Deep Learning for Mask Synthesis and Verification: A Survey (Invited Paper)

Yibo Lin
CECA, CS Department
Peking University
yibolin@pku.edu.cn

ABSTRACT

Achieving lithography compliance is increasingly difficult in advanced technology nodes. Due to complicated lithography modeling and long simulation cycles, verifying and optimizing photomasks becomes extremely expensive. To speedup design closure, deep learning techniques have been introduced to enable data-assisted optimization and verification. Such approaches have demonstrated promising results with high solution quality and efficiency. Recent research efforts show that learning-based techniques can accomplish more and more tasks, from classification, simulation, to optimization, etc. In this paper, we will survey the successful attempts of advancing mask synthesis and verification with deep learning and highlight the domain-specific learning techniques. We hope this survey can shed light on the future development of learning-based design automation methodologies.

1 INTRODUCTION

Mask synthesis and verification are critical to the manufacturability and yield in advanced technology nodes [1]. With the continuous shrinking of feature sizes, lithographic masks need to be carefully designed and verified for good printability before manufacture.

Mask verification takes a mask design as input and simulates the output patterns to verify whether they match the target ones. It usually includes optical and photoresist (use *resist* for short) simulation. Mask synthesis operates in an inverse direction by taking a target layout as input and outputting the actual mask design that can produce the target patterns after lithography. Typical mask synthesis techniques include optical proximity correction (OPC), sub-resolution assist feature (SRAF) generation, and other inverse lithography technologies (ILT). Figure 1 shows a rough flow for mask synthesis and verification. Mask synthesis usually comes before mask verification, while in practice, the former often needs to iteratively interact with the latter for better performance.

Both mask synthesis and verification algorithms are computationally expensive, while they pursue high performance and accuracy at a nanometer scale in advanced nodes. Mask verification requires complicated optical and resist models for accurate simulation of printed patterns. Such lithography simulation also serves as a subroutine in typical mask synthesis techniques like SRAF generation and OPC to guide the optimization. In practice, masks are clipped into local regions with several square micrometers for efficiency. However, the area of integrated circuits nowadays can scale up to hundreds or thousands of square millimeters, consisting of billions of such clips. Thus, developing efficient and high-quality mask synthesis and verification techniques is urgently desired.

Recent advances in deep learning bring new opportunities to speedup mask synthesis and verification, and meanwhile maintaining high quality and accuracy. For example, the techniques developed to solve computer vision tasks are often promising in mask-related problems, as a mask can be naturally represented as image-like data, similar to the case in computer vision. Existing deep learning applications in mask-related problems can be roughly categorized into four major tasks: lithography modeling, mask optimization, printability prediction, and test pattern generation. As illustrated in Figure 1, these tasks cover different stages in the flow, most of which essentially perform cross-stage modeling. In this paper, we review the recent progress in solving such tasks with emerging deep learning techniques, formalizing the problems, highlighting the challenges, and summarizing the current state-of-the-art results.

The rest of the paper will be organized as follows. Section 2 to Section 5 will survey the deep learning applications in lithography modeling, mask optimization, printability prediction, and test pattern generation, respectively. Section 6 will conclude this paper with future directions.

ASPDAC '21, January 18–21, 2021, Tokyo, Japan

ACM ISBN 978-1-4503-7999-1/21/01…$15.00
https://doi.org/10.1145/3394885.3431624

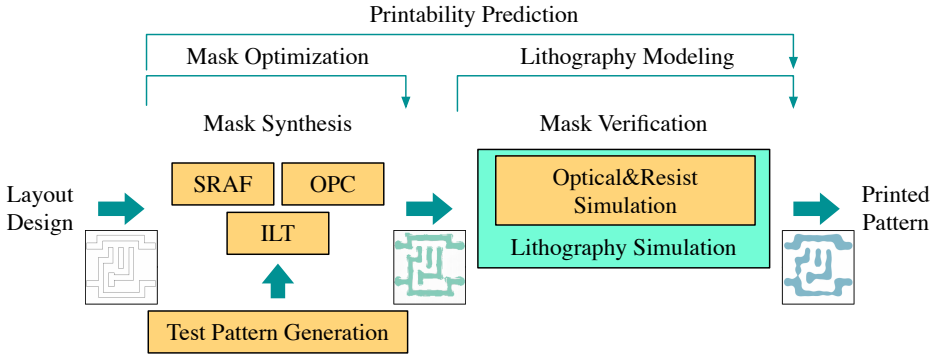


Figure 1: Flow for mask synthesis and verification.

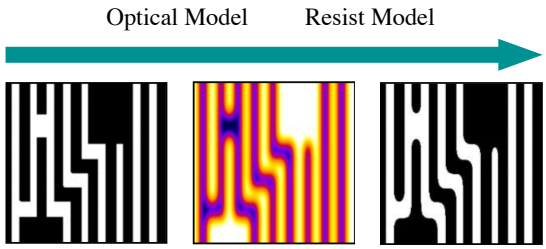


Figure 2: Lithography simulation [2].

2 LITHOGRAPHY MODELING

In this section, we review the lithography modeling problem and survey the recent progress in related deep learning applications.

Lithography simulation requires lithography models to accomplish the computation. In a typical simulation flow, we require optical and resist models to obtain the final printed patterns, as shown in Figure 2. The optical model computes the aerial image from the input mask patterns, i.e., the light intensity map near the surface of the resist, and the resist model determines the slicing thresholds on the aerial image and calculates the resist patterns.

2.1 Optical Models

Aerial image computation adopts thin mask approximation or thick mask approximation [3, 4]. Thin mask approximation (also known as the Kirchhoff approximation) considers the mask as an infinitely thin object, ignoring the impacts on the amplitudes, phases, and polarization of the transmitted light from the three-dimensional (3D) structure of the mask. When the feature sizes become comparable to the light wavelength, the accuracy of thin mask approximation is no longer acceptable. Thick mask approximation is then required to consider various factors like mask topography effects in the lithography process [5]. While thin mask approximation is efficient, thick mask approximation is much more computationally expensive, e.g., 11 seconds v.s. 5 minutes on a $2\mu m \times 2\mu m$ mask clip on average using Synopsys Sentaurus Lithography [6].

To speedup thick mask approximation, Ye et al [4] formulate a 3D aerial image learning problem by taking mask patterns as input and predict a series of 2D aerial images at different resist heights to capture the spatial image intensity distribution inside the resist bulk. The objective for learning is to minimize the pixel-wise normalized root mean square error (NRMSE) defined as follows,

$$\text{NRMSE} = \frac{\left\|\hat{I} - I\right\|_F}{\|I\|_F}, \tag{1}$$

where $\|\cdot\|_F$ represents the Frobenius norm, I represents the original aerial image, and $\hat{I}$ represents the predicted aerial image.

The problem formulation can be interpreted as a multi-domain image translation task from an input image to a series of output images, where the output images are highly correlated and correspond to the spatial light intensity distribution of a 3D aerial image. Ye et al then propose the *TEMPO* framework based on conditional generative adversarial network (GAN) [7, 8], to generate images at different heights. Figure 3 illustrates the overall architecture of the model. The model consists of a generator and a discriminator like the typical GAN architecture. The generator takes a mask pattern and an aerial image from thin mask as input, encodes them into latent space, and decodes the latent code to generate the 2D aerial image at a specific height. The height is controlled by the additional label (a *one-hot* vector) concatenated into the latent code. The generator is trained together with a discriminator identifying whether an aerial image is generated (fake) or not. The loss function to train the model can be written as [8, 9],

$$\begin{aligned} L_{\text{GAN}} = {} & \mathbb{E}_{x,y}[\log D(x,y)] \\ & + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))] \\ & + \lambda \cdot \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1], \end{aligned} \tag{2}$$

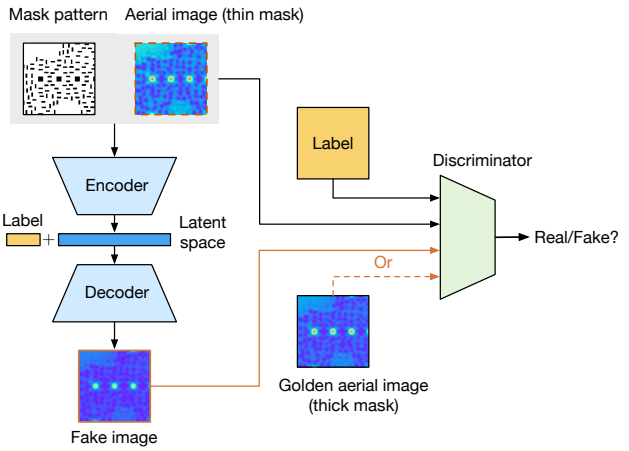


Figure 3: Overview of the TEMPO model [4].

where x is a sample in the input domain, y is its corresponding sample in the output domain, $G(\cdot)$ represents the generator (including encoder and decoder), $D(\cdot)$ represents the probability of a sample being real (not generated by G), $\mathbb{E}_x$ represents the expectation over the input data x, z is a random noise vector as a seed for image generation, and λ is the weight parameter.

Eventually, the model can achieve an average of 27× speedup over rigorous simulation [6] with the NRMSE around 1.79%. Such a speedup includes the runtime required for generating the aerial image using thin mask approximation, which contributes to high accuracy. If the accuracy can be further sacrificed, the framework supports to omit the dependency to the thin mask aerial image and achieve more than 1000× speedup with the average NRMSE of 4.23%.

2.2 Resist Models

Once we obtain the aerial images from the optical models, the task for resist modeling is to compute the resist patterns accurately and efficiently. Watanabe et al [2] discover that convolutional neural networks (CNNs) have the potential to achieve higher accuracy than the conventional compact resist models in Mentor Calibre [10]. However, achieving high accuracy with CNNs requires a large amount of training data, which is often not easy to get at the early stage of process development. Thus, Lin et al [11] formulate a data-efficient learning problem leveraging the data from old technology nodes to assist model training at the target node. In this formulation, they construct CNN models taking an aerial image as input and outputting the slicing thresholds at the boundary of the target pattern. Two techniques are investigated.

Transfer learning. We can regard the old technology node as the source domain with enough data to train an accurate model and the target technology node as the target domain that does not have enough data. The knowledge transfer can be done by finetuning the network weights of the source domain model when training with target domain data. To control the degree of knowledge transfer, we can also fix the first k layers and only finetune the rest ones, where k is a user-defined variable.

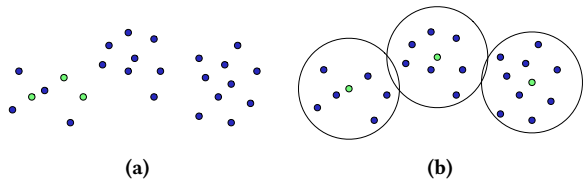


Figure 4: Example of active data selection. (a) Bad data selection and (b) K-Medoids clustering based selection. Three selected points are highlighted. Circles denote three clusters centered by selected points [11].

Active data selection. The distribution of training data also affects the accuracy and generalization of a model. Figure 4(a) illustrates an example of bad data selection, which is likely to cause overfitting. A better approach is to perform clustering and choose the centers of each cluster as the training data, as shown in Figure 4(b). Note that the data samples in the plot denote features, not labels. Thus, after clustering, we can query the labels for the centers to build the training dataset.

With these techniques, a 3-10× reduction on the amount of required training data is reported when achieving the industrial strength of accuracy on datasets equivalent to $10nm$ and $7nm$ technology nodes [11]. Meanwhile, around 10× speedup over rigorous simulation can be achieved by replacing the resist model with neural networks.

2.3 End-to-end Models

While there are individual attempts to introduce deep learning for optical and resist models, it is unclear whether we can directly perform end-to-end modeling, i.e., predicting the resist patterns from the mask patterns, without generating the intermediate aerial images.

Ye et al [12] propose the *LithoGAN* model to tackle this problem. Since both the mask patterns and the resist patterns can be represented as images, one can still formulate the problem into an image translation task and leverage generative models to solve the problem.

The major challenge is the limited resolution in existing learning approaches. Consider the typical critical dimension (CD) around 1-2μm and the target prediction error less than $1nm$. In other words, each mask clip needs to be at least $1\mu m \times 1\mu m$ to capture the peripheral information. Once translated into an image, it should contain at least 1000×1000 pixels to achieve the resolution of one pixel for $1nm$. However,

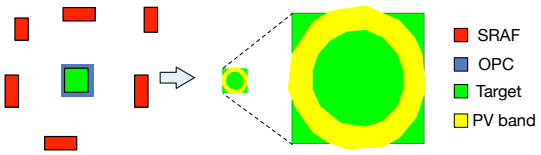


Figure 5: Printing with both OPC and SRAF resulting in better process variation band (PVBand) and edge placement error (EPE) [13].

even with this construction, misprediction of one or two pixels can lead to errors larger than $1nm$, which is extremely challenging for deep learning models.

To tackle this challenge, they zoom in the resist patterns and only predict a small region at the center of the mask clip and represent it with a full image. In this way, super-resolution at the output image can be achieved. They further develop a calibration network to adjust the center of the predicted pattern, as the GAN model is not able to capture the locations of the patterns well. On $7nm$ contact designs, eventually, an average of less than $1nm$ edge displacement errors can be achieved with more than 1800× smaller runtime than rigorous simulation [6, 12].

3 MASK OPTIMIZATION

If we define lithography simulation as a mapping $f : X \rightarrow Y$, where X denotes the mask patterns and Y denotes the printed patterns, then mask optimization essentially solves the inverse mapping $f^{-1} : Y \rightarrow X$, as we are given target resist patterns and try to find the mask design producing the target after lithography process. As the mapping f is very complicated, obtaining f^{-1} is non-trivial. Typical mask optimization techniques include SRAF generation, OPC, and other inverse lithography techniques.

Figure 5 shows an example of printing a contact with both OPC and SRAF that enhance the lithography resolution. The red rectangles in the plot denote SRAFs and the blue rectangle denotes the OPC solution. These SRAFs are too small to be printed, but they will help the imaging of target patterns. OPC adjusts the edges of target patterns for light intensity compensation. Conventional methods for OPC and SRAF generation perform model-based optimization that invokes lithography simulation as a subroutine, which can achieve high solution quality but are in general slow.

3.1 SRAF Generation

Xu et al [14] approach the SRAF generation problem as a pixel-wise prediction task on a mask clip. By representing a mask clip as an image, they first predict the probability of whether each pixel needs to insert an SRAF from its surrounding features and obtain an SRAF probability map as

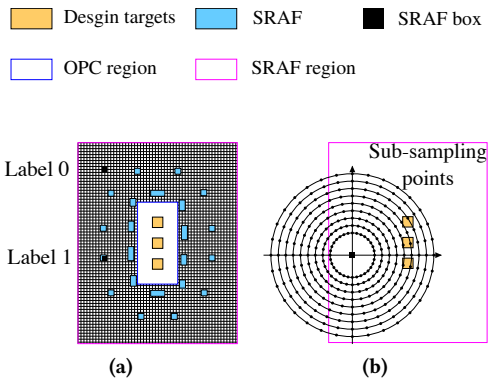
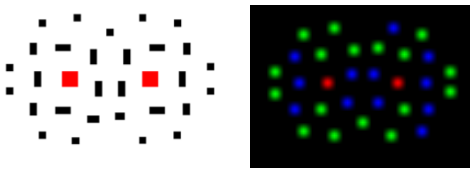


Figure 6: (a) SRAF label sampling; (b) concentric circle area sampling (CCAS) at one pixel [14].

shown in Figure 6. They then insert SRAFs based on the locations of probability maxima at each consecutive insertion region honoring the design rules. The learning problem can be tackled as a classification task with conventional machine learning techniques like logistic regression or support vector machine (SVM). 10× speedup on 1-2μm^2 mask clips and 3× speedup on 100μm^2 clips are reported with 13.5% better EPE and 1.31% PVBand degradation compared with the model-based approach [10]. Geng et al [13] further improve PVBand and EPE by improving the classification accuracy with dictionary learning, and solve an integer linear programming (ILP) problem for the actual SRAF insertion step considering the probability map and design rules.

The above approaches follow the pixel-wise prediction nature that requires to invoke model prediction by m^2 times for an $m \times m$ mask clip, which may not be efficient enough when it comes to complicated models. Alawieh et al [15] cast the learning problem into an image translation problem and develop a conditional GAN based model that can obtain the full probability map with one prediction. However, achieving high-quality SRAF solutions is not easy for GANs, as they usually cannot handle the sharp edges of SRAF shapes well. Thus, Alawieh et al propose a heatmap encoding method that replaces each SRAF shape with a Gaussian-like excitement, as shown in Figure 7. The GAN model first performs image domain translation to the heatmaps for capturing the global distributions of SRAFs, and the SRAF solutions can then be obtained by decoding the predicted heatmaps. Such an encoding method avoids the challenge of generating sharp edges with GAN models and allow efficient parallel decoding. By introducing the CycleGAN model [16], they report 150× speedup over the model-based approach [10] with 2.5% better PVBand and 8.5% EPE degradation. Despite the larger EPE degradation, it is argued that PVBand is a more important metric to judge the quality of SRAF generation, as EPE can be further improved with better OPC strategies [14, 17].



(a) (b)

Figure 7: (a) Original mask clip and (b) the encoded heatmap [15].

3.2 OPC and ILT

Typical OPC algorithms perturb the edge segments of target patterns to enhance lithography resolution. For example, conventional segment-based OPC algorithms iteratively perturb the edge segments of target patterns and perform lithography simulation until convergence. Inverse lithography technologies, on the other hand, refer to more generic optimization that generalizes SRAF generation and OPC, such as the gradient-based optimization on mask pixels [18].

As the mask clips can be represented as images, we can again formulate an image translation task to generate the OPC results from given mask clips. The image translation task can be solved with generative models like GANs [19, 20]. Previously, deep learning models are used to produce intermediate solutions of OPC to reduce conventional OPC iterations. Roughly 2× speedup can be achieved, as the conventional iterations can be reduced by half [20]. Such a methodology is not limited to OPC only. It is compatible with ILT as well since we can replace the training dataset with ILT results and the conventional OPC engines with conventional ILT engines.

Recently, Jiang et al the [21] propose *Neural-ILT* framework that replaces the backbone of ILT with a neural network, i.e., U-Net [22], as shown in Figure 8. They introduce a differentiable lithography simulation layer that can simulate the printed patterns at *nominal*, *min*, *max* process conditions given a mask input. Then, they can train the network with the following loss,

$$L = L_{ilt} + L_{cplx} = \alpha \left\| \mathbf{Z_{norm}} - \mathbf{Z_t} \right\| + \beta \left\| \mathbf{Z_{min}} - \mathbf{Z_{max}} \right\|, \quad (3)$$

where $\mathbf{Z_t}$ denotes the input target pattern, $\hat{\mathbf{M}}$ denotes the optimized mask, and $\mathbf{Z_{norm}}, \mathbf{Z_{min}}, \mathbf{Z_{max}}$ denote the printed patterns at *nominal*, *min*, *max* process conditions, respectively. The first loss term L_{ilt} minimizes the differences between target patterns and printed patterns. The second loss term L_{cplx} optimizes for the mask complexity, i.e., minimizes small shapes, as they observe these shapes usually cannot be printed in the *min* process condition. The hyper-parameters α, β are user-defined. As the lithography simulation layer is differentiable, the network can be trained with backpropagation. With such a setup, training the Neural-ILT framework is

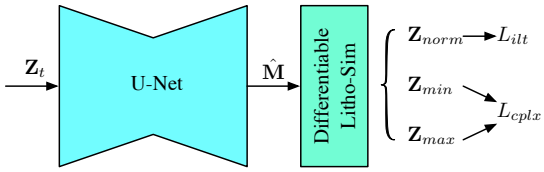


Figure 8: The Neural-ILT framework that replaces the backbone of ILT with a neural network [21].

equivalent to solving ILT with a specially designed objective in Equation (3). Thus, given any new layout clip with target patterns, we can perform training to obtain a high-quality solution $\hat{\mathbf{M}}$ as the optimized mask. Meanwhile, we can also pretrain the U-Net backbone and finetune for new layout clips with faster convergence. Experiments on ICCAD 2013 benchmarks [23] demonstrate 12.5% smaller mean square errors than the conventional ILT engine on CPU [18] with 70× speedup. They also implement a GPU-accelerated ILT engine and show that Neural-ILT can achieve higher solution quality at the same scale of runtime.

Most of the current works focus on small mask clips. However, in practice, we need to optimize the mask for full-chip designs. To tackle mask optimization at a full-chip scale, Chen et al propose the *DAMO* framework with stitchless full-chip splitting for layouts of any size [24]. The algorithm consists of a coarse step based on DBSCAN clustering and a fine step based on KMeans++ clustering to generate clips from a full-chip layout. Figure 9 provides an example of the splitting steps. In the coarse step, the DBSCAN algorithm locates the regions of high via density that are apart from other low-density regions. After DBSCAN clustering, every via pattern is assigned to a coarse cluster. Within each coarse cluster, KMeans++ clustering is performed to find the best splitting windows with the window sizes and a given maximum number of vias in each window. The window size is set to $1024 \times 1024nm^2$. The splitting algorithms enable the deep-learning-based mask optimization algorithms to work at a full-chip scale. They also develop high-resolution mask generation networks and lithography simulation networks for accurate and efficient mask optimization. The experimental results on the via layers of the ISPD 2019 contest [25] demonstrate 9.8% better mean square errors, 0.7% smaller PVBand, and 5× speedup compared with Mentor Calibre [10]. This is the first time for deep-learning-based mask optimization engines demonstrating even better solution quality than industrial tools.

4 PRINTABILITY PREDICTION

To evaluate how good design patterns can be printed, as shown in Figure 1, we need to go through mask synthesis and mask verification, which is often too expensive for feedback at the early stages. Therefore, printability prediction aims

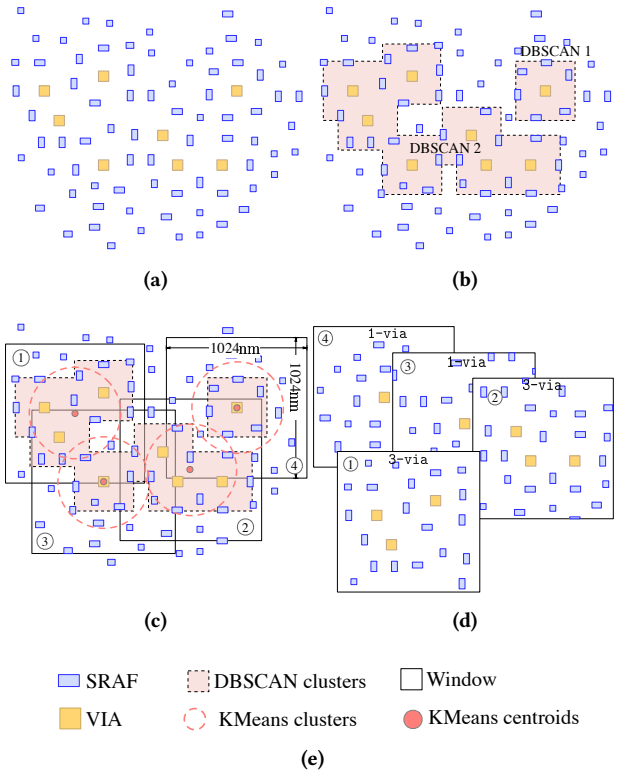


Figure 9: Full-chip splitting algorithm in DAMO [24]. (a) Initial layout; (b) coarse step: full-chip DBSCAN clustering; (c) fine step: KMeans++ on each coarse cluster to get fine clusters; (d) the split clips.

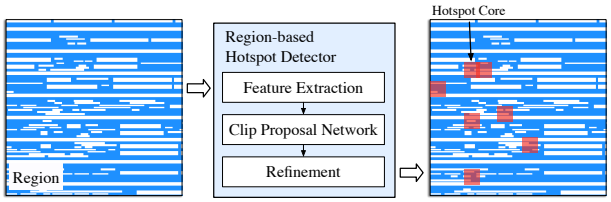


Figure 10: Region-based hotspot detection [38].

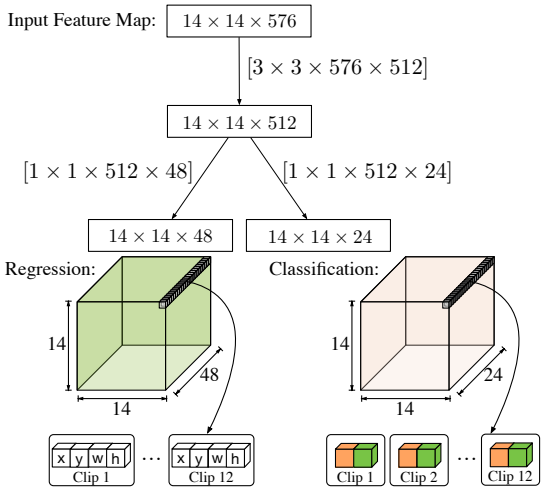


Figure 11: Two branches of the clip proposal network to detect multiple hotspots [38].

at early-stage evaluation of the quality of design patterns without going through mask synthesis and verification.

4.1 Hotspot Detection

A typical printability prediction problem is hotspot detection. That is, given a mask clip before mask synthesis, predict whether the mask contains patterns failed to print, e.g., causing short or open. We denote the failure as a *hotspot*, and those successfully printed patterns as *non-hotspots*. Hotspot detection is an imbalanced classification task, as most of the patterns are non-hotspots and only a small amount of patterns are hotspots. In general, the task requires to detect all hotspots, but allows a few false alarms, i.e., regarding non-hotspots as hotspots. Hence, the key to hotspot detection is how to learn from imbalanced data with a high *true positive rate* (TPR) and a low *false positive rate* (FPR).

There has been a long history of developing machine learning based hotspot detectors, such as support vector machine (SVM) models [26], hierarchical Bayesian models [27]. Yang et al [28, 29] systematically explore CNNs for this imbalanced learning problem. As representing a mask clip without information loss needs a high-definition image, they propose to perform discrete cosine transformation (DCT) to the image and omit the high-frequency components to reduce the image dimensions to 19×19 [28]. Then, they propose a biased training method that encodes the labels of non-hotspots as a one-hot vector $[1-\epsilon, \epsilon]$ and that of hotspots as $[0, 1]$, where $\epsilon \in [0, 0.5)$ is the bias. The purpose of this encoding is to reduce the confidence of predicting non-hotspots during training. They prove this contributes to higher accuracy during inference. They demonstrate 95.5% TPR on average and 14.9% FPR on ICCAD 2012 contest benchmarks [30]. Later, besides exploring better network architectures [31–33], studies have investigated online learning [34], active learning [35, 36], semi-supervised learning [37], etc.

Recently, Chen et al [38] propose to detect multiple hotspots in a mask region, as shown in Figure 10. This problem essentially corresponds to the object detection problem. They propose an inception-based network containing two branches, as shown in Figure 11: a classification branch and a regression branch. The classification predicts whether a clip is a hotspot or non-hotspot, and the regression branch predicts the location and shape of the clip as $[x, y, w, h]$. Eventually, they report an average of 95.8% accuracy on ICCAD 2016 contest benchmarks [39].

4.2 Litho-Aware Layout Design

With printability prediction, we can perform litho-aware layout design at early stages like routing and layout decomposition. For example, Ding et al [40] consider lithography compliance during routing. They build a machine learning model that can predict lithography hotspots with partial routing solutions and use this model to guide the routing algorithm. Zhong et al [41] explore simultaneous layout decomposition and mask optimization, by generating a batch of decomposition solutions and extracting the ones with the best printability.

5 TEST PATTERN GENERATION

All the aforementioned studies assume given layout clips. However, in the real design flow, due to long logic-to-chip cycles, obtaining versatile layout clips is not easy. Therefore, process engineers usually rely on generated layout patterns to evaluate the performance and robustness of a process or mask synthesis algorithms, especially at the stage of exploring new process recipes. To achieve reasonable evaluation, the generated layout patterns need to satisfy the following: 1) no design rule violations; 2) as diverse as possible. In this way, these patterns can be used to evaluate how good a process recipe behaves at various corners.

To tackle this problem, Yang et al [42] link this problem with constrained random image generation. Instead of conditional image generation in previous problems, image generation based on random noise can be exploited. They use a transforming convolutional auto-encoder (TCAE) to encode a layout pattern into latent space and perturb the latent vector to obtain variations in generated images. This approach alone cannot guarantee to generate legal patterns, so they introduce a squish representation to encode the layout topologies rather than the regular pixel representation. The TCAE model also generates squish representations for layout topologies and then they can solve a linear system to obtain the eventual patterns. Figure 12 shows one example of generating 1000 topologies from one topology. Note that the approach still generates illegal patterns that need to be filtered. It is reported that this method can generate patterns with higher diversity than the state-of-the-art industrial tool.

6 CONCLUSION

In this paper, we have surveyed the recent progress in deep-learning-based mask synthesis and verification, especially in mask optimization, lithography modeling, printability prediction, and test pattern generation. As a mask can be naturally represented with an image, deep learning techniques developed for image classification and translation are promising to achieve speedup in orders of magnitude. Meanwhile, due to the uniqueness of mask problems, i.e., rectilinear pattern

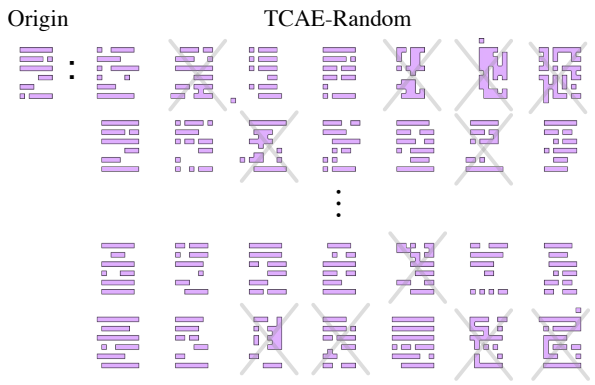


Figure 12: Generate 1000 topologies (~400 legal) from one topology in the existing pattern library [42].

shapes, design rules, high image resolution, and imbalanced label distributions, dedicated customization in feature representation and network architectures is necessary to achieve high-quality solutions. We also see a trend of integrating components from conventional lithography simulation into neural networks for end-to-end learning and optimization, like LithoGAN and Neural-ILT.

In the future, we believe the following directions are worth exploring. 1) With the demonstration of promising results on mask clips, developing algorithms with full-chip support is necessary to show the capability of industrial integration. 2) Most proposed lithography modeling and SRAF generation techniques are demonstrated on contact layers and extending to metal layers is necessary for further adoption. 3) Obtaining enough training datasets is always expensive and thus publicly available large datasets like ImageNet are meaningful to advance the state-of-the-art. 4) Learning with few data and model migration between technology nodes would be practically useful in industrial flows.

REFERENCES

[1] C. A. Mack, "Thirty years of lithography simulation," in *Optical Microlithography XVIII*, vol. 5754. International Society for Optics and Photonics, 2005, pp. 1–12.

[2] Y. Watanabe, T. Kimura, T. Matsunawa, and S. Nojima, "Accurate lithography simulation model based on convolutional neural networks," in *Photomask Japan 2017: XXIV Symposium on Photomask and Next-Generation Lithography Mask Technology*, vol. 10454. International Society for Optics and Photonics, 2017, p. 104540I.

[3] X. Ma, X. Zhao, Z. Wang, Y. Li, S. Zhao, and L. Zhang, "Fast lithography aerial image calculation method based on machine learning," *Applied Optics*, vol. 56, no. 23, pp. 6485–6495, 2017.

[4] W. Ye, M. B. Alawieh, Y. Watanabe, S. Nojima, Y. Lin, and D. Z. Pan, "TEMPO: Fast mask topography effect modeling with deep learning," in *ACM International Symposium on Physical Design (ISPD)*, Taipei, Taiwan, September 2020.

[5] A. K. Wong and A. R. Neureuther, "Mask topography effects in projection printing of phase-shifting masks," *IEEE Transactions on Electron Devices (TED)*, vol. 41, no. 6, pp. 895–902, 1994.

[6] "Synopsys Sentaurus Lithography," https://www.synopsys.com/silicon/mask-synthesis/sentaurus-lithography.html.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Conference on Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976.

[9] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[10] Mentor Graphics, "Calibre verification user's manual," 2008.

[11] Y. Lin, M. Li, Y. Watanabe, T. Kimura, T. Matsunawa, S. Nojima, and D. Z. Pan, "Data efficient lithography modeling with transfer learning and active data selection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 38, no. 10, pp. 1900–1913, 2018.

[12] W. Ye, M. B. Alawieh, Y. Lin, and D. Z. Pan, "LithoGAN: End-to-end lithography modeling with generative adversarial networks," in *ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.

[13] H. Geng, W. Zhong, H. Yang, Y. Ma, J. Mitra, and B. Yu, "SRAF insertion via supervised dictionary learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2019.

[14] X. Xu, Y. Lin, M. Li, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, and D. Z. Pan, "Subresolution assist feature generation with supervised data learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 37, no. 6, pp. 1225–1236, 2017.

[15] M. B. Alawieh, Y. Lin, Z. Zhang, M. Li, Q. Huang, and D. Z. Pan, "GAN-SRAF: Sub-Resolution Assist Feature Generation using Generative Adversarial Networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2020.

[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[17] C. Mack, *Fundamental Principles of Optical Lithography: The Science of Microfabrication*. John Wiley & Sons, 2008.

[18] 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.

[19] S. Lan, J. Liu, Y. Wang, K. Zhao, and J. Li, "Deep learning assisted fast mask optimization," in *Optical Microlithography XXXI*, vol. 10587. International Society for Optics and Photonics, 2018, p. 105870H.

[20] H. Yang, S. Li, Z. Deng, Y. Ma, B. Yu, and E. F. Young, "GAN-OPC: Mask optimization with lithography-guided generative adversarial nets," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 39, no. 10, pp. 2822–2834, 2019.

[21] B. Jiang, L. Liu, Y. Ma, H. Zhang, E. F. Young, and B. Yu, "Neural-ilt: Migrating ilt to nerual networks for mask printability and complexity co-optimizaton," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, November 2020.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[23] 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.

[24] G. Chen, W. Chen, Y. Ma, H. Yang, and B. Yu, "Damo: Deep agile mask optimization for full chip scale," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, November 2020.

[25] W.-H. Liu, S. Mantik, W.-K. Chow, Y. Ding, A. Farshidi, and G. Posser, "ISPD 2019 initial detailed routing contest and benchmark with advanced routing rules," in *ACM International Symposium on Physical Design (ISPD)*, 2019, pp. 147–151.

[26] K.-S. Luo, Z. Shi, X.-L. Yan, and Z. Geng, "SVM based layout retargeting for fast and regularized inverse lithography," *Journal of Zhejiang University SCIENCE C*, vol. 15, no. 5, pp. 390–400, 2014.

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

[28] H. Yang, L. Luo, J. Su, C. Lin, and B. Yu, "Imbalance aware lithography hotspot detection: a deep learning approach," *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol. 16, no. 3, p. 033504, 2017.

[29] H. Yang, J. Su, Y. Zou, Y. Ma, B. Yu, and E. F. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 38, no. 6, pp. 1175–1187, 2018.

[30] J. A. Torres, "Iccad-2012 cad contest in fuzzy pattern matching for physical verification and benchmark suite," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2012, pp. 349–350.

[31] J. Chen, Y. Lin, Y. Guo, M. Zhang, M. B. Alawieh, and D. Z. Pan, "Lithography hotspot detection using a double inception module architecture," *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol. 18, no. 1, p. 013507, 2019.

[32] Y. Jiang, F. Yang, B. Yu, D. Zhou, and X. Zeng, "Efficient layout hotspot detection via binarized residual neural network ensemble," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2020.

[33] V. Borisov and J. Scheible, "Research on data augmentation for lithography hotspot detection using deep learning," in *34th European Mask and Lithography Conference*, vol. 10775. International Society for Optics and Photonics, 2018, p. 107751A.

[34] H. Zhang, B. Yu, and E. F. Young, "Enabling online learning in lithography hotspot detection with information-theoretic feature optimization," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 1–8.

[35] W. Ye, M. B. Alawieh, M. Li, Y. Lin, and D. Z. Pan, "Litho-gpa: Gaussian process assurance for lithography hotspot detection," in *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, Florence, Italy, March 2019.

[36] H. Yang, S. Li, C. Tabery, B. Lin, and B. Yu, "Bridging the gap between layout pattern sampling and hotspot detection via batch active learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2020.

[37] Y. Chen, Y. Lin, T. Gai, Y. Su, Y. Wei, and D. Z. Pan, "Semi-supervised hotspot detection with self-paced multi-task learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, April 2019.

[38] R. Chen, W. Zhong, H. Yang, H. Geng, F. Yang, X. Zeng, and B. Yu, "Faster region-based hotspot detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2020.

[39] R. O. Topaloglu, "Iccad-2016 cad contest in pattern classification for integrated circuit design space analysis and benchmark suite," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2016, pp. 1–4.

[40] D. Ding, J.-R. Gao, K. Yuan, and D. Z. Pan, "AENEID: a generic lithography-friendly detailed router based on post-ret data learning and hotspot detection," in *ACM/IEEE Design Automation Conference (DAC)*, 2011, pp. 795–800.

[41] W. Zhong, S. Hu, Y. Ma, H. Yang, X. Ma, and B. Yu, "Deep learning-driven simultaneous layout decomposition and mask optimization," in *ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[42] H. Yang, P. Pathak, F. Gennari, Y.-C. Lai, and B. Yu, "DeePattern: Layout pattern generation with transforming convolutional auto-encoder," in *ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.