A Fast Machine Learning-based Mask Printability Predictor for OPC Acceleration

Bentian Jiang
The Chinese University of Hong Kong
Hong Kong SAR
btjiang@cse.cuhk.edu.hk

Hang Zhang
Cornell Univeristy
Ithaca, NY, USA
hz459@cornell.edu

Jinglei Yang
University of California, Santa Barbara
Santa Barbara, CA, USA
jingleiyang@ucsb.edu

Evangeline F. Y. Young
The Chinese University of Hong Kong
Hong Kong SAR
fyyoung@cse.cuhk.edu.hk

ABSTRACT

Continuous shrinking of VLSI technology nodes brings us powerful chips with lower power consumption, but it also introduces many issues in manufacturability. Lithography simulation process for new feature size suffers from large computational overhead. As a result, conventional mask optimization process has been drastically resource consuming in terms of both time and cost. In this paper, we propose a high performance machine learning-based mask printability evaluation framework for lithography-related applications, and apply it in a conventional mask optimization tool to verify its effectiveness.

CCS CONCEPTS

• **Hardware → Design for manufacturability**; **Yield and cost modeling**; **Yield and cost optimization**;

KEYWORDS

Design for Manufacturability, Optical Proximity Correction Acceleration, Machine Learning

ACM Reference Format:
Bentian Jiang, Hang Zhang, Jinglei Yang, and Evangeline F. Y. Young. 2019. A Fast Machine Learning-based Mask Printability Predictor for OPC Acceleration. In *ASPDAC '19: 24th Asia and South Pacific Design Automation Conference (ASPDAC '19), January 21–24, 2019, Tokyo, Japan,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3287624.3287682

1 INTRODUCTION

Various modern designs for manufacturability (DFM) technologies aim at tackling the issues caused by continuous shrinking of technology nodes, while the lithography simulation step plays an essential role in physical verification of those technologies. Among all these issues, mask optimization problem becomes extremely critical due to the mismatch between lithography system and smaller feature size.

Optical proximity correction (OPC) is a major resolution enhancement technique (RET) in mask optimization which can significantly improve the mask printability. Conventional OPC approaches include forward model-based OPC [1, 8, 14] and inverse lithography-based OPC (ILT) [6, 13]. Forward model-based OPC usually relies on sub-resolution assist feature (SRAF) insertion [15], edge fragmentation and movement, where the mask is adjusted iteratively based on some mathematical models. Meanwhile the ILT-based OPC flows [6, 13] perform the mask optimization process through optimizing an objective function with certain constrains due to the lithography imaging system. The main issue actually comes from the lithography simulation process. Conventional litho-simulation suffers from large computational overhead especially for advanced technology nodes. More importantly, both model-based and ILT-based methods take the wafer image as a mask update criterion in each optimization iteration, which in other words, multiple rounds of lithography simulation are indispensable in the OPC flow and that makes the whole process drastically time consuming. Thus, it is imperative to derive a fast and accurate lithography model for mask quality evaluation.

1.1 Previous Work

In recent years, machine learning has drawn more attention because of its broad applications in DFM and other areas. The superiority of learning-based solutions has been verified in various lithography-related areas including hotspot detection [10, 19, 21], OPC [11, 18], SRAF insertion [16, 17] and lithography system modeling [9], etc. In hotspot detection, paper [21] proposed a bilinear classification model to tackle matrix form layout feature which can preserve the hidden structural correlations in the lithography process. In lithography system modeling, because a set of manufactured data for a specific lithography configuration is only valid for the training of one single model, paper [9] proposed a new resit modeling framework based on ResNet and transfer learning to improve the data efficiency. As for the OPC area, paper [18] developed an OPC-oriented generative adversarial networks architecture (GAN-OPC) based on the mask solution produced by ILT [6]. GAN-OPC designs a customized flow for model construction, which achieves a reasonably good performance on the mask optimization task. However,

ASPDAC '19, January 21–24, 2019, Tokyo, Japan
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6007-4/19/01…$15.00
https://doi.org/10.1145/3287624.3287682

GAN-OPC still has its own deficiencies. First, its training step is extremely time consuming, which is measured by days, e.g. PGAN-OPC takes more than 28000 seconds to finish training. Clearly, as feature sizes are continuously shrinking, the explosion in training time can significantly affect the scalability of its industrial applications. Second, since GAN-OPC designs a completely new flow which differs from the original ILT [6] flow, it may lead to unexpected non-convergence on new masks. Last, although GAN-OPC facilitates the process by 2X compared with [6], it still needs nearly 400 seconds to perform optimization on a 2x2 um^2 test case.

In general, this work is mainly motivated by 2 issues: (1) the large computational overhead of conventional lithography simulation dominates the runtime of conventional OPC process; (2) previous learning-based OPC works usually have low scalability for new technology node due to the large time-overhead in training step.

1.2 Our Contributions

In this paper, we construct a fast machine learning (ML) based mask printability prediction (MPP) framework for lithography-related applications. Different from previous learning-based works that are designed for specific manufacturing problem like hotspot detection or OPC, we focus on developing an independent mask printability evaluation framework that can be used to improve the scalability for different lithography-related applications. In this work, we used this MPP framework to perform OPC acceleration which verified the effectiveness and flexibility of our proposed framework. The major contributions of this paper are summarized as follows:

- We propose a set of fast and accurate ML-based mask printability prediction models based on machine learning techniques, and the training steps of our models require only 5.48% and 0.44% of previous work's [18] training time.
- We develop with CUDA a matrix-based concentric circle sampling (MCCS) method for feature extraction, which reduces 91.99% runtime comparing to conventional MCCS.
- We propose a novel second order circle subset selection algorithm for feature selection, which improves the prediction accuracy for model by 2.38% and reduces 9.4% false alarms.
- We develop a machine learning-based OPC acceleration framework, which achieves 2.6X-11X runtime speedup while ensuring a comparable printability comparing with previous works [6, 8, 18].

The rest of this paper will be organized as follows. In section 2, we will discuss the preliminaries and formulate the problems, and our proposed algorithm will be presented in section 3. Section 4 presents experimental results, followed by conclusion in section 5.

2 PRELIMINARIES

In this section, we will introduce the background of mask optimization, and then define several useful terms to describe the performance of our prediction model and the OPC acceleration framework. Meanwhile, we will give the problem formulations.

2.1 Lithography Simulation Model

The lithography simulation process of generating the printed image from a given mask M can be modeled in two phases, the optical model and the resist model, and the corresponding output images are I and R, respectively. The optical model produces an aerial image that corresponds to the light intensity distribution on the wafer plane. Theoretically, the Hopkins diffraction model [7] for partially coherent imaging system is applied to analyze the optical model behavior. In practice, a singular value decomposition (SVD) technique [4] is adopted to approximate the Hopkins model, followed by a convoluation with kernels in frequency domain:

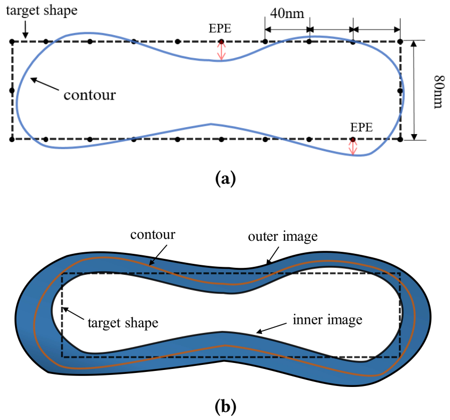


Figure 1: (a) Illustration for EPE Measurement [8]; (b) Illustration for PVBand Measurement [8].

$$I(x,y) = F(M(x,y)) = \sum_{k=1}^{N^2} \omega_k |M(x,y) \otimes h_k(x,y)|^2, \qquad (1)$$

where h_k is the kth kernel in frequency domain and ω_k is the corresponding weight. As mentioned in [6], the N_h^{th} order approximation to the system is used in practice and equation (1) becomes,

$$I(x,y) \approx \sum_{k=1}^{N_h} \omega_k |M(x,y) \otimes h_k(x,y)|^2. \qquad (2)$$

The resist model is then applied to the aerial image produced by the optical model, by setting a constant light intensity threshold I_{th}, the final wafer image R can be obtained by the photo resist model through the following step function,

$$R(x,y) = \begin{cases} 1, & \text{if } I(x,y) \geq I_{th}, \\ 0, & \text{if } I(x,y) < I_{th}. \end{cases} \qquad (3)$$

We pick $N_h = 24$ and $I_{th} = 0.225$ to keep consistence with [2].

2.2 Mask Optimization

Given an input target layout with NxN pixels, current mask optimization process focuses on obtaining a mask solution with NxN pixels such that the wafer image of the mask has the best quality and printability. The mask quality is evaluated by the fidelity of its wafer image with respect to the input target layout. The primary objective of mask optimization is to minimize the number of Edge Placement Error violations, which reflects the fidelity of the printed pattern, and to minimize the Process Variation Band which represents the sensitivity of the optimized mask to process variation.

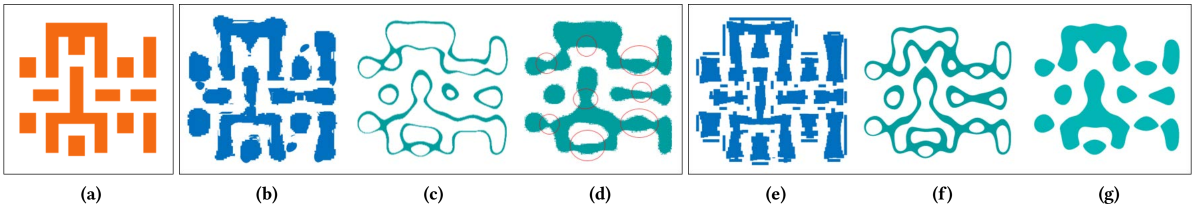


Figure 2: Example of misleading PVBand. (a) target mask; (b) mask solution of PGAN-OPC [18], which tends to produce smaller PVBand; (c) PVBand by mask of PGAN-OPC [18], which is only 94498nm^2; (d) wafer image by mask of [18], whose squared L_2 error is 83663nm, and hotspots are marked in red circle; (e) mask solution of [8], which tends to produce larger PVBand; (f) PVBand by mask of [8], which is 146776nm^2; (g) wafer image by mask of [8], whose squared L_2 error is 79255nm.

2.2.1 *Edge Placement Error (EPE).* EPE is the horizontal and vertical geometric displacement of the image contour from the corresponding edge of the target layout polygon (Fig. 1(a)). A given checkpoint on the polygon edge will be marked as an EPE violation if its EPE exceeds a given displacement threshold.

2.2.2 *Process Variation Band (PVBand).* PVBand measures the area (nm^2) of the XOR region between the lithography contours obtained under two extreme simulation conditions, one at nominal focus and +2% dose, while the other one at defocus and -2% dose[8]. An example of PVBand is shown in Fig. 1(b) (shady area).

2.2.3 *Optical Proximity Correction (OPC) Flow [8].* Optical proximity correction (OPC) is a major resolution enhancement technique, in which the edges of the features are moved to compensate for the distortions due to lithography. Fig. 3(a) shows the flow of a conventional model-based OPC tool [8]. With the input layout polygons, the OPC tool will first segment the edges of the polygons into fragments. Based on the segmentation, sub-resolution assist features (SRAFs) will be inserted to improve the printability. Next, the intensity difference optimization stage and the edge placement error minimization stage will be performed iteratively to obtain the best mask solution with the minimum EPE violations and PVBand.

2.3 Evaluation Metrics

In this paper, we will propose a machine learning-based (ML-based) EPE prediction model and an ML-based intensity prediction model to achieve a fast lithography-guided mask quality evaluation (Fig 3(b)), and these two models will be applied in the EPE minimization stage of the OPC flow (Fig 3(a)) to verify their effectiveness. Given a set of checkpoints on the layout polygons, the EPE prediction model will directly predict the EPE violations from the extracted features on the corresponding checkpoints, while the intensity prediction model will directly predict their intensity values after lithography.

We define several terminologies to quantify the performance of our proposed learning model as follows:

Definition 1 (Accuracy). *The ratio of correctly predicted EPE violations in the set of all actual EPE violations.*

Definition 2 (False Alarm). *The number of incorrectly predicted EPE violations.*

In order to get a good mask solution in our OPC acceleration framework, we need to achieve high accuracy, low false alarms.

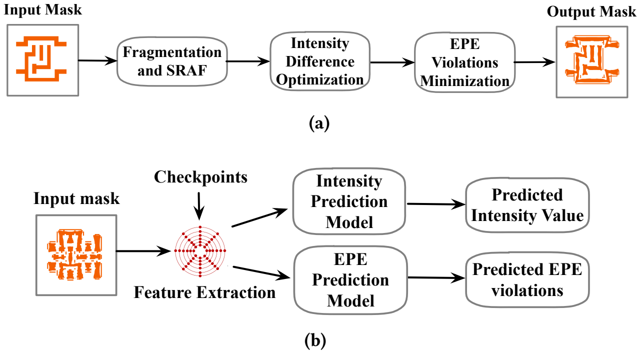


Figure 3: (a) General flow of state-of-the-art OPC tool [8]; (b) Flow of fast lithography-guided mask quality evaluation.

EPE is a commonly used metric in layout printability estimation flow. However, EPE may not be general enough because it only measures the edge displacements at some specific checkpoints. Different choices of checkpoints may result in different EPE violation counts. Considering the objective of mask optimization which is to ensure a better mask printability, it is more suitable to evaluate them base on the squared L_2 error [18]. In general, squared L_2 error is a more comprehensive representation of mask printability, since it can be regarded as the sum of the edge displacement on every edge point.

Definition 3 (Squared $\mathbf{L_2}$ Error). *Let* $\mathbf{M}$ *be the target mask image and* $\mathbf{R}$ *be the wafer image, the squared* $\mathbf{L_2}$ *error is* $||\mathbf{M}-\mathbf{R}||_2^2$.

Meanwhile, PVBand is also considered for manufacturability, since process variation could cause some particular metric to fall below or rise above a specification, and then reduces the overall yield. However, in certain circumstances, PVBand may be misleading, and a smaller PVBand does not always indicate a better mask printability. A clear example for such situation is shown in Fig. 2. The PVBand of the mask in Fig. 2(b) is only 94498nm^2 (Fig. 2(c)), while the PVBand of the mask in Fig. 2(e) is 146776nm^2 (Fig. 2(f)). Although Fig. 2(c) shows smaller PVBand, but its corresponding image (Fig. 2(d)) has much lower fidelity, as we can see many unexpected hotspots (red circles) in the image. This is because PVBand tells the difference of the image in two extreme conditions, but it does not tell directly the quality of the image. There can be hotspots in both extreme conditions, but they got cancelled out by taking the difference. In order to obtain a better criterion on process variation, we normalize the PVBand by its mask image area.

Definition 4 (Normalized PVBand). *The normalized PVBand (NPVBand) is defined as* $\frac{PVBand}{Area}$.

2.4 Problem Formulations

Given a dataset containing information of mask images, wafer images and corresponding EPE labels on checkpoints, we define the following problems.

Problem 1 (ML-based EPE Modeling). *Construct an ML-based EPE detection model that can maximize the accuracy for EPE violation prediction.*

Problem 2 (ML-based Intensity Modeling). *Construct an ML-based intensity model that can minimize the root-mean-square error between the actual intensity value and the predicted intensity value.*

Problem 3 (Optical Proximity Correction). *Given a target mask with NxN pixels, generate a mask solution that has minimized L_2 error and normalized PVBand.*

3 ALGORITHMS

3.1 Overview

As introduced in section 2.2.3 and depicted in Fig. 3(a), the state-of-the-art model-based OPC tool [8] can be generally divided into 3 stages, fragmentation and SRAFs insertion stage, intensity difference optimization stage and edge displacement minimization stage. The last stage uses up more than 80% of the total processing time on average. In this work, we developed a novel machine learning-based (ML-based) OPC acceleration framework (section 3.6), which focuses on accelerating stage 3 of this model-based OPC tool. To build up this framework, we also construct an ML-based EPE prediction model (section 3.5.1) and an ML-based intensity prediction model (section 3.5.2), with an efficient feature extractor (section 3.2) and a powerful feature selection method (section 3.3). Besides OPC, we can also apply our proposed framework in other lithography-related applications like hotspots detection and SRAFs insertion.

3.2 Matrix-based Concentric Circle Sampling

Layout feature extraction plays an important role in keeping the geometric information of layout patterns. Ideally, our extraction method should capture the physical information of light propagation, diffraction, and interference induced by the lithography process. To address this purpose, paper [21] introduced the matrix based concentric circle sampling (MCCS) method. As shown in fig 4(a), for each sample clip, MCCS will extract sampled pixel values on a set of concentric circles, and then concatenates sampling point values within one circle, putting them into a vector forming one row of the feature matrix (fig 4(b)). Let l be the length of the sample clip, n_p denote the number of sample points on each sample circle and r_{in} control the sampling density. In MCCS, circles are sampled up to radius r_{in} in increments of inc_1 and further up to radius $\frac{l}{2}$ using increments of inc_2. For example, under the condition that $l = 600nm$, $r_{in} = 60nm$, $inc_1 = 5$, $inc_2 = 10$, and $n_p = 16$, the dimension of a feature matrix is 37×16 ($37 = 1 + \frac{60}{5} + \frac{300-60}{10}$).

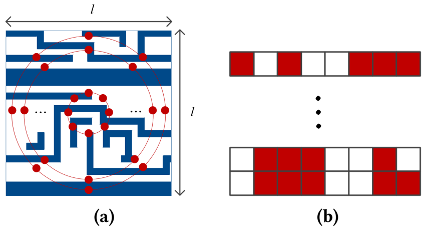


Figure 4: Visualization of MCCS extraction [21]

3.3 Second Order Circle Subset Selection via Mutual Information

The maximal circular mutual information (MCMI) scheme proposed in paper [20] modeled the first-order dependence between circles and their labels, which achieves reasonably good performance on lithography feature selection. However, these circles are mutually dependent because of lithography interference; therefore, it is expected to achieve better performance if we could model the dependence considering full lithography interference. One intuitive way in modeling is that we measure the dependency of all combinations of m features selected from n features, where n is the total number of features as shown below:

$$\mathbf{x}_m \in \mathbf{x}_n, \tag{4}$$

where $\mathbf{x}_n = \{x^i, i = 1, 2, ..., n\}$ is a set of potential circle candidates, and $\mathbf{x}_m$ is the circle subset we would like to select. The problem for the above approach is that there are $\frac{n!}{m!(n-m)!}$ combinations to measure, which is too large to perform in real-world applications. Therefore, it is imperative to develop a model that can approximate the selection procedure with low computational cost. Here we propose our second order maximal circular mutual information scheme (SO-MCMI) to resolve this issue.

We first densely sample n_c circles from each sample clip, and concatenate them into a feature matrix as shown in Fig. 4(b), where each row of the feature matrix is a binary sequence representing one specific sample circle. The dimensions of this feature matrix are n_c x n_p, where n_c is the number of sample circles. We then encode each circle c_i to a more compact representation by converting it into a decimal number,

$$c_i = \sum_{j=0}^{n_p} p_{i,j} \cdot 2^j, \tag{5}$$

where $p_{i,j} \in \{0, 1\}, \forall i, j$ is the j^{th} sample point on the i^{th} sample circle. Note that the primal objective of our feature selection algorithm is to maximize the dependency of selected circle subset with the target variable y. We here adopt mutual information [12], which is proved to be efficient in lithography hotspot detection [20], to define the dependency between circles and target variables. Let the i^{th} and j^{th} circles C_i and C_j be random variables defined in a circle space C and $Y \in \{-1, 1\}$ be a random variable, the mutual information $I(C_i, C_j; Y)$ is defined as:

$$I(C_i, C_j; Y) = \sum_{c_i \in C_i} \sum_{c_j \in C_j} \sum_{y \in Y} p(c_i, c_j, y)\, log \frac{p(c_i, c_j, y)}{p(c_i, c_j)\, p(y)}, \quad (6)$$

where $p(c_i, c_j, y)$ is the joint probability distribution function of random variables C_i, C_j and Y, $p(c_i, c_j)$ and $p(y)$ are the marginal probability distribution functions of $\{C_i, C_j\}$, and Y respectively. We can then describe our goal of SO-MCMI as follows:

$$\mathfrak{I}_{n_c^*} = \arg \max_{\mathfrak{I}_{n_c^*} \subseteq \mathfrak{I}} \sum_{i \in \mathfrak{I}_{n_c^*}} \sum_{j \in \mathfrak{I}_{n_c^*}} I(C_i, C_j; Y), \quad (7)$$

where $\mathfrak{I}_{n_c^*}$ is selected circle indices, $\mathfrak{I} = \{i | 1 \leq i \leq n_c\}$ is the index set of potential circle position candidates, n_c^* is the desired number of circles. Thus, the SO-MCMI is formulated as:

$$\begin{aligned} \max \quad & \mathbf{w}^\top \mathbf{M} \mathbf{w}, \\ \text{s.t.} \quad & \sum_{i=1}^{n_c} w_i = n_c^*, \;\; w_i \in \{0, 1\}, \;\; \forall i. \end{aligned} \quad (8)$$

Variable w is a n_c-dimensional vector, where w_i indicates whether the i^{th} circle is selected, and variable M is a $n_c \times n_c$ matrix, where $M_{i,j}$ represents the mutual information value computed by $I(C_i, C_j; Y)$. Eqn.8 is a mixed-integer quadratic programming (MIQP) problem which can be solved by off-the-shell MIQP solver.

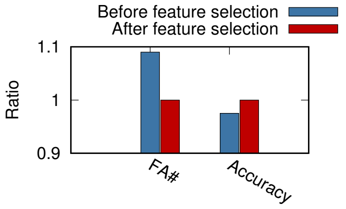
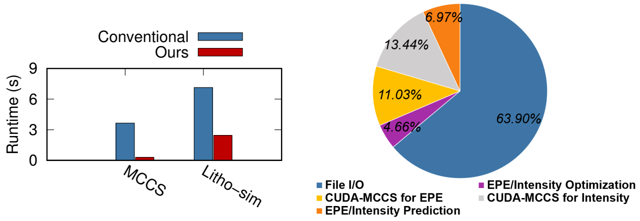


Figure 5: Our feature selection method reduces 9.4% false alarms and improve 2.38% accuracy.

In our implementation, we will densely sample 100 circles for each sample clip, and then perform SO-MCMI to select 37 circles. Experiment result shows that proposed scheme contributes a significant accuracy improvement in EPE modeling (Table 1, Fig. 5).

3.4 CUDA Speedup

Although MCCS shows its great superiority in hotspot detection task comparing to former works, current implementation of MCCS still suffers from long runtime due to the large number of wafer images to be trained on. Clearly, CPU does not have high enough efficiency on image sampling even with multi-threads implementation. Unfortunately, this runtime issue will seriously affect the flexibility and performance of our proposed acceleration framework. As an enhancement version of MCCS [21], we implement it with **CUDA** (C/C++), and conduct our CUDA-based MCCS extractor on GPU to acquire a satisfactory runtime performance. By utilizing CUDA and GPU, we achieve an amazing speedup on the feature extraction step, which plays an essential role on the real-time feature extraction step of our acceleration framework (Fig. 6(b)). To sample a 100 x 16 x n_{cp} feature matrix (n_{cp} is the number of checkpoints) from a 2048 x 2048 input mask, the average extraction time shrinks from **3.652s** to **0.2925s** (Fig. 6(a)).



(a) (b)

Figure 6: (a) MCCS: CUDA-based MCCS reduces 91.99% runtime comparing to conventional CPU-based MCCS; Litho-sim: fast mask printability evaluation scheme reduces 65.8% runtime comparing to conventional litho-simulation. (b) Runtime distribution in stage 3 of acceleration framework.

3.5 Learning Model

In order to capture the hidden structural information induced by the lithography process, we should select a suitable learning model to match the properties of the dataset. Our training dataset is large in scale and with high dimensionality, consisting of more than 30,000 sample clip instances and each instance is a 37×16 matrix. Moreover, the sparse distribution of our sample points will introduce many zero entries in the feature matrices, and the repetitions of similar regular polygons will bring local similarity into the data. Sparsity will induce noise in the feature representation, while local similarity may cause over-fitting. More importantly, large-scale and high-dimensionality increase the complexity of model training.

To overcome these potential impacts on model performance, we apply eXtreme gradient boosting (XGBoost) [3] to train our models. XGBoost is a highly scalable end-to-end tree boosting system which naturally possesses a reasonable linear/non-linear fitting ability. An additional regularization term is applied on the learning objective function to smooth the final learned weights and avoid over-fitting. Moreover, XGBoost introduces a novel sparsity-aware approximate framework for the greedy split finding algorithm [5] (an algorithm that can find the optimal split points), which helps to achieve superfast training speed with large scale dataset.

3.5.1 EPE Classification Modeling. Given a dataset containing information of mask images, wafer images and corresponding EPE labels on checkpoints, our objective is to construct an EPE detection model that can maximize the accuracy of EPE violation prediction. We generate datasets based on the flow described in section 4, and extract the layout features from input masks using MCCS. Corresponding EPE values will then be obtained from the ground truth litho-simulation results. EPE is the geometric displacement of the image contour from the target edge of the layout polygon, and the limit of such displacement is set as 15nm (as set in [2]). So the EPE labels of given checkpoints are encoded as follow.

$$Label_{EPE}(x, y) = \begin{cases} 1, & \text{if displacement} > 15nm, \\ -1, & \text{if displacement} \leq 15nm. \end{cases} \quad (9)$$

Clearly, this is a classification problem, we apply XGBoost to train our EPE detection model, and conduct cross-validation on our trained model. Once we complete the modeling, we can perform EPE violation prediction according to the input mask features, replacing the whole lithography simulation process as a black-box.

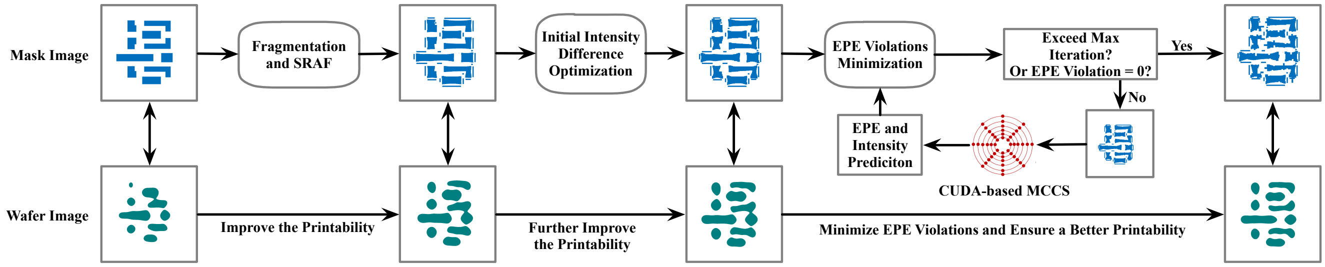

Mask Image
Fragmentation and SRAF
Initial Intensity Difference Optimization
EPE Violations Minimization
Exceed Max Iteration? Or EPE Violation = 0?
Yes
No
EPE and Intensity Prediciton
CUDA-based MCCS
Wafer Image
Improve the Printability
Further Improve the Printability
Minimize EPE Violations and Ensure a Better Printability

Figure 7: Illustration of machine learning based OPC acceleration framework

Algorithm 1 Machine Learning Based OPC Acceleration Framework

```
Input: Original mask description file;
Output: Optimized mask description file;
Initialize $MaxIter \leftarrow iter_{max}$, $StageTwoIter \leftarrow iter_{s2}$;
Segment the edges of the polygons into fragments;
Insert sub-resolution assist features;
for $iter \leftarrow 1$ to $StageTwoIter$ do
    if EPE violation number $\leq EPE_{threshold}$ then
        break;
    else
        Run intensity difference optimization for segments;
    end if
end for
for $iter \leftarrow StageTwoIter$ to $MaxIter$ do
    Extract the feature from optimized mask in last iteration;
    Predict EPE violations and intensity value through feature;
    if EPE violation number = 0 then
        break;
    else
        Run EPE violation optimization for each checkpoint
        according to predicted EPE and intensity value;
    end if
end for
return Optimized mask description file;
```

Note that for the construction of EPE prediction model, we will densely sample 100 circles on each sample clip, and then perform our feature selection algorithm to obtain the selected circle indices set $\mathcal{I}_{n_c^*}$. For the application step, the CUDA-based MCCS extractor will directly use the selected circle indices set $\mathcal{I}_{n_c^*}$.

3.5.2 Intensity Regression Modeling. We perform a similar process as section 3.5.1 to extract data and build the intensity prediction model. Unlike EPE classification, the intensity label from the ground-truth resist image after optical simulation is a continuous value, which makes the intensity prediction modeling a regression problem. We apply XGBoost to train our intensity prediction model, and conduct cross-validation on the trained model. By combining the intensity value, we can determine the optimization direction in the OPC process, and make up the direction information at the checkpoints. The CUDA-based MCCS configuration for intensity model is $l = 600nm$, $r_{in} = 60nm$, $inc_1 = 5$, $inc_2 = 10$, $n_p = 16$.

3.6 Machine Learning-based OPC Acceleration

To accelerate the OPC process, we propose our machine learning-based OPC acceleration framework for the OPC tool in [8] without changing its algorithmic flow. Fig. 7 and algorithm 1 demonstrate the flow of our acceleration scheme. For each input mask, we first perform fragmentation and SRAFs insertion in `stage` 1, followed by an intensity difference optimization process in `stage` 2. Detail algorithms for `stage` 1 and `stage` 2 are described in [8]. After that, `Stage` 3 will iteratively minimize the EPE and the squared L_2 error. In our implementation, `stage` 2 will be performed for 5 iterations and the maximum iteration number for `stage` 3 is 25. For each checkpoint on a vertical or horizontal edge of a polygon, the **original** `Stage` 3 needs to identify the corresponding impact segments, that are the segments whose movement will significantly affect the EPE and intensity value at that checkpoint. A conventional lithography simulation will be conducted in each iteration to calculate the EPE and intensity value for each checkpoint. The impact segments will be adjusted if an EPE violation occurs at the checkpoint. The direction and step size of the adjustment will be determined according to the intensity value obtained by the litho-simulator. We embed our CUDA-based MCCS extractor and pre-trained EPE/intensity prediction models into `stage` 3, to replace the lithography simulation process, and then directly predict the EPE/intensity values from the extracted layout features. As a result, we can achieve a significant runtime improvement in `stage` 3 (Fig 6(a)), which reduces **65.8%** runtime (from **7.14s** to **2.44s**) for a single iteration. The accelerated `stage` 3 will be conducted iteratively until it reaches the optimal or exceeds the max iteration limit.

We can see from the wafer images (Fig. 7) that the printability of the mask gradually increases in the process. Unlike some previous works like [18], our acceleration framework does not change the logic and algorithmic flow of the original method [8], but can still ensure a comparable printability. Note that due to the time limitation, our proposed framework is implemented in multi-platforms (`C/C++`, `CUDA`, `Python`), which brings extra I/O time (63.9% of total time) for cross-platform communication (Fig. 6(b)). We can further improve runtime performance by eliminating I/O time in future works.

4 EXPERIMENTAL RESULTS

Our MCCS feature extractor is implemented in `CUDA` (`C/C++`). The proposed frameworks for feature selection and model learning are implemented in `C++` and `Python`, and our ML-based OPC acceleration framework is implemented in `C` language. All validations are performed on a Linux sever with 3.5GHz Intel i7 CPU and a single Nvidia GTX TITAN X GPU. Due to lack of industrial benchmarks, we conduct experiments on public ICCAD-2013 contest benchmark (each test case is a 2048nm x 2048nm layout clip of 32nm M1 layer) using the lithography simulator provided by IBM [2] to demonstrate our idea. We perform conventional OPC process for each benchmark iteratively to obtain reasonable number of optimized masks. For each optimized mask, our MCCS feature extractor is performed to obtain the layout features, followed by a litho-simulation to obtain the golden intensity and EPE labels.

Table 1: Model Accuracy and Modeling Time

	EPE prediction model			Intensity prediction model		PGAN-OPC[18]
	CPU (s)	FA#	Acccuracy	CPU (s)	RMSE	CPU (s)
Regular feature	225	360	91.45%	122	0.003963	28000
Selected feature	1533	329	93.83%	-	-	-
Ratio	0.146	**1.094**	**0.975**	-	-	-

* *# test instances = 4309, mean value of intensity labels = 0.22436.*

4.1 Effectiveness of the Models

In the first experiment, we evaluate the effectiveness of our learning models and the second order circle subset selection method. We compare the performance of our EPE and intensity prediction model before and after applying the second order circle subset selection method, detailed comparison is shown in Table 1. Columns "**CPU(s)**", "**FA#**", "**Accuracy**", and "**RMSE**" list the runtime of the overall modeling flow in seconds (including feature extraction, model training and testing), the number of false alarms, the accuracy of the EPE prediction model, and the root-mean-square error of intensity prediction model. Note that the **CPU(S)** time do not include lithography-simulation time for mask instances, since it is irrelevant to the effectiveness of the modeling process.

In Table 1, our second order circle subset selection method shows its superiority and makes a significant improvement on the performance of the EPE prediction model, which increases the detection accuracy from **91.45%** to **93.83%**, and reduces the FA# from **360** to **329** respectively. Experimental result shows that EPE and intensity modeling takes only **225** seconds and **122** seconds respectively (1533 seconds after enabling feature selection), including data extraction time and model training time. Our training times are only **5.48%** and **0.44%** of PGAN-OPC's training time (28000s). This rapid modeling speed ensures its flexibility and scalability for new feature sizes. More importantly, our framework has better flexibility in comparison with models trained for specific-tasks. Once we finished the model training, we can easily embed it into other OPC tools or other lithography-guided applications like hotspots detection and SRAF insertion.

4.2 Image Fidelity and Runtime Performance Comparison

In the second experiment, we optimize the ten layout masks in the ICCAD 2013 contest benchmark suite [2] using the proposed machine learning-based OPC acceleration framework. We then compare the image fidelity and runtime performance with our original mask optimizer [8], ILT mask optimizer [6], and GAN/PGAN-OPC [18]. Quantitative results are listed in Table 2 and Fig. 8 shows the mask optimization results of our framework, original optimizer [8] and GAN-OPC [18]. In Table 2, column "**RT**" denotes the total runtime of the whole mask optimization process. Column "$\mathbf{L_2}$" is the squared L_2 error defined in section 2. Although we have a similar flow as [8], our faster EPE/intensity predictors allow the OPC optimization steps to be performed more thoroughly. Thus, it is notable that our proposed framework **outperforms** the previous works in terms of both image fidelity and runtime. We significantly reduces the squared L_2 error by **7.46%**, **12.09%**, **2.11%**, **1.74%** respectively. Most importantly, we achieve **2.68X**, **11.011X**, **5.356X**, **5.193X** runtime speedup respectively, by applying the ML-based models to replace lithography simulator. It's clear that our framework achieves significant runtime speedup as well as ensuring a comparable or even better printability in comparison with previous works.

In Table 3, we further compare the average normalized PVBand (NPVBand) performance of 10 test cases with previous works. It can be observed that our average NPVBand is better than that of our original mask optimizer [8], but slightly worse than GAN/PGAN-OPC [18]. The main reason why we get poorer NPVBand result comparing to the works in [18] is that our original mask optimizer [8] does not target at minimizing the PVBand (already lose 15.6% PVBand score comparing to [18]). However, as discussed in section 2.3, a small PVBand does not infer a high image fidelity since the hotspots in both extreme conditions can get cancelled out in computing the PVBand. Fig. 8 shows the wafer images produced by us and [18], and we can see the superiority of our results in terms of image fidelity.

5 CONCLUSION

In this paper, we study the mask optimization and lithography simulation problems, and present a fast lithography evaluation framework for lithography-related applications, followed by a machine learning-based OPC acceleration framework. Experiments demonstrate that the proposed framework achieves 2.6X-11X runtime speedup while ensuring a comparable or even better printability comparing to the state-of-the-art methods.

ACKNOWLEDGMENT

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK14208914).

REFERENCES

[1] Ahmed Awad, Atsushi Takahashi, Satoshi Tanaka, and Chikaaki Kodama. 2014. A fast process variation and pattern fidelity aware mask optimization algorithm. In *Proc. ICCAD*. 238–245.

[2] Shayak Banerjee, Zhuo Li, and Sani R. Nassif. 2013. ICCAD-2013 CAD contest in mask optimization and benchmark suite. In *Proc. ICCAD*. 271–274.

[3] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. 785–794.

[4] Nicolas Bailey Cobb. 1998. *Fast optical and process proximity correction algorithms for integrated circuit manufacturing.* Ph.D. Dissertation. University of California at Berkeley.

[5] Jerome H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29 (2000), 1189–1232.

[6] Jhih-Rong Gao, Xiaoqing Xu, Bei Yu, and David Z. Pan. 2014. MOSAIC: Mask Optimizing Solution With Process Window Aware Inverse Correction. In *Proc. DAC*.

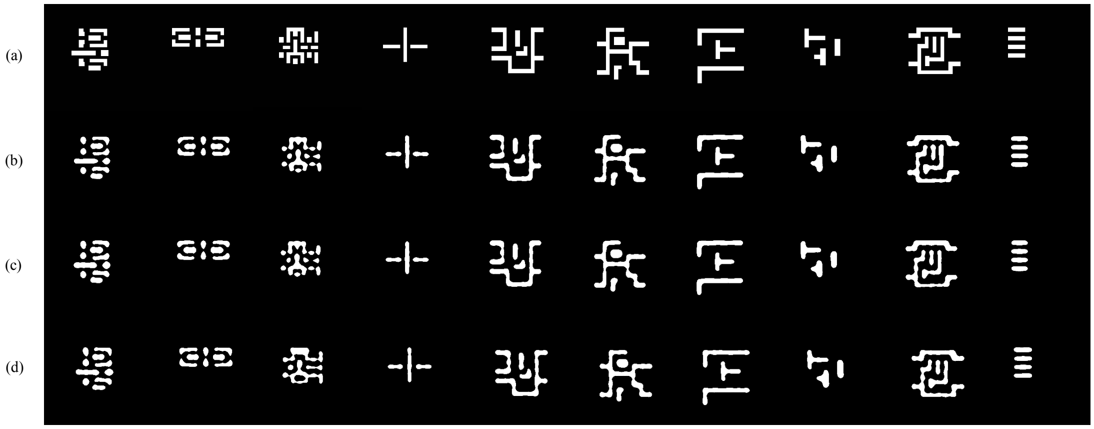


Figure 8: Visualization of wafer images produced by different methods: (a) target masks; (b) wafer images by masks of ours; (c) wafer images by masks of original mask optimizer [8]; (d) wafer images by masks of PGAN-OPC [18].

Table 2: Image Fidelity and Runtime Performance Comparison

Benchmarks		Ours		Original mask optimizer [8]		ILT [6]		GAN-OPC [18]		PGAN-OPC [18]	
ID	Area (nm^2)	RT (s)	L_2 (nm)	RT (s)	L_2 (nm)	RT (s)	L_2 (nm)	RT (s)	L_2 (nm)	RT (s)	L_2 (nm)
case1	215344	69.14	44721	278	53816	1280	49893	380	54970	358	52570
case2	169280	69.09	37418	142	41382	381	50369	374	46445	368	42253
case3	213504	80.86	80491	152	79255	1123	81007	379	88899	368	83663
case4	82560	67.61	19038	307	21717	1271	20044	376	18290	377	19965
case5	281958	71.69	47423	189	48858	1120	44656	378	42835	369	44733
case6	286234	76.53	44762	353	46320	391	57375	367	44313	364	46062
case7	229149	70.18	30400	219	31898	406	37221	377	24481	377	26438
case8	128544	68.87	18200	99	23312	388	19782	394	17399	383	17690
case9	317581	73.38	55767	119	55684	1138	55399	427	53637	383	56125
case10	102400	68.74	14451	61	19722	387	24381	395	9677	366	9990
Average	-	**71.61**	**39267.1**	191.9	42196.4	788.5	44012.7	383.6	40094.6	371.9	39948.9
Ratio	-	**1**	**1**	2.68	1.0746	11.011	1.1209	5.356	1.0211	5.193	1.0174

* *RT denotes total runtime*

Table 3: Average NPVBand Performance Comparison

	Ours	Original mask optimizer [8]	GAN-OPC [18]	PGAN-OPC [18]
Average NPVB	0.2232	0.2298	**0.1939**	0.1948

* *The NPVB of ILT is not listed since the mask data is not available to us.*

52:1–52:6.

[7] HH Hopkins. 1951. The concept of partial coherence in optics. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 208. The Royal Society, 263–277.

[8] Jian Kuang, Wing-Kai Chow, and Evangeline F. Y. Young. 2015. A robust approach for process variation aware mask optimization. In *Proc. DATE*. 1591–1594.

[9] Yibo Lin, Yuki Watanabe, Taiki Kimura, Tetsuaki Matsunawa, Shigeki Nojima, Meng Li, and David Z. Pan. 2018. Data Efficient Lithography Modeling with Residual Neural Networks and Transfer Learning. In *Proceedings of the 2018 International Symposium on Physical Design*. 82–89.

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

[11] Tetsuaki Matsunawa, Bei Yu, and David Z. Pan. 2015. Optical proximity correction with hierarchical Bayes model. In *Proc. SPIE*, Vol. 9426.

[12] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 8 (2005), 1226–1238.

[13] Amyn Poonawala and Peyman Milanfar. 2007. Mask design for optical microlithography–an inverse imaging problem. *IEEE Transactions on Image Processing* 16 (2007), 774–788.

[14] Yu-Hsuan Su, Yu-Chen Huang, Liang-Chun Tsai, Yao-Wen Chang, and Shayak Banerjee. 2016. Fast lithographic mask optimization considering process variation. *IEEE TCAD* 35 (2016), 1345–1357.

[15] Ramya Viswanathan, Jaione Tirapu Azpiroz, and Punitha Selvam. 2012. Process optimization through model based SRAF printing prediction. In *SPIE Advanced Lithography*, Vol. 8326.

[16] X. Xu, Y. Lin, M. Li, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, and D. Z. Pan. 2017. Sub-Resolution Assist Feature Generation with Supervised Data Learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2017), 1–1.

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

[18] Haoyu Yang, Shuhe Li, Yuzhe Ma, Bei Yu, and Evangeline F. Y. Young. 2018. GAN-OPC: Mask Optimization with Lithography-guided Generative Adversarial Nets. In *Proc. DAC*.

[19] Haoyu Yang, Jing Su, Yi Zou, Bei Yu, and Evangeline F. Y. Young. 2017. Layout Hotspot Detection with Feature Tensor Generation and Deep Biased Learning. In *Proc. DAC*. 62:1–62:6.

[20] Hang Zhang, Bei Yu, and Evangeline F. Y. Young. 2016. Enabling Online Learning in Lithography Hotspot Detection with Information-Theoretic Feature Optimization. In *Proc. ICCAD*. 47:1–47:8.

[21] Hang Zhang, Fengyuan Zhu, Haocheng Li, Evangeline F. Y. Young, and Bei Yu. 2017. Bilinear Lithography Hotspot Detection. In *Proc. ISPD*. 7–14.