VLSI Mask Optimization: From Shallow To Deep Learning

Haoyu Yang[1], Wei Zhong[2], Yuzhe Ma[1], Hao Geng[1], Ran Chen[1], Wanli Chen[1], Bei Yu[1]

[1] *Department of Computer Science and Engineering, The Chinese University of Hong Kong*

[2] *ISE, Dalian University of Technology*

Abstract

VLSI mask optimization is one of the most critical stages in manufacturability aware design, which is costly due to the complicated mask optimization and lithography simulation. Recent researches have shown prominent advantages of machine learning techniques dealing with complicated and big data problems, which bring potential of dedicated machine learning solution for DFM problems and facilitate the VLSI design cycle. However, uncertainty nature of state-of-the-art machine learning models have posed great challenges when developing alternative solutions. In this paper, we focus on a heterogeneous OPC framework that assists mask layout optimization. Instead of fitting neural networks for mask optimization tasks directly, a multi-class classification model is developed to capture design characteristics and hence determine the most suitable OPC engines. Experimental results have shown the efficiency and effectiveness of proposed frameworks that have the potential to be alternatives to existing EDA solutions.

Keywords: OPC, DNN, Mask learning, Attention

1. Introduction

VLSI mask optimization is one of the most critical stages in manufacturability aware design, which is costly due to the complicated mask optimization and lithography simulation. Recent studies have shown prominent advantages of machine learning techniques dealing with complicated and big data problems, which bring the potential of dedicated machine learning solution for design for manufacturability (DFM) problems and facilitate the VLSI design cycle [1, 2].

Related researches include layout hotspot detection [3, 4, 5, 6, 7, 8, 9] and mask optimization [10, 11, 9, 12, 13, 14] and pattern generation [15], all of which contribute to high performance mask optimization flow. Among the above, layout hotspot detection tries to identify regions that are sensitive to process variations and require additional care in optical proximity correction (OPC) stage, defect prediction at OPC runtime helps circumvent costly lithography simulation using efficient machine learning engine, and learning-based mask optimization flows directly speed-up OPC by either creating a good mask initialization for legacy OPC engine that requires fewer iterations to converge, or circumventing costly lithography simulation with regression/classification model and yields faster mask update in

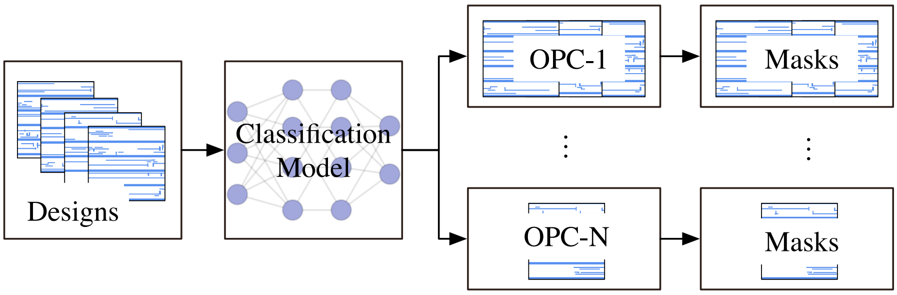


Figure 1 A heterogeneous OPC framework. The classification model identifies whether a design fits different OPC engines.

each iteration. These efforts not only bring benefits for modern OPC flow, but also present the importance of legacy OPC engines, which most, if not all, machine learning solutions still rely on.

Inverse lithography technique (ILT) [16, 17, 11] and model-based OPC [18, 19] are two representative mask optimization methodologies in literature. Compared to model-based OPC, ILTs usually promise good mask printability due to larger solution space. However, the conclusion does not always hold as ILTs require to solve a highly non-convex optimization problem which, sometimes, is hard to converge. Machine learning and deep neural networks are recently investigated to aid modern OPC engines. GAN-OPC [11] developed generative adversarial networks to generates mask starting point for ILT engines to reduce OPC iterations and achieve better convergence. [9] investigated discriminative model to predict edge placement error and hence improves per-cycle OPC efficiency. There are other attempts [12, 20] targeting at direct prediction of edge displaement in mask layouts to reduce overall optimization runtime.

Apparently, different patterns match different OPC engines as can be seen from a simple comparison between [19] and [16]. In this paper, we propose a heterogeneous OPC framework that tackles the possibility of machine learning assisting mask optimization from a different perspective, where a deterministic machine learning model is built to capture design characterestics and identify a better OPC solution for a given design, as shown in Figure 1. The framework also makes the development of machine learning engine less challenging, because misclassification of our neural network model does not pose significant side-effects and we will always have legacy solutions as our backbone.

This paper makes the following contributions:

- We conduct a survey on recent progress of deterministic machine learning models assisting printability estimation and generative models contributing to direct-printable mask synthesis.
- We propose a heterogeneous OPC flow where a deterministic machine learning model decides the proper OPC engine for a given pattern.
- We carefully design a classification model with task-aware loss function to better capture design characterestics and achieve our objectives.

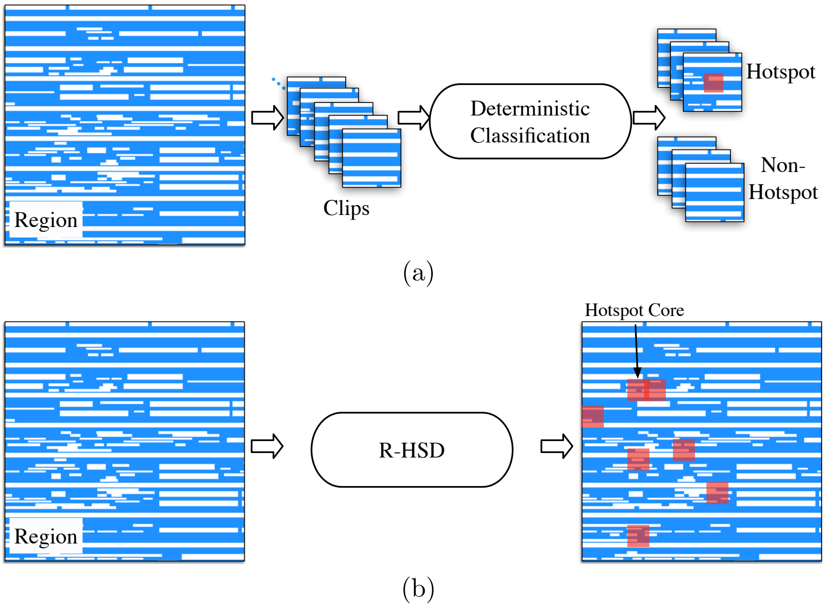


Figure 2 (a) Conventional hotspot detection flow vs. (b) Region based hotspot detection flow as presented in [21].

- Experiments show that the proposed framework takes advantage of both ILT and model-based OPC with trivial model prediction overhead.

Rest of the paper is organized as follows: Section 2 discusses state-of-the-art researches on layout hotspot detection; Section 3 surveys recent progress of OPC and some preliminary machine learning solutions; Section 4 introduces the development of the heterogeneous OPC framework followed by the experiments in Section 5; Section 6 concludes the paper.

2. Hotspot Detection via Machine Learning

2.1. Shallow Machine Learning Solutions

Before the exploding of deep neural networks, traditional machine learning solutions have been deeply investigated to detect lithography hotspots. Representative solutions include decision tree [22], support vector machine (SVM) [23, 24], artificial neural networks [23] and naive Bayes [25], which all follows a standard detection flow as in Figure 2(a).

Ding *et al.* [23] introduced an SVM-based hotspot detection flow, which hierarchically narrows down the search space for hotspot patterns. Layout designs were converted into to feature space by capturing fragment-based features. [24] further enhanced the hotspot detection performance using multiple SVM kernels that focus on difference hotspot clusters. Voting mechanism has made ensemble learning a more promising candidate machine learning framework. [22] incorporated Adaboost and decision tree learner for efficient layout hotspot detection and exhibited good trade-off between detection accuracy and false positive penalty. Another representative ensemble learning framework was proposed in [25], where the information-theoretic approach was applied in the feature extraction module. The problem was solved by a dynamic programming model and embedded into the smooth boosting model with naive Bayes. The lithography simulation overhead was further reduced.

Different from learning-based model designed for specific manufacturing problem on hotspot detection, Jiang *et al.* [9] proposed an independent mask printability evaluation framework which detects hotspots caused by EPE. A second order maximal circular mutual information scheme (SO-MCMI) was introduced to select the circle subset. The SO-MCMI is formulated as

$$\max_{\boldsymbol{w}} \quad \boldsymbol{w}^\top \boldsymbol{M} \boldsymbol{w} \tag{1a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n_c} w_i = n_c^*, w_i \in \{0,1\}, \forall i, \tag{1b}$$

where w_i in n_c-dimensional vector $\boldsymbol{w}$ indicates whether the i^{th} circle is selected. To overcome the potential impacts due to the complicated feature presentations, XGBoost is applied to handle EPE classification and intensity regression modeling.

2.2. Deep Learning Solutions

The fast development of deep neural networks brings new opportunities for hotspot detection solutions. Considering the limitations of conventional machine learning on scalability requirements for printability estimation and feature representation, a novel deep learning based hotspot detection model was proposed in [3]. A feature tensor extraction technology was developed to transform origin features into lower scale representations where spatial information is reserved. To facilitate the training procedure and find a better trade-off between accuracy and false alarm, a batch biased learning (BBL) was presented. BBL can adjust the label penalty for different instances dynamically to improve the model performance, as in Equation (2).

$$\epsilon(l) = \begin{cases} \frac{1}{1+\exp(\beta l)}, & \text{if } l \leq 0.3, \\ 0, & \text{if } l > 0.3, \end{cases} \tag{2}$$

where l is the training loss of the current instance or batch in terms of the unbiased ground truth and β is a manually determined hyper-parameter that controls how much the bias is affected by the loss.

Adaptive squish pattern was proposed in [4] to handle multilayer patterns. Compared with conventional squish patterns presents, the adaptive squish pattern not only reserves the property of lossless representation and store layout topologies and geometry information separately in a storage efficient format, but also provides a fixed size format which is suitable for most manchine learning models.

Imbalance of positve and negative samples of layout patterns are crital problem especially in machine learning based methods. A robust performance metric is needed to evaluate the model performance. ROC curve based measure for hotspot detection algorithm was proposed in [5], which provides a holistic view of imbalance on hotspot detection dataset. Multiple loss functions for neural network models are applied to handle the imbalance problem during training. A general loss function designed for maximize the AUC score can be expressed as

$$\mathcal{L}_{\Phi}(f) = \frac{1}{N_+ N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} \Phi\left(f\left(\boldsymbol{x}_i^+\right) - f\left(\boldsymbol{x}_j^-\right)\right), \tag{3}$$

where $f(\boldsymbol{x}_i^+)$ and $f(\boldsymbol{x}_j^-)$ are the prediction output of positive and negative samples of model f respectively. N_+ and N_- are number of positive and negative samples. The new loss functions present in [5] outperform the traditional cross-entropy loss on the state-of-the-art neural network model.

While these works deal with the patterns in small clips, the large regions with multiple hotspots cannot be handled directly. Recently, a region based method proposed by Chen *et al.* [21] solved this problem by enlarging the small clip into large regions (as depicted in Figure 2(b)). Inspired by the object detection task in computer vision field, a regression and classification multi-task framework is designed to handle multiple hotspots in large regions in a single epoch. The clip proposal network is applied to sample hotspot and non-hotspot regions for both classification and regression training. The loss function for regression on clip i can be written as

$$l_{loc}(l_i, l_i') = \begin{cases} \frac{1}{2}(l_i - l_i')^2, & \text{if } |l_i - l_i'| < 1, \\ |l_i - l_i'| - 0.5, & \text{otherwise}, \end{cases} \tag{4}$$

where l_i and l_i' are the coordinates of prediction and ground truth respectively. The classification loss for clip i can be formulated as

$$l_{hotspot}(h_i, h_i^{'}) = -(h_i \log h_i^{'} + h_i^{'} \log h_i), \tag{5}$$

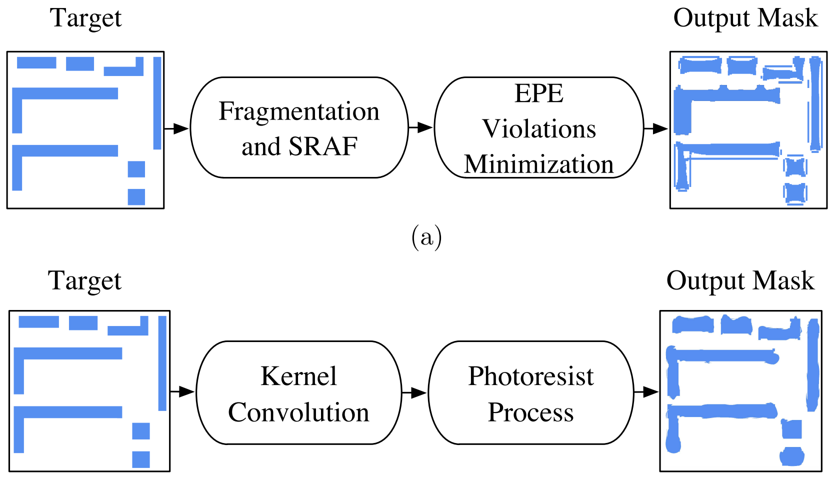
where h_i is the prediction of the model and h_i' is the label. Compared to the deterministic classification flow, the performance in [21] is improved greatly.

2.3. Overcome Imbalance: Pattern Generation

In real VLSI manufacturing scenario, hotspot patterns are usually fetal but rare in a design. This brings challenge for most learning-based solutions which require massive and diverse hotspot data to get a machine learning model well trained. [15] studied the possibility of generating DRC-clean test layout patterns with a generative machine learning model called transforming convolutional auto-encoder (TCAE). Inspired from transforming auto-encoder (TAE) [26], TCAE replaced capsule units with simpler latent vector nodes to represent part-whole feature representation. The identity mapping in TCAE-training allows a neural network to capture certain design rules. Dedicated perturbations on latent vectors promises to create diverse and DRC-clean patterns.

3. Mask Optimization via Machine Learning

Mask optimization ensures good mask printability and hence improves chip manufacturing yield. In advanced technology nodes, the conventional mask optimization processes including model-based and ILT-based approaches consume increasingly more computational resources. The flows of model-based and ILT-based approaches are shown in Figure 3. In this section, we will discuss several machine learning-based alternatives that assist traditional mask optimization flow.



(a)

(b)

Figure 3 The flows of conventional OPC approaches: (a) model-based; (b) ILT-based.

3.1. Machine Learning-based OPC

The superiority of machine learning-based solutions has been evaluated in OPC [12]. However, the lack of scalability under advanced technology nodes becomes the main issue hindering the widespread deployment of a model-based OPC framework. Aiming at addressing the scalability issue, a fast machine learning-based mask printability prediction (MPP) framework [9] for lithography-related applications was proposed. The work can be extended to improve the scalability for different lithography-related applications. To enable the performance of the machine learning-based flow, a matrix-based concentric circle sampling (MCCS) method and a second-order circle subset selection algorithm for feature extraction were designed in [9]. The MPP framework has been demonstrated its effectiveness by embedding into a conventional mask optimization tool.

Existing machine learning models [27, 28, 12] can only perform pixel-wise or segment-wise mask calibration that is not computationally efficient. In accordance with the critical problem, [11] proposed a generative adversarial network (GAN) based mask optimization flow that takes target circuit patterns as input and generates quasi-optimal masks for further inverse lithography technique (ILT) refinement.

To enhance the computational efficiency and alleviate the over-fitting issue, training topologies are synthesized. For the sake of faster training procedure, an ILT-guided pre-training flow was proposed in [11] to initialize the generator with intermediate ILT results. Besides, the authors designed new objectives of the discriminator to ensure the model is trained towards a target-mask mapping instead of a distribution. The new objective function is as follows:

$$\min_{\boldsymbol{G}} \max_{\boldsymbol{D}} \ \mathbb{E}_{\boldsymbol{Z}_t \sim \mathcal{Z}}[1 - \log(\boldsymbol{D}(\boldsymbol{Z}_t, \boldsymbol{G}(\boldsymbol{Z}_t))) + ||\boldsymbol{M}^* - \boldsymbol{G}(\boldsymbol{Z}_t)||_n^n] + \mathbb{E}_{\boldsymbol{Z}_t \sim \mathcal{Z}}[\log(\boldsymbol{D}(\boldsymbol{Z}_t, \boldsymbol{M}^*))], \quad (6)$$

where $\boldsymbol{Z}_t$ represents the target layout, $\mathbf{G}$ for the generator output, $\boldsymbol{D}$ for the discriminator output, p_x for some distribution, $\boldsymbol{M}^*$ for the reference mask, and a set of target patterns $\mathcal{Z} = \{\boldsymbol{Z}_{t,i}, i = 1, 2, \ldots, N\}$ and a corresponding reference mask set $\mathcal{M} = \{\boldsymbol{M}_i^*, i = 1, 2, \ldots, N\}$. Experimental results have verified that this flow can facilitate the mask optimization process as well as ensure a better printability.

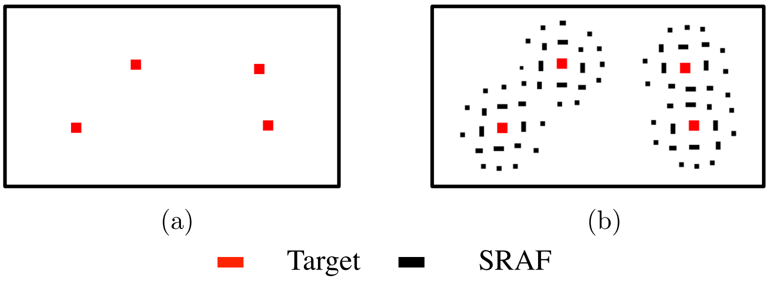
3.2. Machine Learning-based SRAF Insertion

Although conventional OPC can size the mask to give the correct critical dimension (CD) on the wafer, it cannot make the isolated target pattern become dense [29]. As a result, sub-resolution assist feature (SRAF) [30] insertion was proposed. There is a wealth of literature on the topic of SRAF insertion for mask optimization, which can be roughly divided into three categories: rule-based approach, model-based approach, and machine learning-based approach. However, prior machine learning-based approaches [31, 13] lack well-discrimination feature extraction techniques as well as a global view in SRAF designs, which leads to unsatisfied simulation results.

Geng *et al.* firstly revised conventional concentric circle area sampling (CCAS) feature construction method, by proposing a supervised online dictionary learning algorithm for simultaneous feature extraction and dimensionality reduction [10]. In other words, label information is not only utilized in learning stage but also imposed in feature extraction stage, which in turn benefits the learning counterpart. Equation (7) is the main objective function for supervised feature revision, where $\boldsymbol{y}_t \in \mathbb{R}^n$ refers to an input CCAS feature vector, $\boldsymbol{q}_t \in \mathbb{R}^s$ corresponds to discriminative sparse code of t-th input feature vector, $h_t \in \mathbb{R}$ is the label of an input, $\boldsymbol{x}_t \in \mathbb{R}^s$ represents sparse codes, $\boldsymbol{D} = \{\boldsymbol{d}_j\}_{j=1}^s, \boldsymbol{d}_j \in \mathbb{R}^n$ denotes the dictionary made up of atoms to encode input features, $\boldsymbol{A} \in \mathbb{R}^{s \times s}$ is a matrix transforming original sparse code $\boldsymbol{x}_t$ into discriminative sparse code, $\boldsymbol{W} \in \mathbb{R}^{1 \times s}$ is the related weight vector, and α, β represent the balancing hyper-parameters.

$$\min_{\boldsymbol{x},\boldsymbol{D},\boldsymbol{A},\boldsymbol{W}} \frac{1}{N} \sum_{t=1}^{N} \{\frac{1}{2} \left\| \left(\boldsymbol{y}_t^\top, \sqrt{\alpha}\boldsymbol{q}_t^\top, \sqrt{\beta} h_t\right)^\top - \begin{pmatrix} \boldsymbol{D} \\ \sqrt{\alpha}\boldsymbol{A} \\ \sqrt{\beta}\boldsymbol{W} \end{pmatrix} \boldsymbol{x}_t \right\|_2^2 + \lambda \|\boldsymbol{x}_t\|_p \}. \quad (7)$$

To consider SRAF design rules in a global view, the authors construct an integer linear programming (ILP) model in the post-processing stage of their SRAF insertion framework. Experimental results demonstrate the efficacy of the proposed SRAF insertion flow in [10].


(a)

(b)

Figure 4 The visualization of the SRAF image translation in [14]: (a) Original layout with target contacts; (b) SRAFed layout.

However, [10] lies on raw CCAS feature which is manually-crafted but not automatically learnt by the learning model yet. Besides, the grid-based ILP method lacks efficiency, especially for large designs. So there still exists big room to improve. Very recently, GAN-SRAF [14] casted the original SRAF insertion as an image-to-image translation problem where a layout is translated from its original domain to SRAFed layout domain. The visualization of the SRAF image translation is shown in Figure 4. To achieve this formulation, Alawieh *et al.* firstly adopted conditional generative adversarial network (CGAN) in SRAF insertion. In addition, to fit CGAN training, a novel multi-channel heatmap encoding/decoding scheme was proposed to map layouts to images without information loss. The loss function is designed as Equation (8):

$$\min_G \max_D \; \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x},\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x},\boldsymbol{z})))] + \lambda_{L1}\mathbb{E}_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}}\left[\|\boldsymbol{y} - G(\boldsymbol{x},\boldsymbol{z})\|_1\right], \tag{8}$$

where $\boldsymbol{x}$ is an observed image, $\boldsymbol{y}$ is an output image, $\boldsymbol{z}$ denotes a random noise vector, G and D refer to the generator and discriminator in a CGAN, respectively. To further reduce blurring, the authors adopt L_1-norm rather than L_2-norm. With comparable lithographic performance, the GAN-SRAF framework surpasses prior works significantly on insertion speed.

3.3. OPC in Multiple Patterning Scenarios

In advanced technology nodes, layout decomposition and mask optimization are two of the most critical RET stages. In layout decomposition, a target image is divided into several masks, while in mask optimization, each decomposed mask is optimized by some RET techniques like OPC [32].

[33] is a pioneer work that considers multiple exposure effects in ILT framework. To automatically synthesize the masks and then print the desired wafer pattern, [33] first combined ILTs and double-exposure lithography. Via inverting the forward model from mask to wafer, ILTs synthesize the input mask to obtain the required wafer pattern. On the other hand, double-exposure lithography exploits two masks under two illumination settings to print the desired wafer pattern. The objective function of [33] is shown in Equation (9), which is formulated as minimizing the L_2-norm of the difference between the desired pattern

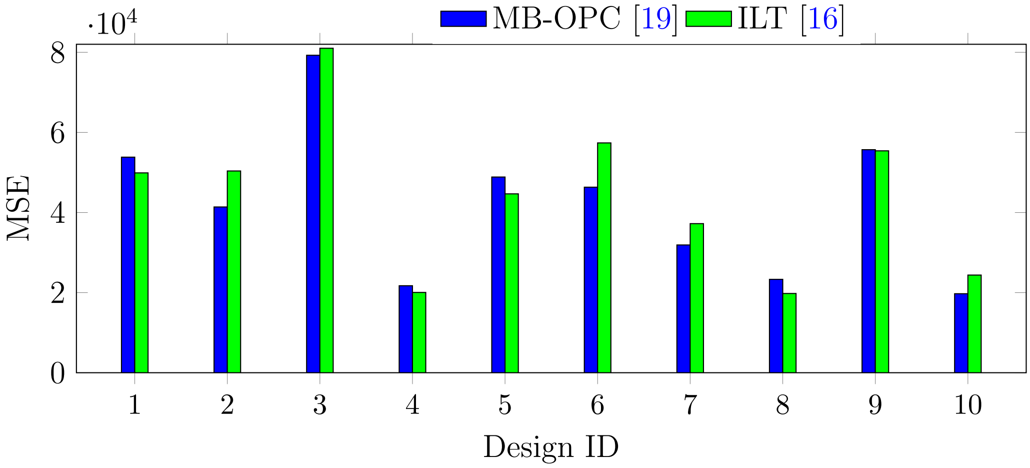


Figure 5 Performance gap between model-based OPC and ILT on ten designs from ICCAD2013 CAD Contest [34].

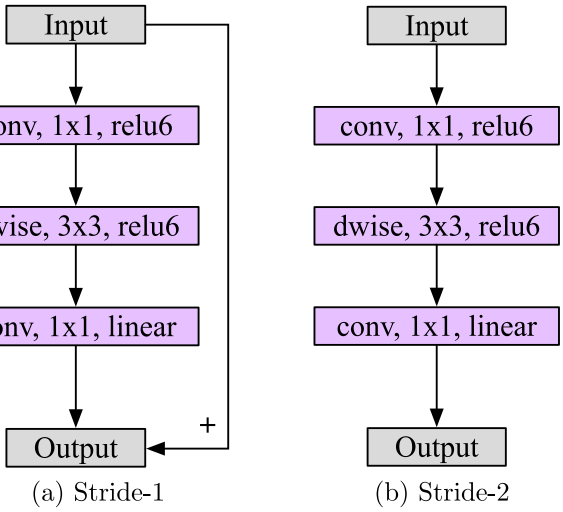
$\boldsymbol{z}^*$ and the aerial image $|\boldsymbol{Ha}|^2 + |\boldsymbol{Hb}|^2$. Here $\boldsymbol{H}$ is a jinc function with cutoff frequency of NA/λ, and $\boldsymbol{a}$, $\boldsymbol{b}$ are sampled from two input masks.

$$\min_{\boldsymbol{a},\boldsymbol{b}} F(\boldsymbol{a},\boldsymbol{b}) = \operatorname*{argmin}_{\boldsymbol{a},\boldsymbol{b}} \left\| \boldsymbol{z}^* - |\boldsymbol{Ha}|^2 - |\boldsymbol{Hb}|^2 \right\|_2^2 . \tag{9}$$

However, [33] has not addressed the layout decomposition problem yet. Ma *et al.* firstly developed a unified optimization framework which solves layout decomposition and mask optimization simultaneously [17]. To compatible with the objective, an unified mathematical formulation $\min_{\boldsymbol{M}_1,\boldsymbol{M}_2} F = \|\boldsymbol{Z}_t - \boldsymbol{Z}\|_2^2$ is proposed in [17], where $\boldsymbol{Z}_t$ represents the target image with $\boldsymbol{Z}$ the printed image, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ for output masks. A gradient-based optimization approach with a set of discrete optimization techniques is also proposed to solve the problem efficiently. The experimental results in [17] demonstrate the efficacy of the unified framework.

4. Heterogeneous OPC

Previous works have shown that different OPC engines exhibit advantages on different designs. [16] and [19] are two representative implementations of ILT and model-based OPC engine. Figure 5 depicts the performance gap of two engines on ten designs from ICCAD2013 CAD Contest [34]. Because in most cases model-based OPC runs faster than ILT, if we can efficiently predict the behavior of different OPC engines and hence choose the best one, meanwhile the throughput of mask optimization flow can be significantly improved. The observation, therefore, inspires the design of a heterogeneous OPC framework, which adopts a deterministic machine learning model identifies the best OPC engine for a given design with negligible overhead.



(a) Stride-1

(b) Stride-2

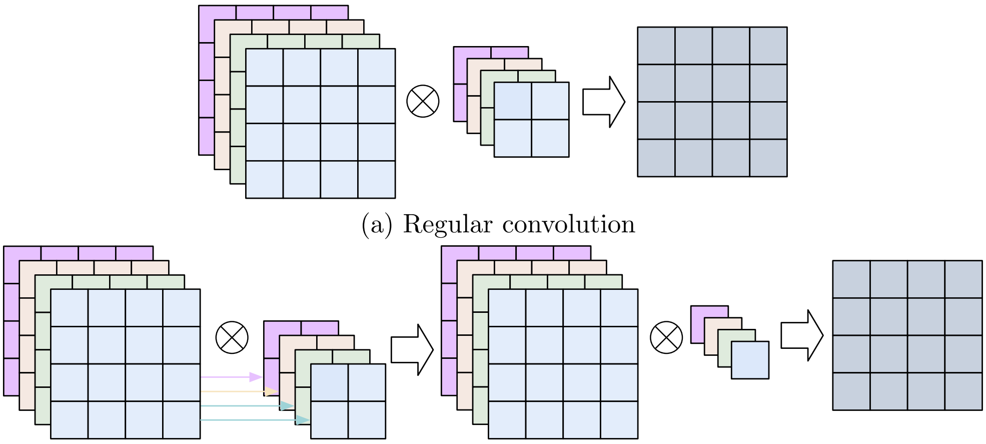
Figure 6 Visualization of bottleneck layer: (a) stride=1; (b) stride=2.

Table 1 Neural Network Configuration.

Layer	Kernel Size	Stride	Output Node #
conv2d	3	2	$112 \times 112 \times 32$
bottleneck	3	1	$112 \times 112 \times 16$
bottleneck	3	1	$56 \times 56 \times 24$
bottleneck	3	1	$28 \times 28 \times 32$
bottleneck	3	1	$14 \times 14 \times 64$
bottleneck	3	1	$14 \times 14 \times 96$
bottleneck	3	1	$7 \times 7 \times 160$
bottleneck	3	1	$7 \times 7 \times 320$
conv2d	1	1	$7 \times 7 \times 1280$
avepooling	7	-	$1 \times 1 \times 1280$
conv2d	1	1	$1 \times 1 \times 1280$
fc	-	-	2

4.1. Efficient OPC Engine Selection with MobileNetV2

Since the explosion of deep neural networks and machine learning, many powerful neural network architectures (e.g. VGG, ResNet, MobileNet, and etc) have been proposed. These neural network designs have been proved to provide satisfactory results on classification tasks. Therefore, the overhead to achieve satisfactory results will be our consideration. Here, we pick the model with smallest cost in computation and storage. To accommodate the demands for efficient back-end design cycle, we adopt the MobileNetV2 [35] as our baseline neural network model, as detailed in Table 1. The basic components of the network is bottleneck depth-separable convolution with residuals (referred as *bottleneck layer*). Bottleneck layers are constructed differently according to their stride settings. Stride-1 blocks come similarly with residue blocks with short cut connections between input and output layers (Figure 6(a)). Stride-2 blocks are traditional feed-forward style by stacking 1×1 and depth-wise 3×3 convolutions with non-linearity.


(a) Regular convolution

(b) Dwise convolution

Figure 7 Depth-wise separable convolution saves computation cost. (a) Regular convolution. (b) Depth-wise separable convolution.

The “dwise” represents the depth-wise separable convolution operation that has been widely used neural network architecture designs. As shown in Figure 7(b), depth-wise convolution performs convolution channel by channel to reduce multiplication operations, and separable convolution decomposes $k \times k$ kernel into $k \times 1$ and $1 \times k$. Basically, such operation can achieve the functionality of standard convolutions (see Figure 7(a)) with significantly reduced parameter number and computation cost by an order of k^2, where k is the convolution kernel size.

4.2. Channel-wise Attention Assists to Capture Design Characteristics

Although the neural networks are designed with reduced computation cost thanks to the efficient depth-wise convolution layer. We can observe that different channels are less correlated than regular convolutions as in Figure 7. Since different channels in intermediate feature maps are activation of corresponding convolution kernels, it is necessary to select semantic meaningful channels for efficient feature extraction and classification.

Attention [36] is originally proposed in sequence-to-sequence transformation applications, aiming to map a query and key-value pairs to an output. In sequence transformation domain, such structure captures the relationships and relative importance of different nodes in a given sequence. Interestingly, this neural network design also exhibits promising results in image understanding [37]. To ensure that our model can capture semantic attributes, we propose the bottleneck-attention layer that embeds channel-wise attention into bottleneck layers. The architecture is visualized in Figure 8, where Φ is a mapping function (that can be composed of pooling and auto-encoders) to acquire attention information. Here, we represent Φ as a combination of poolings and feature learning. For a given input feature map $\boldsymbol{X}$, we first get its abstraction with global average and max pooling (calculate pooling

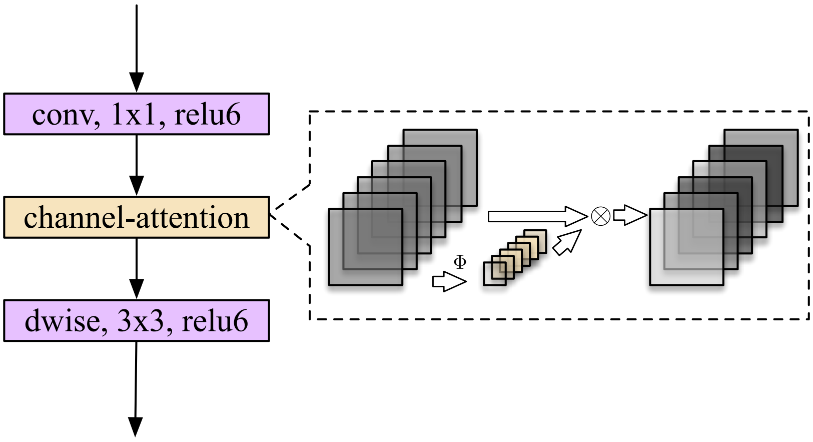


Figure 8 Embedding channel-wise attention into bottleneck layers.

across the entire feature map in each channel),

$$\boldsymbol{p}_{\text{ave}} = \text{gAveragePool}(\boldsymbol{X}), \boldsymbol{p}_{\text{max}} = \text{gMaxPool}(\boldsymbol{X}). \tag{10}$$

The vectors $\boldsymbol{p}_{\text{ave}}$, $\boldsymbol{p}_{\text{max}}$ will be feed into an auto-encoder structure for feature learning and transformation.

$$\boldsymbol{p}'_{\text{ave}} = \text{AE}(\boldsymbol{p}_{\text{ave}}; \boldsymbol{w}), \boldsymbol{p}'_{\text{max}} = \text{AE}(\boldsymbol{p}_{\text{max}}; \boldsymbol{w}). \tag{11}$$

Then the pooling information will be added together to obtain attention map,

$$\boldsymbol{p} = \sigma(\boldsymbol{p}'_{\text{ave}} + \boldsymbol{p}'_{\text{max}}), \tag{12}$$

where σ denotes the activation function. The final output of the attention module will be the element-wise product between $\boldsymbol{p}$ (cast to the same dimension as $\boldsymbol{X}$) and $\boldsymbol{X}$.

5. Experimental Results

5.1. Configurations

As a case study, in this paper, we adopt two OPC engines that are based on ILT and compact model respectively. We adopt the same training design set (with 4000 randomly generated designs) as used to train GAN-OPC [11] which are fed into an ILT engine [16] and a model-based OPC [19]. Each pattern in the training set will be fed into different OPC engines and hence we can obtain the MSE of corresponding mask simulation contours. We then label all instances in the training set with the following rule:

$$y = \begin{cases} 1, & \text{if DATE15 MSE} \geq \text{DAC14 MSE}, \\ 0, & \text{otherwise.} \end{cases}$$

To fit the neural network well, we also each GDSII design is converted into image with a resolution of 1nm/pixel and down-sampled to 224×224.

Table 2 Evaluation of the proposed H-OPC.

ID	MB-OPC [19]		ILT [16]		H-OPC [38]		H-OPC-Attention	
	MSE	Time	MSE	Time	MSE	Time	MSE	Time
1	53816	278	49893	1280	49893	1280	49893	1280
2	41382	142	50369	381	41382	142	41382	142
3	79255	152	81007	1123	79255	152	79255	152
4	21717	307	20044	1271	21717	307	20044	1271
5	48858	189	44656	1120	44656	1120	44656	1120
6	46320	353	57375	391	46320	353	46320	353
7	31898	219	37221	406	31898	219	31898	219
8	23312	99	19782	388	19782	388	19782	388
9	55684	119	55399	1138	55684	119	55684	119
10	19722	61	24381	387	19722	61	19722	61
Avg.	42196.4	191.9	44012.7	788.5	41030.9	414.1	**40863.6**	510.5
Ratio	1.03	0.46	1.07	1.90	1.00	1.00	**0.99**	1.23
Acc.	-		-		80%		**90%**	

5.2. Model Evaluation

We evaluate the proposed framework using ten designs from ICCAD2013 CAD Contest [34]. Each design is fed into the trained CNN model before going through the mask optimization stage. CNN predicts which OPC engine behaves better on the given design. Detailed results are listed in Table 2, where "MB-OPC", "ILT", "H-OPC" and "H-OPC-Attention" list the results of model-based OPC, inverse lithography technique-based OPC, a more efficient neural network model that has similar behavior as used in [38] and the proposed framework with bottleneck-attention layers. In the table, column "ID" represents 10 designs included in the benchmark suite, columns "MSE" indicate the mean square error between the simulated wafer image and the design for each OPC solution, and columns "Time" list the mask optimization runtime of each design using three solutions. As can be seen, the baseline heterogeneous OPC framework can assign better OPC engines to 8 out of ten designs in the benchmark suit, which hence results in better mask optimization performance with average MSE reduced by $\sim 3\%$. Also, the trade-off on runtime overhead is more balanced with the help of a deterministic learning model. With the help of attention layers, our H-OPC framework is able to increase the prediction accuracy from 80% to 90% and hence results in 1% improvement compared to the baseline H-OPC model.

5.3. Discussion

Performance of an OPC engine is affected by many aspects. These include the complexity of target designs and the OPC recipes. Generally speaking, model-based OPC runs faster than ILT because only one forward lithography simulation is required in each iteration and the adjustments only target at shape segments. On the other hand, ILT requires both forward and backward calculations in each optimization step and the mask optimization is in pixel level.

ILT, apparently, offers larger solution space and hence hopefully better mask optimization results. However, we can still observe cases that MBOPC surpasses ILT in terms of contour MSE. This can be explained by the uncertainty in non-convex nature of ILT. These facts make it hard to explicitly distinguish the best engine for a given design and motivate the research to leverage the performance gap between different OPC engines.

The attention component, thanks to its advantage of feature extraction by capturing key information from the learned feature maps, brings improvements on prediction accuracy and average MSE. We can also observe that 20% more runtime overhead is introduced with attention components. Such result is actually what we expected because each design is labeled only according to the simulated contour MSE. Thus, a better trained classification model will result in better MSE only. However, more evaluation metrics can be easily considered by introducing additional label rules when preparing the training set.

6. Conclusion and Future Work

In this paper, we study recent advances of machine learning techniques on VLSI mask optimization problems. We show that both deterministic and generative machine learning models assist to manufacturing-friendly layout design. The former helps to identify process weak regions in a design and can speed-up OPC by circumventing costly lithography simulation. The latter focuses on generation of directly printable masks. Observing the importance of legacy OPC engines in machine learning-based solutions, we propose a new methodology that a machine learning model facilitates modern OPC flow. A deterministic classification model is designed to identify the best OPC engine for a given design with negligible computing overhead. We hope the study can motivate deeper explorations of machine learning solutions for VLSI mask optimization, which should not only include research on machine learning-based OPC engine itself but should also dig into a flow control level. To prototype such framework for the benefits in practical chip manufacturing scenario, future works include (1) supporting large-scale full chip input and (2) stronger classifier design that enable increased number of OPC engines and/or OPC recipes.

Acknowledgment

This work is supported by The Research Grants Council of Hong Kong SAR (No. CUHK24209017 and No. CUHK14209420).

References

[1] D. Z. Pan, B. Yu, J.-R. Gao, Design for manufacturing with emerging nanolithography, IEEE TCAD 32 (2013) 1453–1472.

[2] P. De Bisschop, Optical proximity correction: a cross road of data flows, Japanese Journal of Applied Physics 55 (2016) 06GA01.

[3] H. Yang, J. Su, Y. Zou, Y. Ma, B. Yu, E. F. Y. Young, Layout hotspot detection with feature tensor generation and deep biased learning, IEEE TCAD 38 (2019) 1175–1187.

[4] H. Yang, P. Pathak, F. Gennari, Y.-C. Lai, B. Yu, Detecting multi-layer layout hotspots with adaptive squish patterns, in: Proc. ASPDAC, pp. 299–304, 2019.

[5] W. Ye, Y. Lin, M. Li, Q. Liu, D. Z. Pan, LithoROC: lithography hotspot detection with explicit ROC optimization, in: Proc. ASPDAC, pp. 292–298, 2019.

[6] T. Matsunawa, B. Yu, D. Z. Pan, Laplacian eigenmaps and bayesian clustering based layout pattern sampling and its applications to hotspot detection and OPC, in: Proc. ASPDAC, pp. 679–684, 2016.

[7] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, C. Chiang, Machine-learning-based hotspot detection using topological classification and critical feature extraction, in: Proc. DAC, pp. 671–676, 2013.

[8] Y. Tomioka, T. Matsunawa, C. Kodama, S. Nojima, Lithography hotspot detection by two-stage cascade classifier using histogram of oriented light propagation, in: Proc. ASPDAC, pp. 81–86, 2017.

[9] B. Jiang, H. Zhang, J. Yang, E. F. Young, A fast machine learning-based mask printability predictor for OPC acceleration, in: Proc. ASPDAC, pp. 412–419, 2019.

[10] H. Geng, H. Yang, Y. Ma, J. Mitra, B. Yu, SRAF insertion via supervised dictionary learning, in: Proc. ASPDAC, pp. 406–411, 2019.

[11] H. Yang, S. Li, Y. Ma, B. Yu, E. F. Young, GAN-OPC: Mask optimization with lithography-guided generative adversarial nets, in: Proc. DAC, pp. 131:1–131:6, 2018.

[12] T. Matsunawa, B. Yu, D. Z. Pan, Optical proximity correction with hierarchical bayes model, in: Proc. SPIE, volume 9426, 2015.

[13] X. Xu, Y. Lin, M. Li, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, D. Z. Pan, Subresolution assist feature generation with supervised data learning, IEEE TCAD 37 (2017) 1225–1236.

[14] M. B. Alawieh, Y. Lin, Z. Zhang, M. Li, Q. Huang, D. Z. Pan, GAN-SRAF: Sub-resolution assist feature generation using conditional generative adversarial networks, in: Proc. DAC, pp. 149:1–149:6, 2019.

[15] H. Yang, P. Pathak, F. Gennari, Y.-C. Lai, B. Yu, DeePattern: Layout pattern generation with transforming convolutional auto-encoder, in: Proc. DAC, pp. 148:1–148:6, 2019.

[16] J.-R. Gao, X. Xu, B. Yu, D. Z. Pan, MOSAIC: Mask optimizing solution with process window aware inverse correction, in: Proc. DAC, pp. 52:1–52:6, 2014.

[17] Y. Ma, J.-R. Gao, J. Kuang, J. Miao, B. Yu, A unified framework for simultaneous layout decomposition and mask optimization, in: Proc. ICCAD, pp. 81–88, 2017.

[18] Y.-H. Su, Y.-C. Huang, L.-C. Tsai, Y.-W. Chang, S. Banerjee, Fast lithographic mask optimization considering process variation, IEEE TCAD 35 (2016) 1345–1357.

[19] J. Kuang, W.-K. Chow, E. F. Y. Young, A robust approach for process variation aware mask optimization, in: Proc. DATE, pp. 1591–1594, 2015.

[20] T. Matsunawa, B. Yu, D. Z. Pan, Optical proximity correction with hierarchical bayes model, JM3 15 (2016) 021009.

[21] R. Chen, W. Zhong, H. Yang, H. Geng, X. Zeng, B. Yu, Faster region-based hotspot detection, in: Proc. DAC, pp. 146:1–146:6, 2019.

[22] T. Matsunawa, J.-R. Gao, B. Yu, D. Z. Pan, A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction, in: Proc. SPIE, volume 9427, 2015.

[23] D. Ding, B. Yu, J. Ghosh, D. Z. Pan, EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation, in: Proc. ASPDAC, pp. 263–270, 2012.

[24] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, C. Chiang, Machine-learning-based hotspot detection using topological classification and critical feature extraction, IEEE TCAD 34 (2015) 460–470.

[25] H. Zhang, B. Yu, E. F. Y. Young, Enabling online learning in lithography hotspot detection with information-theoretic feature optimization, in: Proc. ICCAD, pp. 47:1–47:8, 2016.

[26] G. E. Hinton, A. Krizhevsky, S. D. Wang, Transforming auto-encoders, in: Proc. ICANN, pp. 44–51, 2011.

[27] R. Luo, Optical proximity correction using a multilayer perceptron neural network, Journal of Optics 15 (2013) 075708.

[28] A. Gu, A. Zakhor, Optical proximity correction with linear regression, IEEE TSM 21 (2008) 263–271.

[29] C. A. Mack, Scattering bars, Solid State Technology (2003).

[30] C. H. Wallace, P. A. Nyhus, S. S. Sivakumar, Sub-resolution assist features, 2009. US Patent.

[31] X. Xu, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, D. Z. Pan, A machine learning based

framework for sub-resolution assist feature generation, in: Proc. ISPD, pp. 161–168, 2016.

[32] B. Yu, K. Yuan, D. Ding, D. Z. Pan, Layout decomposition for triple patterning lithography, IEEE TCAD 34 (2015) 433–446.

[33] A. Poonawala, P. Milanfar, Double-exposure mask synthesis using inverse lithography, JM3 6 (2007) 043001–043001.

[34] S. Banerjee, Z. Li, S. R. Nassif, ICCAD-2013 CAD contest in mask optimization and benchmark suite, in: Proc. ICCAD, pp. 271–274, 2013.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520, 2018.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proc. NIPS, pp. 5998–6008, 2017.

[37] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T.-S. Chua, Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning, in: Proc. CVPR, pp. 5659–5667, 2017.

[38] H. Yang, W. Zhong, Y. Ma, H. Geng, R. Chen, W. Chen, B. Yu, Vlsi mask optimization: From shallow to deep learning, in: Proc. ASPDAC, IEEE, pp. 434–439, 2020.