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]CSE Department, The Chinese University of Hong Kong
[2]ISE, Dalian University of Technology
{hyyang,byu}@cse.cuhk.edu.hk, zhongwei@dlut.edu.cn

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. In this paper, we focus on a heterogeneous OPC framework that assists mask layout optimization. Preliminary results show the efficiency and effectiveness of proposed frameworks that have the potential to be alternatives to existing EDA solutions.

I 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 DFM problems and facilitate the VLSI design cycle [1, 2].

Related researches include layout hotspot detection [3, 4, 5, 6, 7, 8, 9], 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 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 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. Apparently, different patterns match different OPC engines as can be seen from a simple comparison between [19] and [16]. In this paper, we tackle the possibility of machine learning assisting mask optimization from a different perspective, where a deterministic machine learning model is built to identify a better OPC solution for a given design, as shown in Fig. 1. This paper makes the following contributions:

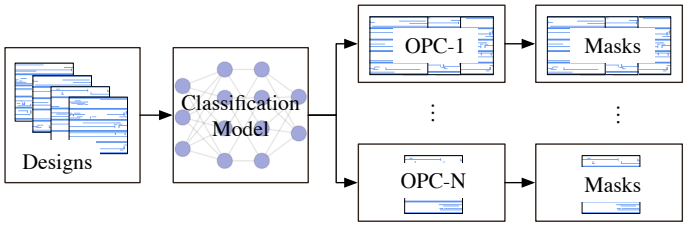
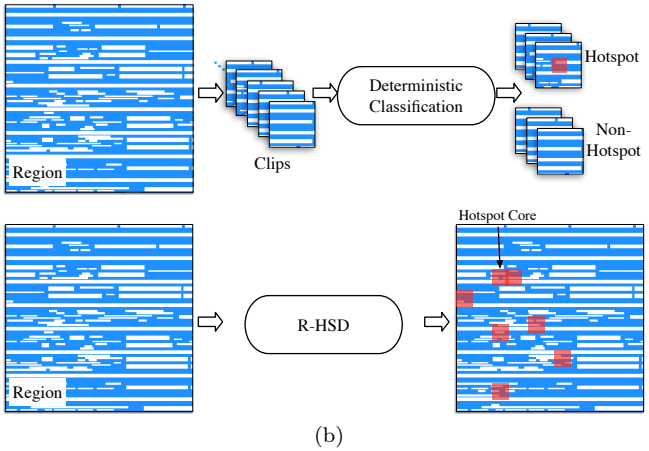


Fig. 1.: A heterogeneous OPC framework. The classification model identifies whether a design fits different OPC engines.

- 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.
- 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 II discusses state-of-the-art researches on layout hotspot detection; Section III surveys recent progress of OPC and some preliminary machine learning solutions; Section IV introduce the development of the heterogeneous OPC framework with preliminary experimental results followed by conclusion in Section V.

978-1-7281-4123-7/20/$31.00



(b)

Fig. 2.: (a) Conventional hotspot detection flow vs. (b) Region based hotspot detection flow as presented in [24].

II Hotspot Detection via Machine Learning

A *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 [20], support vector machine (SVM) [21, 22], artificial neural networks [21] and naive Bayes [23], which all follows a standard detection flow as in Fig. 2(a).

Ding *et al.* [21] introduce an SVM-based hotspot detection flow, which hierarchically narrows down the search space for hotspot patterns. Layout designs are converted into to feature space by capturing fragment-based features. [22] further enhances 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. [20] incorporates Adaboost and decision tree learner for efficient layout hotspot detection and exhibits good trade-off between detection accuracy and false positive penalty. Another representative ensemble learning framework is proposed in [23], where the information-theoretic approach is applied in the feature extraction module. The problem is solved by a dynamic programming model and embedded into the smooth boosting model with naive Bayes. The lithography simulation overhead is 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) is presented 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.

B *Deep Learning Solutions*

The fast development of deep neural networks brings new opportunities for hotspot detection solutions. Yang *et al.* [3] consider the limitation of conventional machine learning on scalability requirements for printability estimation and feature representation, a novel deep learning based hotspot detection model is proposed. A feature tensor extraction technology is approached to transform origin features into lower scale representations where spatial information is reserved. To facilitate the training procedure and find a better tradeoffs between accuracy and false alarm, a batch biased learning (BBL) is presented. BBL adjusts the bias for different instances dynamically which improve the model performance. The bias function is defined as:

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

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 is proposed in [4] to handle the 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 consistent with most manchine learning models. To ensure the layout represented by the squish pattern unchanged, the geometry information δ should be scaled and duplicated. To obtain satisfactory s to change the topology matrix to a desired size as well as attaining low variance δ, the problem can be formulated as

$$\min_{\boldsymbol{s}} \quad ||\boldsymbol{\delta}'||_\infty \tag{3a}$$

$$\text{s.t.} \quad \delta_i' = \delta_i / s_i, \forall i, \tag{3b}$$

$$s_i \in \mathbb{Z}^+, \forall i, \tag{3c}$$

$$\sum_i s_i = d, \tag{3d}$$

where the geometry information before and after scaling are denoted as $\boldsymbol{\delta}$ and $\boldsymbol{\delta}'$. Gradient vanishing problem dur-

ing the training is also considered and a specific residual convolution block is used to enhance the performance.

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 is 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

$$\mathcal{L}_{\Phi}(f) = \frac{1}{N_+ N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} \Phi\left(f\left(\boldsymbol{x}_i^+\right) - f\left(\boldsymbol{x}_j^-\right)\right), \quad (4)$$

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.* [24] solve this problem by enlarging the small clip into large regions (as depicted in Fig. 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

$$l_{loc}(l_i, l_i') = \begin{cases} \frac{1}{2}(l_i - l_i')^2, & \text{if } |l_i - l_i'| < 1, \\ |l_i - l_i'| - 0.5, & \text{otherwise}, \end{cases} \quad (5)$$

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

$$l_{hotspot}(h_i, h_i^{'}) = -(h_i \log h_i^{'} + h_i^{'} \log h_i), \quad (6)$$

where h_i is the prediction of the model and h_i' is the label. Compared to the deterministic classification flow, the performance in [24] got improved greatly.

C 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] studies the possibility of generating DRC-clean test layout patterns with a generative machine learning model called transforming convolutional auto-encoder (TCAE). Derived from transforming auto-encoder (TAE) [25], TCAE replaces 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 create diverse and DRC-clean patterns.

III 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 Fig. 3. In this section, we will discuss several machine learning-based alternatives that assist traditional mask optimization flow.

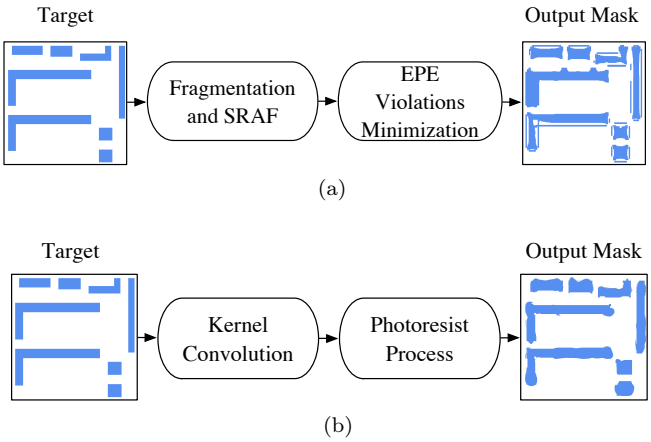


Fig. 3.: The flows of conventional OPC approaches: (a) model-based; (b) ILT-based.

A 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 has bee proposed. What's more, 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 are designed in [9]. The MPP framework has been demonstrated its effectiveness by being applied to a conventional mask optimization tool.

Existing machine learning models [26, 27, 12] can only perform pixel-wise or segment-wise mask calibration that is not computationally efficient. In accordance with the

critical problem, [11] proposes 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 a faster training procedure, an ILT-guided pre-training flow is proposed in [11] to initialize the generator with intermediate ILT results. Besides, the authors design new objectives of the discriminator to ensure the model is trained toward a target-mask mapping instead of a distribution. The new objective function is as follows:

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

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.

B 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 [28]. As a result, sub-resolution assist feature (SRAF) [29] insertion is 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 [30, 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 revise 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, $\boldsymbol{q}_t\in\mathbb{R}^s$ for discriminative sparse code of t-th input feature vector, $h_t\in\mathbb{R}$ for the label of input, $\boldsymbol{x}_t\in\mathbb{R}^s$ for sparse codes, $\boldsymbol{D}=\{\boldsymbol{d}_j\}_{j=1}^s, \boldsymbol{d}_j\in\mathbb{R}^n$ for the dictionary made up of atoms to encode input features, $\boldsymbol{A}\in\mathbb{R}^{s\times s}$ for a matrix transforming original sparse code $\boldsymbol{x}_t$ into discriminative sparse code, $\boldsymbol{W}\in\mathbb{R}^{1\times s}$ the related weight vector, α and β for the balancing hyper-parameters.

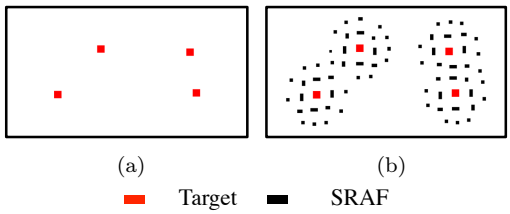


Fig. 4.: The visualization of the SRAF image translation in [14]: (a) Original layout with target contacts; (b) SRAFed layout.

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

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].

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] casts 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 Fig. 4. To achieve this formulation, Alawieh *et al.* firstly adopt conditional generative adversarial network (CGAN) in SRAF insertion. In addition, to fit CGAN training, a novel multi-channel heatmap encoding/decoding scheme is proposed to map layouts to images without information loss. The loss function is designed as Equation (8):

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

where $\boldsymbol{x}$ is an observed image, $\boldsymbol{y}$ an output image, $\boldsymbol{z}$ a random noise vector. G and D refer to the generator and discriminator in CGAN respectively. To further reduce blurring, the authors adopt $L1$-norm rather than $L2$-nom. With comparable lithographic performance, GAN-SRAF framework surpasses prior works on insertion speed.

C 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 [31].

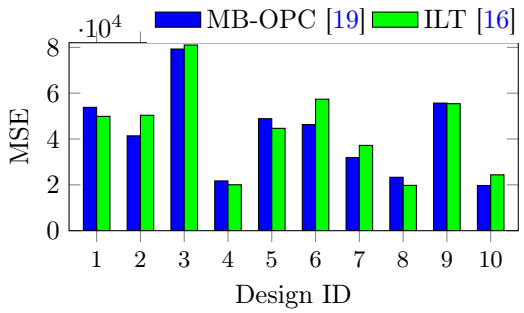


Fig. 5.: Performance gap between model-based OPC and ILT on ten designs from ICCAD2013 CAD Contest [33].

[32] is a pioneer work that considers multiple exposure effects in ILT framework. To automatically synthesize the masks and then print the desired wafer pattern, [32] first combines 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 [32] is shown in Equation (9), which is formulated as minimizing the $L2$-norm of the difference between the desired pattern $\boldsymbol{z}^*$ and the aerial image $|\boldsymbol{Ha}|^2 + |\boldsymbol{Hb}|^2$. $\boldsymbol{H}$ is a jinc function with cutoff frequency NA/λ, and $\boldsymbol{a}$, $\boldsymbol{b}$ are sampled from two input masks.

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

However, [32] has not addressed the layout decomposition problem yet. Ma *et al.* firstly develops 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.

IV 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. Fig. 5 depicts the performance gap of two engines on ten designs from ICCAD2013 CAD Contest [33]. 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.

TABLE I
: Evaluation of the proposed H-OPC.

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

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 as used to train GAN-OPC [11] which are fed into an ILT engine [16] and a model-based OPC [19]. Each design in the training set is labeled according to which OPC engine behaves best. For the classification neural networks, we use the same architecture as in [3]. Layout images are also converted to DCT format accordingly.

We evaluate the proposed framework using ten designs from ICCAD2013 CAD Contest [33]. 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 I, where "MB-OPC", "ILT" and "H-OPC" list the results of model-based OPC, inverse lithography technique-based OPC and the proposed heterogeneous OPC respectively. 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 proposed 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.

V Conclusion and Discussion

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.

ACKNOWLEDGMENT

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

References

[1] D. Z. Pan, B. Yu, and J.-R. Gao, "Design for manufacturing with emerging nanolithography," *IEEE TCAD*, vol. 32, no. 10, pp. 1453–1472, 2013.

[2] P. De Bisschop, "Optical proximity correction: a cross road of data flows," *Japanese Journal of Applied Physics*, vol. 55, no. 6S1, p. 06GA01, 2016.

[3] H. Yang, J. Su, Y. Zou, Y. Ma, B. Yu, and E. F. Y. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," *IEEE TCAD*, vol. 38, no. 6, pp. 1175–1187, 2019.

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

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

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

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

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

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

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

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

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

[13] X. Xu, Y. Lin, M. Li, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, and D. Z. Pan, "Subresolution assist feature generation with supervised data learning," *IEEE TCAD*, vol. 37, no. 6, pp. 1225–1236, 2017.

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

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

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

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

[18] Y.-H. Su, Y.-C. Huang, L.-C. Tsai, Y.-W. Chang, and S. Banerjee, "Fast lithographic mask optimization considering process variation," *IEEE TCAD*, vol. 35, no. 8, pp. 1345–1357, 2016.

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

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

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

[22] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, and C. Chiang, "Machine-learning-based hotspot detection using topological classification and critical feature extraction," *IEEE TCAD*, vol. 34, no. 3, pp. 460–470, 2015.

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

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

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

[26] R. Luo, "Optical proximity correction using a multilayer perceptron neural network," *Journal of Optics*, vol. 15, no. 7, p. 075708, 2013.

[27] A. Gu and A. Zakhor, "Optical proximity correction with linear regression," *IEEE TSM*, vol. 21, no. 2, pp. 263–271, 2008.

[28] C. A. Mack, "Scattering bars," *Solid State Technology*, 2003.

[29] C. H. Wallace, P. A. Nyhus, and S. S. Sivakumar, "Subresolution assist features," Dec. 15 2009, US Patent.

[30] X. Xu, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, and D. Z. Pan, "A machine learning based framework for sub-resolution assist feature generation," in *Proc. ISPD*, 2016, pp. 161–168.

[31] B. Yu, K. Yuan, D. Ding, and D. Z. Pan, "Layout decomposition for triple patterning lithography," *IEEE TCAD*, vol. 34, no. 3, pp. 433–446, March 2015.

[32] A. Poonawala and P. Milanfar, "Double-exposure mask synthesis using inverse lithography," *JM3*, vol. 6, no. 4, pp. 043 001–043 001, 2007.

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