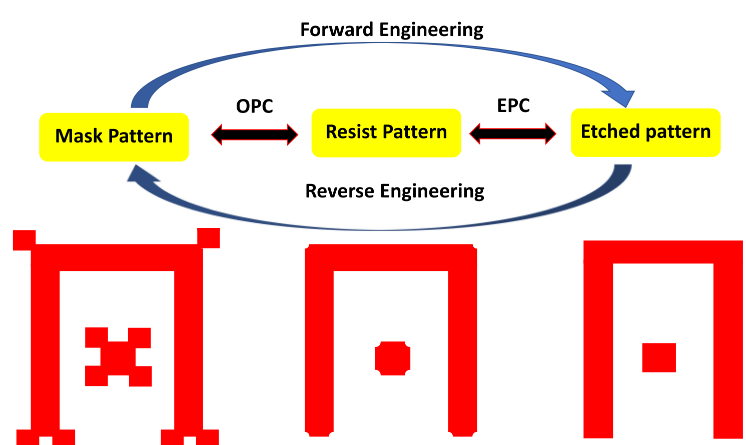
Intelligent Photolithography Corrections Using Dimensionality Reductions

Volume 11, Number 5, October 2019

Parag Parashar
Chandni Akbar
Tejender S. Rawat
Sparsh Pratik
Rajat Butola
Shih H. Chen
Yung-Sung Chang
Sirapop Nuannimnoi
Albert S. Lin



DOI: 10.1109/JPHOT.2019.2938536

Intelligent Photolithography Corrections Using Dimensionality Reductions

Parag Parashar,[1] Chandni Akbar,[1] Tejender S. Rawat,[1] Sparsh Pratik,[1] Rajat Butola,[1] Shih H. Chen,[2] Yung-Sung Chang,[2] Sirapop Nuannimnoi,[1] and Albert S. Lin [2]

[1]Department of Electrical and Computer Science Engineering, National Chiao Tung University, Hsinchu 300, Taiwan
[2]Department of Electronics Engineering, National Chiao Tung University, Hsinchu 300, Taiwan

DOI:10.1109/JPHOT.2019.2938536
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Manuscript received July 2, 2019; revised August 21, 2019; accepted August 26, 2019. Date of publication August 30, 2019; date of current version September 13, 2019. This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant MOST 106-2628-E-009-010-MY3. Corresponding author: Albert S. Lin (e-mail: hdtd5746@gmail.com).

Abstract: With the shrinking of the IC technology node, optical proximity effects (OPC) and etch proximity effects (EPC) are the two major tasks in advanced photolithography patterning. Machine learning has emerged in OPC/EPC problems because conventional optical-solver-based OPC is time-consuming, and there is no physical model existing for EPC. In this work, we use dimensionality reduction (DR) algorithms to reduce the computation time of complex OPC/EPC problems while the prediction accuracy is maintained. Also, we implement a pure machine learning approach where the input masks are directly mapped to the output etched patterns. While one photolithographic mask can generate many experimental patterns at once, our pure ML-based approach can shorten the trial-and-error period in the photolithographic correction. Additionally, we demonstrate the automation in SEM images preprocessing using feature detection, and this facilitates intelligent manufacturing in semiconductor processing. The input vector dimensions are effectively reduced by two orders of magnitude while the observed mean squared error is not affected significantly. The computation runtime is reduced from 4804 s of the baseline calculation to 10 s–200 s The MSE values changed from the baseline 0.037 to 0.037 for singular value decomposition (SVD), to 0.039 for independent component analysis (ICA), and to 0.035 for factor analysis (FA).

Index Terms: Lithography, Diffractive imaging, Technologies for computing.

1. Introduction

Optical lithography is a crucial step in the integrated circuit (IC) technology for transferring patterns from masks to wafers. Nonetheless, in advanced lithography, the patterns printed on wafers are distorted due to optical proximity effects such as resist process effects, diffraction, and interference [1]–[4]. Various resolution enhancement technologies (RETs) have been proposed to improve the pattern fidelity and the performance of the lithographic imaging system [5]. In RETs, the mask is modified according to the fundamental principles of optics and limitations in the lithographic system. Specific to mask side enhancement, optical proximity correction (OPC) is a well-known technique to enhance the resolution of a specific photolithography system operating in a fixed wavelength. OPC techniques compensate for image distortions as this method pre-warps the mask patterns based on the target patterns [6]–[11].

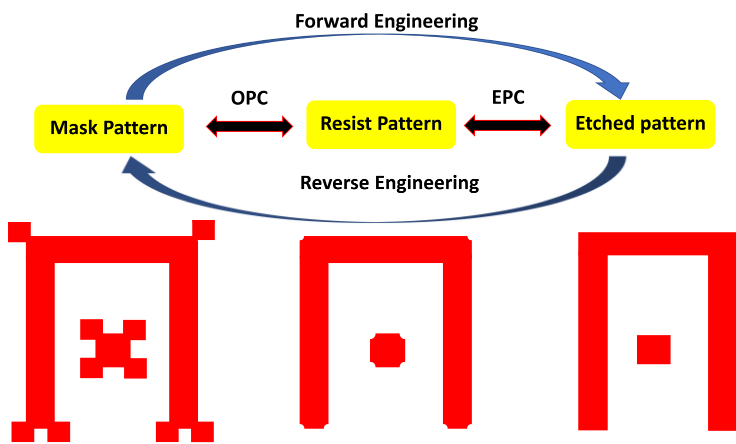


Fig. 1. Illustration of the EPC/OPC procedures and the associated forward and reverse engineering.

Together with OPC effects, etch proximity effects (EPC) also affects the overall critical dimensions for the photolithography and patterning process. Etch proximity effects originate from etching the developed resist patterns. Reference to the photoresist pattern, the etching process can produce an over-etched substrate or an under-etched substrate. The over-etched substrate is known as a negative etch bias while the under-etched substrate is referred to as positive etch bias. Negative etch bias is attributed to the lateral erosion. On the other hand, positive etch bias is caused by the deposited particles, such as ions and radicals, on the surface of the resist and the substrate through the entire etching process [11]. Therefore, in order to obtain the desired patterns and the correct dimensions on the wafer, etch proximity corrections (EPC) have to be applied to compensate the etch bias reference to the resist pattern [12], [13].

The relationship between etched patterns, resist patterns, and mask patterns are shown in Fig. 1. In forward engineering, OPC on the photolithography mask is implemented first to arrive at a resist pattern. Afterward, the resist pattern goes through EPC, and the final etched pattern should match the desired target pattern on the wafer. This whole process is repeated while OPC/EPC is being adjusted until we arrive at the desired pattern on the wafer. On the other hand, in the reverse engineering case, firstly EPC corrections are proposed, and the etched pattern should be the desired pattern. Afterward, the resist pattern is the target pattern after photolithographic exposure and resist development. The OPC is then used to ensure that the mask pattern leads to the desired resist pattern [14], [15].

The OPC process can roughly be classified as model-based OPC (MBOPC) and rule-based OPC (RBOPC). Rules are empirically generated based on the geometrical properties of the layout patterns. This technique is a little inaccurate as compared to MBOPC, but it yields faster results due to less required computation. RBOPC is not suitable beyond 90 nm technology node [14], [16], [17]. On the other hand, MBOPC is based on approximated optics models. With complex masks and volumetric data, MBOPC requires excessive computational time. MBOPC has two separate directions, namely, edge-based (EBOPC) and pixel-based OPC (PBOPC) [18]. In EBOPC, the input mask layout is partitioned into different edges while in PBOPC the input masks are converted into pixels. Either EBOPC or PBOPC is effective, but both of them require enormous time in calculation. As technology shrinks to smaller nodes, more accurate OPC calculations are required, and PBOPC is in general preferred over EBOPC in optics solvers to get more accurate results.

Similarly, EPC can be classified into rule-based EPC (RBEPC) and model-based EPC (MBEPC). Rules are empirically generated by performing a few experiments on certain layouts to calculate etch biases in each case. The results are then put into a tabulated form for future usage if similar patterns or shapes are encountered in IC layout. Nonetheless, these tabulated rules are, in most cases, too simplistic to effectively correct a wide range of patterns that occurs in a real layout. On the other hand, MB-EPC is implemented to model etch bias as a function of a few empirical

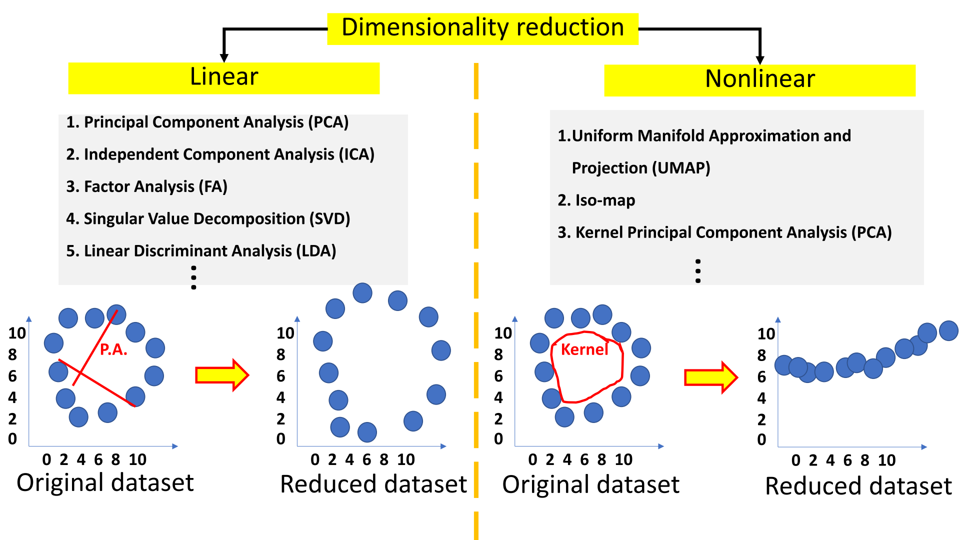


Fig. 2. The linear and non-linear dimensionality reduction techniques.

parameters of a specific shape and its environment. These parameters are density kernel, blocked kernel, and visible kernel [19]–[21]. The coefficients of this polynomial function are determined empirically. Thus, MB-EPC can be more accurate and applied to a wider range of layouts than RB-EPC can, though the results are still unsatisfactory and require excessive time and computations. Furthermore, the nature of the current EPC models is still semi-empirical. To date, there have not been accurate physical models for EPC. Detailed physical models can be difficult to construct due to the high complexity in dry etching processes.

In recent years, machine learning (ML) has emerged as a promising field in bringing new opportunities across all fields. Machine learning has been employed in providing optical proximity corrections (OPC) and etch proximity corrections (EPC) to shorten the trial-and-error cycles and to provide more accurate and faster results [22]–[33]. These ML-based OPC/EPC efforts have pushed the resolution limits of photolithography, reduced image distortions, and achieved less computational time compared to optics model-based OPC. Nonetheless, there have not been many efforts devoted to dimensionality reduction (DR) for photolithography patterning correction problems to date. While DR becomes more and more critical in ML, in this work, we investigate its capability and feasibility to reduce the OPC/EPC problem complexity and computation time.

Dimensionality reduction (DR) has emerged as an effective way to improve the calculation efficiencies and enhance visualization in machine learning problems. The input vector feature sizes are reduced, and in the preferred case, the critical information in the input space is still preserved. Using appropriate DR algorithms, improvement can be observed in data redundancy reduction, intrinsic structure discovery and feature extraction, better visualization of data, enhancing computation, and machine learning efficiency in terms of run time and memory requirements [34]–[38]. DR algorithms can also contribute to removing irrelevant and noisy features, such as independent random noise that is uncorrelated with the input, and the unwanted degrees of freedom. Both linear and nonlinear dimensionality reduction techniques can be implemented depending upon the situation [39]–[42]. Fig. 2 illustrates the basic working mechanism of both linear and non-linear techniques. In a linear dimensionality reduction technique, the high variance directions are located to reflect the key trends in the data set Fig. 2. A new axis is linear and may be non-orthogonal. On the other hand, non-linear dimensionality reduction utilizes nonlinear kernels to locate the key trends in the dataset while preserving the neighborhood relation. In terms of neighborhood preserving, the distance or the distance ranks between the data points in the original space needs to be preserved after the input vector is transformed into the embedding space. Neighborhood preserving is frequently used as a measure of the quality of DR techniques.

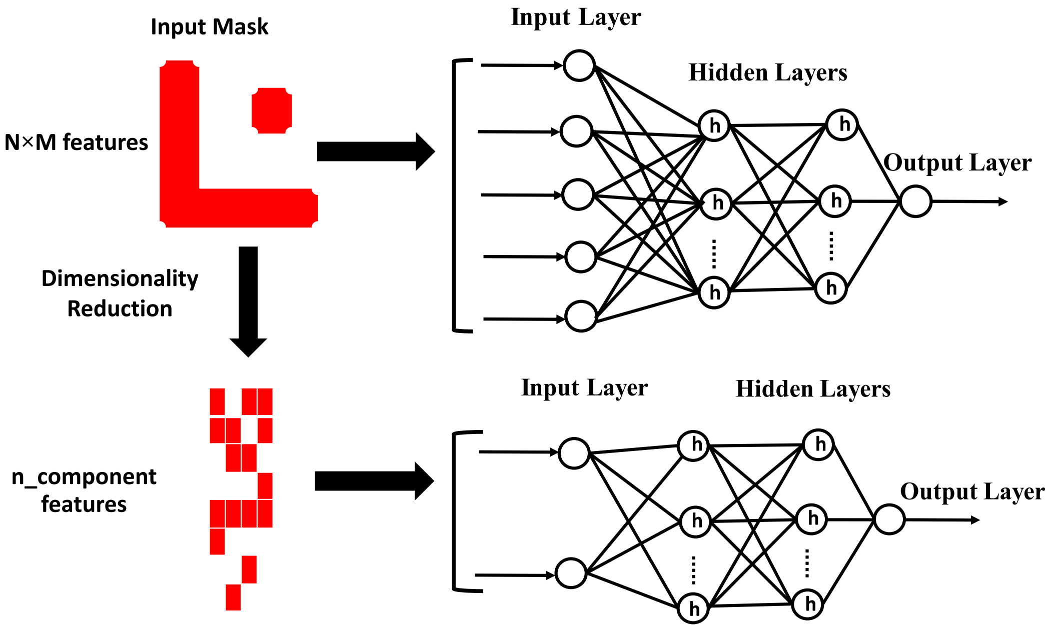


Fig. 3. Illustration of dimensionality reduction in this proposed work. Input mask has N × M feature size, and thus, the number of input nodes of the neural network is high. After dimensionality reduction, the number of input features decreases to n_components.

In this work, the dimensionality reduction technique is exploited to reduce the number of input feature size of pixel-based (PB) OPC/EPC using machine learning. This step induces a little inaccuracy in the data, but these inaccuracies should not contribute significantly to the predicted output etched patterns if DR is effective. Fig. 3 below depicts the overall working mechanism of our work. Various dimensionality reduction techniques have been implemented to decrease the number of input feature size of PBOPC/EPC masks, and this reduced feature set is used to train the neural network (NN). After that, testing of the NN is done by the test dataset, and the comparison has been made between the prediction accuracy using the original input space and the embedding spaces by different DR techniques. We use a pure machine learning based approach, where the input is photolithography mask, and output is the experimental etched pattern. This approach is quite different from most of the current ML-based OPC/EPC in literature [22]–[33]: (1) In most papers, ML is used to tune the relation between input mask and calculated resist pattern using optical solvers to save optical solver calculation time. We train experimental patterns instead. This eliminates the dependency on optical solvers that can be slow. Since one photolithography mask can generate abundant experimental patterns at once, the training dataset can be collected in a short amount of time. Furthermore, DR is used to reduce the neural network training time, and intelligent SEM image preprocessing reduces the data preprocessing time. (2) ML has been mostly applied to OPC, and there are only few papers working on ML EPC [19], [20]. (3) OPC and EPC in our work can be done in one calculation. This is distinct from the current practice of separate efforts in OPC and EPC.

2. Method

2.1 Image Processing

In this work, we use etched pattern as the target output for neural network training, and thus effectively, we implement EPC/OPC together. This approach is new since, in most of the works in literature, OPC and EPC are conducted separately. In most practices in the past, OPC still counts on optics solver while EPC is implemented before OPC using ML or empirical models.

Raw data is generally susceptible to containing noise or unreliability, especially if data collection is from semiconductor manufacturing. In this paper, the data pre-processing involves mainly the following steps: feature detection, feature extraction, geometric transform estimation, binarizing, and centralizing. The preprocessing code is mainly in Matlab [43] scripts. Different mask patterns with varying OPC/EPC mask corrections constitute our dataset. To detect the shape and size of the etched image patterns in the dataset, scale bar in each image has to be detected to correct the slight scale difference during scanning electron microscope (SEM) operation. Feature detection is based on speeded up robust feature (SURF), which is a rotation-invariant algorithm [44]. This descriptor employs Hessian matrix and Laplacian approximation to reduce the CPU time. For extracting the features, the square region centered at the point of interest has been constructed on the clustered scene. The orientations of the square box regions are set according to the orientation of the referenced image [45]–[48]. Afterward, the features on the referenced images, i.e., the scale bar, and the features of the target image with clustered scenes are matched [49]–[52]. Finally, the desired object in the clustered scene is detected using affine transformation [53], [54].

In the next step, the binarization of images is conducted. The purpose of this step is to make the boundary of the etched pattern in SEM images prominent. Firstly, the image is binarized using a locally adaptive threshold algorithm [55], [56]. Afterward, the hole-filling to the enclosed black space by white lines is conducted by morphological reconstruction algorithm [57]. In case of slightly discontinuous lines in the SEM images due to Pt film coating, morphology bridging can be done before hole filling [58]–[61]. Afterward, we rotate the processed images to the upright position since SEM captured image can be at any rotation. The size of each image is reduced by considering each mask at the center position by eliminating the extraneous space in images. Finally, the images are being centered with a consistent size. Fig. 4 illustrates all these steps for the SEM images of the array of 200 nm $\times$ 3 μm lines. All images have been centered with the size of 660 $\times$ 408 pixels. For the actual calculation, the input image size is set as 600 $\times$ 400 and further reduced to n_components = 1–45 based on various dimensionality reduction techniques. The output image size is set as 65 $\times$ 38 to reduce the computation time. Multi-layer perceptron (MLP) with two hidden layers is used in this study. While convolution neural network (CNN) has been the mainstream in image processing and computer vision, the basic MLP performs better based on our test if the neural network output is a predicted SEM image rather than a few labels such as the cases in object detection, motion tracking, or sentiment analysis. This is because the pooling and convolution operation in CNN can lead to blurred input information, which is not desired if an entire output image is to be predicted clearly. In the dimensionality reduction studies, the input layer neuron number is n_components, which ranges from 1 to 45, and the hidden layer neuron number is 50. In the baseline study, the input layer neuron number is 600 $\times$ 400 = 240000, and the hidden layer neuron number has to be increased to 500 to have converged solutions. Directly downscaling from 240000 neurons to 50 neurons lead to no convergence at all. Therefore, the advantage of using DR algorithms lies in not only a reduced input neuron number but also a reduced hidden layer neuron number required for convergence. The output neuron number is uniformly 65 $\times$ 38 = 2470.

The Maxwell equation is linearly scalable. Thus, when the device critical dimension (C.D.) is scaled down, the lithography light source wavelength (λ) has to be scaled down simultaneously. In response to shorter λ, the physical length per pixel is also scaled down proportional to C.D. and λ scaling in digital images. Except the scaling down in the physical length per pixel, given the fact that the intelligent ML-based OPC method in this study only concerns the digitized images with pixels, there is not much difference in applying ML-based OPC to I-line or DUV/EUV lithography.

2.2 Implementation of DR Techniques

Five dimensionality reduction techniques have been implemented in this work namely, singular-value decomposition (SVD), independent component analysis (ICA), factor analysis, isometric mapping (ISOMAP) and uniform manifold approximation and projection (UMAP). We use Python, Scikit Learn [62], [63], Matlab [43], UMAP toolbox by McInnes *et al.* [64], and R programming language in this work.

Fig. 4. Preprocessing of the dataset: (a) Gray-scale SEM image with its corresponding scale bar. (b) Detection of scale bar in the image using feature detection algorithms. (c) Binarization, morphology bridging, hole filling, and rotation. (d) Centered and zero-padded image to adjust the size of each image to achieve a uniform size of 660 $\times$ 408 pixels in the entire dataset.

2.2.1 Singular-Value Decomposition: (SVD) The Singular-Value Decomposition (SVD) reduces a matrix into three matrices, and thus the calculation becomes simpler using the decomposed matrices [63], [65]–[70]. SVD and Principal Component Analysis (PCA) method are related methods where PCA is mainly used for square matrices, and SVD can be used for non-square matrices [71]. The singular value decomposition method leads to a reduced space where, along the new axes, the observations/features have the highest variance. These new directions are known as the principal components [72]. SVD decomposes a matrix as [73]:

$$A = UWV^T \tag{1}$$

where U is the orthonormal eigenvectors of AA^{T}, V^{T} is the transpose of a matrix containing the orthonormal eigenvectors of $A^{\mathrm{T}}A$, and W is a diagonal matrix where its diagonal elements are the square roots of the eigenvalues of $A^{\mathrm{T}}A$ in descending order. In comparison, SVD algorithms are faster and numerically more stable than PCA.

2.2.2 Independent Component Analysis (ICA): Independent component analysis (ICA) is a linear dimension reduction algorithm where the transformation is conducted on the original dataset to arrive at reduced features of independent components [63], [72]–[75]. This method iteratively optimizes and increase the independence between the features in the embedding space. This, in turn, leads to a linear nonorthogonal projection of the original dataset into the embedding space. The observed vectors in the original space are then expanded as the linear combinations of unknown non-gaussian sources S. With the assumption of statistical independence, ICA determines

the sources from the original observation [72], [73]. Using ICA, given that S is a source random vector, $A = W^{-1}$ is a mixing matrix, and X is observation, then ICA finds W and $S = WX$ [73].

2.2.3 Factor Analysis (FA): In factor analysis, relations among a set of random variables is analyzed, and the goal is to determine the correlations among all variables in the original input space [63], [73], [76]–[83]. This is done by postulating a reduced set of common factors where the number of random variables is less than the original set. The new, reduced set of random variables is assumed to be independent with zero mean. The new factors are assumed to be present in the experimental samples despite the fact that they may not be directly observed, counted, or measured. Considering the original space of n random variable, factor Analysis model can be specified by given equations of observed variables Y_i, $i = 1, \ldots,$ n [73], [76]:

$$Y_i = \lambda_{i1} F_1 + \lambda_{i2} F_2 + \cdots + \lambda_{ik} F_k + \delta_i \tag{2}$$

where δ represents variability in each Y_i, and all Y_i are linear functions of F and δ_i. k is the dimension in the embedding space. If there are m observations in our dataset, the factor analysis equation, (2) can be written into matrix form [73]:

$$Y_{n\times 1} = \Lambda_{n\times m} F_{m\times 1} + \delta_{n\times 1} \tag{3}$$

where Λ is the loading matrix. With some manipulation, we can find that the covariance of original space observation, Y, can be written as [63], [73], [76]–[83]:

$$Cov(Y) = \Sigma = \Lambda\Lambda^T + \mathrm{cov}(\delta) \tag{4}$$

Transformation from Y to F is possible if Λ can be solved. The procedure is, in short, to find Λ that reduces the mean square error in the off-diagonal elements in the correlation matrix [73].

2.2.4 Isometric Mapping (ISOMAP): Isometric mapping (ISOMAP) is a manifold learning algorithm for dimensionality reduction. It recovers low-dimensional representation of a non-linear manifold based on the assumptions that the manifold is smooth, and the geodesic distance between two points is equal to the Euclidean distance. The steps of this algorithm are described in [62], [63], [76], [84]–[86]. Basically, k-nearest neighbor (KNN) algorithm is used to extract which are neighboring nodes. In this step, the edge length between the neighboring nodes is the Euclidean distance

$$dist_{ij} = ||x_i - x_j|| \tag{5}$$

Through this procedure, the neighbors of the manifold for all observations are sketched. The neighborhood graph, $G = G(X, \phi)$, is then generated, where X is a set of input data points and ϕ is a set of edges which show the neighborhood relationship between the nodes. Finally, Dijkstra's or Floyd–Warshall algorithm is employed to calculate the shortest distance between any two nodes in the manifold, and multidimensional scaling to calculate the finalized embedding space.

2.2.5 Uniform Manifold Approximation and Projection (UMAP): Uniform Manifold Approximation and Projection (UMAP) is a unique non-linear manifold dimension reduction technique whose visualization quality can be similar to t-SNE [64], [76]. UMAP works in Riemannian geometry where manifold is assumed to be locally connected and metric is locally constant. The implementation of UMAP in its first step is to construct a weighted k-neighbor graph. Let $X = \{x_1, \ldots, x_N\}$ be the input dataset, with a dissimilarity measure d: $X \times X \rightarrow \mathbb{R} \geq 0$. With an input hyperparameter k, for each x_i, the set $\{x_{i1}, \ldots, x_{ik}\}$ of k nearest neighbors of x_i will be computed under the measure d. For each x_i, the following equations are defined as [64]:

$$\rho_i = \min\{d(x_i, x_{ij}) | 1 \leq j \leq k, d(x_i, x_{ij}) > 0\}, \tag{6}$$

and set σ_i is a value to ensure [64]:

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{ij}) - \rho_i)}{\sigma_i}\right) = \log_2(k) \tag{7}$$

Then, a weighted directed graph $G = (V, E, \omega)$ can be constructed and can the set of directed edges $E = \{(x_i, x_{ij})|1 \leq j \leq k, 1 \leq i \leq N\}$ is formed. More details can be found in the decent work by McInnes *et al.* [64].

2.3 Sample Fabrication

Firstly, we prepare the masks of the patterns on Synopsys Laker software. The spacing of each pattern has been set as 10 μm uniformly. We prepared 50 patterns with different sub-resolution-feature (SRF) corrections. Afterward, the laker GDSII is exported as ASCII. Using the coordinate and shape information in the ASCII file, we can convert the mask patterns into binary images. The mask pattern is transferred to the reticle in 1:5 ratio, and the mask is fabricated using laser direct writer and dry plasma etching.

After that, we used TEL CLEAN TRACK MK-8 for the deposition of photoresist on the silicon wafer. After this step, we used Canon FPA-3000i5+ I-Line stepper to expose the wafer for pattern transfer. The frequency of operation is 365 nm for this machine. The resolution and numerical aperture for I-line stepper are 0.35 μm and 0.45–0.63, respectively. Numerical aperture is automatically variable in this step. The reduction ratio and field size for I-line stepper are 5:1 and 20 mm $\times$ 20 mm, respectively. Then, we use TEL CLEAN TRACK MK-8 machine again for removing photoresist and for developing the resist patterns on the Si wafer. Finally, Oxford Estrelas 100 deep reactive ion etching (DRIE) machine has been used for etching of the patterns onto the Si wafer. This machine is based on ion bombardment mechanism. Ion bombardment is manifested mainly in the vertical direction. SF_6 gas is sent into the chamber and plasma is then turned on. The SF_6 molecule will be accelerated towards the Si wafer to impact the exposed surface.

2.4 Sample Measurement

Sample measurement is implemented by using a Hitachi SU8010 Cold Field Emission SEM machine. The electron source for is cathode electron gun. The resolution of the machine is 10 Å (at 15 kV) or 13 Å (at 1 kV). The voltage of operation is 0.1 kV to 30 kV. The sample size suitable for this Hitachi SEM is 50 mm $\times$ 50 mm. The magnification for a quality image for this machine is around 10^5.

3. Results and Discussion

In Table 1, we see reduced program execution time when dimensionality reduction algorithms are applied. The t_{DR} is the execution run time for dimensionality reduction calculation, and the t_{ML} is the time for the training and testing of the neural network. The code is tested on HP Z240 workstation with Intel (R) Xeon (R) CPU E3-1225 v5 3.30 GHz and HP elite desktop. The time shown in Table 1 is on i5 CPU. An apparent reduction in t_{ML} time from 4804s of the baseline calculation to 10 s–200 s is obvious. This is due to the fact that the input image size is reduced from 600 $\times$ 400 pixels to n_components = 1–45. In Table 1, we also observe that in the case of applying linear DR techniques, the mean square error (MSE) does not change significantly reference to the baseline calculation where no dimensionality reduction (DR) is applied to the input images. The MSE values are changed from the baseline value of 0.037 to 0.036 in SVD, to 0.038 in ICA, to 0.034 in FA. In the case of nonlinear DR, the MSE degrades more significantly but still at the same order. The MSE values degrade to 0.058 in Isomap, and to 0.069 in UMAP, for the optimally reduced dimension (n_components) in the embedding space. Satisfactory degradation behavior is attributed to the capability of DR techniques in capturing the key components that govern the outputs in this machine learning OPC/EPC problem. In fact, whenever the dimension of the input vectors is reduced, the information contained in the input space is inevitably lost. Nonetheless, if the input vector space contains a significant portion of less critical information, reasonably efficient DR techniques can be used to exclude this non-significant part of the information and retain the prediction accuracy. It can be noticed that in Table 1 that the execution time for smaller n_components is not necessarily

TABLE 1
The Mean Square Error (MSE) and the Program Execution Time of Dimensionality Reduction (t_{DR}) and Neural Network (t_{ML}) in the Training/Test Set

	(MSE_{train}, MSE_{test}) (n_components, t_{DR}, t_{ML})			
Baseline	(0.0, 0.037) (NA, NA, 4803.63s)			
SVD	(0.043, 0.065) (1, 0.81s, 55.7s)	(0.0, 0.047) (10, 1.1s, 55.5s)	(0.00, 0.037) (20, 1.7s, 48.9s)	(0.00, 0.12) (45, 1.26s, 33.1s)
ICA	(0.043, 0.065) (1, 0.61s,52.4s)	(0.00, 0.048) (10, 0.79s, 63.5s)	(0.00, 0.039) (20, 1.78s, 50.5s)	(0.0, 0.040) (40, 1.7s,41.7s)
FA	(0.040, 0.064) (1, 4.79s, 83.9s)	(0.00, 0.058) (10, 11.7s, 204.1s)	(0.00, 0.058) (30, 50.3s, 66.2s)	(0.00, 0.035) (45, 13s, 28.7s)
ISOMAP	(0.044, 0.06) (1, 0.63s, 68.2s)	(0.0, 0.058) (10, 0.66s, 58.9s)	(0.0, 0.065) (15, 0.67s, 54.1s)	(0.0, 0.0891) (24, 1.02s, 35.7s)
UMAP	(0.040 ,0.087)	(0.0, 0.069)	(0.0, 0.080)	(0.00, 0.073)
	(1, 0.22s,45.9s)	(10, 0.27s, 66.5s)	(20, 0.33s, 67.2s)	(40, 0.34s, 59.1s)

shorter. This is because in the case of n_componets = 1–45 as shown in Table 1, the computation loadings on CPU are all light, and therefore, t_{ML} is mainly determined by the time required for convergence. The convergence is considered achieved when the loss function decreases less than the tolerance $= 1 \times 10^{-4}$ in multilayer perceptron (MLP) training, and thus, longer t_{ML} is observed if more iterations are needed to achieve convergence. As for the relationship between n_componets and MSE_{train}/ MSE_{test}, it can be more complicated. MSE_{train} mainly depends on two factors: embedding quality of DR and the complexity of the function that maps the embedded input space to the output space, and both of these two factors are affected by n_components while their correlation can be difficult to know using a general rule. High embedding quality ensures the input data points are well spaced in the embedding space, and which n_components leads to better embedding varies for different DR algorithms and ML problems. Thus, we can observe in Table 1 that the optimal n_components is, in fact, DR-algorithm dependent. MSE_{test} further depends on the train-test set discrepancy and the test set representation in the embedding space. Without real calculation, it is difficult to know in advance the optimal n_components for MSE_{train} or MSE_{test} since all of the factors are entangled. Fig. 5 plots the real and predicted images in the training and the test set, for using baseline non-reduced input masks and the masks under various DR techniques. It should be emphasized that the reduction in the code execution time is a critical factor to determine whether machine learning is going to be useful in OPC/EPC problems. This is because the bottleneck in current technology is the long execution time associated with the OPC/EPC calculation. Using pure ML-based approach, we not only overcome the problem of solving Maxwell equations in 3D but also the problem of lacking an appropriate model for etching proximity corrections (EPC). While in most photolithography correction cases to data, ML is used to construct a model between the input mask and output optical-solver patterns to reduce optical solver calculation load, in our case we directly train experimental patterns across OPC/EPC problems at once. This can effectively reduce the trial-and-error time since one photolithography mask can generate abundant experimental patterns. Furthermore, DR and intelligent preprocessing, as shown in Section 2.1, reduce the time of neural network training and data warehouse clean up.

Further reduction in code execution time can be done by reducing the output neuron number. This can be done by using feature extraction to highlight the key geometry parameters, as the case in [23]. Alternatively, we can reduce the output neuron number by evaluating the averaged difference between the real images and predicted images at predefined partitions over the entire output images. The number of partitions can be adjusted based on the need. In the extreme case, the output neuron number can be reduced to one where the averaged difference between the entire

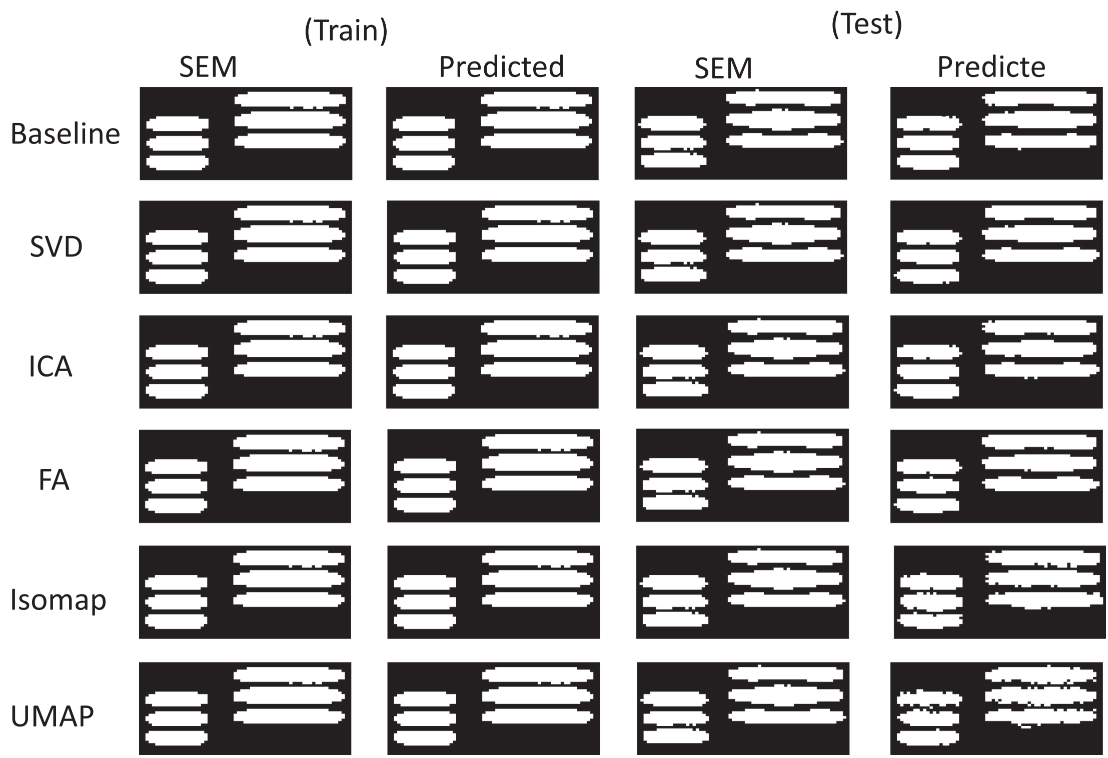


Fig. 5. The real SEM images and predicted SEM images in the training and the test set for different DR algorithms.

real images and the entire predicted images are calculated as a lumped figure of merit. This is feasible because eventually we only want to know whether the real etched pattern is the same as the predicted pattern. The partition and averaged difference scheme can be more feasible then direct down-sampling the images, especially when a high reduction ratio is desired.

Fig. 6(a) shows the embedding spaces after dimensionality reduction using various techniques in the case of optimal n_components values in Table 1 for the five DR algorithms. We plot the training set in blue circles and the test set in red circles. It can be observed that the UMAP leads to a very uniform distribution over the 2D reduced embedding space. On the other hand, the independent component analysis (ICA) maps most of the points onto a crowded region in the embedding space. Solely looking at the data set distribution in the reduced space, as shown in Fig. 6(a), is not sufficient to provide overall concrete information regarding which DR techniques are more suitable for OPC/EPC purposes. A more quantitative analysis is necessary to analyze the quality of vector space embedding. If we solely compare the neighborhood embedding quality, then it has been proposed that several meta-criteria can be used to evaluate the performance of DR algorithms. It has to be emphasized that even we solely discuss neighborhood preservation in vector space embedding, it is still difficult to find a unified measure since this is still highly ML-problem dependent [87]–[89]. Which kind or what degree of neighborhood preserving is better depends on the ML problem under investigation. In literature, the co-ranking matrix has been proposed to show the rank error of DR algorithms. In Fig. 6(b), we plot the co-ranking matrix for SVD, ICA, FA, ISOMAP, and UMAP. Co-ranking matrix is defined as [87]–[90]:

$$Q = [q_{kl}]_{1 \leq k,l \leq N-1} \text{ with } q_{kl} = \left|\{(i,j) : \rho_{ij} = k \text{ and } r_{ij} = l\}\right| \tag{8}$$

where q_{kl} is the matrix element of co-ranking matrix Q. ρ_{ij} and r_{ij} is the matrix element of the distance matrix for the original vectors and embedding vectors.

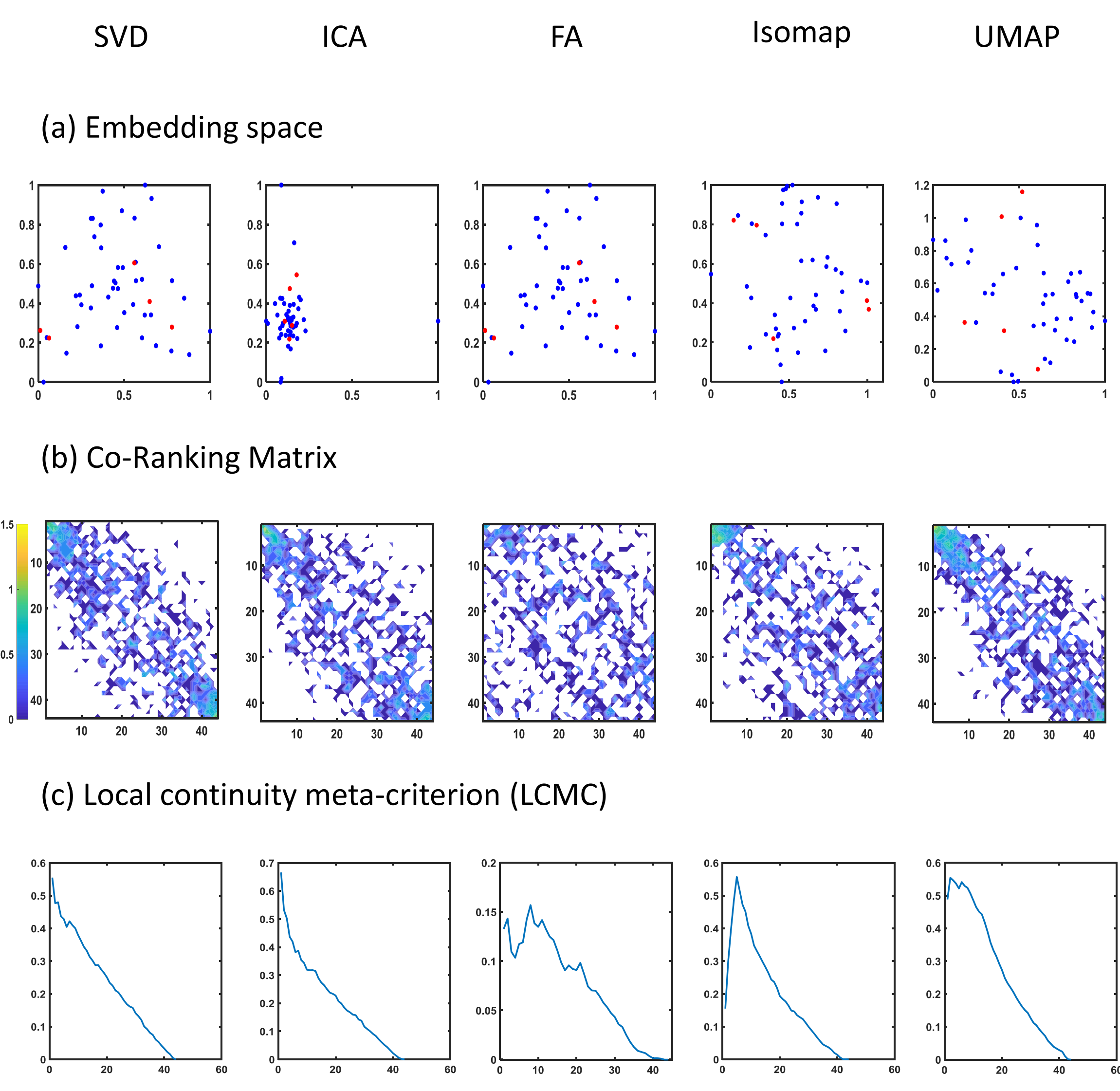


Fig. 6. n_components = 20 for SVD, 20 for ICA, 45 for FA, 10 for ISOMAP, and 10 for UMAP. (a) The embedding space plot of various DR techniques. The first two dimensions are plotted for (b) Co-ranking matrix (c) Local continuity meta criterion (LCMC).

The co-ranking matrix measures the capability of a specific DR algorithm in terms of neighborhood preserving, i.e., the ability to maintain the same distance ranking between the original space and the embedding space. The absolute distance values are not used here since during DR embedding, the distortion and stretching can happen, especially for nonlinear DR algorithms. As a result, the rank of the distance matrix is used to calculate the co-ranking, Q. To further investigate the co-ranking matrix, more pronounced, lumped quantities need to be defined to show the performance of DR algorithms. There have been several different meta-criteria to date [87]–[89]. The prefix meta means the measure is not absolute and ML-problem-dependent. In 2006, Chen *et al.* proposed local continuity meta-criterion (LCMC), which has been cited and used by many researchers in literature. LCMC is defined as following [90]:

$$LCMC(K) = \frac{K}{1-N} + \frac{1}{NK} \sum_{(k,l)\in UL_K} q_{kl} \tag{9}$$

TABLE 2
The Dimensionality Reduction Quality Based on Local Continuity Meta-Criterion and the Associated Parameters

	SVD	ICA	FA	Isomap	UMAP
Kmax	1	1	8	5	2
Q_{local}	0.5551	0.6662	0.1282	0.3794	0.5215
Q_{global}	0.2301	0.2138	0.0636	0.1855	0.2569

where UL_k is the upper left part in the co-ranking matrix [88], N is the number of samples in the dataset, and K runs from 1 to $N-1$, equivalently sweeping the entire coranking matrix Q. NK in the second term is the normalization factor, and $K/1\text{-}N$ is the baseline value for $LCMC(K)$. Higher LCMC values reflect better embedding quality and neighborhood preserving. Fig. 6(c) draws the coranking matrix (Q) and LCMC for various DR algorithms. Since LCMC is a function of K, three parameters (K_{max}, Q_{local}, Q_{global}) are normally defined to characterize the key trends in LCMC curves. K_{max} is the K value where LCMC is at maximum. Q_{local} is the mean over the values at the left of K_{max}, and Q_{global} is the mean over the values at the right of K_{max}. Q_{local} is more important than Q_{global}. Table 2 shows the values of K_{max}, Q_{local}, Q_{global} for various DR algorithms. It can be seen from the co-ranking matrix and LCMC plots, SVD, ICA, and UMAP performs quite well while FA and ISOMAP do not. This is not surprising that UMAP performs well in terms of neighborhood preserving since the proposal of UMAP is to overcome the deficiency of earlier DR algorithms, and using topology to embed vectors can be more advantageous than counting on calculating distance between paired data points in the manifold. Therefore, the neighborhood embedding quality in terms of distance rank preservation is improved using UMAP compared to ISOMAP. At the same time, simple linear DR algorithms such as SVD and ICA can also lead to satisfactory LCMC. On the other hand, it has been observed from Table 1 that in terms of prediction errors, UMAP does not perform well compared to their linear counterparts. In addition, The MSE values of factor analysis (FA) are reasonably low, especially for n_components = 45, but it does not provide low LCMC as shown in Table 2. As a result, it looks like the LCMC in Table 2 does not reflect the entire story of the MSE errors in Table 1. In fact, DR has been a complicated problem, and it can be difficult to find a single figure of merit to quantify the superiority of a specific DR algorithm over the other. Depending on applications and the nature of machine learning problems, one DR algorithm can be better than the other in terms of prediction accuracy. In our application of OPC/EPC, prediction accuracy, as shown in Table 1, is always the ultimate goal while embedding quality in terms of neighborhood preserving only measures the quality of DR in terms of embedding but cannot directly reflect the quality of prediction. Thus, we claim that MSE values in Table 1 should be more important in our case. The underlying reason for the relative performance strength and weakness, for various DR algorithms, will be explained in (10).

In fact, to analyze the relative effectiveness of applying DR to OPC/EPC problems, the total error, ξ, in the prediction can be expressed as:

$$\xi = \xi_{\text{embed}}\left(\xi_{\text{principal}}\right) + \xi_{\text{principal}}\left(\xi_{\text{embed}}\right) + \xi_{\text{fit_DR}}\left(\xi_{\text{embed}}, \xi_{\text{principal}}, \xi_{\text{fit_ML}}\right) + \xi_{\text{fit_ML}}\left(\xi_{\text{embed}}, \xi_{\text{principal}}, \xi_{\text{fit_DR}}\right) \tag{10}$$

The first error ξ_{embed} is the error in neighborhood embedding during dimensionality reduction, as described above, and it is characterized by co-ranking matrix Q and LCMC. The second error $\xi_{\text{principal}}$ represents the information loss during DR, which depends on the complexity of the input vector space. If the input space is highly entangled, applying DR can lead to information loss. In the case that a high complexity space and problems, the input space should not be reduced since every piece of information in the input vector is indispensable. In general, $\xi_{\text{principal}}$ depends on ξ_{embed} because better neighborhood preserving should lead to better input space information preservation. Nevertheless, $\xi_{\text{principal}}$ is more difficult to be assessed using a unified framework such

as in the case of ξ_{embed}. This is due to the fact we do not know what is defined as crucial information in input vector space until we consider output and prediction accuracy. The third error term ξ_{fit_DR} represents the fitting error during dimensionality reduction (DR). This error exists because we only fit the training set during DR and apply the same DR transformation to the test set. To reduce this error, a large training set is beneficial although in many cases, this is unattainable. The fourth error is the error associated with the machine learning model prediction on the test set, ξ_{fit_ML}. The co-ranking matrix and LCMC approach quantify the first error ξ_{embed} and can be regarded as the intrinsic quality of the DR itself. Nonetheless, for other error terms, including $\xi_{principal}$, ξ_{fit_DR}, ξ_{fit_ML}, it can be more difficult to quantify since they are more implicit and can only be known after we consider the prediction accuracy and the test set. As a result, it is not surprising that nonlinear DR algorithms such as UMAP perform worse in Table 1 even if their coranking matrix and LCMC is satisfactory in Table 2. The last thing to note is that ξ_{embed}, $\xi_{principal}$, ξ_{fit_DR}, ξ_{fit_ML} are not independent. Instead, they are entangled, as shown in Eq. 14. Such a complex relation leads to the fact that the assessment should be made when a specific application is targeted.

4. Conclusion

In this work, we demonstrate using image processing and computer vision algorithms to automatically detect, binarize, preprocess, cut, and align the scanning electron micrographs into ML-ready binary images. Such an intelligent manufacturing procedure can promote automation in the photolithography correction process. Implementation of both linear and non-linear dimensionality reduction (DR) algorithms has been performed for pixel-based OPC/EPC using pure machine learning based approaches without counting on Maxwell equations. By training experimental etched pattern directly, we can achieve a consolidated OPC/EPC in one run and reduced trial-and-error time. Pure machine learning approach has the advantage of eliminating the extremely long computation time associated with optical solvers and more importantly, overcoming the problem that no proper physical EPC model is available to date. By applying different DR algorithms, our methodology dramatically reduces the computation time from 4804 s of the baseline calculation to 10 s–200 s. Additionally, the mean square error values have not been significantly increased by incorporating various dimensionality reduction algorithms. The MSE is 0.037 for baseline and changed to 0.035–0.039 for linear DR algorithms and 0.058–0.069 for non-linear DR algorithms. Through real fabrication and model training, we validate the feasibility of using pure-ML based approach and DR algorithms in future advanced photolithography patterning corrections.

References

[1] N. B. Cobb and A. Zakhor, “Fast, low-complexity mask design,” *Proc. SPIE*, vol. 2440, 1995, pp. 313–327.
[2] C. A. Mack, “Reducing proximity effects in optical lithography,” *Japanese J.Appl. Phys.*, vol. 35, pp. 6379–6385, 1996.
[3] J. F. Chen *et al.*, “Practical method for full-chip optical proximity correction,” *Proc. SPIE*, vol. 3051, 1997, pp. 790–803.
[4] R. C. Frye, E. A. Rietman, and K. D. Cummings, “Neural network proximity effect corrections for electron beam lithography,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Conf. Proc.*, 1990, pp. 704–706.
[5] M. Shibuya, “Resolution enhancement techniques for optical lithography and optical imaging theory,” *Opt. Rev.*, vol. 4, Jan. 01, 1997, Art. no. A151.
[6] S. R. J. Brueck and X. Chen, “Spatial frequency analysis of optical lithography resolution enhancement techniques,” *J. Vac. Sci. Technol. B, Microelectronics Nanometer Struct. Proc., Meas., Phenomena*, vol. 17, pp. 908–920, 1999.
[7] L. W. Liebmann, “Layout impact of resolution enhancement techniques: impediment or opportunity?” *Presented at the 2003 Int. Symp. Phys. Des.*, Monterey, CA, USA, 2003.
[8] J. Word and N. B. Cobb, “Enhanced model-based OPC for 65 nm and below,” *Proc. SPIE*, vol. 5567, 2004, pp. 1305–1314.
[9] M. Mukherjee *et al.*, “The problem of optimal placement of sub-resolution assist features (SRAF),” *Proc. SPIE*, vol. 5754, 2005, pp. 1417–1429.
[10] W. C. Huang *et al.*, “Intelligent model-based OPC,” *Proc. SPIE*, vol. 6154, 2006, Art. no. 615436.
[11] I. A. M. Elfadel *et al.*, *Machine Learning in VLSI Computer-Aided Design*. Berlin, Germany: Springer, 2019.
[12] Y. Granik, “Correction for etch proximity: New models and applications,” *Proc. SPIE*, vol. 4346, 2001, pp. 98–112.
[13] S. Shang *et al.*, “Etch proximity correction by integrated model-based retargeting and OPC flow,” *Proc. SPIE*, vol. 6730, 2007, Art. no. 67302G.

[14] A. Poonawala and P. Milanfar, "Mask design for optical microlithography—An inverse imaging problem," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 774–788, Mar. 2007.
[15] Y. Shen *et al.*, "Robust level-set-based inverse lithography," *Opt. Exp.*, vol. 19, pp. 5511–5521, 2011.
[16] X. Ma *et al.*, "A fast and manufacture-friendly optical proximity correction based on machine learning," *Microelectron. Eng.*, vol. 168, pp. 15–26, 2017.
[17] Y. Cai *et al.*, "Application of optical proximity correction technology," *Sci. China Ser. F, Inf. Sci.*, vol. 51, pp. 213–224, Feb. 01, 2008.
[18] X. Ma and G. R. Arce, "Pixel-based OPC optimization based on conjugate gradients," *Opt. Exp.*, vol. 19, pp. 2165–2180, 2011.
[19] S. Shim and Y. Shin, "Etch proximity correction through machine-learning-driven etch bias model," *Proc. SPIE*, vol. 9782, 2016, Art. no. 97820O.
[20] S. Shim and Y. Shin, "Machine learning-guided etch proximity correction," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 1, pp. 1–7, Feb. 2017.
[21] C.-E. Wu *et al.*, "Photoresist 3D profile related etch process simulation and its application to full chip etch compact modeling," *Proc. SPIE*, vol. 9426, 2015, Art. no. 94261Q.
[22] Y. Shen *et al.*, "Efficient optical proximity correction based on semi-implicit additive operator splitting," *Opt. Exp.*, vol. 27, pp. 1520–1528, 2019.
[23] S. Choi, S. Shim, and Y. Shin, "Neural network classifier-based OPC with imbalanced training data," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 38, no. 5, pp. 938–948, May 2019.
[24] X. Ma *et al.*, "Fast optical proximity correction method based on nonlinear compressive sensing," *Opt. Exp.*, vol. 26, pp. 14479–14498, 2018.
[25] Y. Lin *et al.*, "Data efficient lithography modeling with residual neural networks and transfer learning," *presented at the 2018 Int. Symp. Phys. Des.*, Monterey, California, USA, 2018.
[26] S. Lan *et al.*, "Deep learning assisted fast mask optimization," *Proc. SPIE*, vol. 10587, 2018, Art. no. 105870H.
[27] S. Shim, S. Choi, and Y. Shin, "Machine learning (ML)-based lithography optimizations," in *Proc. IEEE Asia Pacific Conf. Circuits Syst.*, 2016, pp. 530–533.
[28] S. Choi *et al.*, "Machine learning (ML)-guided OPC using basis functions of polar Fourier transform," *Proc. SPIE*, vol. 9780, 2016, Art. no. 97800H.
[29] T. Matsunawa *et al.*, "Optical proximity correction with hierarchical Bayes model," *Proc. SPIE*, vol. 9426, 2015, Art. no. 94260X.
[30] K.-S. Luo *et al.*, "SVM based layout retargeting for fast and regularized inverse lithography," *J. Zhejiang Univ. Sci. C*, vol. 15, pp. 390–400, May 01, 2014.
[31] R. Luo, "Optical proximity correction using a multilayer perceptron neural network," *J. Opt.*, vol. 15, 2013, Art. no. 075708.
[32] P. Gao *et al.*, "Optical proximity correction with principal component regression," *Proc. SPIE*, vol. 6924, 2008, Art. no. 69243N.
[33] A. Gu and A. Zakhor, "Optical proximity correction with linear regression," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 2, pp. 263–271, May 2008.
[34] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. "Dimensionality reduction: A comparative review," Tilburg Univ., Tilburg, The Netherlands, Tech. Rep. TiCC-TR 2009-005, 2009. [Online]. Available: https://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf
[35] S. K. Joshi and S. Machchhar, "An evolution and evaluation of dimensionality reduction techniques—A comparative study," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, 2014, pp. 1–5.
[36] V. Laparra, J. Malo, and G. Camps-Valls, "Dimensionality reduction via regression in hyperspectral imagery," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 6, pp. 1026–1036, Sep. 2015.
[37] A. Griparis, D. Faur, and M. Datcu, "A dimensionality reduction approach for the visualization of the cluster space: A trustworthiness evaluation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 2917–2920.
[38] G. S. Reddy, "Dimensionality reduction approach for high dimensional text documents," in *Proc. Int. Conf. Eng. MIS*, 2016, pp. 1–6.
[39] Y. Nozaki and T. Nakamoto, "Nonlinear dimensionality reduction of mass spectrometry data for odor sensing," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst.*, 2015, pp. 190–195.
[40] Y. Pei, "Linear principal component discriminant analysis," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2015, pp. 2108–2113.
[41] X. Jiang *et al.*, "Nonparametrically guided autoencoder with laplace approximation for dimensionality reduction," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 3378–3384.
[42] R. Abdulhammed Alani *et al.*, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, 2019, Art. no. 322.
[43] Matlab Documentation, The Mathworks, 2018.
[44] H. Bay *et al.*, "SURF: Speeded up robust features," *Comput. Vision Image Understanding*, vol. 110, pp. 346–359, 2008.
[45] H. Bay *et al.*, "Speeded-up robust features (SURF)," *Comput. Vision Image Understanding*, vol. 110, pp. 346–359, 2008.
[46] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 510–517.
[47] P. F. Alcantarilla *et al.*, "KAZE features," in *Proc. Eur. Conf. Comput. Vision*, Berlin, Heidelberg, 2012, pp. 214–227.
[48] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Sebastopol, CA, USA: O'Reilly Media, 2008.
[49] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, 2004.
[50] M. Muja and D. G. Lowe, "Fast matching of binary features," *presented at the 2012 9th Conf. Comput. Robot Vision*, 2012.

[51] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *Proc. 4th Int. Conf. Comput. Vision Theory Appl.*, Lisbon, Portugal, 2009, pp. 331–340.
[52] E. Rublee *et al.*, “ORB: An efficient alternative to SIFT or SURF,” in *Proc. Int. Conf. Comput. Vision*, 2011, pp. 2564–2571.
[53] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[54] P. H. S. Torr and A. Zisserman, “MLESAC: A new robust estimator with application to estimating image geometry,” *Comput. Vision Image Understanding*, vol. 78, pp. 138–156, 2000.
[55] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
[56] D. Bradley and G. Roth, “Adaptive thresholding using the integral image,” *J. Graph. Tools*, vol. 12, pp. 13–21, 2007.
[57] P. Soille, *Morphological Image Analysis: Principles and Applications*. Berlin, Germany: Springer, 1999.
[58] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Reading, MA, USA: Addison-Wesley, 1992.
[59] T. Y. Kong and A. Rosenfeld, *Topological Algorithms for Digital Image Processing*. Amsterdam, The Netherlands: Elsevier, 1996.
[60] L. Lam, S.-W. Lee, and C. Y. Suen, “Thinning methodologies—A comprehensive survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 869–885, Sep. 1992.
[61] W. K. Pratt, *Digital Image Processing*, 2nd ed. Hoboken, NJ, USA: Wiley, 1991.
[62] L. Buitinck *et al.*, “API design for machine learning software: experiences from the scikit-learn project,” Sep. 1, 2013, *arXiv:1309.0238*.
[63] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[64] L. McInnes *et al.*, “UMAP: Uniform manifold approximation and projection for dimension reduction,” 2018, *arXiv:1802.03426*.
[65] A. Taufik Fuadi *et al.*, “Singular value decomposition for dimensionality reduction in unsupervised text learning problems,” in *Proc. 2nd Int. Conf. Educ. Technol. Comput.*, 2010, pp. V4-422–V4-426.
[66] J. Liu *et al.*, “Sparse singular value decomposition-based feature extraction for identifying differentially expressed genes,” in *Proc. IEEE Int. Conf. Bioinf. Biomedicine*, 2016, pp. 1822–1827.
[67] A. Winursito *et al.*, “Feature data reduction of MFCC using PCA and SVD in speech recognition system,” in *Proc. Int. Conf. Smart Comput. Electron. Enterprise*, 2018, pp. 1–6.
[68] W. Yan and N. A. Tinker, “Biplot analysis of multi-environment trial data: Principles and applications,” *Can. J. Plant Sci.*, vol. 86, pp. 623–645, 2006.
[69] K. R. Gabriel, “The biplot graphic display of matrices with application to principal component analysis,” *Biometrika*, vol. 58, pp. 453–467, 1971.
[70] T. Postelnicu, J. Biplots, C. Gower, and D. J. Hand, “Chapman & Hall, London, 1996. No. of pages: xvi + 277. Price: £32. ISBN 0-412-71630-5,” *Statist. Med.*, vol. 19, pp. 278–280, 2000.
[71] L. Chen, Y. Liu, and C. Zhu, “Robust tensor principal component analysis in all modes,” in *Proc. IEEE Int. Conf. Multimedia Expo*, 2018, pp. 1–6.
[72] M. R. Islam, R. A. Bhuiyan, N. Ahmed, and M. R. Islam, “PCA and ICA based hybrid dimension reduction model for cardiac arrhythmia disease diagnosis,” in *Proc. IEEE 10th Int. Conf. Humanoid, Nanotechnol., Inf. Technol.,Commun. Control, Environ. Manage.*, 2018, pp. 1–7.
[73] C. O. S. Sorzano and A. Pascual Montano, “A survey of dimensionality reduction techniques,” 2014, *arXiv:1403.2877*.
[74] T. P. Jung, S. Makeig, M. J. McKeown, A. J. Bell, T.-W. Lee, and T. J. Sejnowski, “Imaging brain dynamics using independent component analysis,” *Proc. IEEE*, vol. 89, no. 7, pp. 1107–1122, Jul. 1, 2001.
[75] J. V. Stone, *Independent Component Analysis: A Tutorial Introduction*. Cambridge, MA, USA: MIT Press, 2004.
[76] J. B. Tenenbaum *et al.*, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, pp. 2319–2323, 2000.
[77] D. J. Bartholomew, F. Steele, J. Galbraith, and I. Moustaki, *Analysis of Multivariate Social Science Data*. Evanston, IL, USA: Routledge, 2008.
[78] R. B. Cattell, *Factor Analysis: An Introduction and Manual for the Psychologist and Social Scientist*. Oxford, U.K.: Harper, 1952.
[79] B. Fruchter, *Introduction to Factor Analysis*. Oxford, U.K.: Van Nostrand, 1954.
[80] R. B. Cattell, *The Scientific Use of Factor Analysis in Behavioral and Life Sciences*, New York, NY, USA: Plenum, 1978.
[81] D. Child, *The Essentials of Factor Analysis*, 3rd ed. New York, NY, USA: Bloomsbury Academic, 2006.
[82] R. L. Gorsuch, *Factor Analysis*, 2nd ed. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1983.
[83] R. P. McDonald, *Factor Analysis and Related Methods*. Hillsdale, NJ, USA: Psychology Press, 1985.
[84] M. Gong *et al.*, “Density-sensitive evolutionary clustering,” in *Advances in Knowledge Discovery and Data Mining*. Berlin, Germany: Springer, 2007, pp. 507–514.
[85] G. Xin, D.-C. Zhan, and Z.-H. Zhou, “Supervised nonlinear dimensionality reduction for visualization and classification,” *IEEE Trans. Syst., Man, Cybern., Part B (Cybern.)*, vol. 35, no. 6, pp. 1098–1107, Dec. 2005.
[86] J. Nilsson *et al.*, “Approximate geodesic distances reveal biologically relevant structures in microarray data,” *Bioinformatics*, vol. 20, pp. 874–880, 2004.
[87] J. A. V. Lee and Michel, “Rank-based quality assessment of nonlinear dimensionality reduction,” in *Proc. 16th Eur. Symp. Artif. Neural Netw.*, Bruges, Belgium, 2008, pp. 49–54.
[88] J. A. Lee and M. Verleysen, “Quality assessment of dimensionality reduction: Rank-based criteria,” *Neurocomputing*, vol. 72, pp. 1431–1443, 2009.
[89] G. Kraemer, M. Reichstein, and M. D. Mahecha, “dimRed and coRanking—Unifying dimensionality reduction in R,” *R J.*, vol. 10, pp. 342–358, 2018.
[90] L. Chen, “Local multidimensional scaling for nonlinear dimensionality reduction, graph layout, and proximity analysis,” Ph.D. dissertation, Managerial Sci. Appl. Econom., Univ. Pennsylvania, Philadelphia, PA, USA, 2006.