreme scaling, they also impose additional design constraints and raise challenges in design and manufacturing closure. The major issues come from the following aspects.

Layout-dependent manufacturability. A lithography hotspot is caused not only by a particular mask pattern, but also by the interaction with neighboring patterns inside the lithography influence region. Lithography hotspots still exist after the patterning processes with complex source and mask optimizations for a single exposure. Therefore, in physical verification stage, detecting and fixing these lithography hotspots beforehand play an important role in improving manufacturing yield. Lithography hotspot detection has been extensively studied in the computer-aided design (CAD) community involving various approaches including machine learning. Machine learning approaches are proposed for early and quick detection of lithography hotspots during physical verification and have demonstrated good generalization capability to recognize unseen hotspot patterns. Nevertheless, the general machine learning approaches without certainty estimation are less practical in real-world applications in the sense that they cannot help judge the trustworthiness of the decision. Moreover, the imbalance in hotspot detection problem degrades model performance, increases the cost of data preparation, and slows down the design closure.

Time-consuming process modeling. Smaller and smaller feature sizes require higher process modeling accuracy to guarantee manufacturability. Meanwhile, lithography systems are becoming more and more complex to improve resolution. The increasing concern for physical effects and the introduction of the aforementioned innovative technologies for advanced lithography have greatly augmented the number of parameters whose effects must be analyzed and characterized, increasing the difficulty in process modeling. In IC manufacturing, modeling efficiency is crucial for fast design closure along with modeling accuracy. Detailed computational lithography simulations [77, 112] have been used to obtain accurate pattern images, but they are extremely compute-intensive. Hence, they may not be suitable to be applied on a full-chip scale while having fast turn-around-time to guide early IC physical design. The state-of-the-art lithography modeling techniques still suffer from an exorbitant computational cost.

Another key nanometer IC challenge comes from reliability. With continued feature size shrinking and increased transistor density, reliability issues are more and more severe. The major source of reliability issues on interconnect is EM, which is initiated by current flow and may cause open and short circuit failures over time. Recently, design for reliability (DFR) has obtained more and more attention. Conventionally, designers need to increase design margin to accommodate reliability considerations, which may limit circuit performance, and yet the circuit lifetime uncertainty remains. Incorporation of variation or uncertainty into the design stage, as well as the balance of circuit performance and reliability, is critical in the DFR process.

Design-sensitive reliability. Power grid interconnects have always been susceptible to EM failures as they carry large unidirectional currents. In addition, the continuous drive toward extreme scaling keeps compounding the EM problem for long and thin signal nets that are expected to switch at gigahertz speed. Conventionally, EM checking tools calculate current densities in metal wires and detect EM violations with given design rules; then, these violations are fixed with engineering change order (ECO) efforts. However, traditional post-design fixing approaches such as spacing large-current cells and widening metal wires are ill-equipped when faced with the ever-growing EM violations in advanced technology nodes. Therefore, it is of vital importance to incorporate EM detection and fixing techniques into earlier stages of physical design.

1.3 Dissertation Overview

The objective of this dissertation is to provide integrated design and verification solutions that aim at facilitating fast and reliable manufacturing closure and reliability signoff.

Chapter 2 presents our solutions to the two critical questions faced by most of the previous machine learning approaches for lithography hotspot detection: machine learning model confidence and efficiency. We first propose a lithography hotspot detection framework capable of providing modeling confidence with fewer training data and fewer expensive lithography simulations needed. We further provide a holistic measure for the fundamental class imbalance issue in the lithography hotspot detection task. The proposed measure is directly used as an optimization objective and allows further boosting neural network models for hotspot detection.

Chapter 3 explores the critical trade-off in process modeling between modeling accuracy and efficiency. We propose two distinct lithography modeling engines to improve simulation quality. LithoGAN is the first complete end-to-end lithography modeling framework leveraging generative learning to achieve significant simulation speedup. TEMPO features a generative learning-based framework for efficient and accurate 3D aerial image prediction considering mask topography effects.

Chapter 4 presents two detection and mitigation frameworks for signal and power grid EM, both of which can be seamlessly integrated into standard physical design (PD) flow. We first propose a series of incremental placement techniques for EM in power grid wires with negligible impacts on wirelength and placement density. Next, we further propose a signal EM hotspot detection and mitigation framework. It identifies EM-suspicious signal nets based on information available during placement and addresses those problematic nets at physical design. We demonstrate that such fixing strategy can effectively reduce iterative EM fixing costs and enable faster design closure.

Chapter 5 summarizes this dissertation and discusses potential future research directions.

Chapter 2

Lithography Hotspot Detection for Advanced Manufacturing Processes

2.1 Introduction

With the rapid shrinking of semiconductor process technology nodes, there is a widening gap between design demands and manufacturing capabilities posed by the current mainstream 193-nm lithography. Due to the complexity of lithography systems and process variation, the layout patterns that are hard to print become lithography hotspots. Although numerous design for manufacturability (DFM) techniques have been proposed to improve manufacturing yield, lithography hotspots still exist and need to be identified and eliminated during physical verification. Efficient and accurate lithography hotspot detection is critical for layout finishing and design closure towards yield

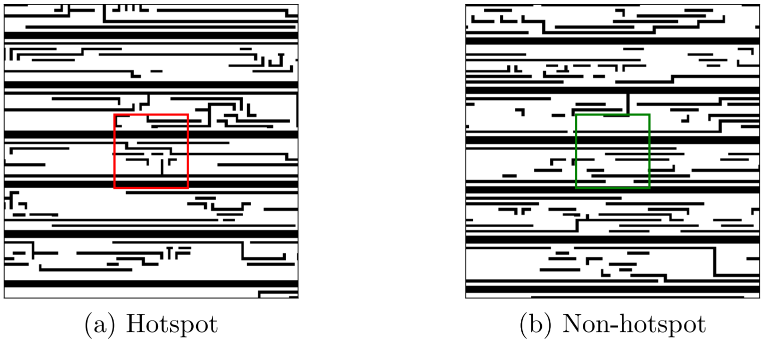
This chapter is based on the following conference papers.

1. Wei Ye, Mohamed Baker Alawieh, Meng Li, Yibo Lin, and David Z. Pan. "Litho-GPA: Gaussian process assurance for lithography hotspot detection." In 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 54-59. IEEE, 2019.
2. Wei Ye, Yibo Lin, Meng Li, Qiang Liu, and David Z. Pan. "LithoROC: lithography hotspot detection with explicit ROC optimization." In Proceedings of the 24th Asia and South Pacific Design Automation Conference (ASPDAC), pp. 292-298. 2019.

I am the main contributor in charge of problem formulation, algorithm development, and experimental validations.

improvement in the physical verification stage.

Figure 2.1 shows an example of two clips where one encompasses lithography hotspot marked by the red rectangle region (a) while the other does not (b). Lithography hotspots can be accurately detected through full-chip lithography simulations which compute the aerial images and contours of printed patterns [54, 102]; though, at a tremendous computational cost [82]. Pattern matching and machine learning based techniques have been proposed for early and quick detection of lithography hotspots during physical verification [71]. Pattern matching is a direct and fast method for hotspot detection [126, 140]. However, pattern matching, including fuzzy pattern matching [70, 121], is still insufficient to handle never-before-seen hotspot patterns. On the other hand, machine learning approaches have demonstrated good generalization capability to recognize unseen hotspot patterns [29, 30, 33, 84, 95, 115, 141–143].


(a) Hotspot

(b) Non-hotspot

Figure 2.1: An example of lithography hotspot clip (a) and non-hotspot clip (b).

In these approaches, a labeled dataset is used to train a machine learning model capable of detecting hotspots in new layout patterns with high accuracy. The primary objective is to achieve high accuracy while minimizing false alarms. Practically, accuracy is given the highest priority; hence, a moderate number of false alarms is typically tolerated for the sake of achieving better accuracy. This is due to the fact that missing any hotspot may result in significant yield degradation.

In the rest of the chapter, Section 2.2 presents our approaches to improve learning confidence and reduce training cost, Section 2.3 explores the effectiveness of the area under the ROC curve (AUC) optimization to boost the model performance when faced with class imbalance.

2.2 Gaussian Process Assurance for Lithography Hotspot Detection

Machine learning approaches have demonstrated good generalization capability to recognize unseen hotspot patterns [29,30,33,84,95,115,141–143]. Recently, deep learning techniques have been actively explored to improve the accuracy of hotspot detection [25,85,107,130,131,137].

Nonetheless, most of the proposed machine learning approaches are not yet able to answer one critical question: how much a hotspot predicted from a machine learning model can be trusted? With efforts mainly tailored towards achieving better accuracy, little attention has been given to this confidence issue. In practice, addressing this concern requires machine learning models to provide confidence guarantees alongside the label predictions. For example, in a deep learning model, the results of the softmax are usually interpreted as probability estimates. However, it has been shown that these probability estimates do not match the correct likelihood [63]; in fact, networks are often too confident about their predictions. In other words, in a classification problem, the output of the softmax can lead to correct labeling of samples; however, the values of the softmax is not a good uncertainty measure.

Bayesian-based methods are the typical option when confidence estimation is needed. In this work, we adopt a Gaussian Process (GP) based classification that can provide a confidence metric for each predicted instance. In practice, a GP prediction is used as a final label only when the confidence level matches a user-defined metric, otherwise, the prediction is marked as untrusted and lithography simulation can be used to further verify the results.

On the other hand, learning based approaches usually require a large amount of training data to obtain models with good generalization, especially for imbalanced datasets, as in the case of the hotspot detection task. This imbalance increases the cost of data preparation and slows down the design closure. This is mainly because each training data sample requires lithography simulation to obtain its label and hotspot samples appear much less often than non-hotspot ones. Therefore, we also propose an active learning scheme with a sequence of weak classifiers to reduce the turnaround time and the cost of data preparation. The combination of GP and active learning scheme is not only able to achieve high accuracy, but also provides a confidence estimation for predictions, with a small amount of training data.

2.2.1 Preliminaries

The hotspot detection task to be solved by machine learning techniques can be formulated as a two-class image classification problem; a classical problem which has been studied extensively in literature. However, the problem at hand has its unique characteristics that should be taken into account. First, despite the fact that the lithography defects are critical, their relative number is significantly small across the whole chip. This poses a major challenge when formulating the task as a learning problem because the two classes are highly imbalanced which necessitates a proper handling to remove the inherent bias in the data.

Second, with such imbalanced data, the number of false alarms is usually comparable to, or even higher than, the number of true hotspots. In practice, the number of false alarms is among the most important metrics to evaluate hotspot detection methods [130]. **Accuracy** (i.e., true positive rate [16]) and the number of **false alarms** (i.e., false positives) are the two prevailing metrics used for detection evaluation. However, to make use of these models, a new criterion should be considered which is *trust*. Among the questions we address in this work is *should we trust all predictions from a highly accurate model?*

2.2.2 Lithography Hotspot Detection

In this section, we explain the details of the proposed Litho-GPA framework for lithography hotspot detection. It consists of two key components:

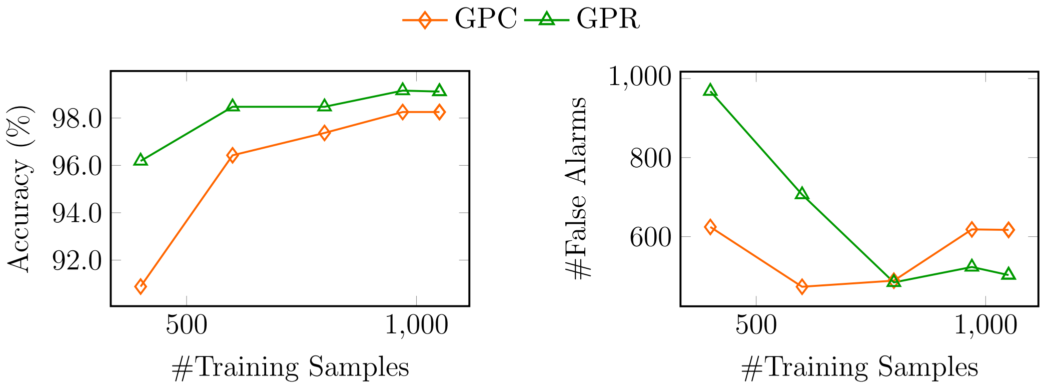
Gaussian Process for hotspot detection and active learning for data preparation.

2.2.2.1 Hotspot Detection using Gaussian Process

Gaussian Process (GP) classification falls under the category of probabilistic classification where test predictions take the form of class probabilities; this contrasts with methods which provide a class label only [92]. Since generalization to test cases inherently involves some level of uncertainty, it is natural to attempt to make predictions in a way that reflects these uncertainties. For hotspot detection, GP can provide, alongside the label, a confidence measure about the label which can help judge the trustworthiness of the obtained classification decision.

In literature, different schemes have been proposed for binary GP classification. Among the most commonly used are those based on logistic or probit mapping where Laplace Approximation is used to estimate the posterior distribution [100]. In other approaches, the binary classification is cast as a regression problem where the objective is to predict a continuous label that can be mapped through thresholding to binary labels. In theory, GP classification with Laplace Approximation (GPC) uses a Bernoulli likelihood in the Bayesian inference, thus incorporating the binary labels into the inference. While such likelihood is an accurate representation of the binary data, it is not conjugate with GP prior; hence it makes the inference intractable and requires approximating the posterior distribution. On the other hand, using a regression based GP for classification (GPR) moves the binary mapping outside the inference; hence, preserving the conjugacy that results in a closed form posterior distribution. In the hotspot detection task, hotspots are assigned to +1 and non-hotspots are assigned to -1; 0 can be a decision boundary which maps the continuous quantity output of GPR to the two discrete classes.

The comparison of GPR and GPC is shown in Figure 2.2. Examining the figure, one can notice that, with the same number of samples, GPR is always achieving higher prediction accuracy and the number of false alarms resulting from GPR is lower than that from GPC when it converges. Moreover, the fact that the posterior distribution can be obtained with no approximation in GPR is reflected in the computational cost where GPC is significantly more expensive computationally than GPR. Based on this comparison, GPR was adopted in this work.



(a) Accuracy

(b) False alarm

Figure 2.2: Comparison between GPC and GPR on `Layout2` where accuracy (a) and the number of false alarms (b) are shown.

For the hotspot detection task, the class label y is assumed to be a continuous noisy version of an underlying GP $f(\mathbf{x})$ with a Radial Basis Function (RBF) kernel. Based on the prior distribution of $f(\mathbf{x})$, the joint distribution of the observed outputs $\mathbf{y}$, and the GP function values for the test outputs, the predictive posterior for the images in the test data $p(\mathbf{f}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*)$ is given by [100]:

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*) &\sim \mathrm{N}(\mu, \Sigma), \\ \mu &= K(\mathbf{X}_*, \mathbf{X})\big[K(\mathbf{X}, \mathbf{X}) + \sigma^2 I\big]^{-1}\mathbf{y}, \\ \Sigma &= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})\big[K(\mathbf{X}, \mathbf{X}) + \sigma^2 I\big]^{-1} K(\mathbf{X}, \mathbf{X}_*), \end{aligned} \tag{2.1}$$

where $\mathbf{X}$ and $\mathbf{X}_*$ are matrices containing the training and testing clip data respectively. σ^2 represents the level of noise in the data, $\mathbf{y}$ is a vector representing the labels for training data, $\mathbf{f}_*$ is a vector representing the function prediction for the testing clip data, and the matrices $K(\cdot, \cdot)$ represent the covariance matrices obtained by evaluating the RBF kernel.

In the GPR approach, the posterior distribution in Equation (2.1) represents the distribution of the class label in the continuous domain. To get a point estimate of the class label, proper thresholding scheme (usually at 0) is used for the mean of the posterior distribution. However, we are interested in a predictive distribution that can provide a confidence to judge upon prediction. This can be achieved by leveraging all information provided by the posterior distribution; i.e., the distribution of the continuous class label. To elaborate on this, we consider the example shown in Figure 2.3 where the posterior distributions for two samples $\mathbf{x}_1$ and $\mathbf{x}_2$ are shown. By looking at the point estimate, both samples have a mean value greater than 0; hence, they will be mapped to label 1. However, it is clear that the uncertainty associated with $\mathbf{x}_2$ is much higher compared to that associated with $\mathbf{x}_1$. In other words, there is a higher probability for the label of $\mathbf{x}_2$ to be less than 0.

Therefore, for a sample with mean greater than 0, a confidence metric can be defined based on the probability that the predicted label is higher than 0. In such case, sample $\mathbf{x}_1$ has a probability of 98% compared to 65% for $\mathbf{x}_2$, which implied higher confidence around the prediction of $\mathbf{x}_1$. To utilize this information, a confidence metric α can be defined to judge upon the validity of the predictions obtained from GPR.

However, while 0 is the intuitive choice for a boundary between the two labels $\{-1, +1\}$ in a classification task, the value of the threshold boundary can be tuned for problems with special characteristics such as class imbalance in the hotspot detection task. Hence, the compromise between accuracy and false alarms can be controlled using a threshold different from zero. In other words, such thresholding scheme can provide control over how *conservative* the model is.

In this hotspot detection task, missing a hotspot can have much more significant consequences when compared to having additional false alarms. With this risk assessment in mind, the threshold can be set to a value κ, where $\kappa < 0$, to bias the prediction towards the hotspot class. Therefore, the labeling processes can be performed according to the following scheme:

$$\hat{y}_i = \begin{cases} +1, & \text{if } p(f_i > \kappa) > \alpha \\ -1, & \text{if } p(f_i < \kappa) > \alpha \\ \text{untrusted}, & \text{otherwise.} \end{cases} \tag{2.2}$$

In Equation (2.2), a class label is given to a particular sample if it meets the user-defined confidence metric α. Otherwise, the prediction for the particular sample is set to untrusted reflecting the low confidence in the model prediction and requiring an actual simulation run to validate this sample. For example, considering the two samples in Figure 2.3 with $\alpha = 0.7$ and $\kappa = 0$, sample $\mathbf{x}_1$ will get a label of +1 while $\mathbf{x}_2$ will not be assigned a label, and a lithography simulation is needed to get the right label.

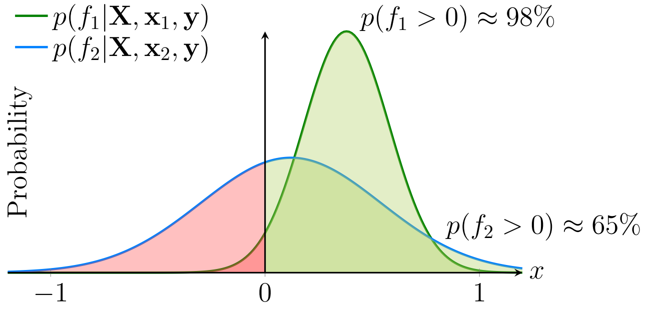


Figure 2.3: The posterior distributions obtained through GPR for two samples are shown. The distributions show higher confidence in the prediction of f_1 compared to that of f_2.

2.2.2.2 Active Learning for Data Selection

When formulating the lithography hotspot detection problem as a learning based classification problem, class imbalance comes forth among the major challenges characterizing the learning task. In practice, there is an abundance in the non-hotspot data on one end and scarceness in the hotspot data on the other. With such setup, a large number of samples is needed to guarantee enough hotspot samples for building accurate classification models. This translates to an enormous computational cost associated with running a large number of lithography simulations. The main reason to endure this cost is based on the fact that given a set of un-simulated data samples, one cannot tell beforehand which ones are hotspots. Hence, the trivial way of collecting data is to randomly select samples for simulation until enough hotspot samples are available.

To address this issue, we propose an active learning framework with the objective of selecting samples that are *likely* to be hotspots and simulating them to get the actual labels. This way, a balanced training dataset, adequate for model training, can be constructed with minimal simulation cost. The main idea is to iteratively select hotspot candidates for simulation based on labels obtained using trained weak classifiers. As a first step, a relatively small set of randomly selected samples, for which simulations are performed and labels are available, is used to build a weak classifier that can point out tentative hotspot samples among the un-simulated ones. These selected samples are then simulated and added to the available training dataset to help improve the performance of the classifier in the next iteration. A weak classifier is adequate here because its training cost is cheaper and the accuracy requirement at this stage is not high.

Here, a weak classifier is one that relies on a simple model; hence, it does not require a large number of samples to train. Although such a classifier may not have a high true positive rate, it can help guide the sampling scheme, especially that the nature of the data will result in a relatively high precision value even with a low true positive rate. Among the possible options, Support Vector Machine (SVM) is used as the weak classifier in this active sampling scheme mainly because of its relatively superior performance and fast training [16]. The details of the active learning method are summarized in Algorithm 2.1.

Algorithm 2.1 Active Learning for Data Selection

```
Input: A pool of unlabeled data samples $\mathcal{P}$, $n$
Output: Labeled training dataset $\mathcal{S}$
1: $\mathcal{S} \leftarrow$ Select $m_0$ samples randomly from $\mathcal{P}$ and obtain their labels through
   simulation;
2: $k \leftarrow 0$;
3: repeat
4:     $k \leftarrow k + 1$;
5:     Train the SVM model with the training dataset $\mathcal{S}$;
6:     $S_k \leftarrow$ Select $m_k$ samples from $\mathcal{P} \setminus \mathcal{S}$ with highest hotspot probability
   by SVM and simulate the labels;
7:     $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_k$;
8: until No hotspot in $\mathcal{S}_k$ or $|\mathcal{S}| \geq n$
9: return $\mathcal{S}$.
```

Algorithm 2.1 takes a pool of unlabeled data samples $\mathcal{P}$ and the maximum allowable size n of final dataset $\mathcal{S}$ as input. An initial training set $\mathcal{S}$ is generated by randomly sampling from the pool, followed by label queries through lithography simulations (line 1). Next, the algorithm builds a sequence of weak classifiers to seek more hotspots with the knowledge of previous sampled and simulated hotspots (lines 3 – 8). Each weak classifier is trained with the obtained labeled dataset $\mathcal{S}$ so far (line 5) and is applied to the remaining unlabeled samples in the pool $\mathcal{P}$, i.e., $\mathcal{P}\backslash\mathcal{S}$. The top m_k samples with the high probability of being hotspot are chosen from $\mathcal{P} \setminus \mathcal{S}$, and their labels are obtained by lithography simulations (line 6). Then, the selected set $\mathcal{S}_k$ is added to the labeled dataset $\mathcal{S}$. Note that the algorithm will return early if no actual hotspot is detected among the m_k samples (line 8). That is because, after several iterations, the trained SVM model is expected to have good accuracy, hence, if no hotspot can be detected by the latest classifier, it is likely that none are still present in the pool. Besides, the algorithm can exit from the iterations of weak classifier building if there are enough samples for the Gaussian Process in Section 2.2.2.1 (line 8).

2.2.2.3 Overall Flow

The proposed Litho-GPA framework is illustrated in Figure 2.4. We first leverage the iterative weak classifier-based sampling scheme to prepare a training set containing enough hotspots (Section 2.2.2.2). A GPR model is trained with the selected data samples. We then apply the GPR model to make predictions with confidence estimation on the testing set (Section 2.2.2.1). If GPR gives the predicted label with high confidence, the result is trusted; otherwise, the unsure testing samples will be verified with lithography simulations.

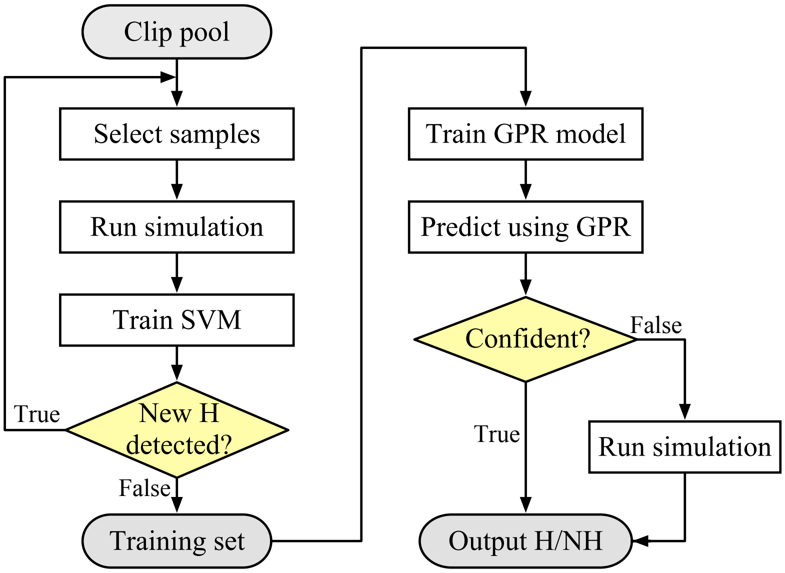


Figure 2.4: Overall flow including data preparation with active sampling and hotspot detection with Gaussian process.

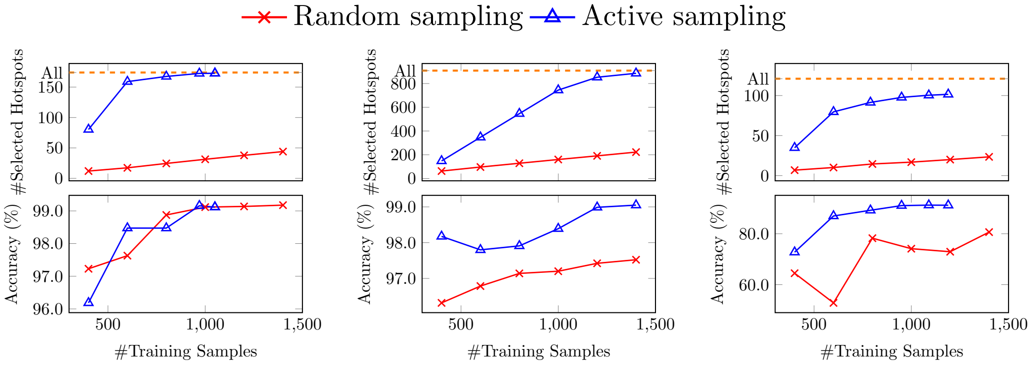
2.2.3 Experimental Results

Our Litho-GPA framework is implemented in Python with the scikit-learn library [96] and validated on the ICCAD 2012 CAD contest benchmark set [116]. `Layout1` is not used because it contains only a few clips and has a different technology node from the rest of the four benchmarks. `Layout5` has a small number of hotspots, and hence we merge it with `Layout4`. Table 2.1 summarizes the benchmark information, the number of all the clips (#All) and the number of hotspot clips (#H) in the training set (Train) and testing set (Test). The input image is downsized to 128×128 by a nearest-neighbor reduction to improve SVM and GPR training time. We run ten experimental trials for each evaluation, each with a different random seed, and report the average results. It is important to note that, although all samples in the training sets are already labeled in these benchmarks, to validate our framework we assume that they are not labeled at the beginning and obtain the labels through simulations in the framework.

Table 2.1: ICCAD 2012 contest benchmark statistics [131].

Design	Train		Test	
	#All	#H	#All	#H
`Layout2`	5,459	174	41,796	498
`Layout3`	5,552	909	48,141	1,808
`Layout4&5`	7,289	121	51,435	218

2.2.3.1 Active Learning for Data Selection



(a) Layout 2

(b) Layout 3

(c) Layout 4&5

Figure 2.5: The number of selected hotspots and testing accuracy (without validation simulations) for different sampling techniques are shown. "All" represents the total number of hotspots in the entire training set.

The purpose of the proposed active sampling approach in Section 2.2.2.2 is to balance the dataset by selectively choosing tentative hotspots to be included in the training set. Here, we compare random data selection and the proposed data selection scheme. In the experiments, we set m_0 to 300 and m_k to 100 in Algorithm 2.1. SVM takes 22.3s at each iteration on average. Table 2.2 displays the number of total sampled data (columns "#All") and the number of hotspots (columns "#H") selected by the two schemes when setting 1400 as the maximum allowable size of training samples for both schemes. It is observed that the active learning scheme converges before reaching the size limit for `Layout2` and `Layout4&5`.

Table 2.2: Comparison of different sampling strategies.

Design	Random				Active			
	#All	(%)	#H	(%)	#All	(%)	#H	(%)
`Layout2`	1,400.0	25.6	44.0	25.3	1,050.0	19.2	172.7	99.3
`Layout3`	1,400.0	25.2	222.3	24.5	1,400.0	25.2	886.3	97.5
`Layout4&5`	1,400.0	19.2	23.4	19.3	1,190.0	16.3	101.5	83.9

Varying the maximum training set size n in Algorithm 2.1, the comparison of the two sampling schemes is shown in Figure 2.5. The figure shows that, with the same number of training samples, the proposed approach can achieve higher accuracy compared to the random sampling. Note that the accuracy is based on the GPR direct prediction results without lithography simulations. This is in fact due to the higher number of hotspots available in the training data when using the active sampling scheme compared to the random sampling strategy as demonstrated also in Figure 2.5. Moreover, one can easily notice that the iterative SVM evaluations are capable of detecting most of the hotspots in the dataset within a few iterations.

2.2.3.2 Validation of Gaussian Process

We demonstrate the effectiveness of the proposed GPR with validation simulations for hotspot detection. Table 2.3 shows the comparison between the state-of-the-art method [131] and our method, in terms of accuracy (ACC) and the number of false alarms (#FA). In this table, "All" denotes model training uses all the training samples in the benchmark, while "Random" and "Active" denote the training data obtained from random sampling and the proposed active sampling scheme in Table 2.2. For the method [131], we strictly use its DCT representation and CNN structure for the comparison. To further demonstrate that softmax output of CNN is not a good uncertainty measure, we compare the performance of CNN and GPR after performing the same number of validation simulations (VS). For GPR, threshold κ in Equation (2.2) is set to -0.2; the confidence metric α is set to 0.682, which is equivalent to one standard deviation confidence interval for a Gaussian distribution. According to this criterion, any untrusted sample needs to be further verified through lithography simulation. Since there is no well-defined metric to quantify confidence interval for CNN, to ensure fairness, we perform the same number of validation simulations to the test samples which has nearly the same softmax probability of being hotspot/non-hotspot and then compare the accuracy and the number of false alarms; that is, we choose the samples which minimize $|\text{softmax}(\text{NH}) - \text{softmax}(\text{H})|$. Column "#Sim" gives the ratio of the number of validation simulations to the testing data size.

Table 2.3 shows that the state-of-the-art work [131] using all the train-

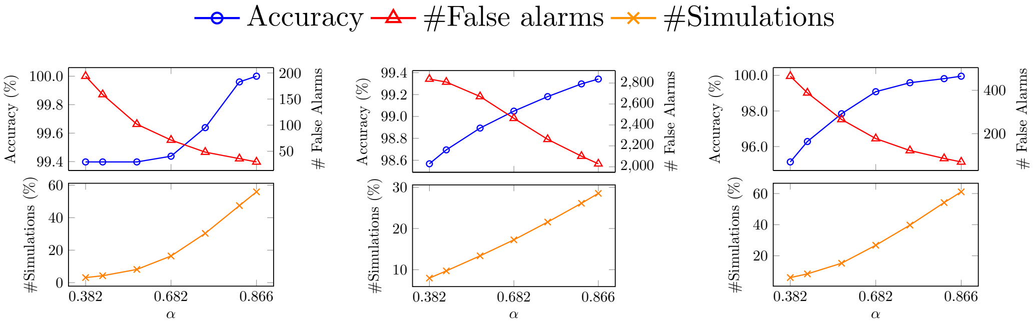
Table 2.3: Comparison of different flows in terms of accuracy and false alarms. The results are averaged over ten runs.

Design	All + [131]		Random + [131]		Active + [131]		Active + [131] + VS		Active + GPR		Active + GPR + VS		#Sim
	#FA	ACC (%)	#FA	ACC (%)	#FA	ACC (%)	#FA	ACC (%)	#FA	ACC (%)	#FA	ACC (%)	(%)
`Layout2`	234.1	97.4	370.9	91.3	1,030.7	99.4	733.3	99.6	502.8	99.1	71.4	99.4	16.4
`Layout3`	3,064.1	98.3	3,333.4	97.7	6,716.3	99.1	5,189.7	99.5	4,443.2	98.3	2,463.4	99.0	17.2
`Layout4&5`	443.4	91.7	512.5	64.2	1,598.4	96.3	1,162.3	98.9	1,130.2	91.2	177.5	99.1	26.8
Average	1,247.2	95.8	1,405.6	84.4	3,115.1	98.2	2,361.8	99.3	2,025.4	96.2	904.1	99.2	20.1
Ratio	1.0	1.0	—	—	—	—	1.89	1.04	—	—	0.72	1.04	—

ing dataset (All + [131]) achieves 95.8% accuracy on average. Our proposed active learning data selection further improves the accuracy of its model to 98.2% (Active + [131]). However, the average number of false alarms of this flow increases from 1247.2 to 3115.1. Active data selection together with our GPR method (Active + GPR) gives a similar accuracy (96.2%) as the state-of-the-art result. Moreover, given the strength of providing confidence of GPR, the accuracy (Active + GPR + VS) is improved to 99.2% after performing validation simulations, and meanwhile, it reduces the number of false alarms by 28% compared with the All + [131] flow. Compared with the Active + [131] + VS flow, the Active + GPR + VS flow obtains comparable accuracy and 2.6× false alarm reduction, which demonstrates the effectiveness of employing confidence measure provided by GPR. In the experiments, GPR training takes 296.6s, 1490.5s and 235.4s on average for the three benchmarks while testing takes 579.4s, 1342.2s and 586.7s.

2.2.3.3 Control of Prediction Confidence

Lastly, we explore the effect of α to control the desired prediction confidence. Figure 2.6 plots the testing accuracy after validation simulations and the percentage of simulated testing samples using different values of α. The accuracy reflects that of the trusted GPR predictions in addition to the instances validated through simulation. As one would expect, larger α values translate to better results in terms of accuracy and false alarms at the expense of higher simulation cost. It is important to note that the choice of α gives



(a) Layout 2 (b) Layout 3 (c) Layout 4&5

Figure 2.6: The testing accuracy, number of false alarms and percentage of simulated testing samples for different α are shown.

the user the flexibility to control the trade-off between the overall detection quality and the number of simulations needed.

2.2.4 Summary

In this section, we have presented Litho-GPA, a hotspot detection framework with Gaussian Process assurance to provide confidence in classifier prediction. The prediction accuracy is improved by exploring both the mean and confidence of prediction. Besides, an active data selection scheme based on weak classifiers is developed to reduce the computational cost in data preparation. Experimental results demonstrate Litho-GPA can achieve comparable accuracy to the state-of-the-art deep learning approaches while obtaining on average 28% reduction in false alarms.

2.3 Explicit ROC Optimization for Lithography Hotspot Detection

One special characteristic of lithography hotspot detection tasks is the imbalance in the layout datasets [132]. Despite the fact that the lithography defects are critical, their relative number is significantly small across the whole chip after various resolution enhancement techniques are applied. Ideally, we would like to have a model with a high true positive rate (TPR) and a low false positive rate (FPR), but in real-world scenarios, there is always a trade-off between the two metrics. Assume there are two classifiers at hand. The first classifier successfully detects more hotspots than the second classifier, but it also generates significantly more false alarms. It is hard to conclude which one is better because we cannot tolerate such a high number of non-hotspot clips falsely identified as hotspots. It is a waste of time and efforts to fix those safe clips. A robust performance evaluation and model selection for imbalanced learning problems have been often accomplished with the support of the receiver operating characteristic (ROC) curve which represents the relationship between the true positive rate and the false positive rate of a family of classifiers resulted from different decision thresholds [110]. Hence, the area under the ROC curve (AUC) is a more proper model evaluation criterion in the sense of being a global metric for all thresholds regardless of class prior probabilities.

Most existing methods still minimize misclassification error such as cross-entropy during training while using certain class balancing techniques.

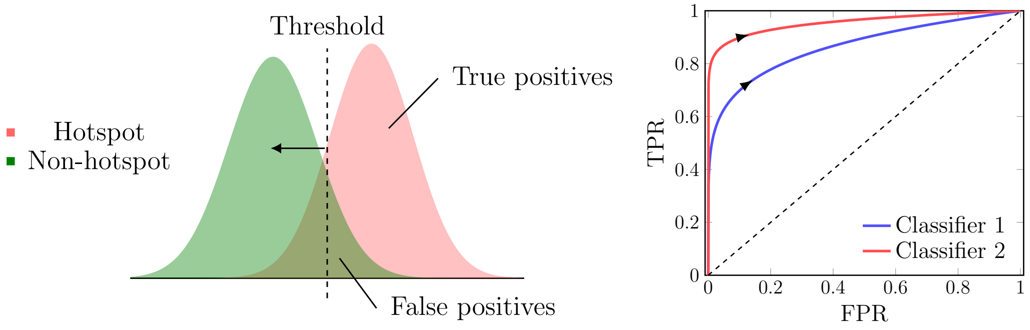
The most straightforward and common approach dealing with imbalance is the use of sampling methods. Undersampling and oversampling methods operate on the training data to improve its balance. Other techniques, including cost-sensitive learning and threshold moving, tackle the class imbalance on the level of the classifier and adjust training or inference algorithms. Since AUC has been widely used to measure performance for binary classification tasks especially on imbalanced datasets, the question then arises: is it possible to use AUC explicitly as the loss function in order to systematically handle the class imbalance problem?

In this work, we examine the effectiveness of directly optimizing a surrogate of AUC to boost the performance of neural network models when facing class imbalance.

2.3.1 Preliminaries

2.3.1.1 ROC Curve and AUC Score

For binary classification tasks, in order to separate the positive class from the negative class, a decision threshold is usually defined to map the continuous predicted score given by the model to a binary category. For each setting of the decision threshold (Figure 2.7(a)), a pair of true-positive rate and false-positive rate values is obtained. By varying the decision threshold over the range [0, 1], the ROC curve showing the relationship between true positive rate and the false positive rate can be obtained (Figure 2.7(b)). Moreover, as Figure 2.7(a) demonstrates, if the predicted score implies the classifier's belief



(a)

(b)

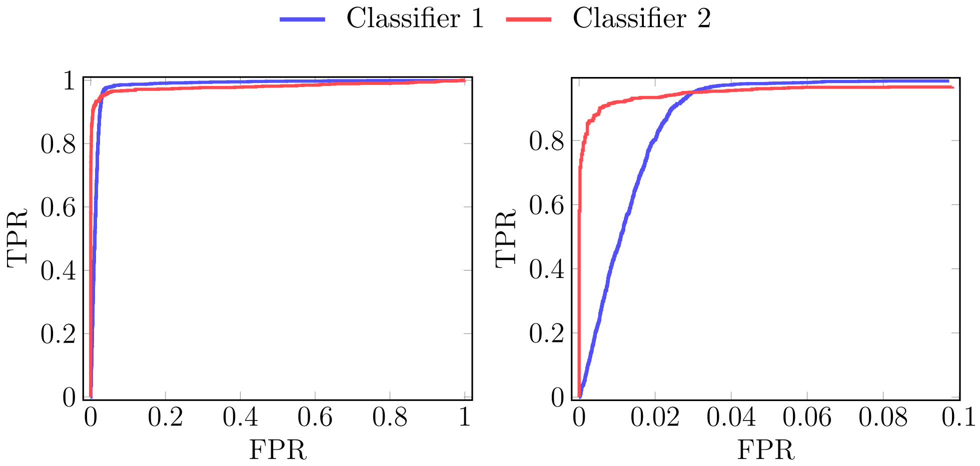
Figure 2.7: (a) An overlapping distribution of predicted scores for positive and negative samples and (b) the ROC curves of two example classifiers. As the threshold in (a) moves to the left, both FPR and TPR in (b) go up accordingly.

that an sample belongs to the positive class, decreasing the decision threshold (e.g., moving the threshold to the left) will increase both true and false positive rates.

AUC is a threshold-independent metric which measures the fraction of times a positive instance is ranked higher than a negative one [39, 110]. Unlike single point metrics, the ROC curve compares classifier performance across the entire range of class distributions, and therefore, the AUC score is a general measure of classifier discrimination performance. Figure 2.7(b) presents two ROC curves. The closer the curve is pulled towards the upper left corner, the better is the classifier's ability to discriminate between the two classes. Therefore, in Figure 2.7(b), classifier 2 has a better performance than classifier 1.

2.3.1.2 Partial AUC Score

The AUC metric traces classifier performance across all thresholds. However, it may summarize over regions of the ROC curve in which one would never operate. For hotspot detection tasks, the primary goal is to detect all possible hotspots. Nevertheless, a practical classifier is not allowed to accomplish the goal at the expense of introducing too many false alarms; the time and money costs associated with fixing those false alarm hotspots render the classifier less favorable than the traditional simulation approach. In this case, our interest is to see the classifier's ability to detect hotspots in the region of the ROC curve corresponding only to acceptably low FPRs.



(a) (b)

Figure 2.8: Comparison of the ROC curves over (a) the entire FPR range and (b) the FPR range of interest.

To elaborate on this, consider the two classifiers shown in Figure 2.8. Classifier 1 has better AUC than classifier 2 according to Figure 2.8(a). But if we zoom into the region of interest (e.g., FPR less than 2%) in Figure 2.8(b), classifier 2 has better overall TPR in this region and it outperforms classifier 1. Therefore, besides measuring the overall AUC score of the classifier, we look into the partial AUC defined in the following way [32, 86]:

$$\widetilde{\mathrm{AUC}}(t_0, t_1) = \int_{t_0}^{t_1} \mathrm{ROC}(t)\mathrm{d}t, \tag{2.3}$$

where the interval (t_0, t_1) denotes the false positive rate region of interest. We can further scale the partial AUC and derive the normalized partial AUC given by [86]

$$\mathrm{AUC}(t_0, t_1) = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} \mathrm{ROC}(t)\mathrm{d}t. \tag{2.4}$$

2.3.1.3 Handling Class Imbalance

Due to the fact that the lithography hotspots are critical, various resolution enhancement techniques are applied to significantly reduce their relative number. Therefore, when a grid scheme is used to extract images from the design, only a small number of images will encompass lithography hotspots while the majority will correspond to sites in the design with no defects. This poses a major challenge when formulating the task as a learning problem.

The class imbalance problem is encountered in many application domains. It has been established that in certain cases, class imbalance hinders the performance of standard classifiers [48], in terms of training convergence and generalization of the model. Sometimes the classifiers even achieve a low error rate by trivially predicting each sample to be negative when the dataset is biased towards the negative class.

Various methods have been proposed to deal with the class imbalance problem. Among them, oversampling and undersampling alter the distribution of training data to make it more balanced. **Undersampling** removes samples from the majority class until all classes have the same amount of data. For example, one-side selection carefully identifies and removes redundant examples close to the boundary between classes [61]. A major disadvantage of undersampling is that it discards potentially useful training samples. Therefore, undersampling is rarely adopted for hotspot detection tasks because those training datasets are highly imbalanced but far from abundance. **Oversampling** is one of the most commonly used methods. It simply replicates randomly selected samples from minority classes, but this approach can increase the time necessary to build a classifier, and may even lead to overfitting [20]. Advanced sampling methods such as SMOTE [23] and its variant [41] create artificial examples by interpolating neighboring data points. In addition, cluster-based oversampling first clusters the dataset and then oversamples each cluster separately [50]. In this way, both between-class and within-class imbalances are reduced. Since the input data samples of hotspot detection tasks are images and optical sources are symmetric, [72, 130] augment the training data with rotation and flipping; besides, although general convolutional neural networks (CNNs) are not rotate invariant, data augmentation by rotation and flipping can help obtain some rotation invariance.

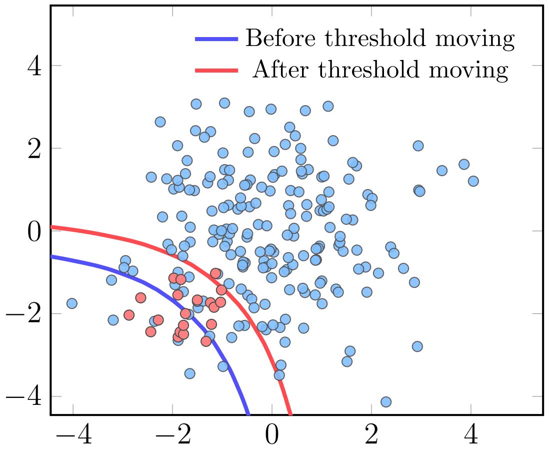


Figure 2.9: Example of threshold moving.

Cost sensitive learning assigns different cost to the misclassification of samples from different classes [34, 120]. For hotspot detection tasks, this is done by associating a greater cost with false negatives than with false positives. [28, 53] study cost sensitive learning of deep neural networks. [118] proposes a new loss function for neural network training to make the networks more sensitive to the minority class. To incorporate the cost sensitivity into neural networks, one can place a heavier penalty on misclassifying the minority class in the loss function such that minority class contributes more to the update of weights. And then, we can train the network by minimizing the misclassification cost instead of the standard loss function.

Threshold moving adjusts the decision threshold of a classifier to cope with the class imbalance problem. This approach is usually applied in the test phase. As demonstrated in Figure 2.9, it moves the threshold toward the majority class such that samples from the minority class become harder

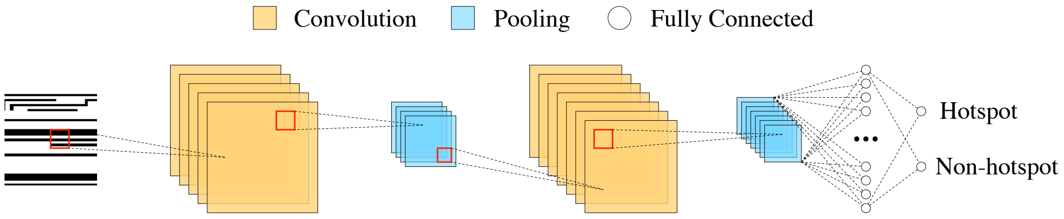


Figure 2.10: Example illustration of convolutional neural network architecture for hotspot detection.

to be misclassified. For traditional machine learning methods, adjustment of the decision boundary is straightforward. For example, it can be done by shifting the bias in a support vector machine (SVM) model. However, it is less practical to move the decision threshold directly when using neural network based classifiers because these networks tend to be overconfident in their prediction; the softmax outputs of the two neurons in last fully-connected layer shown in Figure 2.10 are usually very close to 1 and 0. As it is hard to control the appropriate shift amount, this method may take effect at cost of a large number of false alarms. Instead, [131] biases the ground truth for negative samples from 0 to ϵ during the training phase.

Other approaches explore different training methods specific to neural networks. [44] proposes a two-phase training method which first trains the network on the balanced set and then fine-tunes the output layers. The aforementioned approaches to tackle class imbalance either operate on training data or adjust training or inference methods. As we will demonstrate in the next section, AUC can be interpreted as a ranking measure; that is, the AUC is equal to the probability of ranking a random positive sample over a random negative sample. Therefore, orderings of data samples by the predicted probabilities is consistent even in the face of class imbalance. In this sense, both the shape of the ROC curve and AUC are insensitive to the class distribution. The question then arises, given that AUC is a robust measure of classification performances especially for imbalanced problems, is it possible to develop algorithms that directly optimize this metric during the training phase? In other words, can we optimize the ROC curve explicitly?

2.3.1.4 Problem Formulation

Traditionally, accuracy (i.e., true positive rate [16]) and the number of false alarms (i.e., false positives) are the two prevailing metrics used for detection evaluation. Hence, the traditional hotspot detection problem is usually defined as:

Problem 2.3.1 (Hotspot detection for accuracy optimization)**.** Given a set of layout clips consisting of hotspot and non-hotspot patterns, the object of hotspot detection is to train a classifier that maximizes the accuracy and minimizes the number of false alarms on the testing dataset.

As we demonstrated in Section 2.3.1.1, evaluation of hotspot detection models using accuracy and false alarms separately is not effective, because it is hard to find a good trade-off between the two metrics. Therefore, we propose to assess hotspot detection models using the holistic metric, AUC. Furthermore, the model is trained with the goal of optimizing the ROC curve in the form of maximizing the normalized partial AUC score.

Problem 2.3.2 (Hotspot detection for ROC optimization)**.** Given a set of layout clips consisting of hotspot and non-hotspot patterns, the object of hotspot detection is to train a classifier that maximizes the normalized partial AUC score on the testing dataset.

2.3.2 ROC Optimization

In this section, we derive the AUC with dedicated loss functions for AUC optimization, and compare them with the cross entropy loss.

2.3.2.1 AUC Objective and Loss Functions

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is i-th data sample in the feature space and $y_i \in \{-1, +1\}$ is the true class label of $\mathbf{x}_i$, we can further divide the dataset $\mathcal{D}$ into two sets: the set of positive samples $\mathcal{D}_+ = \{(\mathbf{x}_i^+, +1)\}_{i=1}^{N^+}$ and the set of negative samples $\mathcal{D}_- = \{(\mathbf{x}_i^-, -1)\}_{i=1}^{N^-}$, where $N+$ and $N-$ denote the number of positive and negative samples respectively, and $N = N_+ + N_-$. Let $f(\mathbf{x})$ denote the prediction model. It has been proven that AUC is equivalent to the Wilcoxon-Mann-Whitney (WMW) statistic test of ranks in the following sense [43, 83, 122]:

$$\text{AUC} = \frac{1}{N_+ N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} I(f(\mathbf{x}_i^+) > f(\mathbf{x}_j^-)), \tag{2.5}$$

where $I(f(\mathbf{x}_i^+) > f(\mathbf{x}_j^-))$ is the indicator function given by

$$I(f(\mathbf{x}_i^+) > f(\mathbf{x}_j^-)) = \begin{cases} 1, & \text{if } f(\mathbf{x}_i^+) > f(\mathbf{x}_j^-), \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

AUC averages the score of a positive sample having a higher probability than a negative sample for all between-class pairs; it can also be viewed as the probability that a positive sample is ranked higher than a negative sample. This statistical interpretation led to the capability of computing AUC without building the ROC curve itself, by counting the number of positive-negative example misorderings in the ranking produced by classifier scores [125]. However, AUC defined in Equation (2.5) is a sum of indicator functions which is non-differentiable, to which gradient-based optimization methods cannot be applied. In order to make the problem tractable, it is necessary to apply convex relaxation to the AUC. By replacing $I(f(\mathbf{x}_i^+) > f(\mathbf{x}_j^-))$ in Equation (2.5) with pairwise convex surrogate loss $\Phi(f(\mathbf{x}_i^+) - f(\mathbf{x}_j^-))$, we can minimize the loss defined below as a way to maximize the AUC score:

$$\mathcal{L}_\Phi(f) = \frac{1}{N_+ N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} \Phi(f(\mathbf{x}_i^+) - f(\mathbf{x}_j^-)). \tag{2.7}$$

Various surrogate loss functions can be chosen here. Let $z = f(\mathbf{x}_i^+) - f(\mathbf{x}_j^-)$, then the pairwise squared loss (PSL), one of the most commonly used surrogate loss functions, is given by [31, 35]

$$\Phi_{\text{PSL}}(z) = (1 - z)^2. \tag{2.8}$$

In [109, 144], pairwise hinge loss (PHL) is used as a surrogate function:

$$\Phi_{\text{PHL}}(z) = \max(1 - z, 0). \tag{2.9}$$

Similarly, [103] utilizes the pairwise logistic loss (PLL) to replace the indicator function:

$$\Phi_{\mathrm{PLL}}(z) = \log(1 + \exp(-\beta z)). \tag{2.10}$$

[128] proposes the differentiable function given by the following expression as the surrogate loss:

$$\Phi_{\mathrm{R}^*}(z) = \begin{cases} -(z-\gamma)^p, & \text{if } z > \gamma, \\ 0, & \text{otherwise}, \end{cases} \tag{2.11}$$

where $0 < \gamma \leq 1$ and $p > 1$, and suggests that $p = 2$ or 3 generally achieves the best results. Based on the observation that maximizing the objective with Φ in the form of Equation (2.11) is ineffective to maximize the WMW statistic because it focuses on maximizing the difference between $f(\mathbf{x}_i^+)$ and $f(\mathbf{x}_i^-)$ instead of moving more pairs of $f(\mathbf{x}_i^+)$ and $f(\mathbf{x}_i^-)$ to satisfy $f(\mathbf{x}_i^+) - f(\mathbf{x}_i^-) > \gamma$, the authors further propose another function,

$$\Phi_{\mathrm{R}}(z) = \begin{cases} (-(z-\gamma))^p, & \text{if } z < \gamma, \\ 0, & \text{otherwise}. \end{cases} \tag{2.12}$$

Figure 2.11 demonstrates the comparison of the four surrogate functions. One can notice that the curve of function R is flat in the region $[\gamma, 1]$, which differentiates it from other three curves. The key idea is, during the process of minimizing $\mathcal{L}$ in Equation (2.7), if a positive sample has a higher output than a negative sample by margin γ, this pair of samples will not contribute to the objective.

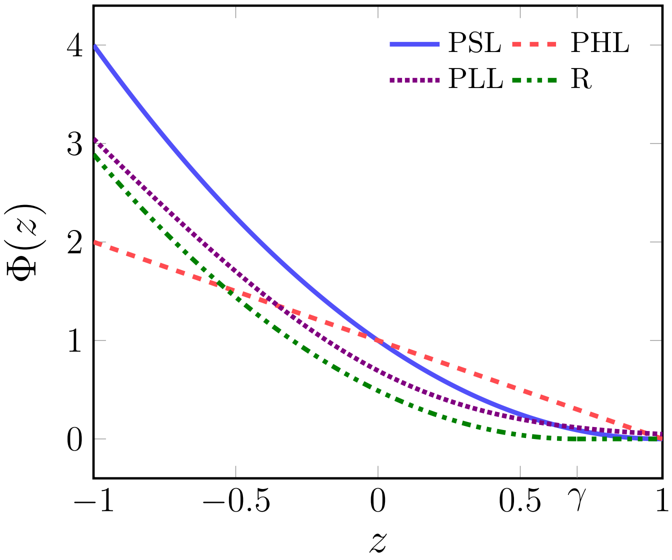


Figure 2.11: Comparison of the four surrogate functions, where $\beta = 3$ in PLL, and $\gamma = 0.7$ and $p = 2$ in R.

2.3.2.2 Comparison with Cross-Entropy Loss

Classifiers such as neural networks typically use cross-entropy (or log-loss) as the cost function. Cross-entropy (CE) loss for binary classifiers is defined as:

$$\mathrm{CE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log f(\mathbf{x}_i) + (1 - y_i)\log(1 - f(\mathbf{x}_i)). \tag{2.13}$$

During the optimization process, CE in Equation (2.13) moves $f(\mathbf{x}_i^+)$ closer to 1 and $f(\mathbf{x}_i^-)$ to 0, while AUC in Equation (2.5) tries to force $f(\mathbf{x}_i^+) > f(\mathbf{x}_i^-)$. One might consider a weak relationship between CE and AUC, but in general the two objectives are quite different. Cross-entropy takes into account the uncertainty of the prediction based on how much the probability estimates vary from the actual labels, and it has been used when calibration is important [36]. Whereas, AUC is a rank statistic and is only affected by the ranking of the samples induced by the predicted probabilities. The order of the samples can be maintained while changing their probability values.

For the hotspot detection problems where positive labels are few but significant, we seek models that are able to predict positive classes more correctly. Table 1 displays an example dataset containing ten data samples with only two positive labels, and two models provide their predicted scores for each sample. As one can see, the two models only behave differently on sample 8 and 9. Model 1 correctly classifies sample 9 as positive, and model 2 correctly classifies sample 8 as negative. Model 1 is better than model 2 for hotspot detection tasks in the sense that it captures all the hotspots correctly even with one false alarm, while model 2 achieves zero false alarms but misses one hotspot.

Here we compare AUC with CE to see how differently they distinguish the two corresponding models when facing class imbalance. The CE scores for the two models are both 0.36. Clearly, cross-entropy believes the two models are performing equally. However, the AUC scores of the two models are 0.94 and 0.75 respectively, and hence, the AUC metric prefers model 1 over model 2. Cross-entropy fails in this case because the loss function in Equation (2.13) is symmetric and does not differentiate between classes. AUC captures the difference in classifying the imbalanced class and thus suits better for class imbalance.

Table 2.4: Comparison of cross-entropy and AUC for model selection on imbalanced dataset.

Sample No.	1	2	3	4	5	6	7	8	9	10
Label	0	0	0	0	0	0	0	0	1	1
Model 1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.8	0.8	0.8
Model 2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.8

2.3.3 Experimental Results

We implement the LithoROC framework in Python with the TensorFlow library [8]. The effectiveness of AUC as the optimization objective for neural networks is validated on the ICCAD 2012 CAD contest benchmark set [116]. Table 2.1 summarizes the benchmark information, the number of all the clips (#All) and the number of hotspot clips (#H) in the training set (Train) and testing set (Test). We configure the CNN architecture in a way similar to [131], which gives the state-from-the-art performance for hotspot detection. Each training process is repeated five times on the same dataset with different random seeds, and the average results on the testing set are shown in this section.

Table 2.5 demonstrates the impact of different loss functions on classification performance. The CNN model in [131] is updated at each step using the mini-batch gradient descent method which randomly picks a group of instances. To overcome the bias towards the majority class during the training process, [131] fixes the batch size to 32 and ensures that the number of positive samples and negative samples are the same in each mini-batch. In addition to following this mini-batch configuration, we explore the impact of imbalanced mini-batches by setting the the class ratio of positive samples per mini-batch to 0.1 and 0.4 respectively. To ensure the number of hotspots is not too small in each batch, the batch size is increased to 64.

There are four convex surrogate loss functions discussed in Section 2.3.2 and we choose the two representative loss functions, the pairwise square loss for AUC maximization (AUC-PSL) in Equation (2.8), and the R loss for AUC maximization (AUC-R) in Equation (2.12). We compare the two losses with the traditional cross-entropy loss (CE). Here we set $\gamma = 0.7$ and $p = 2$ in Equation (2.12). Table 2.5 shows the normalized partial AUC score on the testing data using different loss functions and different mini-batch configurations, where F(α) denote the the normalized partial AUC score given by Equation (2.4) over the FPR range $[0, \alpha]$. We consider $\alpha = 0.01$, 0.02 and 1, because the FPR reported in the recent literature is around 0.01 to 0.02 [131]. Reporting the results for $\alpha = 1$ is to show the difference in the AUC score and the partial AUC score. In Table 2.5, the state-of-the-art classifier from [131] uses CE as the objective function, and sets the batch size to 32 and the ratio of positive examples to 0.5. Its performance for hotspot detection is near saturation, but we can still observe utilizing AUC as the objective function for training the CNN model helps advance the performance of the model under low false positive rates, especially on design `ICCAD3`.

Figure 2.12 presents the ROC curves for design `ICCAD3`. The mean ROC curve of the five runs and the corresponding variance of the curve within

Table 2.5: Comparison between different loss functions for AUC objectives.

Loss	Batch		ICCAD1			ICCAD2			ICCAD3			ICCAD4			ICCAD5		
	Size	Ratio	F(0.01)	F(0.02)	F(1)	F(0.01)	F(0.02)	F(1)	F(0.01)	F(0.02)	F(1)	F(0.01)	F(0.02)	F(1)	F(0.01)	F(0.02)	F(1)
CE	32	0.5	51.0	52.0	88.9	96.6	97.8	99.7	59.9	69.7	98.2	89.9	92.1	98.8	93.0	94.2	98.3
	64	0.1	50.8	51.7	88.1	95.9	97.4	99.7	60.1	70.2	98.2	89.4	92.2	98.9	91.7	92.9	98.2
	64	0.4	50.9	51.8	88.6	95.8	97.3	99.8	61.1	71.7	98.3	90.0	92.7	98.8	92.6	93.7	98.2
AUC-PSL	32	0.5	51.2	52.5	89.2	93.7	95.8	99.6	59.3	68.7	98.0	**90.5**	92.5	98.7	**93.7**	**94.9**	98.6
	64	0.1	51.6	53.3	91.6	88.0	93.4	99.5	58.5	67.0	97.9	86.3	90.8	98.4	92.6	94.7	98.8
	64	0.4	51.6	53.2	91.8	95.6	97.1	99.7	59.9	69.8	98.1	90.2	92.1	98.3	92.3	93.4	98.5
AUC-R	32	0.5	52.9	**55.5**	91.1	96.4	97.7	99.7	60.6	71.0	98.2	**90.5**	**93.0**	98.9	93.4	94.8	98.1
	64	0.1	52.1	53.8	88.9	96.1	97.7	99.7	**64.4**	**74.4**	98.4	89.6	92.2	98.7	93.1	94.6	98.4
	64	0.4	**53.0**	55.1	90.2	**96.9**	**98.0**	99.7	60.1	71.1	98.3	90.0	92.4	98.9	93.1	94.4	98.4

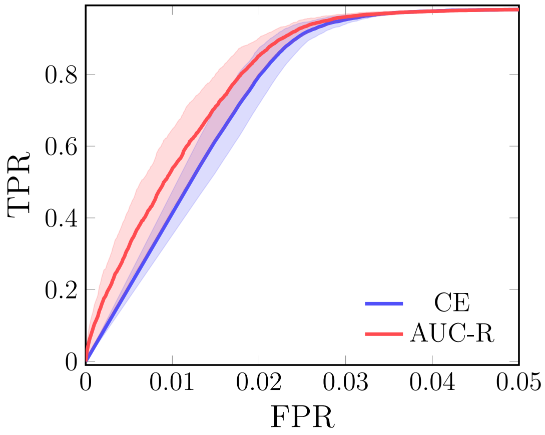
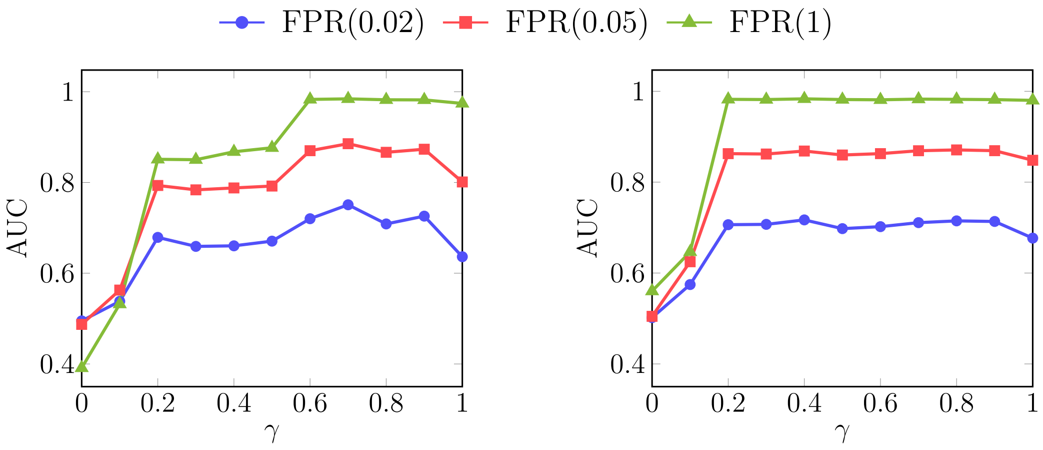


Figure 2.12: Comparison of ROC curves with different loss functions.

±1 standard deviation are shown. One can see that the objective function AUC-R generates a significantly better ROC curve than that of CE. A natural question is then how to choose the margin parameter γ in Equation (2.12). Figure 2.13 plots the AUC score versus the γ for various FPR ranges. To show the difference between curves, instead of using FPR(0.01), FPR(0.02), FPR(1), we use FPR(0.02), FPR(0.05), FPR(1), as the curves for FPR(0.01) and FPR(0.02) are very close. As noted in Figure 2.13, when γ increases from 0 to 0.5, the three AUC scores rise as well. That is because CNN is typically overconfident in its predictions in the sense that the output of the last fully-connected layer after softmax is very close to 0 or 1. In this way, it is over-simple for the between-class sample pairs to satisfy the constraint that a positive sample has as higher output than a negative sample by γ, which actually does not help guide the model to a good optimum. When γ is large enough, the AUC scores for the test data are relatively insensitive.



(a) Hotspot class ratio = 0.1

(b) Hotspot class ratio = 0.4

Figure 2.13: The normalized partial AUC scores at different γ on design `ICCAD3` testing data.

2.3.4 Summary

In this work, we have proposed to use AUC as a robust measure of classifier discrimination performance for hotspot detection tasks. Different surrogate loss functions for AUC maximization are proposed to be used during training to systematically handle the class imbalance problem. Experimental results demonstrate that the new loss functions are promising to outperform the traditional cross-entropy loss when applied to the state-of-the-art neural network model for hotspot detection.

Chapter 3

Lithography Modeling for Efficient Manufacturing Closure

3.1 Introduction

Lithography holds a fundamental position in today's semiconductor manufacturing [124]. It transfers a designed mask pattern into a resist pattern on the top surface of a semiconductor wafer [64, 80]. A typical lithography system consists of four key components: illumination source, mask, lens, and wafer, as shown in Figure 3.1. The illumination source sheds light through the mask and exposes the wafer such that a variety of patterns are printed.

In practice, lithography simulations have been effectively used for process development, performance prediction and a number of other tasks includ-

This chapter is based on the following conference papers.

1. Wei Ye, Mohamed Baker Alawieh, Yibo Lin, and David Z. Pan. "LithoGAN: End-to-end lithography modeling with generative adversarial networks." In 2019 56th ACM/IEEE Design Automation Conference (DAC), pp. 1-6. IEEE, 2019.
2. Wei Ye, Mohamed Baker Alawieh, Yuki Watanabe, Shigeki Nojima, Yibo Lin, and David Z. Pan. "TEMPO: Fast Mask Topography Effect Modeling with Deep Learning." In Proceedings of the 2020 International Symposium on Physical Design (ISPD), pp. 127-134. 2020.

I am the main contributor in charge of problem formulation, algorithm development, and experimental validations.

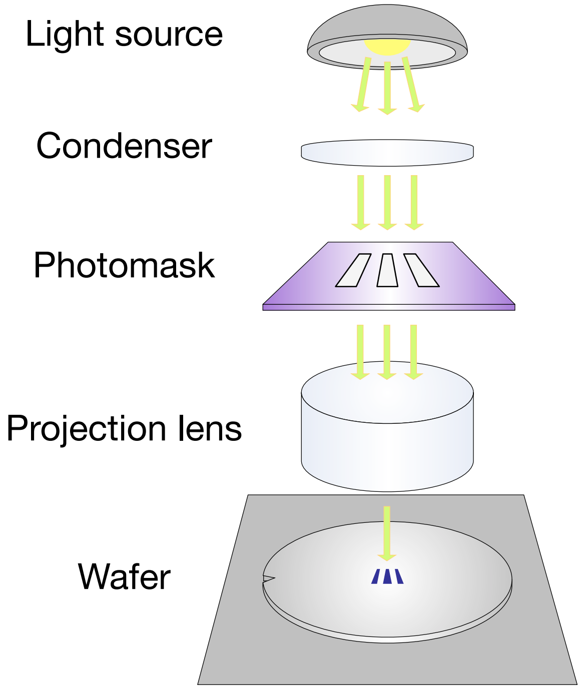


Figure 3.1: Typical lithography system.

ing model-based optical proximity correction (OPC). These simulations are utilized to calculate correct resist shapes that can be used for physical verification such as hotspot detection. However, as the technology node continues scaling down, the trend to print features much smaller than the wavelength of light used has tremendously increased lithographic and manufacturing process complexity, as well as the lithography modeling complexity.

In the rest of this chapter, Section 3.2 introduces lithography modeling leveraging machine learning techniques, and Section 3.3 explores mask topography effects in lithography simulation.

3.2 End-to-End Lithography Modeling

Lithography simulation mainly falls into two categories: physics-level rigorous simulation and compact model-based simulation. Rigorous simulation precisely simulates the physical effects of materials to obtain the printed patterns [77, 112]. In practice, the physical properties of photoresist (resist) and optical systems, the mask patterns, and the process variations are all correlated to the printing. As a rigorous model has to include these cross-related quantities, it is computationally expensive. Also, the calibration of lithography models can take several weeks at advanced technology nodes. In VLSI manufacturing, modeling efficiency is crucial for fast design closure along with modeling accuracy. Therefore, compact models stand as a speedup alternative to rigorous computation with a small sacrifice in accuracy.

Figure 3.2 shows a typical flow of lithography simulation. First, an aerial image is generated from a mask pattern using an optical model which is characterized by the illumination type and projection lenses of an exposure tool. Then a resist model is used to determine the locally varying slicing thresholds [99]. Lastly, the thresholds are processed through extrapolation together with the corresponding aerial image to evaluate the critical dimension (CD) of the printed patterns or to generate the resist contours.

Although conventional variable threshold resist (VTR) models are highly efficient, they fail to keep up their accuracy at advanced technology nodes [119]. To improve simulation quality, machine learning based techniques have been proposed to construct accurate and efficient resist models [71, 72, 105, 119].

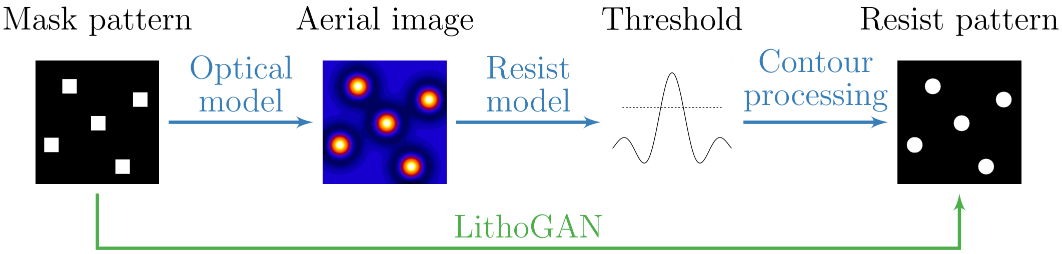


Figure 3.2: Conventional lithography simulation flow consisting of multiple stages and the proposed LithoGAN flow.

These approaches first take a set of training data to train (calibrate) a model and then use this model to make predictions on test data. [105] proposes an artificial neural network (ANN) to predict the height of resist after exposure. However, efforts are spent on determining the appropriate set of features for model training. To overcome the explicit feature extraction, [119] proposes a convolutional neural network (CNN) model that predicts the slicing thresholds in aerial images accurately. Recently, [72] proposed a transfer learning scheme together with an active learning approach to cope with the deficiency in the manufacturing data at advanced technology nodes.

Nevertheless, several drawbacks exist in the mainstream compact models and machine learning approaches. The proposed resist models rely on optical simulation to generate aerial images, which are accompanied by a high computational cost. Additionally, only resist height or slicing threshold is predicted from the proposed models, which requires further processing to finalize the contour patterns. Hence, the state-of-the-art lithography modeling techniques still suffer from an exorbitant computational cost while providing partial modeling schemes that rely heavily on pre- and post-processing procedures.

In spite of various rigorous models and compact models at hand, it is extremely desirable to further improve lithography modeling efficiency without compromising much accuracy. Considering the fact that machine learning based approaches have demonstrated superior efficacy in a particular stage during lithography modeling, a natural question then arises: is it possible to build an end-to-end lithography model with machine learning techniques? Toward this goal, we propose LithoGAN, a novel lithography modeling framework based on conditional generative adversarial network (CGAN) that has demonstrated tremendous success in computer vision over the past few years [37,47,62,89,145]. CGAN manifests itself among numerous generative models with an inherent capability to perform image translation tasks such as image colorization and background masking, where an image in one domain is mapped to a corresponding image in another domain. In addition, CGAN has been adopted for optical proximity correction (OPC) enhancement in IC manufacturing [129].

Our proposed LithoGAN framework is the first complete end-to-end lithography modeling approach mapping the mask pattern at one end to the resist pattern at the other. This approach builds on a CGAN to translate an image from the layout to the resist shape. It turns out that this translation can achieve high accuracy in predicting the shape and size of the resist pattern. Moreover, to further boost the performance of the CGAN, LithoGAN integrates a CNN that can predict the pattern center to help with localization.

3.2.1 Preliminaries

An accurate end-to-end lithography model should produce patterns consistent with the manufactured (golden) ones. In order to evaluate the accuracy of a model, evaluation metrics are required to quantify the critical mismatches. Edge placement error (EPE) is a commonly used metric in lithography to characterize pattern fidelity [80,127]. Technically, EPE measures the Manhattan distances between the printed resist contours and the intended mask patterns at given measurement points. However, our focus is to measure the performance of the proposed LithoGAN framework where we expect a well-trained model to produce contours similar to the golden contours. In other words, the objective is not to optimize EPE, but rather to mimic the golden contours obtained from rigorous simulation. Hence, we propose a new measure, denoted as edge displacement error, which is tailored to our problem.

Definition 3.2.1 (Edge Displacement Error, EDE)**.** Given the bounding boxes of the golden and predicted contours respectively, the edge displacement error for a given edge in the bounding box is defined as the distance between the golden edge and the predicted one.

The definition of EDE is very similar to EPE, except that EDE is defined between two contours, while EPE is defined between a contour and a design target. Figure 3.3 illustrates how EDE measures the edge distance between the model predicted contour and the golden lithography contour. However, this measure is not effective in capturing the details of the mismatch

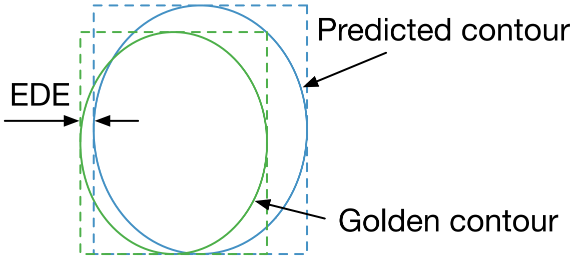


Figure 3.3: An illustration of the EDE evaluation metric.

between the two contours. While evaluating the quality of the contours is still an open problem, we introduce additional metrics to provide a comprehensive evaluation. Considering that the essence of the LithoGAN task is to predict the color of each pixel in a monochrome image, we adopt the metrics commonly used in computer vision tasks such as semantic segmentation [76].

In this work, three metrics are used to evaluate the quality of the synthesized image besides the EDE metric. For the generality of the terminology, we use class i to represent color i of a pixel in the following discussions. Let $p_{i,j}$ be the number of pixels of class i predicted to belong to class j, where $i, j \in \{0, 1\}$. Let $t_i = \sum_j p_{i,j}$ be the total number of pixels of class i.

Definition 3.2.2 (Pixel Accuracy)**.** Pixel accuracy is defined as the percentage of pixels in the image which are correctly classified, $(\sum_i p_{i,i})/(\sum_i t_i)$.

Definition 3.2.3 (Class Accuracy)**.** Class accuracy is defined as the average percentage of pixels in the image which are correctly classified for each class, $\frac{1}{2}\sum_i(p_{i,i}/t_i)$.

Definition 3.2.4 (Mean IoU)**.** Intersection over union (IoU) measures the number of pixels present in both the golden and predicted patterns (intersection) divided by the number of all pixels present in either of them (union). Mean IoU is an average of the IoU scores for all classes, $\frac{1}{2}\sum_i(p_{i,i}/(t_i - p_{i,i} + \sum_j p_{j,i}))$.

The proposed lithography modeling framework first builds a CGAN model using a set of layout clip pairs, where each pair includes a mask pattern and a resist pattern of the center contact as shown in Figure 3.4(a) and Figure 3.4(b) respectively. We define the CGAN-based end-to-end lithography modeling problem as follows.

Problem 3.2.1 (End-to-End Lithography Modeling)**.** Given a dataset containing the pairs of mask patterns and corresponding resist patterns of center contacts, the objective of end-to-end lithography modeling is to train a model that can accurately predict the resist pattern of the center contact based on a given mask pattern.

3.2.2 LithoGAN Framework

3.2.2.1 Data Preparation

For training the proposed framework, a dataset consisting of paired images corresponding to mask patterns and resist patterns is needed. Proper resolution enhancement techniques (RETs) such as sub-resolution assist feature (SRAF) generation and OPC have been applied to the original input mask clips of size 2 µm × 2 µm. Towards a better localization around the target con-

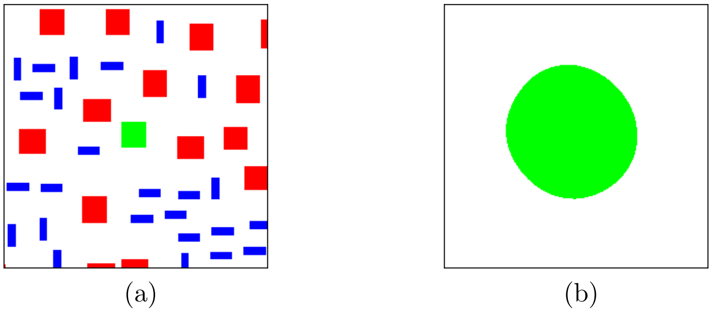


Figure 3.4: (a) Mask pattern and (b) resist pattern of target contact. Green rectangle denotes the center contact after OPC; red rectangles represent other contacts after OPC; blue rectangles denote the SRAFs.

tact, these clips are then cropped to 1 µm × 1 µm such that, in each clip, the target contact is located exactly at the center of the clip.

The obtained clips are converted to RGB images of size 256 × 256 pixels where the target contact of interest is encoded into the green channel, neighboring contacts are encoded into the red channel, and SRAFs are encoded into the blue channel. This coloring scheme, demonstrated by the example in Figure 3.4(a), maps the different types of objects to different colors to help the model discriminate these objects during the learning and inference processes. On the other hand, the target contact is designed to be 60 nm × 60 nm; hence, we use the window size 128 nm × 128 nm to crop the golden resist pattern of the target contact. Although synthesizing a 128 × 128 image might be enough for generating the pattern, the cost of misprediction could be high. For example, mispredicting 1 pixel may result in 1 nm error to the contour, hence, imposing an extremely high requirement to the model. Therefore, we scale the 128 nm × 128 nm clip to a monochrome image of size 256 × 256 pixels as in Figure 3.4(b) such that error from mispredicting 1 pixel is around 0.5 nm. Further improvement to the accuracy is possible by scaling the clip to larger images, but it may cause additional overhead in the modeling effort.

3.2.2.2 CGAN Architecture Design

GANs are deep neural networks that use a training dataset to learn the distribution of the input, typically images, and generate new images from the learned distribution. At the highest level, GANs consist of two networks that compete with each other: a generator and a discriminator [37]. The generator G generates fake samples to fool the discriminator, while the adversarially trained discriminator D distinguishes between real images and fake images generated by the generator. The competition throughout the training process drives both to improve: the discriminator guides the generator on what images to create, while also improving itself by learning what distinguishes real images from the fake ones from the generator.

At the end of the training process, the generator learns the distribution of the training data and is eventually able to generate real-looking images. On the other hand, it is hard for the discriminator to distinguish between training set images and generated images. After the GAN model converges, the role of the discriminator is over, and the main interest is in the generator who is now able to generate high-quality images. In this way, a GAN learns a generative model that maps a random noise vector $\mathbf{z}$ to output image $\widehat{\mathbf{y}}$: $\widehat{\mathbf{y}} = G(\mathbf{z})$.

Unlike the aforementioned unconditional GAN, the goal of a CGAN is to learn how to generate fake samples with a specific condition or characteristics rather than a generic sample purely based on random noise [89]. Specifically, for image translation tasks, both the generator and discriminator observe another input image $\mathbf{y}$. CGAN requires the generated image $G(\mathbf{x}, \mathbf{z})$ to not only fool the discriminator but also to be close to the ground truth output corresponding to the particular input image. Hence, in this work, we adopt this image translation idea proposed in [47].

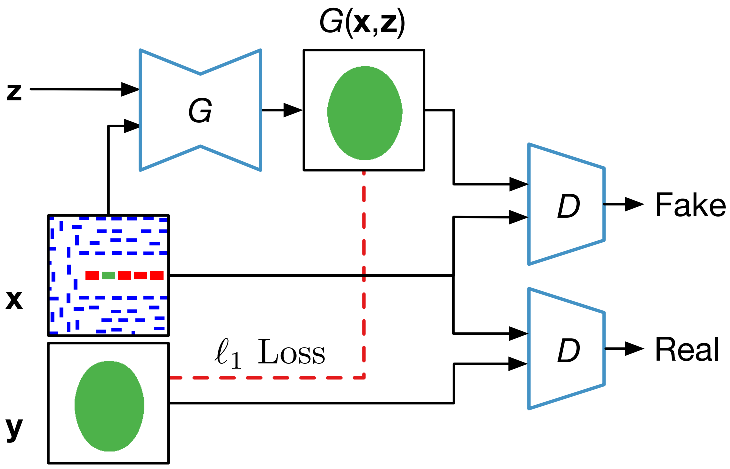


Figure 3.5: CGAN for lithography modeling.

Figure 3.5 shows the training process of our proposed lithography modeling CGAN. $\mathbf{x}$ represents the mask pattern image after SRAF insertion and OPC, and $\mathbf{y}$ represents the golden resist pattern of the target contact given by lithography simulation. The generator generates a fake resist pattern $G(\mathbf{x}, \mathbf{z})$ when fed with the input mask pattern $\mathbf{x}$. The discriminator is responsible for classifying this image pair $(\mathbf{x}, G(\mathbf{x}, \mathbf{z}))$ as fake, and meanwhile, it needs to predict the image pair $(\mathbf{x}, \mathbf{y})$ as real. Here "real" means that $\mathbf{y}$ is the output image corresponding to input $\mathbf{x}$. In other words, the target contact in $\mathbf{x}$ after resist development will become $\mathbf{y}$.

The discriminator outputs a value $D(\mathbf{x}, \mathbf{y})$ indicating the chance that ($\mathbf{x}$, $\mathbf{y}$) is a real pair. As demonstrated in Figure 3.5, the objective of the discriminator is to maximize the chance of recognizing the image pair ($\mathbf{x}$, $\mathbf{y}$) as real and the image pair ($\mathbf{x}$, $G(\mathbf{x}, \mathbf{z})$) as fake. Mathematically, the objective function of D is given by [37]

$$\max_{D} \quad \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{z})))]. \tag{3.1}$$

On the generator side, the objective is to generate images with the highest possible value of $D(\mathbf{x}, G(\mathbf{x}, \mathbf{z}))$ to fool the discriminator. Besides, the generator wishes that the generated image $G(\mathbf{x}, \mathbf{z})$ is close to the ground truth $\mathbf{y}$. The objective of G is defined as [47, 89]

$$\min_{G} \quad \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{z})))] + \lambda \cdot \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}[\|1\|\mathbf{y} - G(\mathbf{x}, \mathbf{z})], \tag{3.2}$$

where ℓ_1 norm is used to quantify the pixel-wise difference between the generated image and the ground truth. In practice, it has been shown that ℓ_1 norm encourages less blurring when compared to ℓ_2 norm [47]. Combining Equation (3.1) and Equation (3.2), we have the following objective function for CGAN,

$$\begin{aligned} \min_{G} \max_{D} \quad & \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{z})))] \\ & + \lambda \cdot \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}[\|1\|\mathbf{y} - G(\mathbf{x}, \mathbf{z})]. \end{aligned} \tag{3.3}$$

Table 3.1: The CGAN architecture.

Generator Encoder			**Generator Decoder**			**Discriminator**		
Layer	Filter	Output Size	Layer	Filter	Output Size	Layer	Filter	Output Size
Input	-	256×256×3	Deconv-BN-LReLU	5×5,2	2×2×512	Input	-	256×256×6
Conv-ReLU	5×5,2	128×128×64	Dropout	-	2×2×512	Conv-LReLU	5×5,2	128×128×64
Conv-BN-ReLU	5×5,2	64×64×128	Deconv-BN-LReLU	5×5,2	4×4×512	Conv-BN-LReLU	5×5,2	64×64×128
Conv-BN-ReLU	5×5,2	32×32×256	Dropout	-	4×4×512	Conv-BN-LReLU	5×5,2	32×32×256
Conv-BN-ReLU	5×5,2	16×16×512	Deconv-BN-LReLU	5×5,2	8×8×512	Conv-BN-LReLU	5×5,1	16×16×512
Conv-BN-ReLU	5×5,2	8×8×512	Deconv-BN-LReLU	5×5,2	16×16×512	FC	-	1
Conv-BN-ReLU	5×5,2	4×4×512	Deconv-BN-LReLU	5×5,2	32×32×256			
Conv-BN-ReLU	5×5,2	2×2×512	Deconv-BN-LReLU	5×5,2	64×64×128			
Conv-BN-ReLU	5×5,2	1×1×512	Deconv-BN-LReLU	5×5,2	128×128×64			
			Deconv-LReLU	5×5,2	256×256×3			

Table 3.2: The CNN architecture.

Layer	Filter	Output Size
Input	-	256×256×3
Conv-ReLU-BN-P	7×7,1	128×128×32
Conv-ReLU-BN-P	3×3,1	64×64×64
Conv-ReLU-BN-P	3×3,1	32×32×64
Conv-ReLU-BN-P	3×3,1	16×16×64
Conv-ReLU-BN-P	3×3,1	8×8×64
FC	-	64
ReLU+Dropout	-	64
FC	-	2

The details of the CGAN architecture are summarized in Table 3.1. The problem that we consider maps a high-resolution input (256×256) to a high-resolution output (256×256), and a common approach to design such a generator is the use of an encoder-decoder network [37,47,89,98]. The encoder passes the input through a series of layers that progressively downsample the input until a bottleneck layer; then the decoder reverses the process by progressively upsampling. In Table 3.1, column "Filter" gives the size and stride of the filter. All convolutional (Conv) and deconvolutional (Deconv) layers have 5×5 filters with a stride of 2. Batch normalization (BN) [46] is selectively applied on certain convolutional layers. The encoder uses leaky ReLU (LReLU) as the activation function, whereas the decoder uses ReLU. The discriminator is a convolutional neural network that performs classification to distinguish between the real image pairs and fake image pairs.

The standard approach to train GANs alternates between one step of optimizing D and one step of optimizing G [37]. In this way, we train both the generator and the discriminator to improve simultaneously, thus avoiding the case where one network is significantly more mature than the other. Here we use mini-batch stochastic gradient descent (SGD) for gradient update and apply the Adam solver [59] during the training stage.

3.2.2.3 LithoGAN

CGAN has demonstrated proven success in image generation tasks [47, 89] where generated images follow the distribution of the training data conditioned on the input images. However, for traditional computer vision tasks, locations of the objects in the generated image are not a major concern. For example, when trained to generate car images, the output of the GAN is judged upon based on the quality of an image as seen by a human while neglecting the exact location of the car in the image. However, for the lithography modeling task, the center of the generated resist pattern is as important as the shape of the pattern. Here the center refers to the center of the bounding box enclosing the resist pattern. In fact, we are interested in predicting a resist pattern which is accurate in both the shape and center.

With these two objectives in mind, and based on our experiments shown in Section 3.2.3, it is evident that CGAN falls short of predicting the correct center location of the resist pattern while demonstrating excellent results predicting the shape of the pattern. Hence, we propose a dual learning framework, referred to as *LithoGAN*, which splits the modeling task into two objectives:

- Resist shape modeling: a CGAN model is used to predict the shape of the resist pattern while neglecting the center;
- Resist center prediction: a CNN model is used to predict the center location of the resist pattern.

The application of the proposed LithoGAN framework is illustrated in Figure 3.6 where two data paths are shown. In the first path, a trained CGAN model is utilized to predict the shape of the resist pattern. During training, the golden pattern is re-centered at the center of the image, and the

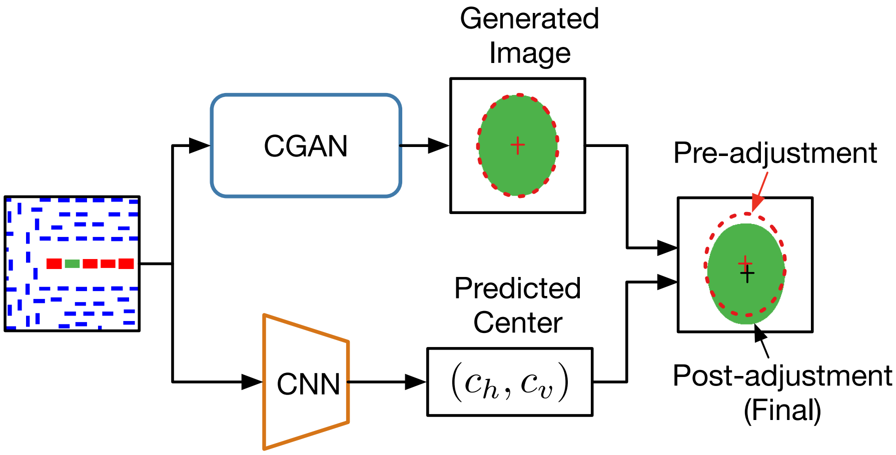


Figure 3.6: The proposed LithoGAN Framework.

coordinates of the original center are saved for CNN training. In other words, the model is trained to predict resist patterns that are always centered at the center of the images. On the other hand, the second path is composed of a CNN trained to predict the center of the resist pattern based on the mask image. The CNN architecture for the resist center prediction task is shown in Table 3.2, where max-pooling (P) with filter size 2 × 2 and stride 2 is applied after each convolutional layer.

In such a way, the shape and the center of the resist pattern are predicted separately. They are combined in the last step before output. As shown in Figure 3.6, the image generated by CGAN is adjusted by recentering the resist shape based on center the coordinates predicted from the CNN. The resulting adjusted image is the final output of the LithoGAN framework.

3.2.3 Experimental Results

The proposed framework for lithography modeling is implemented in Python with the TensorFlow library [8] and validated on a Linux server with 3.3GHz Intel i9 CPU and Nvidia TITAN Xp GPU. The experiments are performed on two benchmarks obtained from [72], where 982 and 979 mask clips are generated at 10nm technology node (N10) and 7nm node (N7) respectively. [72] performed SRAF insertion and OPC using Mentor Calibre [87], and then ran rigorous simulation to generate resist patterns using Synopsys Sentaurus Lithography [111] calibrated from manufactured data. In this work, the resist patterns generated by rigorous simulation are considered as the golden results. To guarantee highly accurate resist patterns, the pattern corresponding to the center contact in a clip is the only one adopted after each simulation. In other words, obtaining the golden resist pattern for each contact in a mask layout requires one rigorous simulation [58], and similarly, predicting this pattern using LithoGAN requires one model evaluation.

Each data sample for model training is a pair of the mask pattern image and the resist pattern image created using the color encoding scheme presented in Section 3.2.2.1. We randomly sample 75% of the data for training different models for N10 and N7 respectively, and the remaining 25% clips are for testing. In our experiments, we set the batch size to 4 and the number of maximum training epochs to 80. The weight parameter λ in Equation (3.3) is set to 100. The learning rate and the momentum parameters in the Adam optimizer are set to 0.0002 and (0.5, 0.999). The training time for each of

CGAN and LithoGAN is around 2 hours. Note that we train the CGAN and LithoGAN models five times each with different random seeds to eliminate random performance variation. The results reported in this section are the average of the five runs.

3.2.3.1 CGAN vs. LithoGAN

To demonstrate the performance of both frameworks discussed in this work: (i) the proposed lithography modeling CGAN and (ii) the improved LithoGAN, we visualize their performance in Figure 3.7. The top two rows are for samples from the N10 dataset, and the bottom two rows are for samples from the N7 dataset. According to [72], there are three types of contact arrays in the dataset, and Figure 3.7 includes at least one sample from each type. One can clearly see that CGAN outputs a shape very close to the golden resist pattern but the resist center can be quite far from the golden center; whereas, LithoGAN predicts both the shape and the center accurately. By examining the histogram showing the distribution of EDE in Figure 3.8, one can notice that LithoGAN can achieve lower EDE values when compared to CGAN; hence, making it closer to the golden solution.

LithoGAN achieves better accuracy compared to CGAN with the assistance of the CNN which predicts the location of the resist shape center. The average Euclidean distance between the golden location of the center and the predicted location on the test set is used to measure the CNN prediction error. The error values for N10 and N7 datasets are 0.43 nm and 0.37 nm respectively.

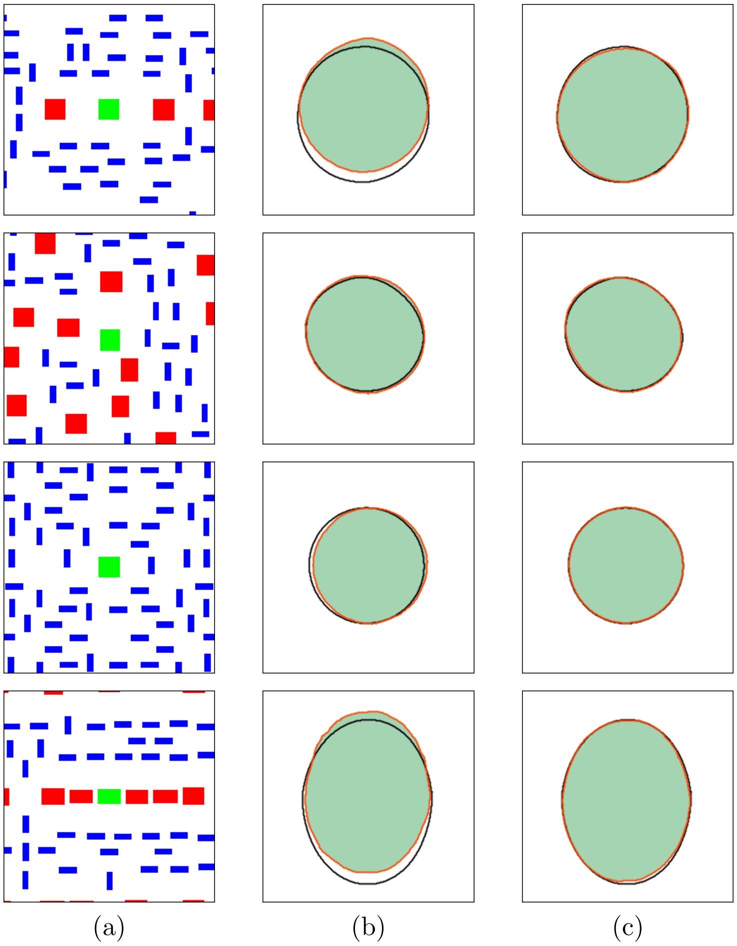


Figure 3.7: (a) Mask pattern input (b) CGAN output and (c) LithoGAN output. Each row represents one clip example. The golden contour is outlined in black. The prediction pattern is filled with green and outlined in red.

Figure 3.9 gives a visualization example of how resist pattern images generated by LithoGAN progressively become more real and closer to the golden results along the training process. Besides, the loss changes of the generator and discriminator are depicted in Figure 3.10. It shows that the model converges after 50 epochs and produces resist patterns of high quality.

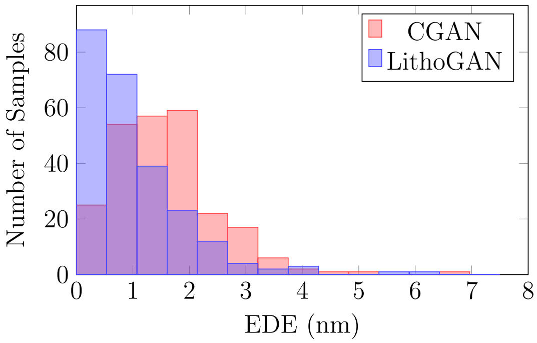


Figure 3.8: EDE distributions for CGAN and LithoGAN.

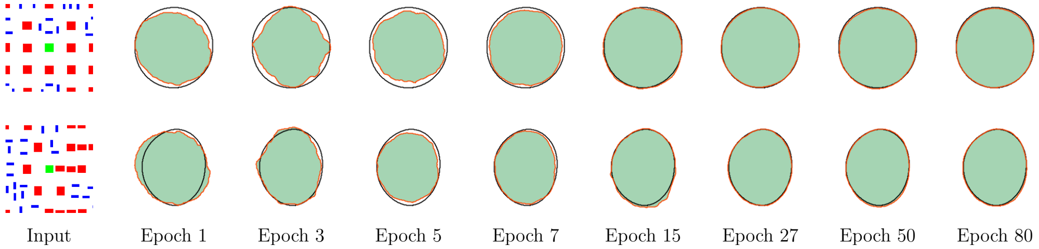


Figure 3.9: Visualization of the model advancement process. The prediction results for two testing samples using the LithoGAN model trained at different numbers of epochs are shown. Each row represents one clip example.

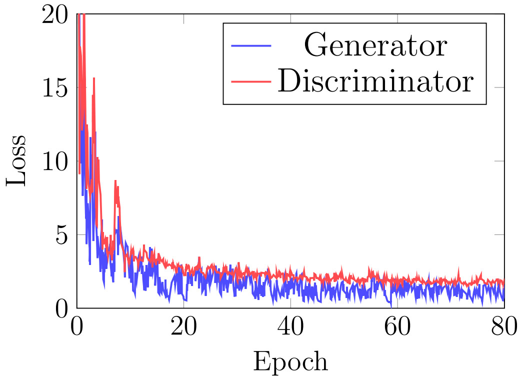


Figure 3.10: Loss curves of the generator and discriminator in LithoGAN.

3.2.3.2 Framework Validation

We first compare the accuracy of our proposed LithoGAN with the state-of-the-art work on lithography modeling [72]. The work [72] first runs the optical simulation with Mentor Calibre [87] on the mask pattern clips. Then it uses the trained CNN model to predict four thresholds for each clip and performs threshold processing to generate the final contours. Instead, the proposed CGAN and LithoGAN for direct lithography modeling only need the mask pattern clips as input and directly output the resist shapes.

Table 3.3 gives a detailed comparison among the three methods using the proposed metrics in Section 3.2.1, where the average results over all the test samples are reported. In this work, the goal is to mimic the results of the rigorous simulation; hence, these results are considered a reference and all metrics are computed with reference to them. In addition to the mean EDE error over all the test samples, we also report the standard deviation for their EDE values. By examining the results in Table 3.3, one can easily find that LithoGAN outperforms CGAN in all the metrics, and the detailed comparison has been shown in Section 3.2.3.1. Besides, although [72] achieves slightly better results, LithoGAN is still competent for lithography usage at advanced technology nodes. That is because the average error of the critical dimension obtained from LithoGAN, 1.99 nm and 1.65 nm for N10 and N7 respectively, fall within the acceptable range (10% of the half pitch for contacts) [72, 119].

Next, we demonstrate the runtime comparison in Table 3.4. It is reported in [72] that the rigorous simulation for both of the two datasets takes

Table 3.3: Comparison of evaluation metrics among different lithography modeling methods.

Dataset	Method	EDE (nm)		Pixel Acc.	Class Acc.	Mean IoU
		Mean	Std. dev.			
N10	Ref. [72]	0.67	0.55	0.98	0.99	0.98
	CGAN	1.52	0.95	0.96	0.97	0.94
	LithoGAN	1.08	0.88	0.97	0.98	0.96
N7	Ref. [72]	0.55	0.53	0.99	0.99	0.98
	CGAN	1.21	0.77	0.98	0.98	0.96
	LithoGAN	0.88	0.67	0.99	0.99	0.97

more than 15 hours. For a fair comparison, we rerun the proposed lithography modeling flow in [72] on our platform. The first step in [72], optical simulation, takes around 80 minutes. We use the same training dataset as that of CGAN and LithoGAN to train their proposed CNN model. Prediction of the four thresholds for each sample in the entire dataset using the CNN model takes 8 seconds. Contour processing is performed on 6 cores in parallel and takes 15 minutes. On the other hand, prediction for an entire N10 or N7 dataset using our CGAN or LithoGAN model takes less than 30 seconds. By comparing the runtime of generating resist patterns for all clips reported in Table 3.4, one can notice that CGAN/LithoGAN can achieve $\sim$1800× runtime reduction when compared to rigorous simulation and $\sim$190× when compared to the flow with machine learning based threshold prediction approach [72]. Hence, the proposed LithoGAN framework achieves significant runtime reduction while obtaining evaluation results that fall within the accepted lithography range.

Therefore, given its compelling speedup, LithoGAN paves the way for a new lithography modeling paradigm that can address the ever-increasing

Table 3.4: Runtime comparison among different methods.

Method	Rigorous Sim	Ref. [72]			Ours (CGAN/LithoGAN)
		Optical Sim	ML	Contour	
Time	> 15h	80m	8s	15m	30s
Ratio	> 1800	190			1

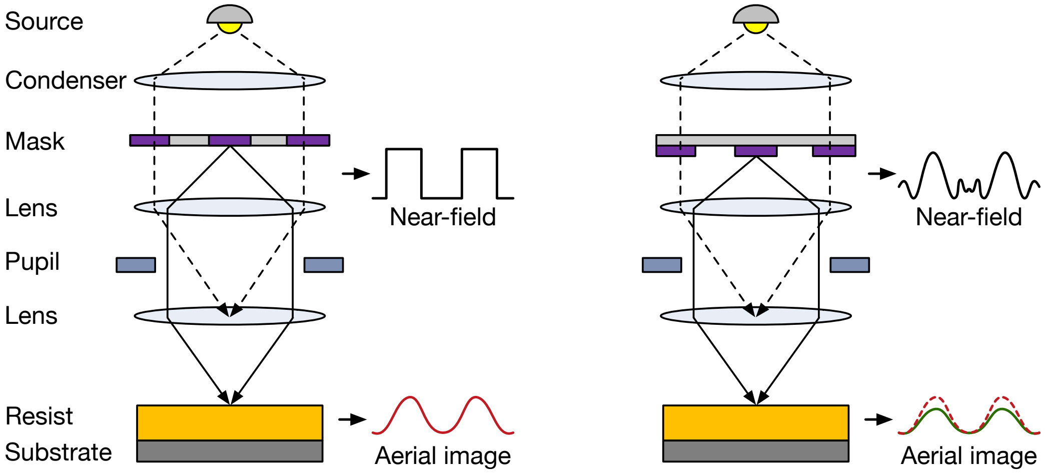
challenge of lithography simulation. This new paradigm can provide an accelerated framework which can perform within the adequate accuracy range for lithography.

3.2.4 Summary

In this work, we have presented the LithoGAN framework for end-to-end lithography modeling. LithoGAN is a dual learning network that predicts the resist shape using a CGAN model and predicts resist center using a CNN model. Experimental results show that the proposed framework predicts resist patterns of high quality while obtaining orders of magnitude speedup compared to conventional lithography simulation and previous machine learning based approach.

3.3 Lithography Modeling Considering Mask Topography Effects

This continuous device scaling has posed the mask topography effects among the major challenges in lithography modeling. In the past, thin mask approximation, or so-called Kirchhoff approximation, was widely used in lithography simulation, as shown in Figure 3.11(a). With such an approx-



(a)

(b)

Figure 3.11: Imaging process of a lithography system. (a) Thin mask model and (b) thick mask model result in different near-fields and aerial images.

imation, the three-dimensional structure of the mask is ignored despite its critical influence on the amplitudes, phases, and polarizations of the transmitted light, as demonstrated in Figure 3.11(b). When the feature sizes start to be comparable to the wavelength, the thin mask approximation is no longer adequate with the increasingly pronounced impacts of thick mask effects on the lithography imaging [38, 104, 123]. As a consequence, the failure to consider mask topography effects in lithography modeling could lead to critical dimension error and focus shift, resulting in the shrinkage of process window and the decrease of the image quality and the process robustness.

In the lithography process, many important properties, such as exposure and development latitude, can be derived from aerial images after optical simulation [81]. These images contain the intensity of the exposure radiation in the plane of the wafer; and hence, the topography effects of the mask can significantly impact their accuracy. Moreover, in lithography simulation, an accurate 3D view of aerial images at different resist heights is crucial to evaluate cross-section views of the resist pattern in order to find defects on the top or bottom position. These defects, if gone undetected, can lead to catastrophic manufacturing failures. Therefore, accurate prediction of 3D aerial images with the mask topography effects considered is important in lithography development and verification.

Conventionally, rigorous simulators capable of capturing mask topography effects have been developed for aerial image calculation. Technically, the precise description of the mask diffraction spectrum in lithography is accomplished by using rigorous algorithms to solve Maxwell's equations for the electromagnetic field [90]. However, despite their superior accuracy, such rigorous methods are prohibitively expensive since performing rigorous calculations at the full-chip level, during OPC for example, is computationally intensive. Under the governing trade-off between accuracy and efficiency, different compact models were formulated as less accurate yet more efficient mask models [9,114]. However, these compact models fail to maintain the accuracy level at advanced technology nodes since newly pronounced lithography effects invalidate several key assumptions in these models as shown in [75, 79].

Recently, advances in machine learning have been leveraged to devise new mask modeling techniques. In [10], an ANN model was proposed to model the rigorous spectrum with respect to the feature vector containing the amplitude and the phase information of the scalar spectrum from different mask patterns. The output of the ANN is used to compute the aerial images using Abbe's method. In [79], for an arbitrary thick mask, its near-field is calculated using the nonparametric kernel regression model and the pre-calculated training libraries; then the aerial image is calculated using Abbe's method as well. The aforementioned machine learning approaches rely on conventional modeling techniques that require intensive feature engineering and depend heavily on post-processing methods which affect the model accuracy.

In the recent past, CGANs have attracted attention due to their wide range of applications in image related tasks [89]. Among the state-of-the-art machine learning models, CGAN stands out due to its inherent capability to perform image translation tasks such as image colorization and background masking, where an image in one domain is mapped to a corresponding image in another domain. In practice, this model has been recently adopted to perform different lithography related tasks [11, 71]. Of particular significance is the application of CGAN in the end-to-end lithography simulation framework, LithoGAN [133]. While LithoGAN has demonstrated impressive efficiency, it only assumes a thin mask model which limits its capability of handling the mask topography effects. Besides, its output format is a monocolor image, while the desired output in the mask modeling task is the intensity map which has a higher accuracy requirement. Moreover, the aerial image estimation requires intensity map prediction at different resist heights. While the default approach is to train different CGAN models for prediction at different heights, such an approach is not efficient both in terms of training time and model size.

In this section, we propose TEMPO as a novel thick mask effect modeling framework using a single, one-fits-all model capable of predicting aerial image intensity at different resist heights. Besides the advantages in terms of the training cost and model size, incorporating the different modeling tasks into a single model can significantly improve the model accuracy. This is mainly due to the fact that various features and information are shared across all heights. Hence, having data from different heights available for training a single model results in a more robust model that has better generalization capabilities when compared to a set of models individually trained on a subset of the available data. To enable such a one-fits-all model, we propose a CGAN architecture that uses the desired prediction height as an additional input appended to the low-level latent representation in the model architecture. With such representation, the height information is efficiently incorporated at the CGAN bottleneck layer where it can have the most powerful impact on output generation.

3.3.1 Preliminaries

3.3.1.1 Mask Topography Effects

As shown in Figure 3.11, in an optical lithography system, the light source illuminates the mask and generates the near-field underneath the mask. Then, the light rays propagate through the projection lens and produce the aerial image on the wafer [79].

In the past, a mask in lithography was mostly considered as an infinitely thin object with homogeneously transparent and opaque areas as demonstrated in Figure 3.11(a). The conventional application of Kirchhoff's boundary conditions on the mask surface provides the so-called thin mask approximation of the near-field.

However, mask topography effects have been observed since the minimum feature size on the mask dropped below the exposure wavelength [104]. The light scattered by mask edges and corners changes the near-field of the light on the mask level. As shown in Figure 3.11(b), the scattering affects both the amplitude and the phase of the incident field, and thus not only changes the aerial image intensity on the wafer level, but also changes the resist profile after resist development. The failure to consider mask topography effects could lead to critical dimension error and focus shift, resulting in the shrinkage of the process window, and the decrease of the image quality and the process robustness. Therefore, mask topography models (thick mask models) have been indispensable since 28 nm tech node and below.

To precisely model the thick mask effects, rigorous simulators have been developed based on fundamental electromagnetism principles. However, they are rather slow and infeasible to apply on full chips within acceptable runtime. Generally, the intensity distribution in the aerial image calculated by a rigorous thick mask simulation is lower than the calculation result by a thin mask simulation because of a waveguide effect due to the topographical structure of the mask [106]. Nevertheless, there is no simple transformation between the outputs of these two kinds of mask models since the magnitude of the mask topography effects varies at different locations on the wafer and is also affected by the design of mask patterns.

There are efforts attempting to construct fast compact models for approximating mask topography effects [9, 114]. However, as shown in [75, 79], newly pronounced lithography effects and conditions keep invalidating some simple assumptions in conventional compact models and render them inapplicable at advanced nodes. The impacts of key factors on the accuracy and efficiency of the compact models need further study and verification, and ad hoc compact model building is incapable of providing models that are adequate for advanced lithography.

3.3.1.2 3D Aerial Image

In order to simplify the analysis of a lithography process, the optical effects of the lithography tool are usually separated from the resist effects of the resist process. As one of the direct outputs of optical analysis, the aerial image is defined as the spatial intensity distribution at the wafer, and is simply the square of the magnitude of the electric field [108]. The aerial image is the source of information that is transferred into the resist, and therefore dictates the quality of the final resist profile. Moreover, from the aerial image, we can easily predict the performance of a given lithographic process in terms of depth of focus, exposure latitude, etc [81].

The spatial image intensity distribution inside the resist bulk is calculated up to the defined resist thickness, and henceforth will be referred to as 3D aerial image or 3D intensity map, as shown in Figure 3.12. 3D aerial image is valuable in evaluating cross-section views of the resist profile in order to find defects on the top or bottom position. Typically, an aerial image simulation extracts the 2D intensity at one specific resist height; thus, the calculation of the entire 3D image is distributed among different threads in rigorous simulation tools [111].

Note that for a pure aerial image setup where the substrate, stack and resist are all set as air, the extraction height of the 2D aerial image does not matter. However, the resist and the stacks are practically composed of one or several non-air like optical materials, which results in standing waves due to interference effects of the incoming and backscattered light in the resist [111]. For the systems where standing waves can be very pronounced, the evaluation of the image intensity at a certain extraction height h must be performed carefully. For example, consider the extraction height $h = 10\,\text{nm}$ and $h = 70\,\text{nm}$ in Figure 3.12. It is obvious that the extraction height $h = 70\,\text{nm}$ will yield a higher image contrast than $h = 10\,\text{nm}$. Therefore, it is necessary to model 3D aerial images.

3.3.1.3 Problem Formulation

For image generation tasks, multiple evaluation metrics are typically used to judge upon model accuracy. Let I denote the golden aerial image and $\hat{I}$ denote the predicted aerial image, where $I, \hat{I} \in \mathbf{R^{n\times n}}$. One of the commonly

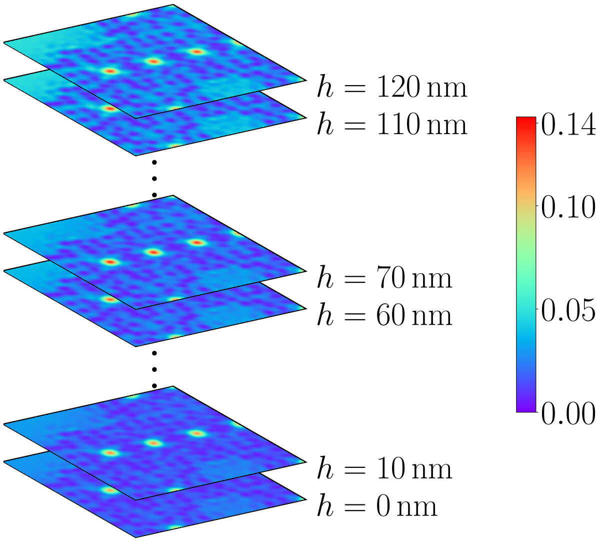


Figure 3.12: Example of the 2D aerial image slices inside a 3D aerial image.

used accuracy metrics is the root-mean-square error (RMSE), which is given by

$$\mathrm{RMSE} = \frac{1}{n}\|F\|\hat{I} - I, \tag{3.4}$$

where $\|F\|A = (\sum_{i,j} A_{i,j}^2)^{1/2}$ represents the Frobenius norm.

Since the overall light intensity of different aerial image samples in the dataset could vary significantly, we also adopt the normalized root-mean-square error (NRMSE) to quantify model performance. The NRMSE between the predicted image and the golden image is defined as the RMSE normalized by the averaged Frobenius norm of the golden image:

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\|F\|I/n} = \frac{\|F\|\hat{I} - I}{\|F\|I}. \tag{3.5}$$

We define the problem studied in this work as follows.

Problem 3.3.1 (3D Aerial Image Learning)**.** Given a training dataset containing mask pattern samples and the corresponding 2D aerial images at m resist heights for each mask pattern sample, the objective is to train a model that can accurately predict the aerial images of a test mask pattern, where the accuracy is measured in terms of the RMSE and the NRMSE.

3.3.2 TEMPO Framework

In a rigorous thick mask simulation flow, the simulator takes as input a mask pattern and generates the corresponding aerial image as shown in Figure 3.13(a). While such an approach is the common practice today, its inordinate runtime hinders its application in the early stages of the process development and mask optimizations. For example, simulating 1000 clips with mask topography effects could take up to 4 days. With this in mind, we propose TEMPO as a fast modeling framework that can significantly speed up the thick mask modeling task and hence, allow the consideration of the mask topography effects in the early stages of the process development. In practice, TEMPO provides in one of its schemes a CGAN model capable of mimicking the rigorous simulation process as shown in Figure 3.13(b). Under the same input/output set as in the rigorous simulation scheme shown in Figure 3.13(a), the CGAN model in TEMPO can translate the image from mask pattern to aerial images with orders of magnitude speedup. Hereafter, this direct translation using our proposed CGAN architecture is referred to as

Scheme 1, and its details will be covered in Section 3.3.2.2.

It is evident that, compared to the rigorous simulation scheme, Scheme 1 in TEMPO is capable of achieving immense speedup at some compromise in accuracy. This accuracy compromise is due to the fact that the optical modeling inside the lithography system is a complicated process; aerial image is the outcome of the interactions among light source, mask pattern and the projection lens. Hence, given the limited information available in the input image containing only the mask patterns, the accuracy of Scheme 1 is not expected to be ideal but can still be acceptable for early exploration stages given its attractive efficiency.

For applications with high accuracy requirements, TEMPO provides an alternative framework, namely Scheme 2 shown in Figure 3.13(c), which represents a compromise between the accurate yet time-consuming rigorous simulation, and the efficient Scheme 1 with imperfect accuracy. Compared to Scheme 1, Scheme 2 sacrifices some additional runtime for better accuracy while still maintaining impressive speedup compared to the rigorous simulation. As a first step, TEMPO in Scheme 2 runs a fast thin mask model to generate aerial images assuming no mask topography effect, and the output aerial image is used along with the mask pattern as the input to the CGAN model. In this way, the aerial image given by the thin mask model provides the CGAN model with additional information not present in the mask pattern image, and hence improves its accuracy. In the next subsections, we first introduce the conventional CGAN model for image translation, then we present

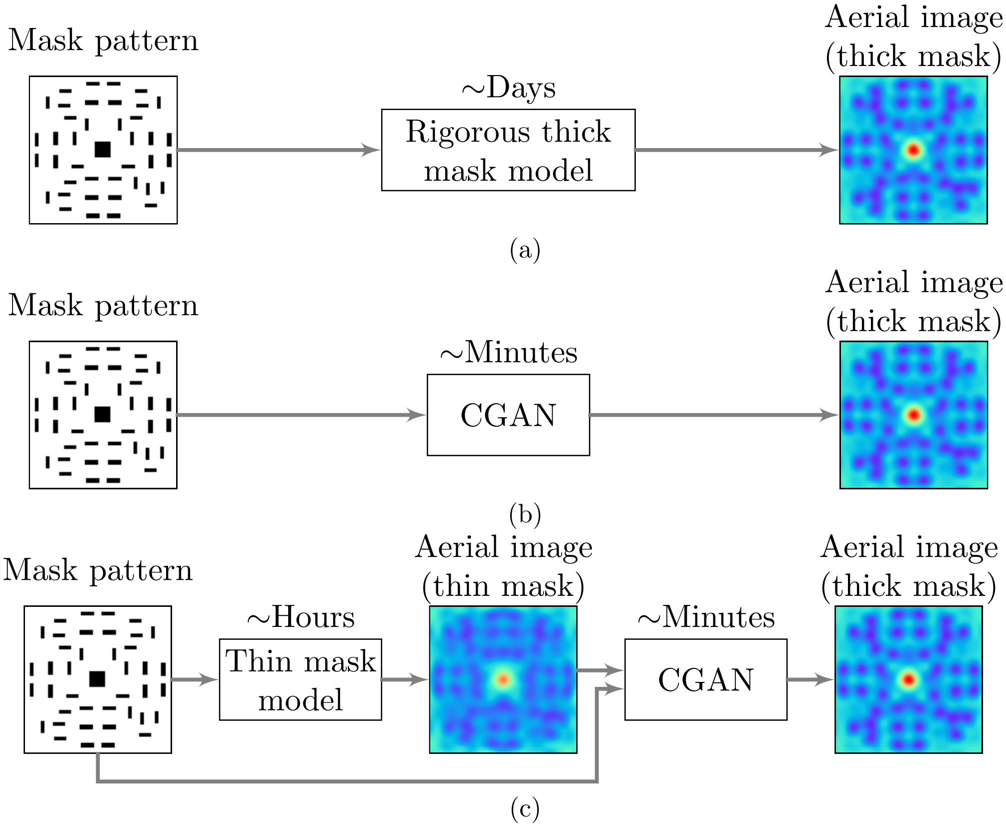


Figure 3.13: (a) Traditional rigorous thick mask simulation flow, (b) proposed Scheme 1 for high efficiency and (c) Scheme 2 for high accuracy in TEMPO.

TEMPO for aerial image generation.

3.3.2.1 Generative Adversarial Networks

Generative adversarial networks (GANs) have demonstrated remarkable success in various computer vision tasks such as image generation [89], image translation [47, 145], and super-resolution imaging [62]. Originally, GANs were developed for the purpose of learning the distribution of a given dataset with the intent of generating new samples from it [37]. A typical GAN model consists of two modules: a generator and a discriminator. The generator is trained to produce samples that cannot be distinguished from real images by the adversarially trained discriminator which is trained to do as well as possible at detecting the generator fakes [37].

The conventional generator in a GAN is basically an encoder-decoder network where the input is passed through a series of layers that progressively downsample it (i.e., encoding), until a bottleneck layer, at which point the process is reversed (i.e,, decoding) [37,89,98]. On the other hand, the discriminator is a convolutional neural network whose objective is to classify fake and real images. Hence, its structure differs from that of the generator and resembles a typical two-class classification network [37,89,98]. This adversarial scheme is represented in the objective function given as:

$$\min_{G} \max_{D} \mathbb{E}_x[\log D(x)] + \mathbb{E}_z[\log (1 - D(G(z)))], \tag{3.6}$$

where $D(\cdot)$ represents the probability of a sample being real; i.e., not generated by G, $\mathbb{E}_x$ denotes the expectation over the input data x, and z is a random noise vector used as a seed for image generation.

A GAN model is typically trained with mini-batch stochastic gradient descent (SGD) [37]. The training alternates between one gradient descent step on the discriminator, and then one step on the generator. After training, the generator part of the GAN is used to generate new samples using random noise vectors while the discriminator is discarded as it is only needed for the training process [37].

Stemming from the core GAN model, different variants of generative neural networks were developed to address challenges in various fields of study, especially computer vision. Technically, many tasks in computer vision and graphics can be thought of as translation problems where an input image is to be translated from domain A to another domain B. Isola et al. [47] introduced an image-to-image translation framework that uses GANs in a conditional setting where the generator transforms images conditioned on the input image. Instead of randomly generating images from the learned distribution, it transfers an input image into another domain, hence, acting as an image translator. To train such a model, a paired training dataset is needed where each sample is a pair of an input image (i.e., image in the input domain) and its corresponding output image (i.e., translated image in the target domain).

Mathematically, the loss function used for training the CGAN can be given as [47, 89]:

$$\begin{aligned} L_{\text{CGAN}} = {} & \mathbb{E}_{x,y}[\log D(x,y)] \\ & + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))] \\ & + \lambda \cdot \mathbb{E}_{x,y,z}[\|1\|y - G(x,z)], \end{aligned} \tag{3.7}$$

where x is a sample in the input domain, y is its corresponding sample in the output domain, and λ is the weight parameter. Comparing equations (3.6) and (3.7), one can notice the addition of the loss term which penalizes the difference between the generated sample $G(x, z)$ and its corresponding golden reference y.

3.3.2.2 TEMPO Architecture Design

Image translation using CGAN was proposed as a means for domain transfer between two distinct domains. However, different applications require more comprehensive translation schemes with one-to-many domain transfers. Aerial image generation requires domain transfer from the single mask pattern domain to multiple resist height domains. Another popular application of such a scheme is facial image translation, where an input facial image is translated into different target domains representing different facial expressions or appearances [26]. For aerial image generation and other similar tasks, the most straightforward option is to train multiple domain-to-domain models. So, for m target domains, m such models are needed.

Clearly, the approach of building m individual models has multiple drawbacks. Most evident is the size of the model that scales with the number of target domains. This also requires a large dataset from all domains to train different independent models. Besides, when assuming that different target domains are independent, an opportunity for information sharing between those slightly different tasks is missed. In terms of the data, since we model the light intensity in a 3D continuous space in the aerial prediction task, the intensity values change continuously. The aerial images extracted from discrete resist heights should be highly correlated. In terms of the model, the input encoding performed by the generator's encoder is very similar across different domains in many applications. This is true in the aerial image generation as well as the facial translation scenario. Mainly, the important features for the translation tasks are common across different target domains, and the target specification is rather important in the decoder that generates the images. Hence, if an adequate information-sharing scheme is developed, the performance can be enhanced by exploiting the high correlation between images in different domains. Therefore, model scalability and information sharing render the setup of multiple individual models ineffective.

To overcome these two drawbacks, new variants of CGAN have been proposed, such as ComboGAN [14] and StarGAN [26]. In ComboGAN, information sharing is addressed through a joint training scheme for the m different 2-domain transfer models [14]. On the other hand, StarGAN tries to address the scalability issue by building a single generator and incorporating the target domain into its input. However, the target domain representation in StarGAN still carries high redundancy since it requires m additional channels in the input image to one-hot encode the chosen k-th target domain out of m domains. In other words, the size of the input image scales linearly with the number of target domains. Better scalability necessities a more compact input domain encoding scheme.

Towards the goal of a compact model with an information-sharing scheme, two important features of the one-to-many domain transfer task in this work should be noted. First, the target information is not necessary for the input encoding task. It is fair to assume that the features that are needed from the input image to generate the aerial image at different heights are the same. It is the way these features are later decoded that is impactful on the image generation. Hence, the target information is not needed as an input to the encoder in the generator network. The second feature is that the bottleneck layer in the generator carries the most critical information as it represents the latent representation of the input upon which the output image is generated; thus, the information in this layer is of significant impact on the result. Therefore, we propose within TEMPO a new *one-fits-all* model where a one-hot encoding vector of length m carrying the target domain information is appended to the latent space representation in the bottleneck layer, as shown in Figure 3.14. This way, the information is appended at a critical location in the network where it can guide the output image generation while having a compact representation. Compared to that used in StarGAN where each one extra input channel is needed for each domain, the encoding scheme in TEMPO requires only a single channel for all the domains. This can significantly improve the scalability of TEMPO when faced with a significant increase in the number of target domains.

In the next subsections, the details of both the generator and discriminator used in TEMPO are shown. These implementations are adapted from the deep convolutional generative adversarial networks framework proposed in [98].

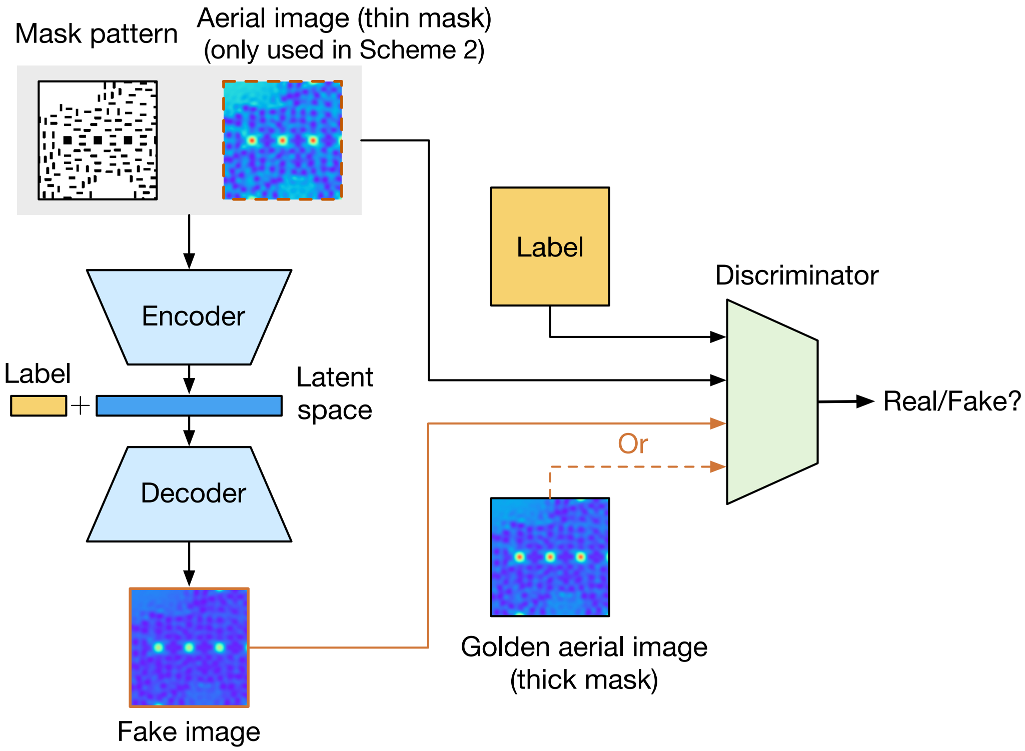


Figure 3.14: Overview of the TEMPO model.

3.3.2.3 Generator

We adopt the encoder-decoder network which is commonly used to design a generator [37, 47, 89, 98]. The input is passed through a series of layers in the encoder that progressively downsamples it, until a bottleneck layer, at which point the process is reversed in the decoder. The details of the encoder and decoder are summarized in Table 3.5. Specifically, eight convolutional and deconvolutional layers are used for the encoder and decoder, respectively. In Table 3.5, the column "Size" and the column "Stride" give the size and stride of each filter, and the number of layers sharing the same filter setting is shown in the column “Count”. “Additional” indicates the additional layers for normalization and activation function. Here, batch normalization (BN) [46] is selectively applied on certain convolutional layers both in the encoder and decoder. The encoder uses leaky ReLU (LReLU) as the activation function, whereas the decoder uses ReLU. The input of the generator is the images of 200×200 pixels, and can have single channel (mask pattern) in Scheme 1 or two channels (mask pattern and thin-mask aerial image) in Scheme 2. “Concat” denotes the concatenation of the one-hot label vector of size m and the latent space vector of size 512. For image translation tasks using CGAN, a significant amount of information is shared between the input and the output, and we followed the design of U-Net [101] with skip connections between encoder layers and decoder layers.

3.3.2.4 Discriminator

On the other hand, the discriminator is a convolutional neural network that performs classification to distinguish between the real image pairs and fake image pairs. Meanwhile, the target domain information is fed into the discriminator that is trained to discriminate image pairs from different target domains. Here, the target information is encoded by appending to the input image a single channel whose pixel values reflect the target domain. In practice, since the different domains in this application correspond to different resist heights, there exists a true ordering for the target domains themselves. Therefore, an ordinal encoding scheme is used to encode the ID of the target domain k ($k \in \{0, 1, \ldots, m-1\}$) on the additional input channel whose pixel values are the same and are set as follows:

$$\frac{p_{\max} - p_{\min}}{m-1} \cdot k + p_{\min}, \tag{3.8}$$

where $p_{\min}$ and $p_{\max}$ denote the minimum and maximum possible values in the additional input channel. Commonly used settings include $p_{\min} = 0, p_{\max} = 255$ or $p_{\min} = -1, p_{\max} = 1$.

Table 3.5 summarizes the details of the discriminator which constitutes of four convolutional layers and one fully connected layer (FC) whose output is the binary classification results.

3.3.3 Experimental Results

In this work, we explore mask topography effects on contacts as according to the existing studies and reports, the mask topography effect should be considered more carefully for contact hole patterns than line and space patterns [88]. We generate 966 clips of size $2\,\mu\text{m} \times 2\,\mu\text{m}$ containing various contact patterns following the clip generation method described in [72]. Each contact is designed to be $60\,\text{nm} \times 60\,\text{nm}$, and the contact pitch is 128 nm. We perform sub-resolution assist feature (SRAF) insertion and OPC on contact patterns using Mentor Graphics Calibre [87].

We run rigorous optical simulation to generate 3D aerial images using Synopsys Sentaurus Lithography [111]. A quasar light source is used for this experiment. The wavelength of the light source is set to 193 nm, and the

Table 3.5: Network architecture of the proposed TEMPO.

Network	Layer	Count	Channel	Size	Stride	Additional
Generator Encoder	Input	—	1 (2) [a]	—	—	—
	Conv	1	64	5	2	LReLU,BN
	Conv	1	128	5	2	LReLU,BN
	Conv	1	256	5	2	LReLU,BN
	Conv	5	512	5	2	LReLU,BN
Generator Decoder	Concat	1	$512 + m$	—	—	—
	Deconv	4	512	5	2	ReLU,BN
	Deconv	1	256	5	2	ReLU,BN
	Deconv	1	128	5	2	ReLU,BN
	Deconv	1	64	5	2	ReLU,BN
	Deconv	—	1	5	2	ReLU
Discriminator	Input	—	3 (4)	—	—	—
	Conv	1	64	5	2	LReLU
	Conv	1	128	5	2	LReLU
	Conv	1	256	5	2	LReLU
	Conv	1	512	5	2	LReLU
	FC	1	1	—	—	Sigmoid

[a] $(\cdot)$ denotes the number of channels in Scheme 2.

numerical aperture (NA) of the imaging system is 1.2. The simulation window of 1.5 µm × 1.5 µm is configured as nonperiodic and centers each of the clips. Since the resist thickness is 120 nm and simulation resolutions in X, Y and Z directions are set to 7.5 nm, 7.5 nm and 10 nm respectively, we got 2D aerial images of 200 × 200 pixels at 13 different resist heights for each clip, i.e., $n = 100$ in Equation (3.4) and Equation (3.5), and $m = 13$.

In this work, the aerial images generated by rigorous simulation considering mask topography effects are used as the golden data for TEMPO training. Each sample in the training set is a collection of the mask pattern image and the corresponding aerial images at 13 different resist heights. Note that the mask pattern clip within the simulation window is 1.5 µm × 1.5 µm and the grid unit in the original layout is 1 nm, so we size it down to a grayscale image of 200 × 200 pixels using average filtering. Each pixel in the aerial image is an intensity value stored in the 32-bit single-precision format.

The proposed TEMPO is implemented in Python with the Tensorflow library and validated on a Linux server with 3.3GHz Intel i9 CPU and Nvidia TITAN Xp GPU. In our experiments, we randomly sample 75% of the data for training the model and the remaining 25% clips are for testing. We set the batch size to 4 and the number of maximum training epochs to 70. The weight parameter λ in Equation (3.7) is set to 1000. We also build 13 individual models to predict 2D aerial images at each resist height separately which work as the baseline approach. Each of the individual models takes as input the dataset of aerial images at only one resist height and is trained with the same hyperparameter setting as TEMPO. Note that the 13 individual models have a total of 1.17×10^9 trainable parameters (weights and biases), whereas TEMPO has 1.03×10^8. Therefore, TEMPO effectively reduces the model size for the 3D aerial image prediction task.

We first demonstrate the accuracy of our proposed TEMPO. Table 3.6 gives a detailed comparison between the individual models and our TEMPO under Scheme 1 and Scheme 2 using the proposed RMSE and NRMSE metrics in Section 3.3.1.3. The number shown in the table is the average of all the test samples on each resist height. One can easily see that TEMPO outperforms

Table 3.6: Comparison of evaluation metrics among different modeling methods.

Height (nm)	RMSE ($\times 10^{-4}$)				NRMSE (%)			
	Scheme 1		Scheme 2		Scheme 1		Scheme 2	
	Baseline	TEMPO	Baseline	TEMPO	Baseline	TEMPO	Baseline	TEMPO
0	11.88	10.87	4.96	4.12	4.55	4.15	1.96	1.62
10	12.53	11.48	5.41	4.21	4.55	4.15	2.03	1.57
20	13.50	12.63	5.24	4.50	4.51	4.19	1.79	1.54
30	15.30	13.26	6.11	4.74	4.97	4.25	2.02	1.56
40	14.26	13.32	5.53	4.79	4.63	4.31	1.82	1.58
50	14.36	13.11	5.96	4.93	4.71	4.29	1.98	1.63
60	14.37	13.22	7.99	5.23	4.63	4.24	2.63	1.70
70	15.18	13.61	7.32	5.71	4.62	4.13	2.27	1.76
80	15.58	14.52	7.71	6.24	4.48	4.17	2.26	1.81
90	16.42	15.25	8.00	6.79	4.57	4.23	2.26	1.90
100	16.79	15.59	8.40	7.42	4.62	4.28	2.34	2.05
110	17.16	15.75	8.96	8.17	4.68	4.29	2.46	2.23
120	17.11	15.74	13.27	8.66	4.63	4.26	3.67	2.34
Average	14.96	13.72	7.30	5.81	4.63	4.23	2.27	1.79
Max	17.16	15.75	13.27	8.66	4.97	4.31	3.67	2.34
Std. dev.	1.69	1.58	2.24	1.52	0.12	0.06	0.49	0.27

the individual modeling approach (denoted as Baseline) under both schemes. Besides, whether using the 13 individual GAN models or the proposed TEMPO approach, Scheme 2 always gives better accuracy than Scheme 1. Moreover, TEMPO improves the RMSE from 14.96×10^{-4} to 13.72×10^{-4} on average, and NRMSE from 4.63% to 4.23% in Scheme 1, while improving the RMSE from 7.3×10^{-4} to 5.81×10^{-4} and NRMSE from 2.27% to 1.79% in Scheme 2. Clearly, Scheme 2 in TEMPO can help gain better improvement in accuracy because the aerial image produced by the fast thin mask simulation, as an additional input in Scheme 2, provides more information about the lithography system, and hence TEMPO is able to achieve notable improvement under such situation. To visually examine the accuracy difference between the two schemes in TEMPO, the aerial images for two samples of distinct pattern designs are shown in Table 3.7.

As one of the most important outputs of optical models, the aerial image can be used together with resist models to simulate final resist profiles. Therefore, in addition to the direct comparison of aerial images, we also evaluate the effectiveness of our proposed methods based on the quality of generated resist patterns. We calculated the CD value of the resist pattern for the center contact in each sample using the average of the aerial images at 13 resist heights. Using the CD values derived from the golden aerial images as reference, Table 3.8 shows the comparison of CD errors in the X and Y directions among different mask topography effect modeling methods. The row "thin mask sim." represents the CD errors when using the aerial images without considering mask topography effects, and the errors could go up to more than 20 nm. Our proposed TEMPO in Scheme 2 gives very small CD errors, for example, 0.38 nm in the X direction and 0.45 nm in the Y direction, which qualifies it for practical lithography usage. Besides, TEMPO gives smaller CD errors when compared with the baseline with 13 individual GAN models, which further demonstrates the advantages of our one-fits-all approach.

Last, we demonstrate the runtime comparison in Table 3.9, where the total runtime of generating the 3D aerial images for all the test samples, i.e., 242 samples, are shown. We can clearly see that the two schemes in TEMPO satisfy different needs for speed and accuracy at lithography development phases. Scheme 2 in TEMPO achieves $\sim 26.5\times$ runtime reduction when compared to rigorous thick mask simulation while achieving satisfactory

Table 3.7: Aerial image results for two test clips using Scheme 1 and Scheme 2 in TEMPO.

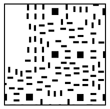
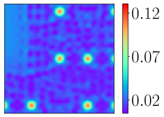
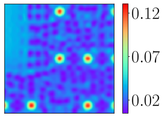
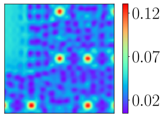
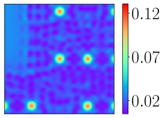
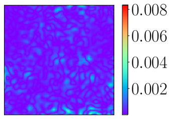
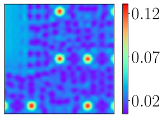
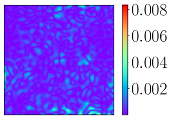
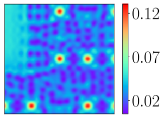
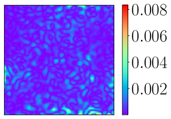
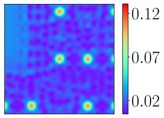
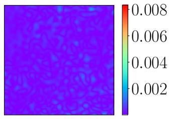
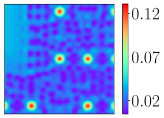
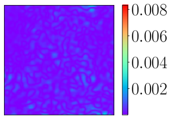
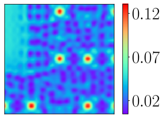
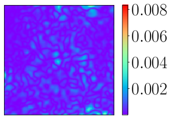
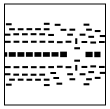
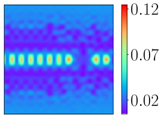
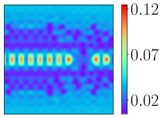
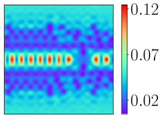
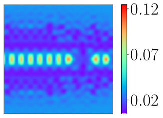
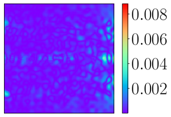
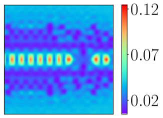
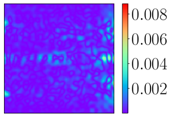
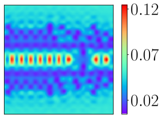
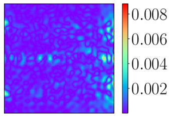
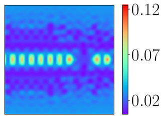
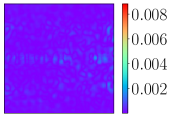
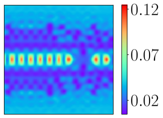
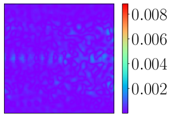
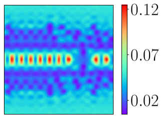
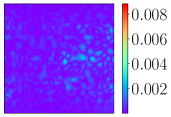
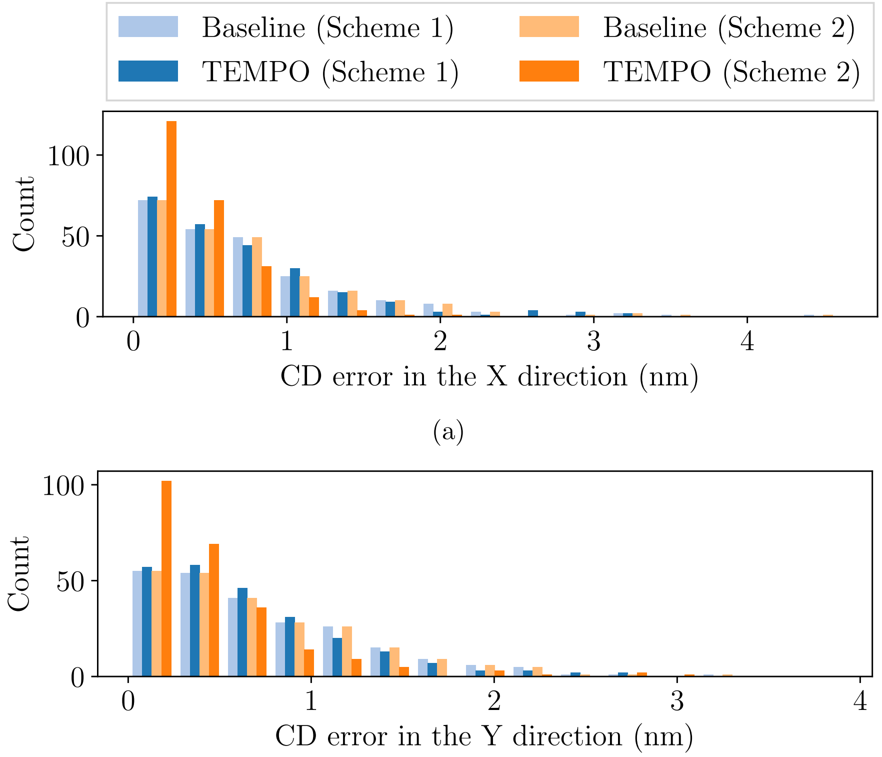
Input	Method	0 nm		60 nm		120 nm	
		Aerial image	Diff w.r.t. golden	Aerial image	Diff w.r.t. golden	Aerial image	Diff w.r.t. golden
	Golden						
	TEMPO (Scheme 1)						
	TEMPO (Scheme 2)						
	Golden						
	TEMPO (Scheme 1)						
	TEMPO (Scheme 2)						

Table 3.8: Comparison of CD errors in the X and Y directions among different methods.

Method		CD error X (nm)		CD error Y (nm)	
		Average	Max	Average	Max
Thin mask sim.		2.77	20.67	3.93	33.49
Scheme 1	Baseline	0.75	4.64	0.73	3.19
	TEMPO	0.72	3.38	0.67	2.82
Scheme 2	Baseline	0.48	2.05	0.50	3.89
	TEMPO	0.38	1.88	0.45	3.11



(a)

(b)

Figure 3.15: Distribution of CD errors using different methods: (a) error in the X direction and (b) error in the Y direction.

accuracy. Considering the acceptable CD degradation in Scheme 1 compared to Scheme 2 while being 50× faster, Scheme 1 in TEMPO is suitable for the early exploration stages where speed is favored over high accuracy.

Table 3.9: Runtime comparison between rigorous simulation and the proposed TEMPO framework.

	Rigorous mask sim.	TEMPO (Scheme 1)	TEMPO (Scheme 2)		
			Thin mask sim.	GAN	Total
Runtime	20.5 h	1.1 m	45.3 m	1.1 m	46.4 m
Ratio	26.51	0.02	—	—	1.00

3.3.4 Summary

In this work, we have presented TEMPO, a novel and scalable framework which is capable of generating 3D aerial images efficiently and accurately for modeling mask topography effects. Essentially, TEMPO comprises a one-fits-all CGAN model for multi-domain image-to-image translation, with the accuracy and compactness further boosted by across-domain information sharing. Besides, the two flexible schemes of operations in TEMPO provide different trade-offs between accuracy and efficiency, which promotes the wider application of TEMPO in different stages of process development. The experimental results demonstrate that TEMPO can achieve superior performance in both speed and accuracy for advanced lithography usage.

Chapter 4

Analysis and Optimization for Electromigration Reliability

4.1 Introduction

As IC technologies continue to scale, electromigration (EM) comes forth as one of the prominent reliability issues challenging the design of robust circuits [68]. Complex chip functionalities have been made possible by virtue of increasing transistor densities and aggressive scaling of interconnects. However, these two factors bring along higher current densities in metal wires, a phenomenon that further exacerbates EM. Particularly, high current densities lead to the migration of atoms in metal wires resulting in opens and shorts over time [17]. Hence, the continuous drive toward extreme scaling will

This chapter is based on the following conference papers.

1. Wei Ye, Yibo Lin, Xiaoqing Xu, Wuxi Li, Yiwei Fu, Yongsheng Sun, Canhui Zhan, and David Z. Pan. "Placement mitigation techniques for power grid electromigration." In 2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 1-6. IEEE, 2017.
2. Wei Ye, Mohamed Baker Alawieh, Yibo Lin, and David Z. Pan. "Tackling signal electromigration with learning-based detection and multistage mitigation." In Proceedings of the 24th Asia and South Pacific Design Automation Conference (ASPDAC), pp. 167-172. 2019.

I am the main contributor in charge of problem formulation, algorithm development, and experimental validations.

keep compounding the EM problem, making EM design closure a challenging task [1, 51].

Addressing the EM challenge requires a two-step process: (i) violation detection and (ii) EM mitigation. Conventionally, EM checking tools are invoked after the detailed routing stage [2, 65]. These tools compare the current densities in metal wires with technology-specific design rules to detect EM violations. Next, the violations are fixed with engineering change order (ECO) efforts [5]. EM checking tools leverage post-routing information to detect violations, which consequently limits the efficiency of their mitigation techniques. In the routing phase, the locations of standard cells and the corresponding current distribution are already fixed and the traditional fixing approaches such as wire widening and cell resizing are not effective enough to handle the ever-growing number of violations [1]. In fact, the methodology of "EM-analysis-then-fix" is becoming obsolete at advanced nodes [45], which makes it of vital importance to incorporate EM detection and fixing techniques into earlier stages of physical design (PD).

Two clear benefits are associated with such early stage EM handling. First, the number of EM violations can be decreased as a result of using a larger set of mitigation techniques. Second, introducing early stage mitigation techniques can help reduce the resulting overhead when compared to post-routing fixing techniques. Thus, moving the EM detection and resolving steps to earlier stages of the physical design can help in reducing runtime or the number of iterations needed for design closure.

In the rest of the chapter, Section 4.2 describes placement mitigation techniques for power grid EM, and Section 4.3 presents the signal EM detection and mitigation framework.

4.2 Placement Mitigation for Power Grid Electromigration

Due to unidirectional current flow and high current density, power grid is one of the most vulnerable interconnect structures to EM failure. EM checking tools calculate current densities in metal wires and detect EM violations in power grids with given design rules; then these violations are fixed with engineering change order (ECO) efforts [134]. However, traditional fixing approaches such as spacing large-current cells and widening metal wires are not effective enough to handle the ever-growing number of violations in power grids. Therefore, it is necessary to mitigate EM degradation in power grids at earlier design stages, such as placement. The locations of standard cells and the corresponding current distribution are determined during placement stage and the placement solution can directly affect the final quality of EM design closure.

Power grid consists of horizontal power rails connecting standard cells together, and these rails are connected with wider vertical power stripes [136]. As illustrated in Figure 4.1(a), *power tile* is the region between two adjacent VDD (or VSS) power stripes and the adjacent power rails [45], and the chip region is partitioned into multiple power tiles. It is observed that lower-level

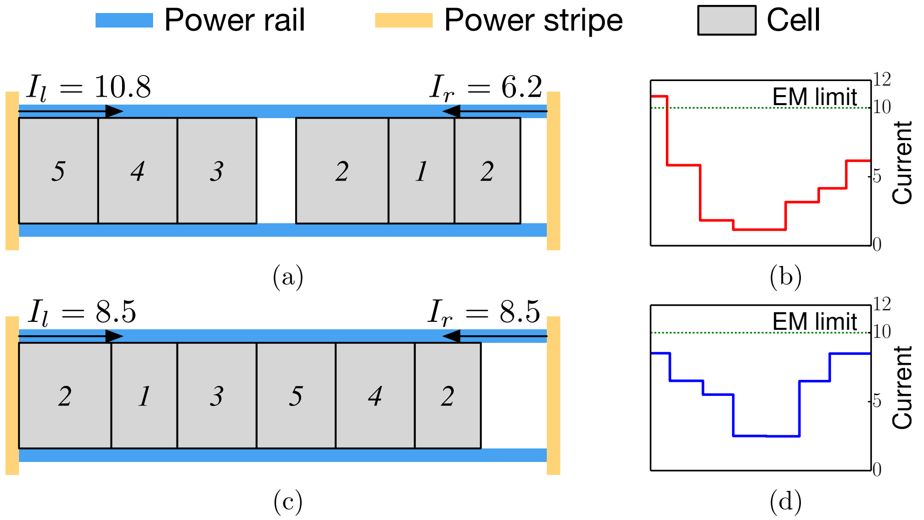


Figure 4.1: (a) An initial cell placement in a power tile and (b) the corresponding current distribution in the power rail; (c) An EM-friendly placement and (d) the corresponding current distribution.

metal layers of power grids are more susceptible to EM failures due to smaller wire width, and EM violations are most likely to occur around weak power grid connections, which deliver current to high power-consuming regions. [45] proposes a global placement problem with a bin-packing formulation to constrain power consumption of each power tile, so that high-power cells are forced to spread across the placement region and current densities are flattened over the chip.

However, placement with globally balanced current density does not guarantee EM friendliness. As illustrated in Figure 4.1, the metal wire segments touching vias on both sides carry the largest currents in a power rail. They feed all the cells in the tile and are the weakest points to EM. Current densities of these segments may still exceed the current limit even if the total current density of the power tile is below the threshold. Figure 4.1(a) shows a placement within a power tile, and the number on each cell denotes its normalized current. Suppose the DC current limit for power rail is 10. Figure 4.1(b) shows the simulation result of the current distribution on the power rail. The total current drawn by all the cells is 17, which is less than the power tile total current limit 20. But an EM violation occurs on the left side because the current exceeds the EM limit. The placement shown in Figure 4.1(c) guarantees that the maximum current in the power tile will not exceed the EM current limit.

Globally balancing current density over chip cannot completely resolve EM violations because the maximum current constraint for power tiles is more strict than the total current constraint. Therefore, as illustrated in Figure 4.1, besides determining which cells to be placed in the power tile, we need to figure out the order and spacing of these cells under the EM current limit.

4.2.1 Problem Formulation

Hsu et al. [45] propose an average power-based model to evaluate power grid static EM at placement stage. With given supply voltage, we use the DC current limit I_{limit} of power rail metal wires to evaluate power grid EM violations. For a standard cell, we consider the sum of the dynamic current and leakage current at this stage, which is calculated as:

$$I = \alpha \cdot C \cdot V_{\text{DD}} \cdot f + I_{\text{leak}},$$

where α is the cell activity factor, V_{DD} is the supply voltage and f is the system clock frequency. C is the sum of the load capacitance and the output pin capacitance. Load capacitance further includes downstream gate capacitance and interconnect capacitance. Since nets have not been routed at this stage, we use half-perimeter wirelength (HPWL), which is widely adopted in placement [13], to estimate interconnect capacitance.

Figure 4.2 demonstrates how we calculate the maximum current in the local power rails within a power tile. P_l and P_r are the left and right endpoints of the VDD power rail. d_i^l and d_i^r are the distances from the midpoint of the i-th cell to P_l and P_r, respectively. R_i^l and R_i^r are the wire resistances of the corresponding metal segments, which are proportional to d_i^l and d_i^r. The following equations hold:

$$I_i^l \cdot \ R_i^l = I_i^r \cdot \ R_i^r, \quad I_i^l + I_i^r = I_i.$$

Thus,

$$I_i^l = \frac{d_i^r}{d_i^l + d_i^r} I_i, \quad I_i^r = \frac{d_i^l}{d_i^l + d_i^r} I_i. \tag{4.1}$$

The currents drawn by all the cells in the power tile from P_l and P_r are computed as:

$$I^l = \sum_i I_i^l, \quad I^r = \sum_i I_i^r, \quad I^l + I^r = \sum_i I_i. \tag{4.2}$$

Since the cells in the tile only draw current via P_l and P_r, the peak current that occurs in the local power rail is $\max\{I^l, I^r\}$.

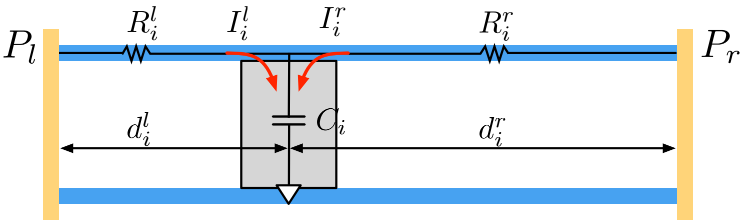


Figure 4.2: The model for current calculation in a power tile.

Definition 4.2.1 (EM Violation)**.** There is an EM violation in the power tile if $\max\{I^l, I^r\} > I_{\text{limit}}$.

According to Equation (4.1) and Equation (4.2), it is necessary to know the current and location of each cell in the power tile to compute I^l and I^r. Therefore, once a cell is relocated, we update its current because the load capacitance is also changed. When a cell is moved to a new power tile, we need to predict whether this movement will cause an EM violation in the new power tile. Since we may not decide the new location of the cell immediately, we derive a total current constraint of the power tile to estimate the EM violation. Combining Equation (4.2) with the property that an EM-friendly power tile satisfies $I^l \leq I_{\text{limit}}$ and $I^r \leq I_{\text{limit}}$, there is a relaxed constraint on the total current of power tile as follows:

$$\sum_i I_i = I^l + I^r \leq 2I_{\text{limit}}. \tag{4.3}$$

Satisfying the above constraint is a necessary condition for the power tile to be free from EM violation. When the total current of the cells in the power tile is less than $2I_{\text{limit}}$, cell placement in the power tile further determines

I^l and I^r.

In this work, we follow ICCAD 2013 placement contest [57] to use scaled half-perimeter wirelength (sHPWL) defined below to quantify the quality of placements:

$$\text{sHPWL} = \text{HPWL} \cdot (1 + \text{ABU}),$$

where HPWL is the wirelength metric, and average bin utilization (ABU) is used to evaluate placement density [56]. The bin used for ABU density calculation contains multiple rows of power tiles.

Problem 4.2.1 (EM-Aware Detailed Placement)**.** Given an initial legalized detailed placement and the EM DC current limit I_{limit}, we seek a legal placement to minimize the number of EM violations and further reduce sHPWL.

4.2.2 Algorithms

4.2.2.1 Cell Move

The major objective of the first technique is to achieve cell current balance among power tiles from a global scope. We use the cell move approach [27,97] and try to move cells out from current-overfilled tiles to other tiles which can accommodate them and help improve sHPWL.

Power tiles with total currents greater than the threshold $2I_{\text{limit}}$ will definitely have EM violations and such violations cannot be fixed by local permutation. In addition, an EM violation may exist for a power tile even if its total current density is below the threshold. Therefore, we define a current threshold parameter t_c and the tiles with total cell current greater than $2I_{\text{limit}} \cdot t_c$ are regarded as the source tiles for cell move. The cell with the largest current will be moved to a target tile in the search region [73] that has enough area and current capacity. Note that the total current of the target tile after the cell movement should be less than the predefined $2I_{\text{limit}} \cdot t_c$ to avoid cell move loops. We sort the candidate target tiles according to their distances to the optimal region [94] and the tile with the minimum cost will be chosen. The cost of moving cell i to target tile j is defined as:

$$\text{cost}(i, j) = \Delta\,\text{sHPWL}(i, j) + \beta \cdot \text{density}(j),$$

where $\Delta\,\text{sHPWL}(i, j)$ denotes the sHPWL change and $\text{density}(j)$ denotes the total current density of tile j after the cell movement. Both of them are normalized in the scale of site half-perimeter.

We only decide the target tile for the cell to move into by the above procedure. The cell is temporarily put in the center of the target tile, and its precise location and possible overlaps will be solved by the subsequent techniques. We iteratively repeat the above procedure until we cannot find cells to move or the maximum number of iterations is reached.

4.2.2.2 Single Row Placement

After cell move, we perform the ordered single row placement that minimizes wirelength under the total current constraint for each power tile in a row. The ordered single row placement for minimizing wirelength has been well-studied [19, 52, 113, 117]. Under the maximum displacement m for each cell, the problem can be transferred to the shortest path problem, and a DP-based algorithm is able to solve it in $O(m^2 n)$ [113, 139]. However, the additional constraint of total current makes the problem more complicated, which cannot be solved by the above approaches. We define the ordered single row placement problem under the total current constraint in Problem 4.2.2. We provide our main SINGLEROWDP algorithm in Theorem 4.2.1 which invokes SINGLETILEDP in Theorem 4.2.2. Note that our entire algorithm is still optimal even if we replace SINGLETILEDP by other algorithms that are able to output optimal solutions, which makes our single row placement algorithm widely applicable. The straightforward way to implement SINGLETILEDP has quadratic dependence in m, but we can achieve linear dependence in m by using some standard tricks [60]. For any positive integers n, m, we use $[n]$ to denote set $\{1, 2, \cdots, n\}$, and $[n, m]$ to denote set $\{n, n+1, \cdots, m\}$. To give a more general formulation, we use *cost* to denote wirelength and *value* to denote current.

Problem 4.2.2 (Fixed Order Single Row Placement)**.** Given n ordered cells, M locations and B tiles, m denotes the maximum displacement and L_i denotes a set of feasible locations[1] where the i-th cell can be placed, i.e., $\max_{i \in [n]}$ $L_i = m$. Let $c_{i,j}$ and $v_{i,j}$ denote the cost and value corresponding to placing the i-th cell at the j-th location, $\forall i \in [n], j \in L_i$. Let $v_{\max}(= 2I_{\text{limit}} \cdot t_c)$ denote

[1] Note that L_i is a set of consecutive integers (i.e., $L_i = [x, x+1, \cdots, y-1, y]$) in the problem as we claimed. Our algorithm is also working for the general case that L_i contains gaps.

the value threshold. The goal is to find a feasible, non-overlapping placement that keeps the initial order ($\forall i \in [n-1], \pi(i) < \pi(i+1)$, where $\pi(i) \in L_i$ is the new location for i-th cell), such that the value (current) constraint is satisfied, and the total cost (wirelength) is minimized.

Theorem 4.2.1 (Single Row Dynamic Programming)**.** *There is an algorithm (Procedure* SINGLEROWDP *in Algorithm 4.1) running in* $O(Bt^3m + Bn)$ *time*[2] *that is able to output a placement* $\pi : [n] \to [M]$ *such that* $\forall b \in [B]$, $\sum_{i,\pi(i)\in J_b} v_{i,\pi(i)} \leq v_{\max}$ *holds, and* $\sum_{i\in[n]} c_{i,\pi(i)}$ *is minimized, where* J_b *denote the set of locations belong to tile* $b \in [B]$*, and* t *denote the maximum number of cells per tile.*

Proof. Let $f_{i,j}$ denote the cost that all the first i cells are placed in the first j tiles if there is no violation over all the first j tiles, otherwise $f_{i,j} = \infty$. Let $\widehat{f}_{i_1,i_2,j}$ denote the cost that for placing from the i_1-th cell to the i_2-th cell to tile j if there is no violation, otherwise $\widehat{f}_{i_1,i_2,j} = \infty$. The total running time consists of three parts. The first part (lines 2–5) is computing all Q_j and $\widehat{L}_i^j$ in $O(Bn)$ time, where Q_j is the set of cells that can be placed in tile j, and $\widehat{L}_i^j$ denotes the feasible locations for cell i in tile j. Before we define $t = \max_{j\in[B]} |Q_j|$. The second part (lines 6–8) is from calling SINGLETILEDP $O(Bt^2)$ times and the running time of SINGLETILEDP is $O(tm)$. Thus, the running time for the second part is $O(Bt^3m)$. The third part (lines 9–11) is

[2] Our current result assumes that $v_{i,j} = v_{i,j'}$ for any j, j' in the same tile. Our algorithm can be extended to the case without that assumption, the running time becomes $O(Bt^3m \cdot v_{\max} + Bn)$.

dominated by computing set $S_{i_2,j}$ $O(Bt)$ times, and computing $S_{i_2,j}$ takes $O(t)$ time. Overall, the running time is $O(Bt^3m + Bn)$. The correctness can be proved by induction. Let $S_{i_2,j}$ denote a set of possible states i_1 such that i_2 is coming from i_1. For each iteration, we update $f_{i_2,j}$ by taking the minimum from $|S_{i_2,j}|$ states. For each state, suppose we transform from some $i_1 \in S_{i_2,j}$, the cost contains two parts: the first part is the cost of placing all the first i_1 cells in the first $j-1$ tiles, i.e., $f_{i_1,j-1}$; the second part is the cost of placing cells $i_1+1, \cdots, i_2$ in the tile j, i.e., $\widehat{f}_{i_1+1,i_2,j}$. □

Remark 4.2.1. For simplicity, our DP algorithms (in Algorithm 4.1) demonstrate the case where each cell has the same unit site, all the tiles have the same length, and the maximum displacement for each cell is the same. It is easy to extend it to the general setting. We also omit the details of backtracking to output the optimal solution.

Theorem 4.2.2 (Single Tile Dynamic Programming)**.** *Given t cells and a tile with ℓ locations, let L_i denote a set of feasible locations for cell $i \in [t]$ and $m = \max_i |L_i|$. Let $c_{i,j}$ denote the cost of placing cell i at the j-th location. There is an optimal algorithm*[3] *(Procedure* SINGLETILEDP *in Algorithm 4.1) running in $O(tm)$ that is able to output a placement $\pi : [t] \rightarrow [\ell]$ such that $\sum_{i\in[t]} c_{i,\pi(i)}$ is minimized.*

Proof. Let $g_{k,l}$ denote the optimal cost of cells $i_1, \cdots, k$ being placed in the

[3] The running time is optimal, because the input size (the number of feasible locations) is already $\Omega(tm)$.

Algorithm 4.1

```
procedure SingleRowDP($(c, v)$)                                        ▷ Theorem 4.2.1
    for $j = 1 \rightarrow B$ do
        $Q_j \leftarrow \{i | L_i \cap J_j \neq \emptyset, i \in [n]\}$
        for $i \in Q_j$ do
            $\widehat{L}_i^j \leftarrow L_i \cap J_j$
        end for
    end for
    for $j = 1 \rightarrow B; i_1 \in Q_j; i_2 \geq i_1, i_2 \in Q_j$ do
        if $\sum_{i=i_1}^{i_2} v_{i,j} \leq v_{\max}$ then
            $\widehat{f}_{i_1,i_2,j} \leftarrow$ SingleTileDP$(c, \widehat{L}^j, i_1, i_2)$
        end if
    end for
    for $j = 1 \rightarrow B; i_2 \in Q_j$ do
        $S_{i_2,j} \leftarrow \{i_1 | i_1 \in Q_j, i_1 \leq i_2, f_{i_1,j-1} \neq \infty, \widehat{f}_{i_1+1,i_2,j} \neq \infty\}$
        $f_{i_2,j} \leftarrow \min_{i_1 \in S_{i_2,j}} (f_{i_1,j-1} + \widehat{f}_{i_1+1,i_2,j})$
    end for
    return $\min_{j \in [B]} f_{n,j}$
end procedure
procedure SingleTileDP$(c, L, i_1, i_2)$                              ▷ Theorem 4.2.2
    for $k = i_1 \rightarrow i_2$ do
        $\tau(k)$ to be the last location of $L_k$
        for  $l \in L_k$ do
            if $l - 1 \notin L_{k-1}$ then
                $l' \leftarrow \tau(k-1)$
            else
                $l' \leftarrow l - 1$
            end if
            $g_{k,l} \leftarrow \min(g_{k,l-1}, g_{k-1,l'} + c_{k,l})$
        end for
    end for
    return $g_{i_2,\tau(i_2)}$
end procedure
```

first l locations of the tile. L_k denotes a set of feasible locations that cell k can be placed, we use $\tau(k)$ denote the last location of L_k. Because we have two nested for loops, one is going through all the cells, and the other is going through all the feasible locations of the cell, thus the running time is

$O(\sum_{i=i_1}^{i_2}|L_i|) = O((i_2 - i_1 + 1)m) = O(tm)$. By induction, we can show that the optimal solution is $g_{i_2,\tau(i_2)}$. In each iteration, we update the $g_{k,l}$ by taking the minimum of two states. One is placing cell k at location l, whose cost is $g_{k-1,l'} + c_{k,l}$. Note that we cannot set l' to be $l-1$ directly, because it is possible that $l-1$ is not a feasible location for cell $k-1$. The other state is not placing cell k at location l, whose cost is $g_{k,l-1}$, and $g_{k,l-1} = \infty$ if $l-1 \notin L_k$. $\square$

Our SingleTileDP algorithm finds the optimal solution by checking the states whether the current cell is placed at a certain location. Thus, the time complexity of it is $O(tm)$, which is the same as the work in [74] by pruning solution spaces. In most cases, the single row placement can help reduce the number of the current-overfilled tiles to zero. However, in some extreme cases with tight maximum displacement or existence of blockages, it may fail because the cells cannot be shifted much in the row.

4.2.2.3 Single Tile Placement

After the steps mentioned above, we determine the cells in each power tile. We now present the single tile placement which helps address the maximum current violation in a power tile if the total current constraint has been satisfied.

Problem 4.2.3 (Single Tile Placement)**.** Given the cells within a power tile, find a non-overlapping placement for these cells so that HPWL is minimized under the constraint that the maximum current in the power tile is less that the EM current limit I_{limit}.

In this section, we use $\mathcal{C}$ to denote the set of cells in the power tile and $\mathcal{N}$ to denote the set of nets. Let $\mathcal{L}$ denote set of all possible site locations in the tile. For the i-th cell in $\mathcal{C}$, we use W_i to denote its width and I_i to denote the its current. W denotes the width of the entire power tile. Let $p_{i,k}$ denote the horizontal distance between the i-th cell and the pin corresponding to the i-th cell associated with the k-th net.

MILP Formulation A power tile with an EM violation usually contains more than ten cells and thus the sliding window approach for cell reordering [21] cannot be applied. Placement problems using mixed integer programming (MIP) [22] and MILP [42, 66], by contrast, are more scalable by applying branch-and-cut approach. Hence, we propose an MILP formulation to determine the order and locations of cells in the power tile and consider the no-overlap and maximum current constraints simultaneously. Since cells are only moved inside the power tile, the y-coordinate of each cell is fixed. Assuming the cells in other tiles is also fixed for the time, the total HPWL can be formulated by the sum of the difference between the left and right boundary of the bounding box for each net. We define the MILP model minimizing the total HPWL as follows:

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{y},\mathbf{l},\mathbf{r}} \quad & \sum_{k\in\mathcal{N}}(\mathbf{r}_k - \mathbf{l}_k) && \text{(4.4a)}\\
\text{s.t.} \quad & \mathbf{x}_{i,j}, \mathbf{y}_{i,j} \in \{0,1\},\ \forall i \in \mathcal{C}, j \in \mathcal{L}, && \text{(4.4b)}\\
& \sum_{j\in\mathcal{L}} \mathbf{x}_{i,j} = 1,\ \forall i \in \mathcal{C}, && \text{(4.4c)}\\
& \sum_{i\in\mathcal{C}} \mathbf{y}_{i,j} \leq 1,\ \forall j \in \mathcal{L}, && \text{(4.4d)}\\
& \sum_{j\in\mathcal{L}} \mathbf{y}_{i,j} = W_i,\ \forall i \in \mathcal{C}, && \text{(4.4e)}\\
& \sum_{j=j'}^{j'+W_i} \mathbf{y}_{i,j} - W_i \cdot x_{i,j'} \geq 0,\ \forall i \in \mathcal{C}, \forall j' \in \mathcal{L}, && \text{(4.4f)}\\
& \sum_{i\in\mathcal{C}}\sum_{j\in\mathcal{L}} I_i \cdot (j + W_i/2) \cdot \mathbf{x}_{i,j} \leq I_{\text{limit}} \cdot W, && \text{(4.4g)}\\
& \sum_{i\in\mathcal{C}}\sum_{j\in\mathcal{L}} I_i \cdot (W - j - W_i/2) \cdot \mathbf{x}_{i,j} \leq I_{\text{limit}} \cdot W, && \text{(4.4h)}\\
& \mathbf{l}_k \leq \sum_{j\in\mathcal{L}} j \cdot \mathbf{x}_{i,j} + p_{i,k} \leq \mathbf{r}_k, \forall i \in \mathcal{C}, \forall k \in \mathcal{N}. && \text{(4.4i)}
\end{aligned}
$$

Formulation (4.4) is optimized over four kinds of variables, where $x_{i,j}, y_{i,j}$ are binary variables and l_k, r_k are continuous variables. If the lower left corner of cell i locates at site j then $x_{i,j} = 1$, otherwise $x_{i,j} = 0$. If cell i occupies site j then $y_{i,j} = 1$, otherwise $y_{i,j} = 0$. Constraint (4.4c) makes sure that each cell is placed at only one location, and Constraint (4.4d) guarantees that cells do not overlap. Constraints (4.4e) and (4.4f) ensure that each cell occupies the same number of total sites as its width, and all occupied site locations for the cell are contiguous. According to Equation (4.1) and Equation (4.2), we calculate the currents in the leftmost and rightmost power rail segments and force them to be smaller than the maximum current limit in Constraint (4.4g) and (4.4h). The left boundary l_k and right boundary r_k of the bounding box of each net are defined in Constraint (4.4i).

Speedup Techniques The aforementioned MILP formulation is optimal as it considers the orders of cells and spacing simultaneously, but may suffer from long runtime overhead. Here we propose a set of speedup techniques that breaks the single tile placement into two phases, where cells are reordered and shifted subsequently.

Figure 4.3 illustrates the fast way to solve the single tile placement problem. Given the cells in the problematic power tile (Figure 4.3(a)), we pack all the cells to the center (Figure 4.3(b)) and run the MILP algorithm shown in Formulation (4.5) to determine the order of cells and ignore whitespaces in the tile temporarily under the maximum current constraint (Figure 4.3(c)). After that, the tile-based ordered placement SingleTileDP (in Algorithm 4.1) is run to determine the locations of cells if the wirelength can be improved further under the maximum current constraint. The final placement is shown in Figure 4.3(d).

Let D denote the distance of the leftmost cell to left power stripe, we define FastMILP as follows:

$$\min_{\mathbf{z},\mathbf{s},\mathbf{l},\mathbf{r}} \sum_{k\in\mathcal{N}} (\mathbf{r}_k - \mathbf{l}_k) \tag{4.5a}$$

$$\text{s.t. } \mathbf{z}_{i,j} \in \{0,1\},\ \forall i \neq j \in \mathcal{C}, \tag{4.5b}$$

$$\mathbf{z}_{i,j} + \mathbf{z}_{j,i} = 1,\ \forall j > i \in \mathcal{C}, \tag{4.5c}$$

$$\mathbf{z}_{i,j} + \mathbf{z}_{j,k} - \mathbf{z}_{i,k} \leq 1,\ \forall i \neq j \neq k \in \mathcal{C}, \tag{4.5d}$$

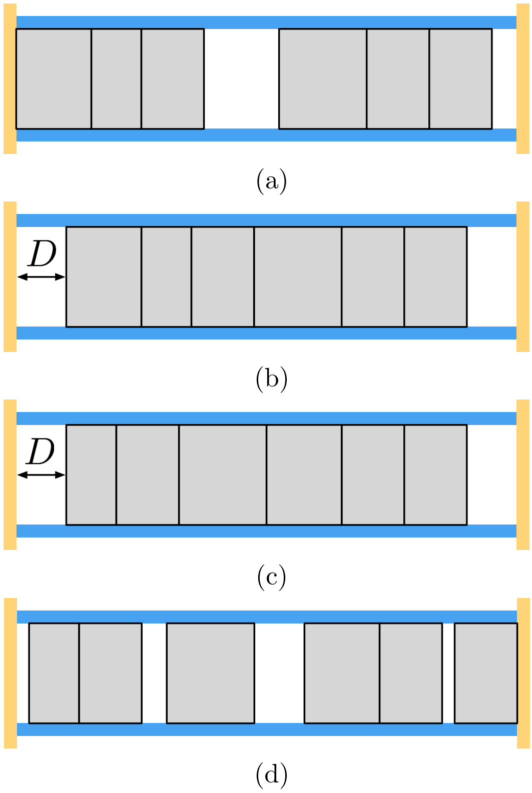
$$\mathbf{s}_i = \sum_{j\neq i} \mathbf{z}_{j,i} \cdot W_j + D,\ \forall i \in \mathcal{C}, \tag{4.5e}$$

$$\sum_{i\in\mathcal{C}} I_i \cdot (\mathbf{s}_i + W_i/2) \leq I_{\text{limit}} \cdot W, \tag{4.5f}$$

$$\sum_{i\in\mathcal{C}} I_i \cdot (W - \mathbf{s}_i - W_i/2) \leq I_{\text{limit}} \cdot W, \tag{4.5g}$$

$$\mathbf{l}_k \leq \mathbf{s}_i + p_{i,k} \leq \mathbf{r}_k, \forall i \in \mathcal{C}, \forall k \in \mathcal{N}. \tag{4.5h}$$

Formulation (4.5) is optimized over four kinds of variables, where $\mathbf{z}_{i,j}$ is a binary variable, and $\mathbf{s}_i$, $\mathbf{l}_k$ and $\mathbf{r}_k$ are continuous variables. We use variable $\mathbf{z}_{i,j}$ to represent the relative order of cell i and j. If cell i is on the left of cell j then $\mathbf{z}_{i,j} = 1$, else $\mathbf{z}_{i,j} = 0$. Variable $\mathbf{s}_i$ denotes the placement site of lower left corner of that cell. For any three cells i, j and k, we also need to make sure that if cell i is on the left of cell j and cell j is on the left of cell k, then cell i must be on the left of cell k, which is guaranteed by Constraint (4.5d). Constraint (4.5e) transfers the relative orders of a group of cells to their site locations. The maximum current limit constraint is addressed by Constraint (4.5f) and Constraint (4.5g). Constraint (4.5h) is identical to Constraint (4.4i) in Formulation (4.4) to formulate HPWL.



(a)

(b)

(c)

(d)

Figure 4.3: Speedup techniques for single tile placement. (a) The initial placement, (b) the placement after packing cells, (c) the placement after cell reordering and (d) the final placement after SingleTileDP.

Notice that the number of binary variables in Formulation (4.4) is $2 \cdot |\mathcal{C}| \cdot |\mathcal{L}|$, and the number of binary variables in Formulation (4.5) is $|\mathcal{C}|^2 - |\mathcal{C}|$. It is observed that $|\mathcal{L}| \gg |\mathcal{C}|$ in our benchmarks, which is the reason for that solving Formulation (4.5) is much faster than Formulation (4.4) in practice. Although these speedup techniques cannot guarantee an optimal solution of the single tile placement problem, experimental results demonstrate that it can achieve noticeable runtime speedup without sacrificing too much performance.

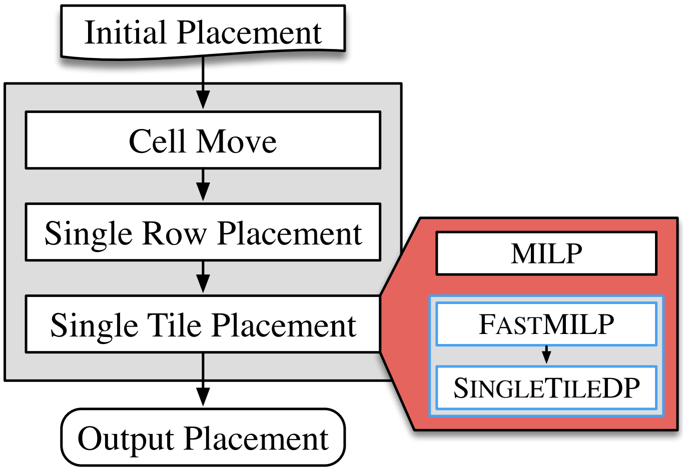


Figure 4.4: Overall flow of our detailed placement techniques.

There are some corner cases where the maximum current constraint cannot be satisfied by any cell placement within the tile, even if the total current of the tile is less than the threshold. Our MILP models become infeasible in this situation and will report that no feasible solution can be found.

4.2.2.4 Overall Flow

The proposed detail placement flow for power grid EM mitigation is shown in Figure 4.4. The first two stages are cell move and single row placement to reduce the total currents in current-overfilled tiles. The third stage is single tile placement to reduce the maximum current in each of power tiles with EM violations. Single tile placement has two available algorithms, including the original MILP algorithm and a series of speedup techniques (FastMILP and SingleTileDP).

4.2.3 Experimental Results

Our placement framework was implemented in C++ and run on a 3.40 GHz Linux machine with 32 GB memory. Since EM violations are more noticeable at advanced nodes, we validated our algorithms on the set of benchmarks from [74], which integrated NanGate 15*nm* standard cell library [7] into ICCAD 2014 placement benchmarks [55] and used RippleDP [27] to generate the initial detailed placements. Table 4.1 presents the characteristics of this set of benchmarks. ICCAD 2014 placement contest defines two maximum displacement limits for each design, and we chose the smaller one for less perturbation to the original placements, as listed in column “Disp. (um)”. GUROBI [40] was used as the MILP solver. We set the user-defined parameters t_c and β to 0.7 and 5. Note that for some extremely dense power tiles, it takes a long time to find the optimal solution. Thus, we set a time limit for each run of the MILP solver to 200s.

We set the supply voltage, operating temperature and clock frequency to 0.88V, 125°C and 1GHz in the experiments. Cell current was calculated from the NLDM file in NanGate 15nm standard cell library [7]. We studied the typical values of the metal width of power grids at 16-nm nodes and set power rail and power stripe wire width to 0.09um and 0.32um, respectively. The power tile width was set to 5.76um. The EM DC current limit under this setting was 0.067mA. To demonstrate the effectiveness of the proposed flow, we set the EM DC current limit tighter to 0.026mA for benchmarks *b19*, *mgc_edit_dist* and *mgc_matrix_mult* with small initial violation numbers

Table 4.1: Benchmark Characteristics

Design	#cells	#nets	#blk	Density	Target util.	Disp.(um)
vga_lcd	165K	165K	0	68.94%	70%	10
b19	219K	219K	0	44.85%	70%	20
leon3mp	649K	649K	0	72.02%	75%	30
leon2	794K	795K	0	84.19%	90%	40
mgc_edit_dist	131K	133K	13	67.26%	70%	30
mgc_matrix_mult	155K	159K	16	59.31%	65%	30
netcard	959K	961K	12	66.29%	70%	50

Table 4.2: Result comparison between two flows.

Design	Initial			MILP flow				Fast flow			
	EM vio.	HPWL	sHPWL	EM vio.	ΔHPWL	ΔsHPWL	CPU	EM vio.	ΔHPWL	ΔsHPWL	CPU
	#	$\times 10^6$	$\times 10^6$	#	%	%	s	#	%	%	s
vga_lcd	127	1.421	1.874	0	0.076	0.012	52.72	0	0.073	0.009	7.99
b19	363	0.965	1.139	2	0.279	0.113	32.93	4	0.290	0.086	15.04
leon3mp	738	5.340	6.837	3	0.125	0.135	254.22	7	0.125	0.131	99.13
leon2	2076	13.092	14.487	35	0.561	0.637	4445.77	39	0.558	0.626	405.26
mgc_edit_dist	100	1.525	1.880	0	0.185	-0.273	5.66	0	0.185	-0.273	5.46
mgc_matrix_mult	144	0.912	1.126	0	0.149	-0.534	80.01	0	0.151	-0.533	7.14
netcard	2424	14.570	20.189	197	0.773	1.304	5314.47	206	1.040	1.430	678.35
avg.	853.1	5.404	6.790	33.9	0.307	0.199	1455.11	36.6	0.346	0.211	174.05
ratio	1			0.040			1	0.043			0.12

(≤ 40).

Table 4.2 shows the detailed placement results. "MILP flow" and "Fast flow" denote the flows using the original MILP algorithm (Section 4.2.2.3) and the speedup techniques (Section 4.2.2.3) respectively in the single tile placement step. "EM vio." denotes the number of EM violations. "Δ HPWL" and "Δ sHPWL" denote the change in wirelength and scaled wirelength. "CPU" denotes the total runtime in second. The MILP flow is effective to fix 96% of the EM violations on average and its impacts on wirelength (0.31%) and placement density (0.2%) are very small. Furthermore, compared with the MILP flow, the fast flow is at least 8× faster while achieving comparable results, regarding EM violation reduction (95.4%), wirelength (0.35%), and density (0.21%). It is worth mentioning that the fast flow performs better than the MILP flow for the *vga_lcd* benchmark. It is because although the MILP approach theoretically gives an optimal solution to the power tile placement, the optimal solutions to the local sub-problems do not necessarily lead to the final globally optimal solution.

The key ideas of our work are first reducing the total currents for current-overfilled power tiles by cell move and single row placement, and further reducing the maximum current in each power tile by single tile placement. Figure 4.5 demonstrates that the two types of placement techniques are indispensable for EM violation reduction. The first two techniques for total current reduction can remove a part of the EM violations, but more EM violations are fixed by single tile placement. As shown in Figure 4.6, the initial number of

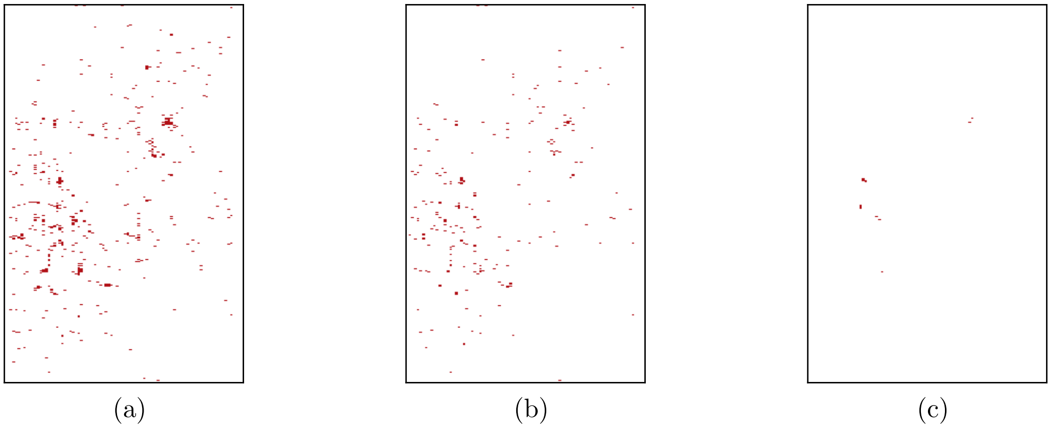


Figure 4.5: The EM violation map of design *leon2*. The tiles with violations are marked in red. (a) The initial design, (b) after single row placement, and (c) after single tile placement.

EM violations is largely dependent on the EM current limit, and our placement flow can achieve zero EM violations in a wide range of current limits. The proposed placement techniques are designed to serve as an incremental placement step that can be used in ECO stage as well. It can be easily integrated into the existing physical design flow to eliminate EM violations iteratively.

4.2.4 Summary

This work presents a set of detailed placement mitigation techniques to handle power grid EM, including cell move, single row placement, and single tile placement. Experimental results show that these techniques achieve high-quality placement results regarding EM violation reduction, total wirelength, and placement density.

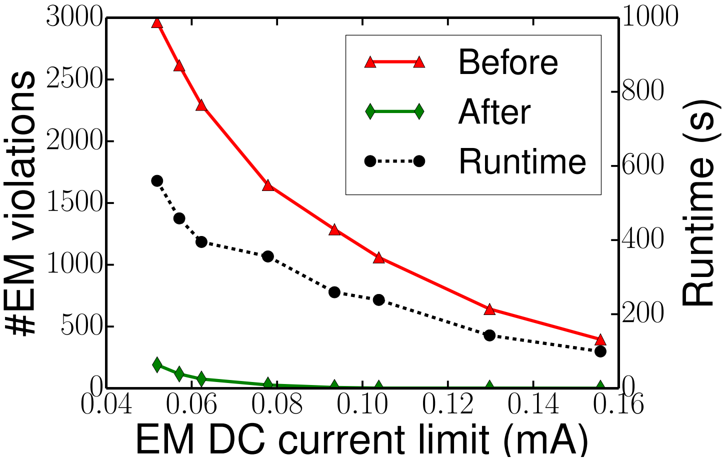


Figure 4.6: The number of EM violations before and after placement and runtime vs. EM DC current limit for design *leon2*.

4.3 Learning-Based Detection and Multistage Mitigation for Signal Electromigration

While EM and its impact on design reliability have long been understood, in the past, its effects were limited and were not a severe concern for signal nets. Signal wires were considered to be less susceptible to EM failures than power grids, as they usually carry small average currents and benefit from the healing effect of bidirectional currents [69]. However, the scaling trends in advanced technology nodes, along with strict and complex EM rules, make advanced designs more susceptible to signal EM. Driven by the continuous push for better performance, signal nets that are usually thin and long are expected to switch at gigahertz speed, a scenario that further exacerbates the effects of signal EM in advanced technology nodes [1, 51]. Besides, EM rules have also become more complex to take into consideration the physical effects of advanced technologies. In mature technology nodes, the rule-specified limits provided by the foundry were primarily metal width and temperature. At 28 nm and below, we see the addition of more dependencies, such as interconnect length, via dimensions, and temperature increase.

EM failure has been dealt with at different design stages, including placement [138] and routing [24, 49, 67, 93]. However, the focus has been concentrated on applying optimization techniques after EM violations are already detected, or using approximation methods to guess possible violations. In this work, we propose a novel signal EM hotspot detection and mitigation framework based on information available at the placement phase. In particular, three main steps constitute our proposed approach. As a first step, a classification model is trained through machine learning techniques to detect signal EM hotspots based on features extracted from the placement scheme. This model can be trained using data obtained from designs where EM hotspots are already known, and then, it can be applied to detect hotspots in new designs. In addition to its main role in hotspot prediction, the model helps identify the placement-based features that are critical for hotspot identification. Knowing these features is fundamental for constructing effective EM adjustment techniques at the placement stage.

In the second step, the placement scheme is adjusted by incorporating signal EM hotspot mitigation mechanism in the cost function of the placement problem. This mechanism incorporates the detection model information about critical features to address the EM hotspots. At the end of this step, a new placement is obtained. As a last step, the classification model is used again to detect hotspots still present after placement adjustment and non-default routing (NDR) rules are applied to address these hotspots in the routing stage.

4.3.1 Problem Formulation

The key idea of our proposed approach is to leverage machine learning techniques to detect EM hotspots at placement stage and exploit the trained models to guide EM mitigation. An overview of the framework is presented in Figure 4.7.

Figure 4.7(a) shows the process of training an EM hotspot prediction model. Starting from the input netlist of the training set, a PD tool is used to get the placement result. Next, routing and EM evaluation are performed to get the EM hotspots in the designs. Finally, the placement information is used along with the EM hotspot results to train a classification model for EM detection.

Figure 4.7(b) demonstrates the application of the EM detection and mitigation framework. After having a trained model for EM detection, PD tool is used to do placement, and then EM hotspots are predicted using the classification model given the placement-related features. Next, placement is incrementally updated to mitigate predicted EM hotspots. Then, the EM detection model is used again to detect remaining hotspots that are finally routed using NDR rules.

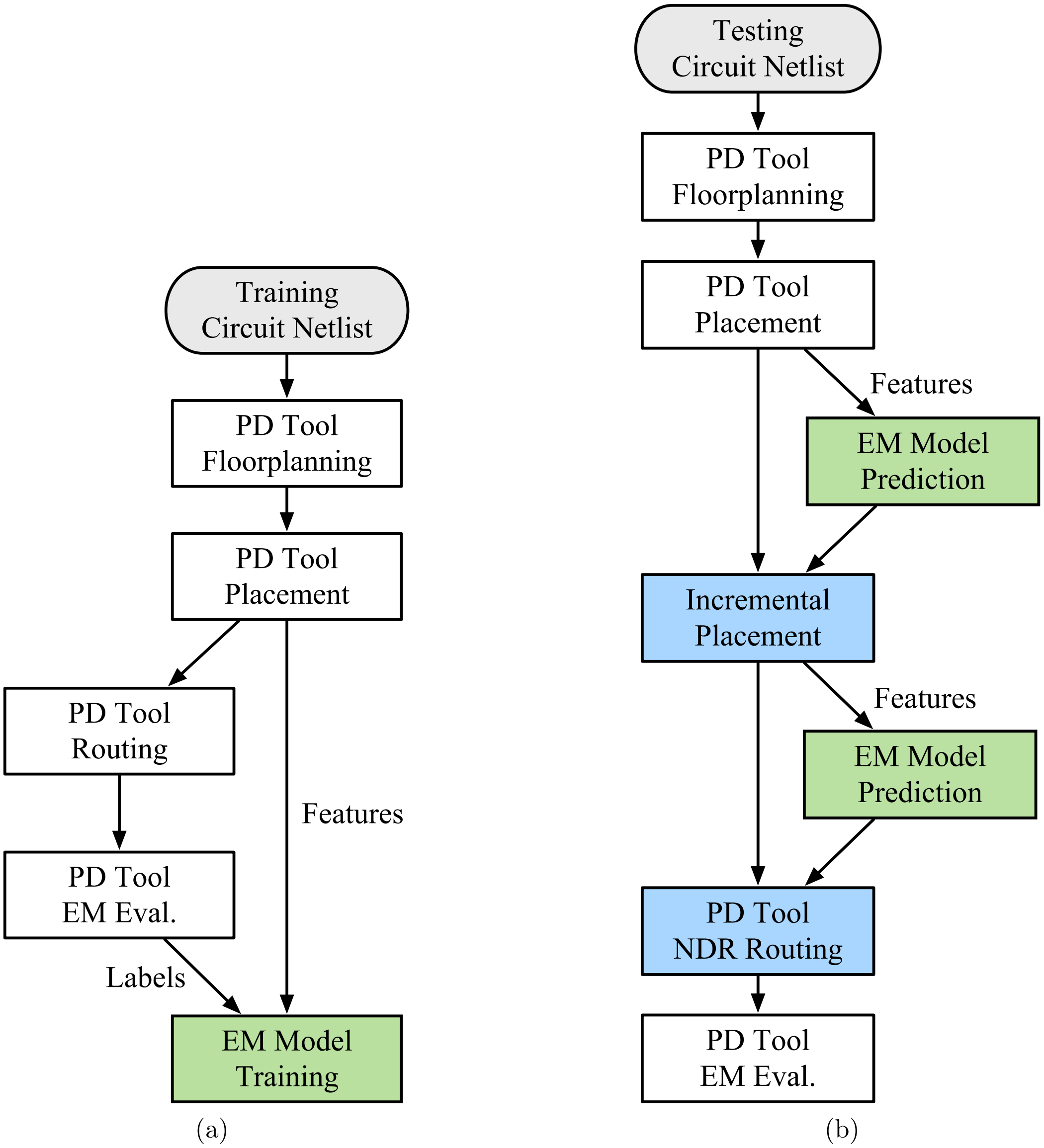


Figure 4.7: An overview of the hotspot detection model training (a) and its application in EM detection and mitigation (b) is shown.

4.3.2 Machine Learning for EM Detection

4.3.2.1 Features Extraction

Despite the fact that the current profile for the design is not available at the placement stage, multiple features that are highly correlated with the current can be crafted. To elaborate on this, we consider the three nets in Figure 4.8, A, B and C. One can expect net A to have the highest current density. This is mainly because, unlike the 2-pin nets B and C, A is a 6-pin net connected with two large cells. On the other hand, net C is the one least prone to EM. In practice, although both B and C are 2-pin nets, Figure 4.8 clearly shows that the neighborhoods around the pins of net B are more congested (i.e., high pin density). This in turn can lead to detours when routing net B; hence, longer wires, large wire capacitance and higher current.

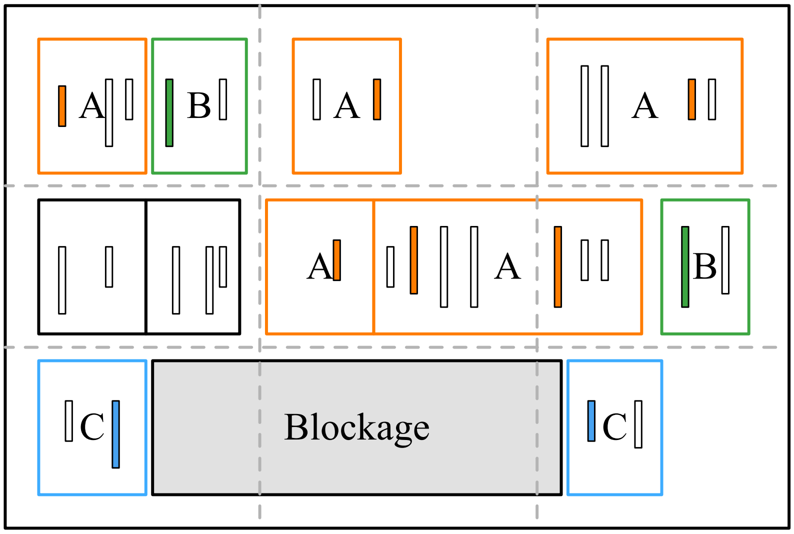


Figure 4.8: An illustration of a placement scheme with three nets is shown.

In our approach, we extract a set of features from the placement to be used for training the model for EM detection. These features can be divided into two categories: (i) net-specific features and (ii) neighborhood related features. The net-specific features used are the following:

1. Net half-parameter wirelength (HPWL)
2. Number of net pins
3. Net switching activity
4. Maximum fall transition time
5. Maximum capacitance
6. Circuit frequency

On the other hand, neighborhood related features are used to capture information about possible congestion around net pins. To define these features, the placement region is divided into a grid with fixed window size as shown by the gray-colored grid in Figure 4.8. Then, for each net, a set of features is defined over all grid windows containing pins connected to the net.

Using Figure 4.8 as an example, we consider the feature defined as the average number of pins. To compute this feature for net A, we first identify the grid windows containing pins of net A which are the three windows in the first row, and second and third window in the second row counting from the left. Then, we average the number of pins in the five windows counting all pins in the windows, not only those connected to net A. This results in a feature value equal to $\frac{21}{5}$ for net A. Computing the same feature for nets B and C gives 5 and 2 respectively. The full list of neighborhood related features used is as follows:

1. Average number of pins
2. Average number of cells
3. Average cell area
4. Average area capacity (space not occupied by blocks)
5. Average number of placement sites

It is important to note that all the features mentioned above can be extracted without any knowledge about the final routing scheme. Moreover, with the exception of switching activity that can be obtained through high-level hardware simulation, all features can be extracted from the placement scheme.

4.3.2.2 Data Preparation

Starting from the labeled training set, features defined in the previous section are extracted resulting in a feature vector with a Boolean class label for each net in the design. Two important characteristics of the resulting dataset should be examined. First, the dataset is significantly imbalanced. In other words, the EM hotspot class (H) is enormously outnumbered by the non-hotspot class (NH). Secondly, the different features have different ranges of values. For instance, HPWL has a wider range of possible values compared to the number of pins. These two characteristics can affect the training process and the interpretability of the model, and hence, they should be addressed before training.

In the scenario where the two classes are imbalanced, the training is expected to be biased towards the objective of learning the larger class while neglecting the errors in predicting the smaller one. Among the methods used to address such bias is class weighting where higher weights are given to instances in the smaller class when formulating the training objective. This can be done by associating different costs with mispredicting instances from different classes; i.e., mispredicting an instance from the smaller class is associated with higher cost compared to mispredicting an instance from the larger one.

On the other hand, having features with different ranges of values can affect both the model training and its interpretability. During training, numerical issues arising from such case can cause convergence problems. In addition, in distance-based classification models, different ranges of values can result in unwanted weighting for the features. Moreover, having features with different ranges makes the task of interpreting any model more challenging. For example, important features in a trained model are usually inferred from the weight given to each feature after the training phase. For the case where all features have similar ranges, it suffices to compare the absolute values of the weights to judge upon the importance of the features. However, with features taking values in different ranges, this comparison does not hold any more. Therefore, a normalization step is done to map all features to the $[0, 1]$ interval to ensure they all have the same weight when training the EM detection model.

4.3.2.3 Cascaded Model for False Alarm Avoidance

The EM detection problem can be cast into a classification problem. In practice, a wide range of classification models are available for use, and these models vary in their complexity and application space [16]. Two important characteristics of the EM detection application contribute to the decision upon the classification model to use. First, the problem is a binary classification problem (i.e., two class problem) with relatively small number of features. Secondly, the EM detection model is a part of an EM detection and mitigation framework. Hence, in addition to the detection task, we are interested in analyzing the trained model to arrive at the features contributing the most to the prediction decision. Knowing these features plays a significant role in the EM mitigation process described in the next section. Therefore, the interpretability of the trained model is critical from this perspective.

In practice, as the complexity of the classification model increases, interpretabilty becomes more challenging. And since the problem at hand is low-dimensional, we choose to use logistic regression [16, 18] as the classification model. Such model is known to behave well with binary classification problems and its regression coefficients can be used to interpret the importance of the different features.

As will be demonstrated in the result section, logistic regression can achieve high EM detection accuracy at a small false alarm rate. However, by examining the overall flow of the EM detection and migration framework and the relative number of H and NH instances, false alarm rate should be addressed from a different perspective. Technically, in a general classification problem, correctly labeling 99% of the target group (H in our case) with 3% false alarm rate can be acceptable. However, given that the two groups are unbalanced, even a 3% false alarm rate can result in a number of false alarms that is a multiple of that of H instances.

Hence, with such number of false alarms, mitigation techniques will perform a large number of unnecessary changes to the placement and routing schemes; thus, introducing additional overheads. To address this issue, we introduce the two-stage detection approach shown in Figure 4.9. In the first stage, a classification model M1 is trained to detect EM hotspots using all the nets in the training dataset. After the first stage, all nets with NH prediction will be labeled as NH without further processing. For nets labeled H by M1, a new model, M2, is trained to prune out false alarms. M2 is trained using nets in the training dataset labeled H by M1. For those nets going through the second stage, the final label will be the prediction of M2.

In practice, when two models are trained, inference for new nets can be done in a way analogous to the training process. First, an initial prediction is obtained by applying M1, and if the prediction is NH the net is given that as the final label. Otherwise, a new prediction is obtained from M2, and the final label is that generated by M2.

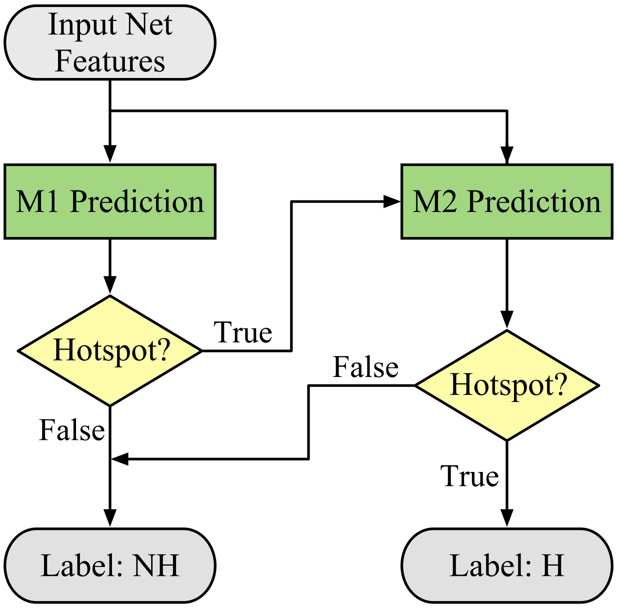


Figure 4.9: The flow of the two stage detection approach is shown.

This proposed approach helps reduce the number of false alarms while preserving the interpretability characteristic of the model. This translates to reducing the overhead incurred by the mitigation process.

4.3.3 Machine Learning guided EM Mitigation

4.3.3.1 Placement Adjustment

Besides its main role in detecting nets susceptible to EM failures, the trained EM detection model points out the potential directions to mitigate them The coefficients in the trained model indicate that wirelength and cell density, the two features that can be optimized in the given placement, contribute significantly to EM severity. Therefore, we propose an incremental placement approach to mitigate signal EM violations with minimal perturbation to the layout. The major purpose of this technique is to achieve selective wirelength reduction and cell density improvement.

Similar to a timing-driven placement [13], a net n_i is assigned a weight w_i based on its EM criticality. The higher the weight assigned to a given net is, the more is the push by the placer to reduce its wirelength. Considering cell density as well, the cost for a cell move is defined as:

$$\text{wHPWL}(1 + \alpha \cdot c_d), \tag{4.6}$$

where wHPWL is the weighted wirelength sum of all the nets connected to this cell, i.e., $\text{wHPWL} = \sum_i w_i \, \text{HPWL}(i)$, and c_d denotes the cell density cost computed according to [57, 57].

The incremental placement scheme is summarized in Algorithm 4.2. After detailed placement, PD tools are able to output high-quality placement results in term of timing, power, and routability of a design, which serve as the starting point for our signal EM optimization. As a first step, the trained EM prediction model is used to detect the set of EM hotspot nets $\mathcal{H}$ in the input placement scheme. Next, the set of cells $\mathcal{C}$ connected by the nets $\mathcal{H}$ is identified and reordered by their area. At this stage, the objective of the proposed incremental placer is to move the cells in $\mathcal{C}$ in a way to minimize the wirelength of the nets in $\mathcal{H}$ and mitigate the cell density around the target cells.

Algorithm 4.2 Cell Move

```
Input: Initial placement, predicted EM hotspot nets $\mathcal{H}$;
$\mathcal{C} \leftarrow$ set of cells connected by $\mathcal{H}$;
Reorder $\mathcal{C}$;
repeat
    for $c_i \in \mathcal{C}$ do
        Determine the search region of $c_i$;
        Move $c_i$ to the position that minimizes the objective;
    end for
until converged or maximum iteration reached
Legalize placement;
```

The principal idea is to find a search region for a cell in the placement region and move the cell to the best location in this region. Different from the classical optimal region calculation method [94], we use the weighted median to compute the optimal region for cell move since the nets have different weights in the current scheme. Then, the optimal region is extended to larger search region.

We perform wirelength optimization to improve both wirelength and density until less than 1% of the target cells are moved in an iteration or the maximum number of iterations is reached. After that, legalization is performed to remove possible overlaps.

4.3.3.2 Non-Default Routing for EM Adjustment

While the aforementioned incremental placement algorithm is tailored to address the EM hotspots, it does not guarantee the mitigation of all detected hotspots. In other words, some EM hotspots can be still present after the incremental placement adjustment stage. For example, for the nets with high fanouts, the current flowing through the main metal branch drives large capacitive loads, therefore, improving wirelength is not effective enough to solely resolve the current issue.

However, we can still utilize the EM prediction model after the proposed incremental placement. That is, we can set the router to route those predicted hotspot nets with wider widths to avoid iterative fixing. Practically, this option is readily available in many PD tools through the non-default routing (NDR) rule option. As the name implies, NDR applies non-default routing geometries to those selected nets in the design based on user specification; i.e., instead of the default single-width single-spacing (1W1S) scheme, a user set scheme can be used to route specific nets in the design. This option is leveraged to address the EM hotspots detected by the model in Section 4.3.2.3 after incremental placement using a double-width single-spacing (2W1S) NDR rule.

4.3.4 Experimental Results

Throughout the experiments, TSMC 40nm CMOS physical design kit (PDK) [6] was used for evaluating the efficacy of our proposed framework. Moreover, slow process, voltage and temperature (PVT) corners were used to generate a worst-case EM environment. The five benchmark circuit netlists used are taken from ICCAD 2014 placement contest [57] and OpenCores [4] respectively. In addition, physical design was performed using Synopsys IC Compiler (ICC) 2017 [5].

4.3.4.1 EM Prediction Model Comparison

Among the five available designs, three were used to train the hotspot detection model, while the remaining two were used for testing. The training data set consisting of designs `b19`, `ecg`, and `mmm` contains a total of 426152 nets of which 2681 are hotspots. Meanwhile, designs `med` and `vga` with 298197 nets, including 648 hotspots, are used for testing.

The confusion matrix summarizing the evaluation of the model on the testing data when a single stage logistic regression (M1) was used is shown in Table 4.3. While the results demonstrate high true positive rate, the number of false alarms is more than 10× the number of actual hotspots. On the other hand, Table 4.4 shows the confusion matrix when the cascaded model (M1+M2) described in Section 4.3.2.3 was used. One can notice a reduction of 65% in the number of false alarms at the cost of missing 21 of the hotspots. This cascaded model provides a compromise between the high accuracy of the hotspot detection and the overhead induced from fixing nets wrongly labeled as hotspots. The details will be demonstrated in Section 4.3.4.3.

Table 4.3: Confusion matrix of M1.

	NH	H
$\widehat{\text{NH}}$	290084	2
$\widehat{\text{H}}$	7465	646

Table 4.4: Confusion matrix of M1+M2.

	NH	H
$\widehat{\text{NH}}$	295178	23
$\widehat{\text{H}}$	2371	625

4.3.4.2 Incremental Placement + NDR

As mentioned earlier, our mitigation flow consists of two steps: incremental placement and NDR. We performed the two mitigation techniques on the five designs to verify the effectiveness. The placement algorithm in Section 4.3.3.1 was implemented in C++. During EM mitigation at the placement stage, at most 6 incremental placement iterations were allowed in the experiments, and the parameter α in formulation (4.6) was set to 1. We set the same weight w for all the hotspot nets as $w = 2000/\left|\mathcal{H}\right|$, while keeping unity weight for NH nets. The information of the detected hotspots is provided to ICC for performing NDR.

To demonstrate the efficacy of each individual component in the proposed multistage mitigation framework, we run several flows as described below and the results are summarized in Table 4.5. We first run clock tree synthesis (CTS) and default routing on the initial placement generated by the PD tool, and the number of the final EM violations without any repair approaches is shown under the column "Initial". Second, we performed incremental placement with the actual EM hotspots being known, and then run CTS and routing. The number of final EM violations under this flow is under column "Incr. place". Lastly, we run the incremental placement, CTS and NDR routing and reported the final number of EM violations under the column "Incr. place + NDR route". Note that the target EM violations to repair for all the flow shown in Table 4.5 are reported from ICC EM evaluation. It can be observed that incremental placement solely reduces 37.1% of the violations on average, and incremental placement and NDR routing reduces 74.1% violations, which is about the same performance achieved when using EM fixing in PD tool (73.1%).

Table 4.5: Comparison of PD tool EM repair, the incremental placement flow, and incremental placement combined with NDR flow in terms of final EM violations is shown.

Design	Initial	PD tool	Incr. place	Incr. place + NDR
b19	302	104	260	108
ecg	225	3	70	21
mmm	2,154	1,637	1,997	1,320
med	252	6	34	6
vga	396	80	360	83
Avg. improve	—	73.1%	37.1%	74.1%

4.3.4.3 Framework Validation

The EM detection model was trained on three benchmarks, b19, ecg and mmm, and we integrated the trained model into the proposed framework, which is applied to the five benchmarks. Table 4.6 reports the number of EM violations, routed wirelength (Wirelength), net area (Area), overall runtime, worst negative timing slack (WNS), and total negative timing slack (TNS) at the end of detailed routing in different flows. "Initial" denotes the default PD

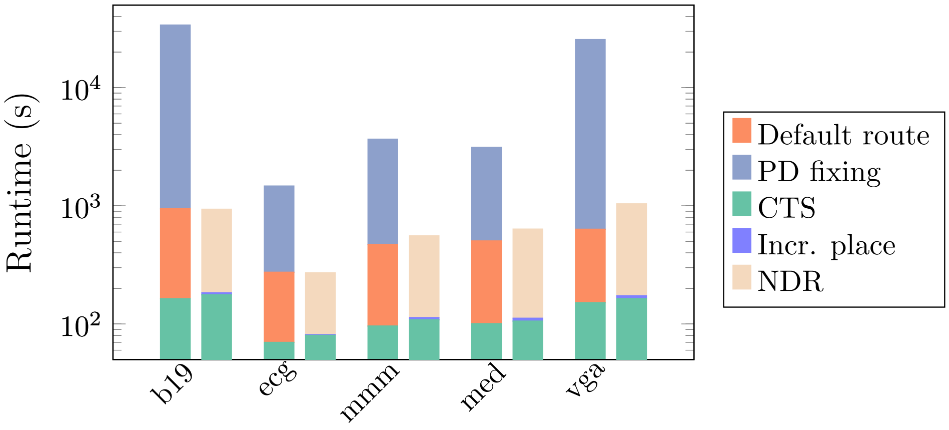


Figure 4.10: Runtime comparison between the PD fixing flow (CTS + default route + PD fixing) and the proposed M1+M2 flow (incr. place + CTS + NDR) is shown.

flow without any EM fixing attempts, while “PD fixing” denotes using wire widening throughout during the fixing stage. “M1” and “M1+M2” represent the proposed EM detection and mitigation flow with M1 and M1+M2 as the EM prediction model respectively.

We can see that the proposed M1+M2 flow fixes about the same number of EM violations as the PD fixing flow. It also achieves 15× speedup compared with the PD fixing flow. The runtime decomposition for the PD tool fixing flow and our proposed flow with the cascaded model is shown in Figure 4.10. One can see that, compared to PD fixing flow whose runtime is dominated by post-route EM fixing, the M1+M2 flow can perform the incremental placement in less than 10 seconds and NDR takes nearly the same runtime as the default routing.

Table 4.6: Comparison of EM violation reduction, metal wirelength and area overhead, and timing impact for the designs produced by the conventional EM fixing flow and our proposed methodology using machine learning trained model.

Design	Flow	#EM Vio.	Wirelength (um)	Area (um^2)	Runtime (s)	WNS (ns)	TNS (ns)
b19 Nets: 219,289	Initial	302	2,242,990	165,284	943.4	-0.11	-0.85
	PD fixing	104	2,260,090	188,171	33,865.1	-0.09	-3.83
	M1	116	2,368,201	221,096	1,293.5	-0.09	-0.61
	M1+M2	120	2,248,412	174,432	935.5	-0.10	-0.77
ecg Nets: 48,337	Initial	225	873,557	63,050	274.1	-0.17	-81.00
	PD fixing	3	874,541	63,420	1,470.1	-0.17	-81.97
	M1	3	996,912	67,072	420.6	-0.23	-83.2
	M1+M2	9	884,589	65,727	270.7	-0.20	-91.88
mmm Nets: 158,526	Initial	2,154	1,823,239	132,646	471.8	-0.15	-9.43
	PD fixing	1,637	1,824,374	138,799	3,663.9	-0.15	-9.43
	M1	1,245	1,872,680	191,545	724.4	-0.16	-10.52
	M1+M2	1,364	1,847,248	143,328	556.5	-0.16	-10.83
med Nets: 133,222	Initial	252	2,638,638	190,443	504.3	-0.19	-135.53
	PD fixing	6	2,642,620	199,231	3,124.5	-0.19	-141.38
	M1	11	2,746,344	260,535	12,971.9	-0.23	-161.12
	M1+M2	11	2,655,013	207,188	635.2	-0.22	-148.43
vga Nets: 164,975	Initial	396	3,169,437	227,432	633.7	-0.21	-116.74
	PD fixing	80	3,227,050	268,042	25,529.2	-0.23	-144.39
	M1	84	3,306,214	300,918	1,113.9	-0.17	-84.71
	M1+M2	87	3,228,308	272,295	1,038.8	-0.18	-86.02
Ratio wrt initial	PD fixing	0.269	1.006	1.083	19.10	0.983	1.760
	M1	0.246	1.062	1.307	6.38	1.052	0.955
	M1+M2	0.267	1.011	1.093	1.21	1.087	1.004

4.3.5 Summary

In this work, we propose a novel EM hotspot detection and mitigation framework using learning-based detection and multistage mitigation. Utilizing features extracted from the placement, a classification model is proposed to detect EM hotspots in the design. In addition, an incremental placement strategy is proposed to mitigate the detected EM hotspots. EM hotspots still present after the placement-stage mitigation are addressed through the NDR scheme in the routing stage. Contrary to conventional EM mitigation flows, the proposed approach addresses the EM problem at an earlier stage in the PD process resulting in faster closure and versatile mitigation techniques.

Chapter 5

Conclusions

This dissertation has proposed a set of physical design algorithms and modeling methodologies to enhance design manufacturability and reliability. The major contributions include:

- Chapter 2 has explored machine learning approaches to improve accuracy and efficiency for lithography hotspot detection. (a) Section 2.2 has presented LithoGPA, a hotspot detection framework with Gaussian Process assurance to provide confidence in classifier prediction. Besides, an active data selection scheme based on weak classifiers is developed to reduce the computational cost in data preparation. (b) Section 2.3 has investigated the usage of AUC as a robust measure of classifier discrimination performance. Different surrogate loss functions for AUC maximization are proposed to be used during training to systematically handle the class imbalance problem. Experimental results demonstrate that the proposed loss functions are promising to outperform the traditional cross-entropy loss when applied to the state-of-the-art neural network model for hotspot detection.

- Chapter 3 has studied effective and efficient lithography modeling methodologies to enable fast design closure and improve manufacturing yield. (a) Section 3.2 has presented the LithoGAN framework for end-to-end lithography modeling. LithoGAN is a dual learning network that predicts the resist shape using a CGAN model and predicts resist center using a CNN model. The experiments on N10 and N7 datasets demonstrate that the proposed framework predicts resist patterns of high quality while obtaining orders of magnitude speedup compared to conventional lithography simulation and previous machine learning based approach. (b) Section 3.3 has described TEMPO, a lithography modeling framework for mask topography effects that is capable of generating 3D aerial images efficiently and accurately. The two flexible schemes of operations in TEMPO provide different trade-offs between accuracy and efficiency, which promotes the wider application of TEMPO in different stages of process development.

- Chapter 4 has focused on physical design for emerging reliability challenges and design constraints. (a) Section 4.2 has presented a set of detailed placement mitigation techniques to handle power grid EM, including cell move, single row placement, and single tile placement. Experimental results show that these techniques achieve high-quality placement results in terms of EM violation reduction, wirelength, and placement density. (b) Section 4.3 has demonstrated the ineffectiveness of the traditional design-then-fix flow in advanced nodes. A novel EM hotspot detection and mitigation framework has been proposed based on the information available at the placement stage. A multistage EM mitigation approach has been proposed to address the problematic nets detected by the classification model.

With the above explorations and discussions, this dissertation has demonstrated the effectiveness of advanced modeling and optimization techniques such as machine learning and optimization algorithms. Especially with the rising of deep learning, many conventional and emerging issues in VLSI design can be relieved, enabling further scaling, faster design closure, and more cost reduction. In particular, the following future research directions and open problems are interesting:

- DFM with machine learning. While conventional machine learning models are typically applied in the scope of regression and classification, generative learning has taken one step further in DFM to act as a standalone simulator as in LithoGAN [133] and TEMPO [135] and as an optimizer in GAN-OPC [129] and GAN-SRAF [12]. It is expected that such models will be introduced into different early exploration stages in DFM to achieve orders of magnitude speedup compared to traditional approaches. Besides, there are several practical issues needed to considered: pattern coverage, data preparation cost, and model efficiency. Active learning, adversarial learning, and transfer learning are possible candidates to resolve those issues and revolutionize the DFM flow.

- DFR with machine learning. This dissertation has demonstrated the benefits of bringing machine learning to the EM fixing flow. It enables the incorporation of EM detection and fixing techniques into earlier stages of physical design. Machine learning methods can be potentially applied to address other reliability issues, including aging due to NBTI and HCI, to be identified and automatically fixed concurrently with the design process. In this way, not only are the reliability issues addressed earlier and entirely in the design process, but the trade-offs between timing, power, signal integrity, and reliability are considered simultaneously. Designs that are truly optimized and meet all performance and reliability requirements can be obtained.

It is worthwhile to mention that the manufacturability and reliability problems shall not be considered separately, as they do affect each other, e.g., the interconnect printability and signal EM failures. Future process modeling and design methodologies are expected to involve more cross-layer optimization, where synergistic modeling and optimizations of joint manufacturability and reliability effects will be needed. At the same time, VLSI circuits are likely to be highly optimized according to different application domains for performance, power, and cost. Such domain specifications require CAD tools to learn different design objectives from application domains and perform automatic adjustments to optimization strategies in the design flow.

Bibliography

[1] Addressing signal electromigration (EM) in today's complex digital designs. https://www.eetimes.com/document.asp?docid=1280370.

[2] ANSYS RedHawk. https://www.ansys.com/products/semiconductors/ansys-redhawk.

[3] ITRS. http://www.itrs.net.

[4] OpenCores. https://opencores.org.

[5] Synopsys IC Compiler. http://www.synopsys.com.

[6] TSMC 40nm Technology. http://www.tsmc.com/english/dedicatedFoundry/technology/40nm.htm.

[7] NanGate FreePDK15 Open Cell Library. http://www.nangate.com/?page_id=2328, 2015.

[8] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[9] Konstantinos Adam and Andrew R Neureuther. Domain decomposition methods for the rapid electromagnetic simulation of photomask scattering. *JM3*, 1(3):253–270, 2002.

[10] Viviana Agudelo, Tim Fühner, Andreas Erdmann, and Peter Evanschitzky. Application of artificial neural networks to compact mask models in optical lithography simulation. *JM3*, 13(1):011002, 2013.

[11] Mohamed Baker Alawieh, Yibo Lin, Wei Ye, and David Z Pan. Generative learning in VLSI design for manufacturability: Current status and future directions. *Journal of Microelectronic Manufacturing*, 2, 2019.

[12] Mohamed Baker Alawieh, Yibo Lin, Zaiwei Zhang, Meng Li, Qixing Huang, and David Z. Pan. GAN-SRAF: Sub-resolution assist feature generation using conditional generativeadversarial networks. In *Proc. DAC*, 2019.

[13] Charles J. Alpert, Dinesh P. Mehta, and Sachin S. Sapatnekar. *Handbook of Algorithms for Physical Design Automation.* CRC press, 2008.

[14] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. ComboGAN: Unrestrained scalability for image domain translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 783–790, 2018.

[15] Lucile Arnaud, G Tartavel, T Berger, D Mariolle, Y Gobil, and I Touet. Microstructure and electromigration in copper damascene lines. *Microelectronics Reliability*, 40(1):77–86, 2000.

[16] Christopher M Bishop et al. *Pattern Recognition and Machine Learning.* Springer New York, 2006.

[17] James R Black. Electromigration — a brief survey and some recent results. *IEEE TED*, 16(4):338–347, 1969.

[18] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[19] Ulrich Brenner and Jens Vygen. Faster optimal single-row placement with fixed ordering. In *Proc. DATE*, pages 117–121, 2000.

[20] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[21] Andrew E Caldwell, Andrew B Kahng, and Igor L Markov. Optimal partitioners and end-case placers for standard-cell layout. *IEEE TCAD*, 19(11):1304–1313, 2000.

[22] Stephen Cauley, Venkataramanan Balakrishnan, Y Charlie Hu, and Cheng-Kok Koh. A parallel branch-and-cut approach for detailed placement. *ACM TODAES*, 16(2):18, 2011.

[23] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[24] Xiaodao Chen, Chen Liao, Tongquan Wei, and Shiyan Hu. An interconnect reliability-driven routing technique for electromigration failure

avoidance. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 9(5):770–776, 2012.

[25] Ying Chen, Yibo Lin, Tianyang Gai, Yajuan Su, Yayi Wei, and David Z. Pan. Semi-supervised hotspot detection with self-paced multi-task learning. In *Proc. ASPDAC*, 2019.

[26] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. ICCV*, pages 8789–8797, 2018.

[27] Wing-Kai Chow, Jian Kuang, Xu He, Wenzan Cai, and Evangeline F. Y. Young. Cell density-driven detailed placement with displacement constraint. In *Proc. ISPD*, pages 3–10, 2014.

[28] Yu-An Chung, Hsuan-Tien Lin, and Shao-Wen Yang. Cost-aware pre-training for multiclass cost-sensitive deep learning. In *Proc. IJCAI*, pages 1411–1417, 2016.

[29] Duo Ding, J. Andres Torres, and David Z. Pan. High performance lithography hotspot detection with successively refined pattern identifications and machine learning. *IEEE TCAD*, 30(11):1621–1634, 2011.

[30] Duo Ding, Bei Yu, Joydeep Ghosh, and David Z. Pan. EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation. In *Proc. ASPDAC*, pages 263–270, 2012.

[31] Yi Ding, Peilin Zhao, Steven C. H. Hoi, and Yew-Soon Ong. An adaptive gradient method for online AUC maximization. In *Proc. AAAI*, pages 2568–2574, 2015.

[32] Lori E. Dodd and Margaret S. Pepe. Partial auc estimation and regression. *Biometrics*, 59(3):614–623, 2003.

[33] Dragoljub G. Drmanac, Frank Liu, and Li-C. Wang. Predicting variability in nanoscale lithography processes. In *Proc. DAC*, pages 545–550, 2009.

[34] Charles Elkan. The foundations of cost-sensitive learning. In *Proc. IJCAI*, pages 973–978, 2001.

[35] Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass AUC optimization. In *Proc. ICML*, pages III–906–III–914, 2013.

[36] Irving John Good. Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 107–114, 1952.

[37] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014.

[38] Ronald L Gordon and Chris A Mack. Mask topography simulation for EUV lithography. In *Proc. SPIE*, volume 3676, pages 283–297, 1999.

[39] David Marvin Green and John Arthur Swets. *Signal detection theory and psychophysics.* New York : Wiley, 1966.

[40] Gurobi Optimization Inc. Gurobi optimizer reference manual. `http://www.gurobi.com`, 2014.

[41] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, pages 878–887, 2005.

[42] Kwangsoo Han, Andrew B. Kahng, and Hyein Lee. Scalable detailed placement legalization for complex sub-14nm constraints. In *Proc. ICCAD*, pages 867–873, 2015.

[43] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[44] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

[45] Meng-Kai Hsu, Nitesh Katta, Homer Yen-Hung Lin, Keny Tzu-Hen Lin, King Ho Tam, and Ken Chung-Hsing Wang. Design and manufacturing process co-optimization in nano-technology. In *Proc. ICCAD*, pages 574–581, 2014.

[46] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[47] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, pages 5967–5976, 2017.

[48] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[49] Iris Hui-Ru Jiang, Hua-Yu Chang, and Chih-Long Chang. WiT: optimal wiring topology for electromigration avoidance. *IEEE VLSI*, 20(4):581–592, 2012.

[50] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *ACM Sigkdd Explorations Newsletter*, 6(1):40–49, 2004.

[51] Andrew B Kahng, Siddhartha Nath, and Tajana S Rosing. On potential design impacts of electromigration awareness. In *Proc. ASPDAC*, pages 527–532, 2013.

[52] Andrew B Kahng, Paul Tucker, and Alex Zelikovsky. Optimization of linear placements for wirelength minimization with free sites. In *Proc. ASPDAC*, pages 241–244, 1999.

[53] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature

representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3573–3587, 2018.

[54] Juhwan Kim and Minghui Fan. Hotspot detection on Post-OPC layout using full chip simulation based verification tool: A case study with aerial image simulation. In *Proc. SPIE*, volume 5256, 2003.

[55] Myung-Chul Kim, Jin Hu, and Natarajan Viswanathan. ICCAD-2014 CAD contest in incremental timing-driven placement and benchmark suite. In *Proc. ICCAD*, pages 361–366, 2014.

[56] Myung-Chul Kim, Natarajan Viswanathan, Charles J. Alpert, Igor L. Markov, and Shyam Ramji. MAPLE: multilevel adaptive placement for mixed-size designs. In *Proc. ISPD*, pages 193–200, 2012.

[57] Myung-Chul Kim, Natarajan Viswanathan, Zhuo Li, and Charles Alpert. ICCAD-2013 CAD contest in placement finishing and benchmark suite. In *Proc. ICCAD*, pages 268–270, 2013.

[58] Taiki Kimura, Tetsuaki Matsunawa, Shigeki Nojima, and David Z Pan. Hybrid hotspot detection using regression model and lithography simulation. In *Proc. SPIE*, volume 9781, 2016.

[59] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[60] Jon Kleinberg and Eva Tardos. *Algorithm design.* Pearson Education India, 2006.

[61] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proc. ICML*, pages 179–186, 1997.

[62] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. CVPR*, pages 4681–4690, 2017.

[63] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *Proc. ICLR*, 2018.

[64] Harry J Levinson. *Principles of lithography*, volume 146. SPIE press, 2005.

[65] Baozhen Li, Paul Muller, James Warnock, Leon Sigal, and Dinesh Badami. A case study of electromigration reliability: From design point to system operations. In *Proc. IRPS*, pages 2D.1.1–2D.1.6, 2015.

[66] Shuai Li and Cheng-Kok Koh. Mixed integer programming models for detailed placement. In *Proc. ISPD*, pages 87–94, 2012.

[67] Jens Lienig. Introduction to electromigration-aware physical design. In *Proc. ISPD*, pages 39–46, 2006.

[68] Jens Lienig. Electromigration and its impact on physical design in future technologies. In *Proc. ISPD*, pages 33–40, 2013.

[69] B-K Liew, Nathan W Cheung, and Chenming Hu. Projecting interconnect electromigration lifetime for arbitrary current waveforms. *IEEE TED*, 37(5):1343–1351, 1990.

[70] Sheng-Yuan Lin, Jing-Yi Chen, Jin-Cheng Li, Wan-Yu Wen, and Shih-Chieh Chang. A novel fuzzy matching model for lithography hotspot detection. In *Proc. DAC*, pages 68:1–68:6, 2013.

[71] Yibo Lin, Mohamed Baker Alawieh, Wei Ye, and David Z Pan. Machine learning for yield learning and optimization. In *Proc. ITC*, pages 1–10, 2018.

[72] Yibo Lin, Meng Li, Yuki Watanabe, Taiki Kimura, Tetsuaki Matsunawa, Shigeki Nojima, and David Z Pan. Data efficient lithography modeling with transfer learning and active data selection. *IEEE TCAD*, 2018.

[73] Yibo Lin, Bei Yu, Xiaoqing Xu, Jhih-Rong Gao, Natarajan Viswanathan, Wen-Hao Liu, Zhuo Li, Charles J. Alpert, and David Z. Pan. MrDP: Multiple-row detailed placement of heterogeneous-sized cells for advanced nodes. In *Proc. ICCAD*, 2016.

[74] Yibo Lin, Bei Yu, Yi Zou, Zhuo Li, Charles J. Alpert, and David Z. Pan. Stitch aware detailed placement for multiple e-beam lithography. In *Proc. ASPDAC*, pages 186–191, 2016.

[75] Peng Liu, Yu Cao, Luoqi Chen, Guangqing Chen, Mu Feng, Jiong Jiang, Hua-yu Liu, Sungsoo Suh, Sung-Woo Lee, and Sukjoo Lee. Fast and ac-

curate 3D mask model for full-chip OPC and verification. In *Proc. SPIE*, volume 6520, 2007.

[76] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, pages 3431–3440, 2015.

[77] Kevin D Lucas, Hiroyoshi Tanabe, and Andrzej J Strojwas. Efficient and rigorous three-dimensional model for optical lithography simulation. *Journal of the Optical Society of America A*, 13(11):2187–2199, 1996.

[78] Xu Ma and Gonzalo R Arce. *Computational lithography*, volume 77. John Wiley & Sons, 2011.

[79] Xu Ma, Xuejiao Zhao, Zhiqiang Wang, Yanqiu Li, Shengjie Zhao, and Lu Zhang. Fast lithography aerial image calculation method based on machine learning. *Applied Optics*, 56(23):6485–6495, 2017.

[80] Chris Mack. *Fundamental Principles of Optical Lithography: The Science of Microfabrication*. John Wiley & Sons, 2008.

[81] Chris A Mack. Understanding focus effects in submicrometer optical lithography: a review. *Optical Engineering*, 32(10):2350–2363, 1993.

[82] Chris A Mack. Thirty years of lithography simulation. In *Proc. SPIE*, volume 5754, pages 1–13, 2004.

[83] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Statist.*, 18(1):50–60, 1947.

[84] Tetsuaki Matsunawa, Jhih-Rong Gao, Bei Yu, and David Z. Pan. A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction. In *Proc. SPIE*, volume 9427, 2015.

[85] Tetsuaki Matsunawa, Shigeki Nojima, and Toshiya Kotani. Automatic layout feature extraction for lithography hotspot detection based on deep neural network. In *SPIE Advanced Lithography*, volume 9781, 2016.

[86] Donna Katzman McClish. Analyzing a portion of the roc curve. *Medical Decision Making*, 9(3):190–195, 1989.

[87] Mentor Graphics. Calibre verification user's manual, 2008.

[88] Akiko Mimotogi, Masamitsu Itoh, Shoji Mimotogi, Kazuya Sato, Takashi Sato, and Satoshi Tanaka. Mask topography effects of hole patterns on hyper-na lithography. In *Proc. SPIE*, volume 6607, 2007.

[89] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[90] MG Moharam, Drew A Pommet, Eric B Grann, and TK Gaylord. Stable implementation of the rigorous coupled-wave analysis for surface-relief gratings: enhanced transmittance matrix approach. *Journal of the Optical Society of America A*, 12(5):1077–1086, 1995.

[91] Gordon E Moore. Lithography and the future of moore's law. In *Integrated Circuit Metrology, Inspection, and Process Control IX*, volume 2439, pages 2–17. International Society for Optics and Photonics, 1995.

[92] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[93] Jiwoo Pak, Bei Yu, and David Z. Pan. Electromigration-aware redundant via insertion. In *Proc. ASPDAC*, pages 544–549, 2015.

[94] Min Pan, Natarajan Viswanathan, and Chris Chu. An efficient and effective detailed placement algorithm. In *Proc. ICCAD*, pages 48–55, 2005.

[95] J. W. Park, A. Torres, and X. Song. Litho-aware machine learning for hotspot detection. *IEEE TCAD*, 37(7):1510–1514, 2018.

[96] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct.):2825–2830, 2011.

[97] Sergiy Popovych, Hung-Hao Lai, Chieh-Min Wang, Yih-Lang Li, Wen-Hao Liu, and Ting-Chi Wang. Density-aware detailed placement with instant legalization. In *Proc. DAC*, pages 122:1–122:6, 2014.

[98] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[99] John Randall, Kurt G Ronse, Thomas Marschner, Anne-Marie Goethals, and Monique Ercken. Variable-threshold resist models for lithography simulation. In *Optical Microlithography XII*, volume 3679, pages 176–182. International Society for Optics and Photonics, 1999.

[100] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. MIT press, 2006.

[101] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[102] Ed Roseboom, Mark Rossman, Fang-Cheng Chang, and Philippe Hurat. Automated full-chip hotspot detection and removal flow for interconnect layers of cell-based designs. In *Proc. SPIE*, volume 6521, 2007.

[103] Cynthia Rudin and Robert E Schapire. Margin-based ranking and an equivalence between adaboost and rankboost. *Journal of Machine Learning Research*, 10(Oct):2193–2232, 2009.

[104] Johannes Ruoff. Impact of mask topography and multilayer stack on high NA imaging of EUV masks. In *Photomask Technology 2010*, volume 7823, page 78231N, 2010.

[105] Youngsoo Shin Seongbo Shim, Suhyeong Choi. Machine learning-based 3d resist model. In *Proc. SPIE*, volume 10147, 2017.

[106] Seong-bo Shim, Young-chang Kim, Suk-joo Lee, Seong-woon Choi, and Woo-sung Han. Study of the mask topography effect on the OPC modeling of hole patterns. In *Proc. SPIE*, volume 6924, 2008.

[107] Moojoon Shin and Jee-Hyong Lee. Accurate lithography hotspot detection using deep convolutional neural networks. *JM3*, 15(4):043507, 2016.

[108] Bruce W Smith and Kazuaki Suzuki. *Microlithography: science and technology*. CRC press, 2018.

[109] Harald Steck. Hinge rank loss and the area under the ROC curve. In *Proc. ECML*, pages 347–358. Springer Berlin Heidelberg, 2007.

[110] John Arthur Swets and Ronald M. Pickett. *Evaluation of diagnostic systems: methods from signal detection theory*. New York : Academic Press, 1982.

[111] Synopsys. Sentaurus Lithography. `https://www.synopsys.com/silicon/mask-synthesis/sentaurus-lithography.html`, 2016.

[112] Allen Taflove and Susan C Hagness. *Computational electrodynamics: the finite-difference time-domain method*. Artech house, 2005.

[113] Taraneh Taghavi, Charles Alpert, Andrew Huber, Zhuo Li, Gi-Joon Nam, and Shyam Ramji. New placement prediction and mitigation techniques for local routing congestion. In *Proc. ICCAD*, pages 621–624, 2010.

[114] Jaione Tirapu-Azpiroz, Paul Burchard, and Eli Yablonovitch. Boundary layer model to account for thick mask effects in photolithography. In *Proc. SPIE*, volume 5040, 2003.

[115] Yoichi Tomioka, Tetsuaki Matsunawa, Chikaaki Kodama, and Shigeki Nojima. Lithography hotspot detection by two-stage cascade classifier using histogram of oriented light propagation. In *Proc. ASPDAC*, pages 81–86, 2017.

[116] Andres J. Torres. ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite. In *Proc. ICCAD*, 2012.

[117] Jens Vygen. Algorithms for detailed placement of standard cells. In *Proc. DATE*, pages 321–324, 1998.

[118] Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J Kennedy. Training deep neural networks on imbalanced data sets. In *Proc. IJCNN*, pages 4368–4374, 2016.

[119] Yuki Watanabe, Taiki Kimura, Tetsuaki Matsunawa, and Shigeki Nojima. Accurate lithography simulation model based on convolutional

neural networks. In *SPIE Advanced Lithography*, pages 101470K–101470K. International Society for Optics and Photonics, 2017.

[120] Gary M Weiss. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19, 2004.

[121] Wan-Yu Wen, Jin-Cheng Li, Sheng-Yuan Lin, Jing-Yi Chen, and Shih-Chieh Chang. A fuzzy-matching model with grid reduction for lithography hotspot detection. *IEEE TCAD*, 33(11):1671–1680, 2014.

[122] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[123] Alfred K Wong and Andrew R Neureuther. Mask topography effects in projection printing of phase-shifting masks. *IEEE TED*, 41(6):895–902, 1994.

[124] Alfred Kwok-Kit Wong. *Resolution Enhancement Techniques in Optical Lithography*, volume 47. SPIE press, 2001.

[125] Shaomin Wu, Peter Flach, and Cèsar Ferri. An improved model selection heuristic for AUC. In *Proc. ECML*, pages 478–489, 2007.

[126] Jingyu Xu, Subarna Sinha, and Charles C. Chiang. Accurate detection for process-hotspots with vias and incomplete specification. In *Proc. ICCAD*, pages 839–846, 2007.

[127] Xiaoqing Xu, Tetsuaki Matsunawa, Shigeki Nojima, Chikaaki Kodama, Toshiya Kotani, and David Z. Pan. A machine learning based framework for sub-resolution assist feature generation. In *Proc. ISPD*, pages 161–168, 2016.

[128] Lian Yan, Robert H Dodier, Michael Mozer, and Richard H Wolniewicz. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proc. ICML*, pages 848–855, 2003.

[129] Haoyu Yang, Shuhe Li, Yuzhe Ma, Bei Yu, and Evangeline F.Y. Young. GAN-OPC: Mask optimization with lithography-guided generative adversarial nets. In *Proc. DAC*, pages 131:1–131:6, 2018.

[130] Haoyu Yang, Luyang Luo, Jing Su, Chenxi Lin, and Bei Yu. Imbalance aware lithography hotspot detection: a deep learning approach. *JM3*, 16(3):033504, 2017.

[131] Haoyu Yang, Jing Su, Yi Zou, Bei Yu, and Evangeline F. Y. Young. Layout hotspot detection with feature tensor generation and deep biased learning. In *Proc. DAC*, pages 62:1–62:6, 2017.

[132] Wei Ye, Mohamed Baker Alawieh, Meng Li, Yibo Lin, and David Z Pan. Litho-GPA: Gaussian process assurance for lithography hotspot detection. In *Proc. DATE*, pages 54–59, 2019.

[133] Wei Ye, Mohamed Baker Alawieh, Yibo Lin, and David Z Pan. LithoGAN: End-to-end lithography modeling with generative adversarial networks.

In *Proc. DAC*, page 107, 2019.

[134] Wei Ye, Mohamed Baker Alawieh, Yibo Lin, and David Z. Pan. Tackling signal electromigration with learning-based detection and multistage mitigation. In *Proc. ASPDAC*, pages 167–172, 2019.

[135] Wei Ye, Mohamed Baker Alawieh, Yuki Watanabe, Shigeki Nojima, Yibo Lin, and David Z Pan. TEMPO: Fast mask topography effect modeling with deep learning. In *Proc. ISPD*, page 127–134, 2020.

[136] Wei Ye, Meng Li, Kai Zhong, Bei Yu, and David Z Pan. Power grid reduction by sparse convex optimization. In *Proc. ISPD*, pages 60–67, 2018.

[137] Wei Ye, Yibo Lin, Meng Li, Qiang Liu, and David Z. Pan. LithoROC: Lithography hotspot detection with explicit ROC optimization. In *Proc. ASPDAC*, pages 292–298, 2019.

[138] Wei Ye, Yibo Lin, Xiaoqing Xu, Wuxi Li, Yiwei Fu, Yongsheng Sun, Canhui Zhan, and David Z. Pan. Placement mitigation techniques for power grid electromigration. In *Proc. ISLPED*, 2017.

[139] Bei Yu, Xiaoqing Xu, Jhih-Rong Gao, Yibo Lin, Zhuo Li, Charles Alpert, and David Z. Pan. Methodology for standard cell compliance and detailed placement for triple patterning lithography. *IEEE TCAD*, 34(5):726–739, May 2015.

[140] Yen-Ting Yu, Ya-Chung Chan, Subarna Sinha, Iris Hui-Ru Jiang, and Charles Chiang. Accurate process-hotspot detection using critical design rule extraction. In *Proc. DAC*, pages 1167–1172, 2012.

[141] Yen-Ting Yu, Geng-He Lin, Iris Hui-Ru Jiang, and Charles Chiang. Machine-learning-based hotspot detection using topological classification and critical feature extraction. *IEEE TCAD*, 34(3):460–470, 2015.

[142] Hang Zhang, Bei Yu, and Evangeline F. Y. Young. Enabling online learning in lithography hotspot detection with information-theoretic feature optimization. In *Proc. ICCAD*, pages 47:1–47:8, 2016.

[143] Hang Zhang, Fengyuan Zhu, Haocheng Li, Evangeline F. Y. Young, and Bei Yu. Bilinear lithography hotspot detection. In *Proc. ISPD*, pages 7–14, 2017.

[144] Peilin Zhao, Steven C. H. Hoi, Rong Jin, and Tianbao Yang. Online AUC maximization. In *Proc. ICML*, pages 233–240, 2011.

[145] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV*, 2017.

Vita

Wei Ye received the B.S. degree in microelectronics from Zhejiang University, Hangzhou, China, in 2015, and the M.S.E. degree in engineering from the University of Texas at Austin, Texas, U.S., in 2017. She started her Ph.D. program at the University of Texas at Austin in 2015, with research advisor David Z. Pan. She has interned at Synopsys, Sunnyvale in 2016 summer, Cadence, Austin in 2018 summer, and Kioxia Corporation (Toshiba Memory Corporation), Japan in 2019 summer.

Wei Ye's research interests include physical design and machine learning applications in VLSI CAD.

Permanent address: yeweizju@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.