SRAF Insertion via Supervised Dictionary Learning

Hao Geng, Wei Zhong, Haoyu Yang, Yuzhe Ma, Joydeep Mitra, and Bei Yu

***Abstract*—In modern VLSI design flow, sub-resolution assist feature (SRAF) insertion is one of the resolution enhancement techniques (RETs) to improve chip manufacturing yield. With aggressive feature size continuously scaling down, layout feature learning becomes extremely critical. In this paper, for the first time, we enhance conventional manual feature construction, by proposing a supervised online dictionary learning algorithm for simultaneous feature extraction and dimensionality reduction. By taking advantage of label information, the proposed dictionary learning framework can discriminatively and accurately represent the input data. We further consider SRAF design rules in a global view, and design two integer linear programming models in the post-processing stage of SRAF insertion framework. Experimental results demonstrate that, compared with a state-of-the-art SRAF insertion tool, our framework not only boosts the performance of the machine learning model, but also improves the mask optimization quality in terms of edge placement error (EPE) and process variation (PV) band area.**

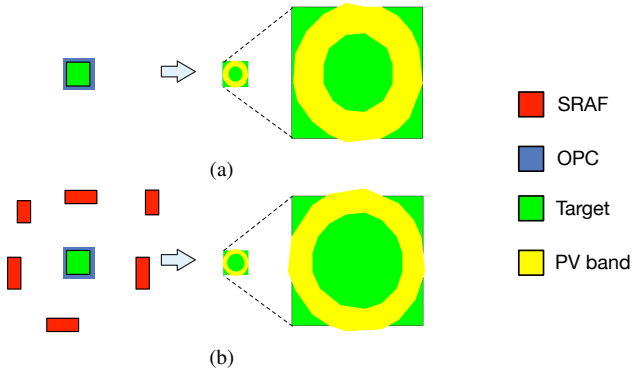


Fig. 1 (a) Printing with OPC only (2688 nm^2 PV band area); (b) Printing with both OPC and SRAF (2318 nm^2 PV band area).

I. Introduction

As feature size of semiconductors enters nanometer era, lithographic process variations are emerging as more severe issues in chip manufacturing process. That is, these process variations may result in manufacturing defects and a decrease of yield. Besides some design for manufacturability (DFM) approaches such as multiple patterning and litho-friendly layout design [1], [2], a de facto solution alleviating variations is mask optimization through various resolution enhancement techniques (RETs) (e.g. [3], [4]). Optical proximity correction (OPC) is the most successful representative strategy among numerous RETs, which aims at compensating lithography proximity effects by correcting mask pattern shapes and inserting assist features.

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 [5]. As a result, sub-resolution assist feature (SRAF) [6] insertion was proposed. Without printing SRAF patterns themselves, the small SRAF patterns can transfer light to the positions of target patterns, and therefore SRAFs are able to improve the robustness of the target patterns under different lithographic variations. A lithographic simulation example demonstrating the benefit of SRAF insertion is illustrated in Fig. 1. Here process variation (PV) band [7] (i.e. yellow circuit) area is applied to measure the performance of lithographic process window. As a matter of fact, the smaller area of the PV band, the better printing performance. In Fig. 1(a), only OPC is conducted to improve the printability of the target pattern, while in Fig. 1(b) both SRAF insertion and OPC are applied. We can see that, through SRAF insertion, the PV band area of the printed target pattern is effectively reduced from 2688 nm^2 as in Fig. 1(a) to 2318 nm^2 as in Fig. 1(b).

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 [8], model-based approach [9], and machine learning-based approach [7]. Rule-based approach is able to achieve high performance on simple designs, but it cannot handle complicated target patterns. Although model-based approach has a better performance, it is unfortunately very time-consuming. Recently, Xu *et al*. investigated an SRAF insertion framework based on machine learning techniques [7]. After trained on the training data set, the machine learning model draws inferences that can guide SRAF insertion on testing data. However, on account of coarse feature extraction techniques and lack of a global view of SRAF designs, the lithographic simulation results may not be good enough.

In a learning-based SRAF insertion flow, firstly, features are extracted from raw layout clips. One of the key take-aways of previous arts is the importance of features extracted from clips that leverage prior gained knowledge to achieve expected results. Namely, with more representative, generalized and discriminative layout features, the calibrated model can perform better. In this paper, we argue that the label information utilized in learning stage can be further imposed in feature extraction stage, which in turn will benefit the learning counterpart. In accordance with this argument, we

The preliminary version has been presented at the IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC) in 2019. This work is supported in part by The Research Grants Council of Hong Kong SAR (Project No. CUHK24209017).

H. Geng, H. Yang, Y. Ma and B. Yu are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, NT, Hong Kong.

W. Zhong is with International School of Information Science and Engineering, Dalian University of Technology, China.

J. Mitra is with the Cadence Design Systems, San Jose, CA, USA.

propose a supervised online dictionary learning algorithm, which converts features from a high-dimension space into a low-dimension space with label information integrated into the feature representations at the same time. To the best of our knowledge, this is the first layout feature extraction work seamlessly combining with label information. There is no prior art in applying the dictionary learning techniques or further supervised dictionary approaches into SRAF insertion issue. Our main contributions are listed as follows.

- Leverage supervised online dictionary learning algorithm to handle a large amount of layout patterns.
- Our proposed feature is more discriminative and representative, and is embedded into SRAF insertion framework.
- The SRAF insertion with design rules is modeled as an integer linear programming (ILP) problem, and two ILP models are proposed.
- Experimental results show that our method not only boosts the performance of the machine learning model, but also improves the mask optimization quality.

The rest of this paper is organized as following. Section II introduces some preliminaries on metrics, problem formulation in the paper and illustrates the whole working flow of our framework to insert SRAFs. Section III describes the specific feature extraction method, and our supervised online dictionary learning algorithm. Section IV reveals two ILP models in post-processing stage. Section V presents the experiment results, followed by the conclusion in Section VI.

II. Preliminary

A. Problem Formulation

Given a machine learning model, F_1 score is used to measure its accuracy. Specifically, the higher, the better. In addition, we employ PV band area and edge placement error (EPE) to quantify lithographic simulation results.

Definition 1 (F_1 Score [10]). *F_1 score is the harmonic mean of two metrics, precision and recall. Precision is the number of true positive results divided by the number of all positive results, while recall is the ratio between number of true positive results and the number of positive results should be returned.*

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{1}$$

Definition 2 (PV Band). *Given a lithography simulation contours at a set of process conditions, the process variation (PV) band is the area between the outer contour and inner contour.*

Definition 3 (EPE [7]). *Given a lithography simulation contour at the nominal condition, the edge placement error (EPE) is defined as the displacement between the target pattern contour and the nominal contour.*

Based on the above metrics, we define the SRAF insertion problem as follows.

Problem 1 (SRAF Insertion). *Given a training set of layout clips and specific SRAF design rules, the objective of SRAF insertion is to place SRAFs in the testing set of layout clips such that the corresponding PV band and the EPE are minimized.*

B. Overall Flow

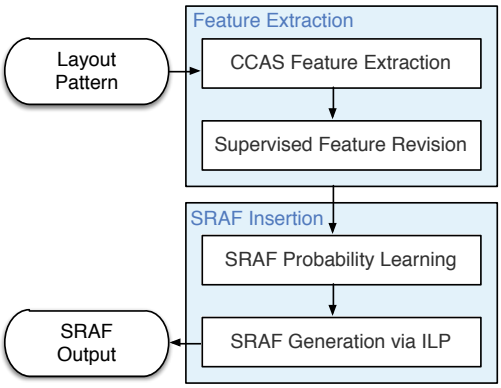


Fig. 2 The proposed SRAF generation flow.

The overall flow of our proposed SRAF insertion is shown in Fig. 2, which consists of two stages: feature extraction and SRAF insertion. In the feature extraction stage, after feature extraction via concentric circle area sampling (CCAS), we propose supervised feature revision, namely, mapping features into a discriminative low-dimension space. A dictionary consists of atoms which are the representatives of the original features. Sparsely encoded over a well-trained dictionary, the original features are described as combinations of atoms. Due to space transformation, the new features (i.e. sparse codes) are more abstract and discriminative with negligible information loss for classification. Therefore, proposed supervised feature revision is expected to avoid over-fitting of a machine learning model. In the second stage, based on the predictions inferred by learning model, SRAF insertion can be treated as a mathematical optimization problem accompanied by taking design rules into consideration.

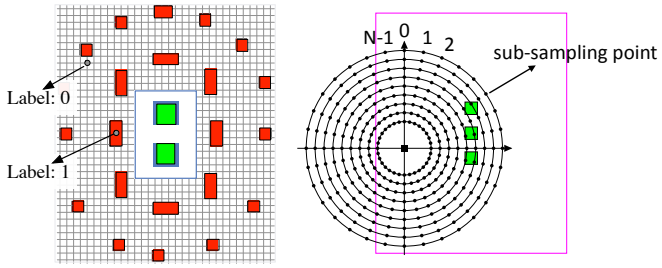
III. Feature Extraction

In this section, we firstly introduce the CCAS feature extraction method, and supervised feature revision. By the end, we give the details about our supervised online dictionary learning algorithm and corresponding analysis.

A. CCAS Feature Extraction

With considering concentric propagation of diffracted light from mask patterns, recently proposed CCAS [11] layout feature is used in SRAF generation domain.

In SRAF insertion, the raw training data set is made up of a set of layout clips which include a set of target patterns and model-based SRAFs. Each layout clip is put on a 2-D grid plane with a specific grid size so that real training samples can be extracted via CCAS method at each grid. For every sample, according to the model-based SRAFs, the corresponding label is either "1" or "0". As Fig. 3(a) illustrates, "1" means inserting an SRAF at this grid, while "0" indicates there is no



(a) (b)

Fig. 3 (a) SRAF label; (b) CCAS for feature extraction in machine learning model-based SRAF generation.

assist feature. Fig. 3(b) shows the CCAS feature extraction method.

However, since adjacent circles contain similar information, the CCAS feature has much redundancy. In fact, the redundancy will hinder the fitting of a machine learning model.

B. Supervised Feature Revision

With CCAS features as inputs, the dictionary learning model is expected to output the discriminative feature of low-dimension. In the topic of data representation [12], a self-adaptive dictionary learning model can sparsely and accurately represent data as linear combinations of atoms (i.e., columns) from a dictionary matrix. This model reveals the intrinsic characteristics of raw data.

In recent arts, sparse decomposition and dictionary construction are coupled in a self-adaptive dictionary learning framework. As a result, the framework can be modeled as an unconstrained optimization problem. The joint objective function of a self-adaptive dictionary model for feature revision problem is proposed as Equation (2):

$$\min_{\boldsymbol{x},\boldsymbol{D}} \frac{1}{N}\sum_{t=1}^{N}\{\frac{1}{2}\left\|\boldsymbol{y}_t-\boldsymbol{D}\boldsymbol{x}_t\right\|_2^2+\lambda\left\|\boldsymbol{x}_t\right\|_p\}, \tag{2}$$

where $\boldsymbol{y}_t \in \mathbb{R}^n$ is an input CCAS feature vector, and $\boldsymbol{D}=\{\boldsymbol{d}_j\}_{j=1}^{s}, \boldsymbol{d}_j \in \mathbb{R}^n$ represents the dictionary made up of atoms to encode input features. $\boldsymbol{x}_t \in \mathbb{R}^s$ denotes sparse codes (i.e. sparse decomposition coefficients) with p referring to the type of norm. Meanwhile, N is the total number of training data vectors in memory. The above equation, illustrated in Fig. 4, consists of a series of reconstruction error, $\left\|\boldsymbol{y}_t-\boldsymbol{D}\boldsymbol{x}_t\right\|_2^2$, and a regularization term $\left\|\boldsymbol{x}_t\right\|_p$. In Fig. 4, every grid represents a numerical value, and dark grids of $\boldsymbol{x}_t$ indicate zero. It can be seen that the motivation of dictionary learning is to sparsely encode input CCAS features over a well-trained dictionary.

1) Latent Supervised Information: However, from Equation (2), it is easy to discover that the main optimization goal is minimizing the reconstruction error in a mean squared sense, which may not be compatible with the goal of classification. Therefore, we try to explore the latent label information, and then propose our joint objective function as Equation (3). As aforementioned, a label indicates whether a grid is occupied with an SRAF or not. Here, an assumption

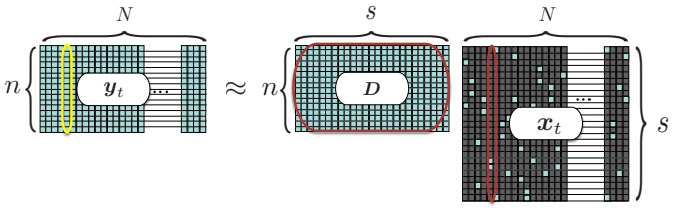


Fig. 4 The overview of dictionary learning.

has been made in advance that every atom is associated with a particular label. It is true because in initialization, atoms are trained by CCAS features for each class.

$$\min_{\boldsymbol{x},\boldsymbol{D},\boldsymbol{A}} \frac{1}{N}\sum_{t=1}^{N}\{\frac{1}{2}\left\|\left(\boldsymbol{y}_t^\top,\sqrt{\alpha}\boldsymbol{q}_t^\top\right)^\top-\begin{pmatrix}\boldsymbol{D}\\ \sqrt{\alpha}\boldsymbol{A}\end{pmatrix}\boldsymbol{x}_t\right\|_2^2+\lambda\left\|\boldsymbol{x}_t\right\|_p\}. \tag{3}$$

In Equation (3), α is a hyper-parameter balancing the contribution of each part to reconstruction error. $\boldsymbol{q}_t \in \mathbb{R}^s$ is defined as discriminative sparse code of t-th input feature vector. Thus, $\boldsymbol{A} \in \mathbb{R}^{s\times s}$ transforms original sparse code $\boldsymbol{x}_t$ into discriminative sparse code. Filled with constants ("0.0" or "1.0"), $\boldsymbol{q}_t$ reflects the categorical relationship between corresponding atoms and t-th input. For example, "1.0" means the input shares the same label with the corresponding atom, while "0.0" vice versa. Given the input and the dictionary $\boldsymbol{D}$, it is obvious that $\boldsymbol{q}_t$ is not undetermined. On the contrary, it has some fixed types and once an input is given, one type is selected.

For example, assume $\boldsymbol{D}=\{\boldsymbol{d}_1,\boldsymbol{d}_2,\boldsymbol{d}_3,\boldsymbol{d}_4\}$ with $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ from class 1, $\boldsymbol{d}_3$ and $\boldsymbol{d}_4$ from class 2, then $\boldsymbol{q}_t$ for the corresponding input $\boldsymbol{y}_t$ is supposed to be either $(1.0,1.0,0.0,0.0)^\top$ or $(0.0,0.0,1.0,1.0)^\top$. For further explanation, we merge different types of $\boldsymbol{q}_t$ as a $\boldsymbol{Q}$ matrix:

$$\boldsymbol{Q}=\begin{pmatrix}\boldsymbol{q}_1, & \boldsymbol{q}_2\end{pmatrix}=\begin{pmatrix}1.0 & 0.0\\ 1.0 & 0.0\\ 0.0 & 1.0\\ 0.0 & 1.0\end{pmatrix}, \tag{4}$$

where $(1.0,1.0,0.0,0.0)^\top$ means input sample shares the same label with $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$, and $(0.0,0.0,1.0,1.0)^\top$ indicates that the input, $\boldsymbol{d}_3$ and $\boldsymbol{d}_4$ are from the same class.

To illustrate physical meaning of Equation (3) clearly, we can also rewrite it via splitting the reconstruction term into two terms within l_2-norm as Equation (5):

$$\min_{\boldsymbol{x},\boldsymbol{D},\boldsymbol{A}} \frac{1}{N}\sum_{t=1}^{N}\{\frac{1}{2}\left\|\boldsymbol{y}_t-\boldsymbol{D}\boldsymbol{x}_t\right\|_2^2+\frac{\alpha}{2}\left\|\boldsymbol{q}_t-\boldsymbol{A}\boldsymbol{x}_t\right\|_2^2+\lambda\left\|\boldsymbol{x}_t\right\|_p\}. \tag{5}$$

The first term $\left\|\boldsymbol{y}_t-\boldsymbol{D}\boldsymbol{x}_t\right\|_2^2$ is still the reconstruction error term. The second term $\left\|\boldsymbol{q}_t-\boldsymbol{A}\boldsymbol{x}_t\right\|_2^2$ represents discriminative error, which imposes a constraint on the approximation of $\boldsymbol{q}_t$. As a result, the input CCAS features from same class share quite similar representations.

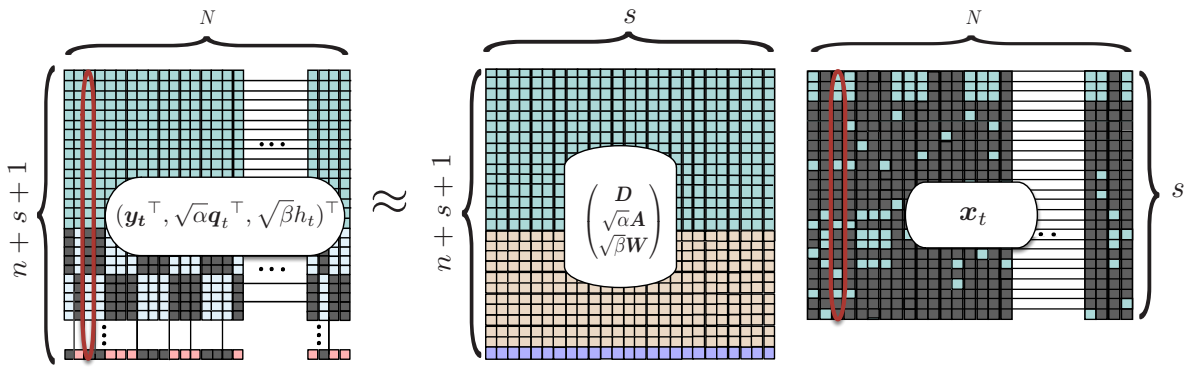


Fig. 5 The illustration of supervised dictionary learning.

2) Direct Supervised Information: Since the latent class information has been exploited, the label information can also be directly harnessed. After adding the prediction error term into initial objective function Equation (3), we propose our final joint objective function as Equation (6):

$$\min_{\boldsymbol{x},\boldsymbol{D},\boldsymbol{A},\boldsymbol{W}} \frac{1}{N}\sum_{t=1}^{N}\{\frac{1}{2}\left\|\left(\boldsymbol{y}_t^\top, \sqrt{\alpha}\boldsymbol{q}_t^\top, \sqrt{\beta}h_t\right)^\top - \begin{pmatrix} \boldsymbol{D} \\ \sqrt{\alpha}\boldsymbol{A} \\ \sqrt{\beta}\boldsymbol{W} \end{pmatrix}\boldsymbol{x}_t\right\|_2^2 + \lambda\|\boldsymbol{x}_t\|_p\}, \tag{6}$$

where $h_t \in \mathbb{R}$ is the label with $\boldsymbol{W} \in \mathbb{R}^{1\times s}$ the related weight vector, and therefore $\|h_t - \boldsymbol{W}\boldsymbol{x}_t\|_2^2$ refers to the classification error. α and β are hyper-parameters which control the contribution of each term to reconstruction error and balance the trade-off. It can be seen that Equation (6) restricts the representation of original data. Furthermore, it is expected to let the original data vectors from the same class share similar representations, thus benefits the calibration of a machine learning model. The illustration for supervised dictionary learning is shown in Fig. 5, where the supervised information is appended to the end of the unsupervised data vector and new feature vectors from same class are similar in terms of structure.

The discriminative property in low-dimension of the new feature is demonstrated in Fig. 6, where blue color refers to the non-zero number. More than 1000 CCAS features and their corresponding new features are selected as exemplars. Fig. 6(a) shows the original CCAS feature which is sparse due to benchmark layouts. The dimensionality of the CCAS features is over 1000. On the other hand, as shown in Fig. 6(b), the new features are in low-dimension space (dimensionality only 200) meanwhile they can be roughly divided into two classes.

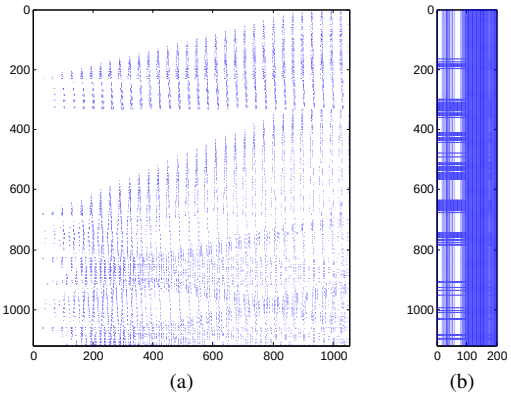


Fig. 6 Feature map comparison: (a) CCAS based; (b) The same features after supervised feature revision.

C. Online Algorithm

Recently, some attempts which explore label information are proposed in succession such as Discriminative K-SVD [13], kernelized supervised dictionary learning [14], label consistent K-SVD [15] (LCK-SVD) dictionary learning and supervised K-SVD with dual-graph constraints [16].

However, most of them are based on K-SVD [17] which belongs to batch-learning method. They are not suitable for dealing with large dataset since the computation overhead (e.g. computing the inverse of a very large matrix) may be high. Online learning method applied in dictionary learning model [18], [19] is a good idea, yet these algorithms are unsupervised.

Therefore, we develop an efficient online learning method, which seamlessly combines aforementioned supervised dictionary learning. Unlike the batch approaches, online approaches process training samples incrementally, one training sample (or a small batch of training samples) at a time, similarly to stochastic gradient descent.

According to our proposed formulation (i.e. Equation (6)), the joint optimization of both dictionary and sparse codes is non-convex, but sub-problem with one variable fixed is convex. Hence, Equation (6) can be divided into two convex sub-problems. Note that, in a taste of linear algebra, our new input with label information, i.e. $\left(\boldsymbol{y}_t^\top, \sqrt{\alpha}\boldsymbol{q}_t^\top, \sqrt{\beta}h_t\right)^\top$ in Equation (6), can be still regarded as the original $\boldsymbol{y}_t$ in Equation (2). So is the new merged dictionary consisting of $\boldsymbol{D}$, $\boldsymbol{A}$ and $\boldsymbol{W}$. For simplicity of description and derivation, in following analysis, we will use $\boldsymbol{y}_t$ referring to $\left(\boldsymbol{y}_t^\top, \sqrt{\alpha}\boldsymbol{q}_t^\top, \sqrt{\beta}h_t\right)^\top$ and $\boldsymbol{D}$ standing for merged dictionary with $\boldsymbol{x}$ as the sparse codes.

The two stages, i.e. sparse coding and dictionary constructing, will be alternatively performed in iterations. Thus, in t-th iteration, the algorithm firstly draws the input sample $\boldsymbol{y}_t$ or

a mini-batch over the current dictionary $\boldsymbol{D}_{t-1}$ and obtains the corresponding sparse codes $\boldsymbol{x}_t$. Then use two updated auxiliary matrices, $\boldsymbol{B}_t$ and $\boldsymbol{C}_t$ to help compute $\boldsymbol{D}_t$.

The objective function for sparse coding is showed in Equation (7):

$$\boldsymbol{x}_t \triangleq \arg\min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{y}_t - \boldsymbol{D}_{t-1}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1. \tag{7}$$

If the regularizer adopts l_0-norm, solving Equation (7) is NP-hard. Although there exist some classical algorithms like matching pursuit (MP) [20] and orthogonal matching pursuit (OMP) [21] to address problem with l_0-norm regularizer, nowadays, adopting l_1-norm is much more popular. The reason is that l_1-norm is a convex replacement of l_0-norm, and thus many more fancy optimization algorithms can be exploited. In statistical community, Equation (7) is a model called Lasso [22], which is short for the least absolute shrinkage and selection operator. As we can see in Equation (7), Lasso is a penalized least squares technique that puts l_1 constraint on the estimated regression coefficients. It is mainly based on the following two concepts. One is that l_1 regularization approximates l_0 regularization. The other is that shrinkage operation improves prediction performance. Lasso can be solved by coordinate descent [23], [24] and some other first-order algorithms which are critical in large-scale machine learning and deep learning [25]. For example, fast iterative shrinkage-threshold algorithm (FISTA) [26] is a fancy Lasso solver, which achieves the optimal convergence rate of first-order algorithms. Although various variants exist, gradient descent and mirror descent are the foundations of first-order methods [27].

Two auxiliary matrices $\boldsymbol{B}_t \in \mathbb{R}^{(n+s+1)\times s}$ and $\boldsymbol{C}_t \in \mathbb{R}^{s\times s}$ are defined respectively in Equation (8) and Equation (9):

$$\boldsymbol{B}_t \leftarrow \frac{t-1}{t}\boldsymbol{B}_{t-1} + \frac{1}{t}\boldsymbol{y}_t\boldsymbol{x}_t^\top, \tag{8}$$

$$\boldsymbol{C}_t \leftarrow \frac{t-1}{t}\boldsymbol{C}_{t-1} + \frac{1}{t}\boldsymbol{x}_t\boldsymbol{x}_t^\top. \tag{9}$$

The objective function for dictionary construction is:

$$\boldsymbol{D}_t \triangleq \arg\min_{\boldsymbol{D}} \frac{1}{t}\sum_{i=1}^{t}\{\frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}_i\|_2^2 + \lambda\|\boldsymbol{x}_i\|_1\}. \tag{10}$$

Algorithm 1 summarizes the algorithm details of the proposed supervised online dictionary learning (SODL) algorithm. First, in initialization stage, we randomly pick some data of two categories from training dataset as inputs $\boldsymbol{Y}$. To initialize $\boldsymbol{D}$, several iterations of K-SVD [17] for each class are computed, and then all the outputs of each K-SVD are combined. As a result, atoms are naturally allocated to corresponding classes. After initialization of $\boldsymbol{D}$, initial $\boldsymbol{Q}$ is ready. For initialization of $\boldsymbol{A}$, we adopt ridge regression model [28], as showed in Equation (11):

$$\arg\min_{\boldsymbol{A}} \|\boldsymbol{Q} - \boldsymbol{A}\boldsymbol{X}\| + \lambda_2\|\boldsymbol{A}\|_2^2. \tag{11}$$

Similarly, for initial $\boldsymbol{W}$, the ridge regression model is again exploited. Then, we use coordinate descent algorithm [23], [24] as the solving scheme to Equation (7) (line 3). To accelerate the convergence speed, Equation (10) involves the computations of past signals $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t$ and the sparse codes $\boldsymbol{x}_1, ..., \boldsymbol{x}_t$. One way to efficiently update dictionary is that introduce some sufficient statistics, i.e. $\boldsymbol{B}_t \in \mathbb{R}^{(n+s+1)\times s}$ (line 4) and $\boldsymbol{C}_t \in \mathbb{R}^{s\times s}$ (line 5), into Equation (10) without directly storing the past information, namely input data sample $\boldsymbol{y}_i$ and corresponding sparse codes $\boldsymbol{x}_i$ for $i \leq t$. These two auxiliary variables play important roles in updating atoms, which aggregates the past information during computations. We further exploit block coordinate method with warm start [24] to resolve Equation (10) (line 6). As a result, through some gradient calculations, we bridge the gap between Equation (10) and sequentially updating atoms based on Equations (12) and (13).

Algorithm 1 Supervised Online Dictionary Learning (SODL)

Require: Input merged features $\boldsymbol{Y} \leftarrow \{\boldsymbol{y}_t\}_{t=1}^N, \boldsymbol{y}_t \in \mathbb{R}^{(n+s+1)}$ (including original CCAS features, discriminative sparse code $\boldsymbol{Q} \leftarrow \{\boldsymbol{q}_t\}_{t=1}^N, \boldsymbol{q}_t \in \mathbb{R}^s$ and label information $\boldsymbol{H} \leftarrow \{h_t\}_{t=1}^N, h_t \in \mathbb{R}$).

Ensure: New features $\boldsymbol{X} \leftarrow \{\boldsymbol{x}_t\}_{t=1}^N, \boldsymbol{x}_t \in \mathbb{R}^s$, dictionary $\boldsymbol{D} \leftarrow \{\boldsymbol{d}_j\}_{j=1}^s, \boldsymbol{d}_j \in \mathbb{R}^{(n+s+1)}$.

1: **Initialization**: Initial merged dictionary $\boldsymbol{D}_0, \boldsymbol{d}_j \in \mathbb{R}^{(n+s+1)}$ (including initial transformation matrix $\boldsymbol{A}_0 \in \mathbb{R}^{s\times s}$ and initial label weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{1\times s}$), $\boldsymbol{C}_0 \in \mathbb{R}^{s\times s} \leftarrow \boldsymbol{0}$, $\boldsymbol{B}_0 \in \mathbb{R}^{(n+s+1)\times s} \leftarrow \boldsymbol{0}$;
2: **for** $t \leftarrow 1$ to N **do**
3: Sparse coding $\boldsymbol{y}_t$ and obtaining $\boldsymbol{x}_t$; ▷ Equation (7)
4: Update auxiliary variable $\boldsymbol{B}_t$; ▷ Equation (8)
5: Update auxiliary variable $\boldsymbol{C}_t$; ▷ Equation (9)
6: Update dictionary $\boldsymbol{D}_t$; ▷ Algorithm 2;
7: **end for**

$$\boldsymbol{u}_j \leftarrow \frac{1}{\boldsymbol{C}[j,j]}(\boldsymbol{b}_j - \boldsymbol{D}\boldsymbol{c}_j) + \boldsymbol{d}_j. \tag{12}$$

$$\boldsymbol{d}_j \leftarrow \frac{1}{\max\left(\|\boldsymbol{u}_j\|_2, 1\right)}\boldsymbol{u}_j. \tag{13}$$

For each atom $\boldsymbol{d}_j$, the updating rule is illustrated in Algorithm 2. In Equation (12), $\boldsymbol{D}_{t-1}$ is selected as the warm start of $\boldsymbol{D}$. $\boldsymbol{b}_j$ indicates the j-th column of $\boldsymbol{B}_t$, while $\boldsymbol{c}_j$ is the j-th column of $\boldsymbol{C}_t$. $\boldsymbol{C}[j,j]$ denotes the j-th element on diagonal of $\boldsymbol{C}_t$. Equation (13) is an l_2-norm constraint on atoms to prevent atoms becoming arbitrarily large (which may lead to arbitrarily small sparse codes). [29] proves that in the stage of constructing dictionary, the convex optimization problem allowing separable constraints in the updated blocks (columns) will guarantee the convergence to a global optimum.

D. Convergence Analysis

Proposed algorithm deals the non-convex optimization problem in an alternative framework. As a result, our algorithm converges to a stationary point of the objective function, while finding the global optimum is not guaranteed [30]. In fact, for practical applications, stationary points are enough empirically.

Algorithm 2 Rules for Updating Atoms

Require: $\boldsymbol{D}_{t-1} \leftarrow \{\boldsymbol{d}_j\}_{j=1}^{s}, \boldsymbol{d}_j \in \mathbb{R}^{(n+s+1)}$,
$\boldsymbol{B}_t \leftarrow \{\boldsymbol{b}_j\}_{j=1}^{s}, \boldsymbol{b}_j \in \mathbb{R}^{(n+s+1)}$,
$\boldsymbol{C}_t \leftarrow \{\boldsymbol{c}_j\}_{j=1}^{s}, \boldsymbol{c}_j \in \mathbb{R}^{s}$.
Ensure: dictionary $\boldsymbol{D}_t \leftarrow \{\boldsymbol{d}_j\}_{j=1}^{s}, \boldsymbol{d}_j \in \mathbb{R}^{(n+s+1)}$.
1: **for** $j \leftarrow 1$ to s **do**
2: Update the j-th atom $\boldsymbol{d}_j$; ▷ Equations (12) and (13)
3: **end for**

Before our analysis, some definitions should be made in advance. The optimal value of the sparse coding problem is defined as Equation (14).

$$l\left(\boldsymbol{y}_i, \boldsymbol{D}\right) \triangleq \min_{\boldsymbol{x}} \{\frac{1}{2}\left\|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \lambda\left\|\boldsymbol{x}\right\|_1\}. \tag{14}$$

As illustrated in Equation (15), we denote the empirical cost function (whose value is supposed to be small) to check whether D is good at representing the input data.

$$f_t\left(\boldsymbol{D}\right) \triangleq \frac{1}{t}\sum_{i=1}^{t} l\left(\boldsymbol{y}_i, \boldsymbol{D}\right). \tag{15}$$

Actually, according to [31], the minimization of empirical cost $f_t\left(\boldsymbol{D}\right)$ with high accuracy is not the real spotlight. By contrast, the minimization of expected cost $f\left(\boldsymbol{D}\right)$, i.e. Equation (16) , is what we want to explore.

$$f\left(\boldsymbol{D}\right) \triangleq \mathbb{E}_{\boldsymbol{y}_t}\left[l\left(\boldsymbol{y}_t, \boldsymbol{D}\right)\right] = \lim_{t\to\infty} f_t\left(\boldsymbol{D}\right). \tag{16}$$

We define the surrogate function, $\hat{f}_t$ in Equation (17), for f_t. It can be seen that it upperbounds the empirical cost $f_t\left(\boldsymbol{D}_t\right)$.

$$\hat{f}_t\left(\boldsymbol{D}\right) \triangleq \frac{1}{t}\sum_{i=1}^{t}\{\frac{1}{2}\left\|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}_i\right\|_2^2 + \lambda\left\|\boldsymbol{x}_i\right\|_1\}. \tag{17}$$

The $\hat{f}_t\left(\boldsymbol{D}_t\right)$ and $f_t\left(\boldsymbol{D}_t\right)$ converge almost to the same limit and hence that $\hat{f}_t$ plays a role as the surrogate function for f_t. As a result, $\boldsymbol{D}_t$ is close to $\boldsymbol{D}_{t-1}$ for large value of t, which motivates us to use $\boldsymbol{D}_{t-1}$ as a warm start when computing $\boldsymbol{D}_t$.

Theorem 1. *Assume the surrogate function $\hat{f}_t$ are strictly convex with lower-bounded Hessians, when t goes infinity, the distance between the stationary points of proposed dictionary learning problem and dictionary $\boldsymbol{D}_t$ will converge to* 0.

The detailed proof is provided in Appendix B.

IV. SRAF Insertion

A. SRAF Probability Learning

After feature extraction via CCAS and revision through SODL framework, the discriminative, low-dimension features of grids on a layout are obtained. Therefore, a machine learning model is calibrated by new features of training set, and predicts the probabilities of inserting SRAF patterns into gridded testing layout. This procedure is named as SRAF Probability Learning, which not only guides the SRAF insertion, but also bridges the gap between the previous continuous optimization problem and the consequent discrete optimization issue. For fair comparison, we exploit the same classifier, logistic regression, as used in [7]. This widely used model harnesses the logistic function as the core to predict the probabilities for classes.

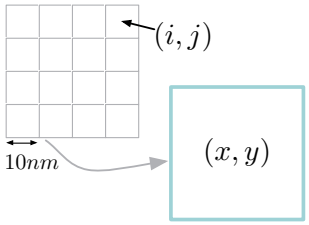


Fig. 7 SRAF grid model construction.

B. SRAF Insertion via ILP

Through SODL model and classifier, the probabilities of 2-D grids can be obtained. Based on design rules for the machine learning model, the label for a grid with probability less than the threshold is "0". It means that the grid will be ignored when doing SRAF insertion. However, in [7], the scheme to insert SRAFs is a little naive and greedy. Actually, combined with some SRAF design rules such as maximum length and width, minimum spacing, the SRAF insertion can be modeled as an integer linear programming problem. With ILP model to formulate SRAF insertion, we will obtain a global view of SRAF generation.

In the objective of the ILP approach, we only consider valid grids whose probabilities are larger than the threshold. The probability of each grid is denoted as $p(i,j)$, where i and j indicate the index of a grid. For simplicity, we merge the current small grids into new bigger grids, as shown in Fig. 7. Then we define $c(x,y)$ as the value of each merged grid, where x, y denote the index of a merged grid. The rule to compute $c(x,y)$ follows Equation (18).

$$c(x,y) = \begin{cases} \sum_{(i,j)\in(x,y)} p(i,j), & \text{if } \exists\ p(i,j) \geq \text{threshold}, \\ -1, & \text{if all } p(i,j) < \text{threshold}. \end{cases} \tag{18}$$

The motivation behind this approach is twofold. One is to speed up the ILP via reducing the problem size. Because we can pre-determine some decision variables whose values are negative. The other is to keep the consistency of predictions in SRAF probability learning.

Although the grid mergence may lead to some quality degradation, it is acceptable compared to the magnificent efficiency increase. The quality degradation is caused by the a few location deviations of SRAF patterns under grid models with different sizes. However, in SRAF insertion, the lithographic performance mainly depends on the number of SRAF patterns and SRAF pattern shapes, while few location deviations affects the lithographic performance minorly. On the other hand, according to the design rules, the minimum size of an SRAF pattern is $40nm \times 40nm$, and the size of a merged grid is also $40nm \times 40nm$. It is more natural to exploit grid mergence, and thus boosts the ILP. Otherwise, it would be time-consuming to find solutions for ILP since constraints are much more complicated.

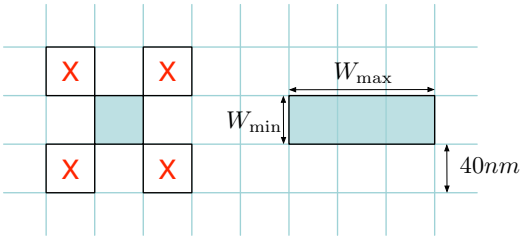


Fig. 8 Relaxed design rules for SRAF insertion under the grid model.

1) ILP with Relaxed Design Rules: In ILP for SRAF insertion, our real target is to maximize the total probability of valid grids with feasible SRAF insertion. Accordingly, it is manifest to put up with the objective function, which is to maximize the total value of merged grids. With considering relaxed design rules (e.g. relaxing maximum length from 90nm to 120nm), the ILP formulation is shown in Formula (19).

$$\max_{a(x,y)} \quad \sum_{x,y} c(x,y) \cdot a(x,y) \tag{19a}$$

$$\text{s.t.} \quad a(x,y) + a(x-1,y-1) \le 1, \quad \forall (x,y), \tag{19b}$$

$$a(x,y) + a(x-1,y+1) \le 1, \quad \forall (x,y), \tag{19c}$$

$$a(x,y) + a(x+1,y-1) \le 1, \quad \forall (x,y), \tag{19d}$$

$$a(x,y) + a(x+1,y+1) \le 1, \quad \forall (x,y), \tag{19e}$$

$$a(x,y) + a(x,y+1) + a(x,y+2) + a(x,y+3) \le 3, \quad \forall (x,y), \tag{19f}$$

$$a(x,y) + a(x+1,y) + a(x+2,y) + a(x+3,y) \le 3, \quad \forall (x,y), \tag{19g}$$

$$a(x,y) \in \{0,1\}, \quad \forall (x,y). \tag{19h}$$

Here $a(x,y)$ refers to the insertion situation at the merged grid (x,y). According to the rectangular shape of an SRAF and the spacing rule, the situation of two adjacent SRAFs on the diagonal is forbidden by Constraints (19b) to (19e); e.g. Constraint (19b) requires the $a(x,y)$ and the left upper neighbor $a(x-1,y-1)$ cannot be 1 at the same time, otherwise which will lead to the violation against design rules. Constraints (19f) to (19g) restrict the maximum length of SRAFs. The Fig. 8 actively illustrates these linear constraints coming from design rules.

2) ILP with General Design Rules: Previous case is a relaxed version. However, in most scenarios, more general design rules for inserting SRAFs are considered. Especially, the off-grid pattern is allowed. Now 90nm as maximum length for SRAFs and $40nm$ still the minimum length is taken into account, and thus we choose 5 most representative types for SRAF patterns which are shown in Fig. 9.

After considering the flexibility for more general rules, the new ILP model is proposed in Formula (20).

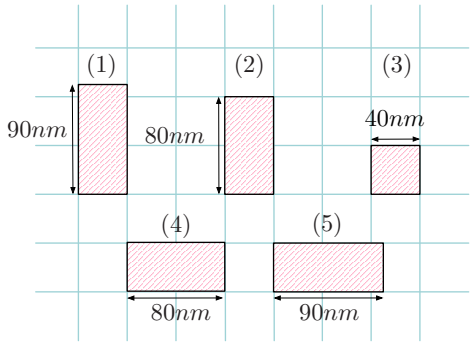


Fig. 9 Five basic types of SRAF patterns under general design rules.

$$\max_{a(x,y,i)} \quad \sum_{x,y,i} w_i \cdot a(x,y,i) \cdot v(x,y,i) \tag{20a}$$

$$\text{s.t.} \quad v(x,y,1) = c(x,y) + c(x-1,y) + 0.25 \cdot c(x-2,y), \quad \forall (x,y), \tag{20b}$$

$$v(x,y,2) = c(x,y) + c(x-1,y), \quad \forall (x,y), \tag{20c}$$

$$v(x,y,3) = c(x,y), \quad \forall (x,y), \tag{20d}$$

$$v(x,y,4) = c(x,y) + c(x,y+1), \quad \forall (x,y), \tag{20e}$$

$$v(x,y,5) = c(x,y) + c(x,y+1) + 0.25 \cdot c(x,y+2), \quad \forall (x,y), \tag{20f}$$

$$\sum_{i} a(x,y,i) \le 1, \quad \forall (x,y), \tag{20g}$$

$$a(x_1,y_1,i) + a(x_2,y_2,j) \le 1, \quad \forall (x_1,y_1,i),(x_2,y_2,j) \in \mathbb{C}, \tag{20h}$$

$$a(x,y,i) \in \{0,1\}, \quad \forall (x,y,i). \tag{20i}$$

The decision variable $a(x,y,i)$ is extended to three dimensions, which indicates whether the i-th type of SRAF pattern should be inserted at the merged grid (x,y). $v(x,y,i)$ means the contribution of inserting the i-th type into the merged grid (x,y) to the objective function, while w_i is the corresponding weight parameter. The values for w_1, w_2, w_3, w_4, w_5 are set to be 1.4, 0.1, 1.0, 0.1, 1.4 respectively, which indicates that SRAF patterns of 90nm in length are more encouraged to insert over patterns of 80nm. Since the regular SRAFs around the target pattern benefit lithography, we set the same value for w_1 and w_5, and the same principle of assignment works on w_2 and w_4. The different values for different SRAF patterns are defined in Constraints (20b) to (20f). Constraint (20g) sets the restriction that at most only one SRAF pattern can be inserted in one grid. We introduce a conflict set $\mathbb{C}$ in the formulation (see Constraint (20h)), and it contains the conflict pairs of SRAF patterns between one grid and its neighbors. Here, the "conflict" means the distance between two patterns violates the spacing design rule. For instance, $(x,y,1)$ and $(x-3,y+1,5)$ form one conflict pair, which is exemplified in Fig. 10. To address the problem efficiently, we can assume the main directions for inserting an SRAF pattern are up and right. Hence, under current spacing design rule, we only consider the partial neighborhood of the merged grid (x,y) (i.e. the

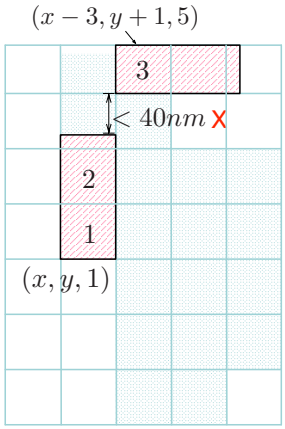


Fig. 10 The illustration of new ILP model.

TABLE I F_1 Score (%) Comparison with [7].

Benchmark	ISPD'16 [7]	SODL
Dense1	95.37	96.69
Dense2	94.77	97.00
Dense3	93.70	96.87
Dense4	93.89	96.49
Dense5	93.54	96.16
Dense6	93.02	96.86
Dense7	94.22	97.13
Dense8	93.24	96.85
Sparse1	90.51	93.62
Sparse2	87.65	94.03
Sparse3	85.75	91.68
Sparse4	85.56	93.35
Sparse5	85.69	90.48
Sparse6	84.65	91.57
Sparse7	85.00	93.01
Sparse8	84.05	92.72
Sparse9	84.71	90.21
Sparse10	84.03	92.60
Average	89.41	94.30
Ratio	1.000	**1.055**

grid labeled with "1" in Fig. 10), which is the area filled with small green dots and the grids labeled with "2" and "3". It can be seen that by introducing the concept of conflict set, the new ILP model is more flexible than previous one and able to to handle new spacing design rule.

V. Experimental Results

We implement the framework using `python` on an 8-core 3.7GHz Intel platform. To verify the effectiveness and the efficiency of our SODL algorithm, we employ the same benchmark set as applied in [7], which consists of 8 dense layouts and 10 sparse layouts with contacts sized $70nm$. The spacing for dense and sparse layouts are set to $70nm$ and $\geq 70nm$ respectively.

TABLE I and TABLE II compare our results with a state-of-the-art machine learning based SRAF insertion tool [7]. Column "Benchmark" lists all the test layouts in both tables. In TABLE I, Column "ISPD'16" refers to the performance of the machine learning framework in [7], while Column "SODL" denotes the results of our model. Within TABLE II, Columns "SODL+Greedy" corresponds to the results of our supervised online dictionary learning framework without ILP model in post-processing, while "SODL+ILP" and "SODL+NewILP" indicate the results from different ILP models with and without relaxed SRAF rules. Columns "F_1 score", "PV band", "EPE" and "CPU" are the evaluation metrics in terms of the learning model performance, the PV band area, the EPE, and the total runtime. Note that in "SODL+Greedy", the same greedy SRAF generation approach as in [7] is utilized.

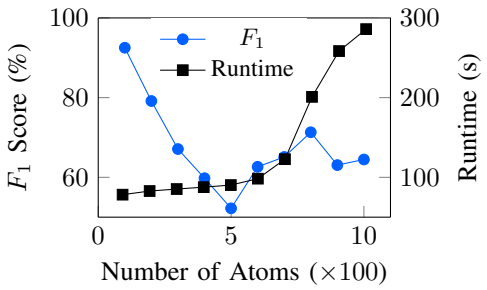


Fig. 11 The trend of changing number of atoms.

It can be seen from TABLE I that the SODL algorithm outperforms [7] in terms of F_1 score by 5.5%. This indicates the predicted SRAFs by our model match the reference results better than [7]. In other words, the proposed SODL based feature revision can efficiently improve machine learning model generality.

We exemplify the trends of runtime and F_1 score with respect to the changing of number of atoms, which is depicted in Fig. 11. With an increment in number of atoms, runtime ascends. Meanwhile, F_1 score goes down until number of atoms reaches a threshold. According to the theory of dictionary learning, the number of atoms is the dimension of the new feature. Dimension of features impacts the fitting of a machine learning model. Inappropriate dimensionality may cause under-fitting or over-fitting issues. That is the main reason why F_1 scores fluctuate. On the other hand, with the dimension increase of new features, the machine learning model suffers the running time issue. The fitting time of some machine learning models like SVM, increases non-linearly. One reason is the time complexities of algorithms. More importantly, the non-converge problem affects the running time.

We also feed the SRAFed layouts into Calibre [32] to go through a simulation flow that includes OPC and lithography simulation, which will generate printed contours under a given process window. The simulation results summarized in TABLE II show that we get better PV band and EPE results. In particular, after incorporating relaxed SRAF design rules with an ILP solution, proposed algorithm behaves even much better with an average PV band of $2.609\times10^{-3}\mu m^2$ and an average EPE of $0.774nm$ that surpass [7] with 2% less PV band and 3% less EPE. With considering general SRAF design rules in a new ILP model, our method also performs better in terms of 3.5% less PV band area and 11.8% less EPE within an acceptable runtime. The reason why the running time increases is that general design rules are much more complicated. For example, we need to consider the off-grid patterns. On the other hand, the neighborhood considered in

TABLE II Lithographic Performance Comparison with [7].

Benchmark	ISPD'16 [7]			SODL+Greedy			SODL+ILP			SODL+NewILP		
	PV band ($.001\mu m^2$)	EPE (nm)	CPU (s)	PV band ($.001\mu m^2$)	EPE (nm)	CPU (s)	PV band ($.001\mu m^2$)	EPE (nm)	CPU (s)	PV band ($.001\mu m^2$)	EPE (nm)	CPU (s)
Dense1	1.891	0.625	1.46	1.840	0.625	1.12	1.823	0.750	1.26	1.850	0.667	2.22
Dense2	1.960	0.313	1.35	2.033	0.500	1.06	2.003	0.438	1.20	1.987	0.438	2.03
Dense3	2.677	1.625	1.25	2.718	1.375	0.97	2.445	1.375	1.07	2.545	1.750	1.88
Dense4	2.426	1.313	1.47	2.288	1.125	1.13	2.459	1.625	1.29	2.363	1.125	2.21
Dense5	2.445	1.250	1.47	2.428	1.375	1.18	2.336	1.375	1.28	2.413	0.938	2.26
Dense6	2.933	0.750	1.17	2.871	1.000	0.91	2.886	0.250	1.00	2.538	0.000	1.78
Dense7	2.426	1.500	1.42	2.409	1.333	1.09	2.318	1.500	1.21	2.277	1.083	2.10
Dense8	2.354	1.417	1.40	2.436	1.417	1.10	2.366	1.167	1.20	2.445	1.000	2.14
Sparse1	2.937	0.438	2.61	2.866	0.563	2.04	2.803	0.375	2.18	2.813	0.500	4.01
Sparse2	2.870	0.625	7.02	2.872	0.516	5.25	2.873	0.594	5.63	2.803	0.625	10.44
Sparse3	2.882	0.556	14.07	2.911	0.535	10.74	2.829	0.528	11.50	2.764	0.563	21.77
Sparse4	2.896	0.566	23.81	2.891	0.496	18.44	2.830	0.547	19.66	2.785	0.547	37.00
Sparse5	2.889	0.565	28.96	2.931	0.571	23.18	2.850	0.580	24.80	2.799	0.633	45.86
Sparse6	2.875	0.558	41.87	2.852	0.630	32.43	2.787	0.572	34.29	2.789	0.552	65.28
Sparse7	2.881	0.540	56.95	2.921	0.611	44.87	2.841	0.575	47.58	2.786	0.536	90.51
Sparse8	2.899	0.564	74.56	2.860	0.573	59.70	2.835	0.560	63.08	2.780	0.610	117.52
Sparse9	2.885	0.586	94.93	2.940	0.549	75.34	2.833	0.568	79.58	2.801	0.573	151.87
Sparse10	2.884	0.599	106.33	2.915	0.512	82.90	2.836	0.560	88.00	2.790	0.555	165.34
Average	2.667	0.799	25.67	2.666	0.795	20.19	2.609	0.774	21.43	2.574	0.705	40.35
Ratio	1.000	1.000	1.000	0.999	0.994	**0.787**	0.978	0.969	0.835	**0.965**	**0.882**	1.572

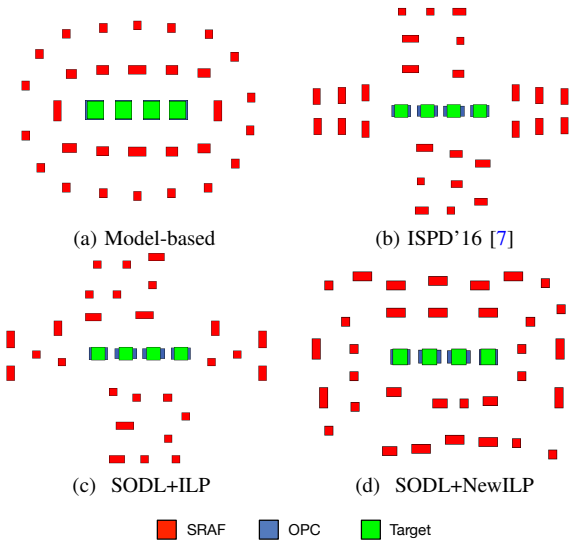


Fig. 12 Different SRAF insertion methods on a dense layout example.

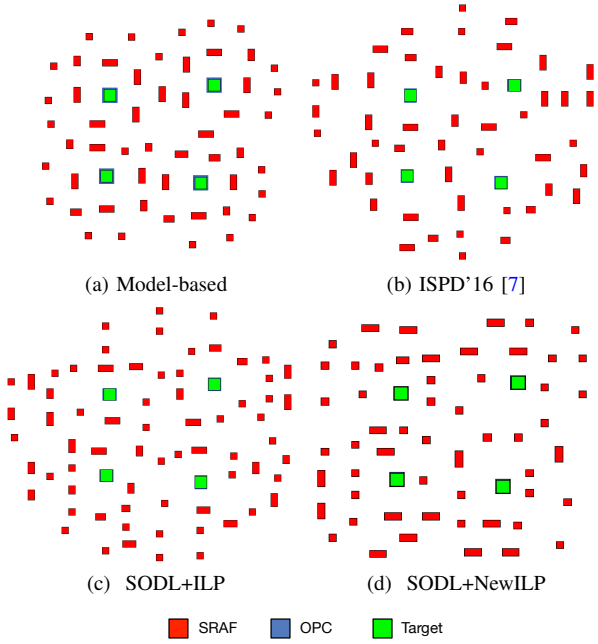


Fig. 13 Different SRAF insertion methods on a sparse layout example.

the new ILP model also affects the running time. For one grid, the neighborhood area where SRAF patterns may conflict with others becomes much larger than in the original ILP (see Fig. 10). Although the runtime of new ILP model is almost as twice as the original one's, it is worth adopting the model considering the increase of lithographic performance and size of benchmarks.

To further demonstrate the effectiveness and efficiency of proposed ILP models, scalability is explored. In TABLE II, the area of different benchmarks ranges from 1070×1070 nm^2 to 10630×10670 nm^2. We drew the relationship curve between the layout area and the runtime, which is visualized in Fig. 14 to clarify the scalability issue further. Note that the benchmarks of same area are removed in the illustration. For example, the benchmarks "Dense4" and "Dense5" are of the same size, and we only keep "Dense4".

The visualizations of simulation results for dense and sparse patterns are exemplified in Fig. 12 and Fig. 13 respectively. The differences between Figs. 12(c) and 12(d), Figs. 13(c) and 13(d) are mainly due to different ILP models following different design rules. Although the predictions from the machine learning model guide the inserting, however, they do not dominate the inserting process. The learning model only filters the invalid small grids. Hence, after solving, the feasible solutions of two ILPs to insert SRAFs can differ a lot.

TABLE III The Comparisons of total Area (μm^2) of Inserted SRAFs.

Benchmark	ISPD'16 [7]	SODL+Greedy	SODL+ILP	SODL+NewILP
Dense1	0.072	0.078	0.090	0.122
Dense2	0.099	0.087	0.118	0.149
Dense3	0.123	0.127	0.142	0.120
Dense4	0.138	0.131	0.157	0.157
Dense5	0.107	0.107	0.146	0.148
Dense6	0.146	0.142	0.133	0.140
Dense7	0.138	0.153	0.157	0.136
Dense8	0.116	0.117	0.142	0.145
Sparse1	0.387	0.336	0.408	0.311
Sparse2	1.213	1.299	1.150	0.958
Sparse3	2.350	1.977	2.502	1.943
Sparse4	4.004	4.028	4.027	3.123
Sparse5	4.832	4.909	4.950	3.875
Sparse6	6.922	6.937	7.136	5.510
Sparse7	9.688	7.602	9.658	7.776
Sparse8	12.61	11.73	12.64	9.925
Sparse9	16.18	10.46	16.37	12.54
Sparse10	17.97	14.00	17.68	13.94
Average	4.283	3.568	4.311	3.389
Ratio	1.000	0.833	1.007	0.791

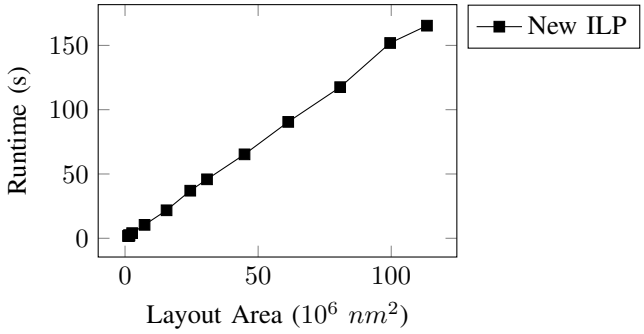


Fig. 14 The scalability of proposed New ILP model.

In addition, the comparisons of total area of inserted SRAFs among different methods are also summarized in TABLE III. According to the results in TABLE II and TABLE III, we can infer that the total area of inserted SRAF is not the critical factor affects the lithography performance.

VI. Conclusion

In this paper, for the first time, we have introduced the concept of dictionary learning into the layout feature extraction stage and further proposed a supervised online algorithm constructing dictionary. This algorithm has been exploited into a machine learning-based SRAF insertion framework. To get a global view of SRAF generation, combined with design rules, two ILP models have been built to generate SRAFs. The experimental results show that the F_1 score of machine learning model in SRAF insertion has been boosted and runtime overhead is also acceptable compared with a state-of-the-art SRAF insertion tool. More importantly, the results of lithography simulations demonstrate the promising lithography performance in terms of PV band area and EPE. With the transistor size shrinking rapidly and the layouts becoming more and more complicated, we expect to apply our ideas into general VLSI layout feature learning and encoding.

References

[1] D. Z. Pan, B. Yu, and J.-R. Gao, "Design for manufacturing with emerging nanolithography," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 32, no. 10, pp. 1453–1472, 2013.

[2] S. Shim, S. Choi, and Y. Shin, "Light interference map: A prescriptive optimization of lithography-friendly layout," *IEEE Transactions on Semiconductor Manufacturing (TSM)*, vol. 29, no. 1, pp. 44–49, 2016.

[3] S. Banerjee, Z. Li, and S. R. Nassif, "ICCAD-2013 CAD contest in mask optimization and benchmark suite," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013, pp. 271–274.

[4] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "MOSAIC: Mask optimizing solution with process window aware inverse correction," in *ACM/IEEE Design Automation Conference (DAC)*, 2014, pp. 52:1–52:6.

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

[6] C. H. Wallace, P. A. Nyhus, and S. S. Sivakumar, "Sub-resolution assist features," Dec. 15 2009.

[7] 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 *ACM International Symposium on Physical Design (ISPD)*, 2016, pp. 161–168.

[8] J. Jun, M. Park, C. Park, H. Yang, D. Yim, M. Do, D. Lee, T. Kim, J. Choi, G. Luk-Pat *et al.*, "Layout optimization with assist features placement by model based rule tables for 2x node random contact," in *Proceedings of SPIE*, vol. 9427, 2015.

[9] S. D. Shang, L. Swallow, and Y. Granik, "Model-based SRAF insertion," Oct. 11 2011.

[10] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in *European Conference on Information Retrieval (ECIR)*. Springer, 2005, pp. 345–359.

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

[12] M. J. Gangeh, A. K. Farahat, A. Ghodsi, and M. S. Kamel, "Supervised dictionary learning and sparse representation-a review," *arXiv preprint arXiv:1502.05928*, 2015.

[13] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2691–2698.

[14] M. J. Gangeh, A. Ghodsi, and M. S. Kamel, "Kernelized supervised dictionary learning," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4753–4767, 2013.

[15] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1697–1704.

[16] Y. Yankelevsky and M. Elad, "Structure-aware classification using supervised dictionary learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4421–4425.

[17] M. Aharon, M. Elad, and A. Bruckstein, "k-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[18] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *International Conference on Machine Learning (ICML)*, 2009, pp. 689–696.

[19] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.

[20] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[21] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Asilomar Conference on Signals, Systems, and Computers*, 1993, pp. 40–44.

[22] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, pp. 267–288, 1996.

[23] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.

[24] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, p. 1, 2010.

[25] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint*, 2016.
[26] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences (SIIMS)*, vol. 2, no. 1, pp. 183–202, 2009.
[27] Z. Allen-Zhu and L. Orecchia, "Linear coupling: An ultimate unification of gradient and mirror descent," in *Innovations in Theoretical Computer Science Conference (ITCS)*, 2017.
[28] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 1, pp. 185–194, 1999.
[29] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.
[30] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear gauss–seidel method under convex constraints," *Operations research letters*, vol. 26, no. 3, pp. 127–136, 2000.
[31] L. Bottou and O. Bousquet, "The trade-offs of large scale learning," in *Conference on Neural Information Processing Systems (NIPS)*, 2008, pp. 161–168.
[32] Mentor Graphics, "Calibre verification user's manual," 2008.
[33] J. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer Science & Business Media, 2010.

APPENDIX

A. Calculation of Gradient

In t-th iteration to update atoms of proposed algorithm, with $\boldsymbol{x}$ fixed, Equation (10) can be rewritten as in Equations (21) to (25).

$$\boldsymbol{D}_t \triangleq \arg\min_{\boldsymbol{D}} \frac{1}{2t} \left\|\boldsymbol{Y}_t - \boldsymbol{D}_{t-1}\boldsymbol{X}_t\right\|_F^2 \tag{21}$$

$$\triangleq \arg\min_{\boldsymbol{D}} \frac{1}{2} tr\left[\left(\boldsymbol{Y}_t - \boldsymbol{D}_{t-1}\boldsymbol{X}_t\right)^\top \left(\boldsymbol{Y}_t - \boldsymbol{D}_{t-1}\boldsymbol{X}_t\right)\right] \tag{22}$$

$$\triangleq \arg\min_{\boldsymbol{D}} \frac{1}{2} tr\left(\boldsymbol{D}_{t-1}^\top \boldsymbol{D}_{t-1}\boldsymbol{X}_t\boldsymbol{X}_t^\top - 2\boldsymbol{D}_{t-1}^\top \boldsymbol{Y}_t\boldsymbol{X}_t^\top\right) \tag{23}$$

$$\triangleq \arg\min_{\boldsymbol{D}} \left(\frac{1}{2} tr\left(\boldsymbol{D}_{t-1}^\top \boldsymbol{D}_{t-1}\boldsymbol{C}_t\right) - tr\left(\boldsymbol{D}_{t-1}^\top \boldsymbol{B}_t\right)\right) \tag{24}$$

$$\triangleq \arg\min_{\boldsymbol{D}} \left(\frac{1}{2}\sum_k \boldsymbol{d}_k^\top \sum_i \boldsymbol{d}_i \boldsymbol{c}_{ik} - \sum_k \boldsymbol{d}_k^\top \boldsymbol{b}_k\right). \tag{25}$$

In the stage of updating atoms in new algorithm, block coordinate descent algorithm, which means updating one atom (i.e., $\boldsymbol{d}_j$) while fixing other atoms, is still exploited with warm start mechanism. Therefore, the updating rule for atoms can be derived from the following equation.

$$\frac{\partial(25)}{\partial \boldsymbol{d}_j} = \frac{\partial\left(\frac{1}{2}\sum_k \boldsymbol{d}_k^\top \sum_i \boldsymbol{d}_i \boldsymbol{c}_{ik} - \sum_k \boldsymbol{d}_k^\top \boldsymbol{b}_k\right)}{\partial \boldsymbol{d}_j} \tag{26}$$

$$= \frac{\partial\left(\sum_{k\neq j} \boldsymbol{d}_j^\top \boldsymbol{d}_k \boldsymbol{c}_{kj} + \frac{1}{2}\boldsymbol{d}_j^\top \boldsymbol{d}_j \boldsymbol{c}_{jj} - \sum_k \boldsymbol{d}_k^\top \boldsymbol{b}_k\right)}{\partial \boldsymbol{d}_j} \tag{27}$$

$$= \sum_{k\neq j} \boldsymbol{d}_k \boldsymbol{c}_{kj} + \boldsymbol{d}_j \boldsymbol{c}_{jj} - \boldsymbol{b}_j \tag{28}$$

$$= \boldsymbol{D}\boldsymbol{c}_j - \boldsymbol{b}_j. \tag{29}$$

B. Proof of Theorem 1

Proof. As $\boldsymbol{C}_t$ and $\boldsymbol{B}_t$ are in a compact set, extracting converging sequences becomes possible. Therefore, the assumption could be made that two sequences converge to $\boldsymbol{C}_\infty$ and $\boldsymbol{B}_\infty$. As a result, $\boldsymbol{D}_t$ converges to $\boldsymbol{D}_\infty$. Assuming that $\boldsymbol{V} \in \mathbb{R}^{(n+s+1)\times(s)}$, $\hat{f}_t$ upperbounds the empirical cost f_t, i.e. $\hat{f}_t\left(\boldsymbol{D}_t + \boldsymbol{V}\right) \geq f_t\left(\boldsymbol{D}_t + \boldsymbol{V}\right)$. When $t \to \infty$, the inequality, $\hat{f_\infty}\left(\boldsymbol{D}_\infty + \boldsymbol{V}\right) \geq f\left(\boldsymbol{D}_\infty + \boldsymbol{V}\right)$, still holds.

Introduce a sequence $a_t > 0$ which converges to 0. With harnessing the Taylor expansion and using $\hat{f_\infty}\left(\boldsymbol{D}_\infty\right) = f\left(\boldsymbol{D}_\infty\right)$, the inequality (30) exists.

$$\begin{aligned} &f\left(\boldsymbol{D}_\infty\right) + tr\left(a_t \boldsymbol{V}^\top \nabla \hat{f_\infty}\left(\boldsymbol{D}_\infty\right)\right) \\ \geq &f\left(\boldsymbol{D}_\infty\right) + tr\left(a_t \boldsymbol{V}^\top \nabla f\left(\boldsymbol{D}_\infty\right)\right). \end{aligned} \tag{30}$$

This inequality holds for all $\boldsymbol{V}$, $\nabla \hat{f_\infty}\left(\boldsymbol{D}_\infty\right) = \nabla f\left(\boldsymbol{D}_\infty\right)$. A first-order necessary optimality condition for $\boldsymbol{D}_\infty$ being an optimum of $\hat{f_\infty}$ is that $-\nabla \hat{f_\infty}$ is in the normal cone of the convex set of dictionary matrices at $\boldsymbol{D}_\infty$ [33]. So the first-order necessary optimality condition for $\boldsymbol{D}_\infty$ being an optimum of f is also validated. Since $\boldsymbol{B}_t$ and $\boldsymbol{C}_t$ asymptotically get close to their accumulation points, $-\nabla f\left(\boldsymbol{D}_t\right)$ will be close to the normal cone at $\boldsymbol{D}_t$. □

Hao Geng received the M.E. degree from the Department of Electronic Engineering and Information Sciences, University of Science and Technology of China in 2015, and the M.Sc. degree in machine learning from the Department of Computing, the Imperial College London in 2016. He is currently pursuing his Ph.D. degree at the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His research interests include machine learning and deep learning in VLSI design for manufacturing.

Wei Zhong received the B.S. degree from Dalian University of Technology, Dalian, China, in 2008, the M.S. and Ph.D. degrees from Waseda University, Tokyo, Japan, in 2010 and 2014, respectively. He served as a Research Assistant in the Information, Production and Systems Research Center of Waseda University, Kitakyushu, Japan, from 2010 to 2011, an Associate Specialist in the Central Research Laboratory of Ricoh Company, Tokyo, Japan, from 2011 to 2014, a Chief Designer and Director of the Institute of Image Processing Technology in the State Key Laboratory of Digital Multimedia Technology, Hisense Group, Qingdao, China, from 2014 to 2018. From 2015 to 2017, he also served as a Postdoctoral Researcher in the University of Science and Technology of China, Hefei, China. He is currently an Associate Professor in the International School of Information Science and Engineering, Dalian University of Technology, Dalian, China. His research interests include several aspects of computer vision and image processing algorithms, VLSI design automation, networks on chips and hardware-software co-design of embedded systems.

Haoyu Yang received his B.E. degree from Qiushi Honors College, Tianjin University in 2015. He is currently pursuing his Ph.D. Degree at the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His research interests include VLSI design for manufacturability and machine learning.

Yuzhe Ma received his B.E. degree from the Department of Microelectronics, Sun Yat-sen University in 2016. He is currently pursuing his Ph.D. degree at the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He has interned at Cadence Design Systems, San Jose, CA, USA and NVIDIA Research, Austin, TX, USA. His research interests include VLSI design for manufacturing, physical design and machine learning on chips.

Joydeep Mitra currently works at Cadence Design Systems with a focus on bringing Machine Learning into Electronic Design Automation. He has held senior engineering and R&D positions at several EDA and semiconductor companies including Siemens (Mentor Graphics), Synopsys (Magma), Oracle (Sun), and AMD. His background includes the domains of mask synthesis, circuit simulation, signal integrity, DFM, and physical design automation. Joydeep has a Ph.D. in Electrical and Computer Engineering from the University of Texas, Austin. His research interests include advanced patterning using directed self-assembly and physical design automation for 3-D ICs.

Bei Yu (S'11–M'14) received the Ph.D. degree from The University of Texas at Austin in 2014. He is currently an Assistant Professor in the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He has served as TPC Chair of ACM/IEEE Workshop on Machine Learning for CAD, and in many journal editorial boards and conference committees. He is Editor-of-Chief of IEEE TCCPS Newsletter. He received five Best Paper Awards from Integration, the VLSI Journal in 2018, International Symposium on Physical Design 2017, SPIE Advanced Lithography Conference 2016, International Conference on Computer Aided Design 2013, and Asia and South Pacific Design Automation Conference 2012, and five ICCAD/ISPD contest awards.