Automatic correction of lithography hotspots with a deep generative model

Woojoo Sim[ab*], Kibok Lee[a*], Dingdong Yang[ac], Jaeseung Jeong[b], Ji-Suk Hong[b], Sooryong Lee[b], and Honglak Lee[a]

[a]University of Michigan, 2260 Hayward St., Ann Arbor, MI, USA
[b]Samsung Electronics, 1 Samsungjeonja-ro, Hwaseong-si, Gyeonggi-do, Korea
[c]Brown University, 115 Waterman St., Providence, RI, USA

ABSTRACT

Deep learning has recently been successfully applied to lithography hotspot detection. However, automatic correction of the detected hotspots into non-hotspots has not been explored. This problem is challenging because the standard supervised learning requires a training dataset with pairs of hotspots and non-hotspots, which is impractical to collect because lithography hotspots involve diverse and complicated lithographic pattern properties. In this paper, we propose a new framework for lithography hotspot correction with a deep generative network combined with a learning strategy optimized for lithography patterns. Our key idea is to learn to translate hotspots to non-hotspots and vice versa, simultaneously. In this way, the training dataset does not have to be paired, and hotspot patterns in variety of background can be learned. Our method does not require the understanding of the cause of hotspots and can correct hotspots that are difficult to recognize by conventional approaches. For evaluation, we propose to synthesize a training dataset that reflects a variety of real-world lithography patterns. Experimental results show that our framework can correct hotspot images with comparable quality as a conventional complicated process, while significantly reducing the overall processing time.

Keywords: Hotspot correction, deep learning, deep generative model, image-to-image translation

1. INTRODUCTION

Recently, deep learning has been applied to various subjects in computational lithography including optical and resist modeling [1–3] and mask correction, especially in assist feature generation [4–9]. However, manipulation of lithography hotspots, such as simultaneously detecting and correcting hotspots to *coldspots*,[1] with deep learning has been unexplored. Hotspot manipulation is complicated and expensive in the conventional approach, which consists of the series of processes including modeling physical and chemical phenomena, applying the model to mask correction, and verifying the mask using that model. Moreover, for hotspot correction, the whole process should be iterated many times to check the feasibility of modified design. Therefore, replacing such a complicated process with a deep neural network has many advantages, because it can be time-efficient and would not require a complicated sequential processing and understanding of lithographic phenomena.

Hotspot detection with deep convolutional neural network (CNN) has been explored in prior works [10–15]. However, the correction of those hotspots has not been explored yet, because how to correct the hotspots into corresponding coldspots is not apparent in data obtainable from the real world. In particular, conventional supervised learning approaches are not easily applicable to hotspot correction, because of the difficulty in collecting paired hotspot and coldspot data. Furthermore, prior works on detection do not localize hotspots in the given images, which may cause unnecessary change on surrounding non-hotspot region around hotspots, which we refer to as *background*. In summary, we have the following desiderata in hotspot correction: (a) the mapping from hotspots to coldspots should be learned without explicitly pairing hotspot images and coldspot images and (b) background around hotspots should not be modified by the hotspot correction method.

*Equal contributions. Correspondence to Woojoo Sim (space.sim@samsung.com) and Kibok Lee (kibok@umich.edu).

[1]Hotspots are not printed properly on the wafer, resulting in the deformation of the physical patterns that ruins the function of circuits. In this paper, we refer to non-hotspots as coldspots.

To satisfy such desiderata, we propose a new framework where an advanced deep generative network is trained with synthetic data optimized to lithography patterns. Specifically, we build upon the recently proposed cycle-consistent generative adversarial network (CycleGAN) [16]. With this approach, hotspot and coldspot data do not have to be paired for training hotspot correction models, because the model implicitly understands the lithographic pattern properties by learning bidirectional mappings between hotspots and coldspots with cycle consistency. For example, cycle consistency requires that, when a hotspot image is mapped to coldspot and mapped back to hotspot, then the resulting image would be similar to the original image. To achieve this, only the hotspots (not the backgrounds) should be corrected; otherwise, the model would fail to recover the background around the hotspots. Thus, cycle consistency ensures that only the hotspots are corrected while the background remains the same. Moreover, this approach requires neither any prior knowledge on pre-defined lithographic pattern types nor any pre-processing like classification or segmentation. This enables the model to correct any hotspots even when the cause of them is unclear; we provide an example of this with experimental results in Figure 8. Further, we emphasize that our framework only contains a single neural network that does both detection of hotspots and the correction of them in one shot, such that the overall process is not a complicated combination of sub-processes using different neural networks.

To evaluate our model, we propose to synthesize a dataset with various types of hotspots. As we do not have a corresponding coldspot image for each hotspot image in the real world cases, the generated hotspot images are not paired with their desired corrections. Rather, we provide a dataset of unpaired hotspot and coldspot images, such that our model learns to understand how to correct hotspots into coldspots. In addition, only a small subset of possible geometries and locations of hostpots are represented in the training data. Further, multiple hotspot types may exist in the same images with random backgrounds. Combined with all these properties, the synthesized dataset effectively reflects the challenging real-world scenarios.

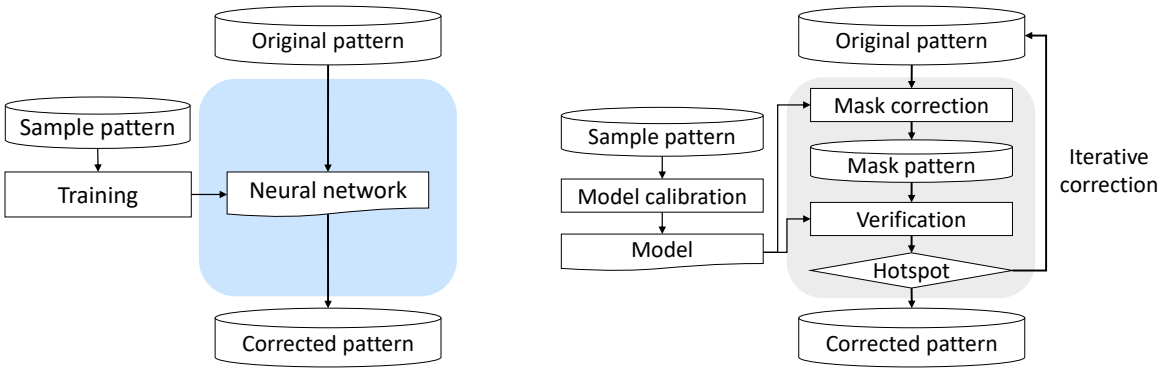
We measure the performance of our model under the pixel-based precision and recall for each corrected image. For both simple and complicated hotspot types, significant portion of the corrected images have high precision and recall values as they can be accepted as proper coldspot images. The experimental results show that the corrected images are comparable with the results of the conventional method with the iterations of lithography modeling, mask correction, and verification process, while ours significantly outperforms them in the whole process time.[2] Moreover, for the cases of complicated hotspot types and unknown hotspot types, our method detects their locations, extracts their properties and corrects them simultaneously, which cannot be automatically done by the conventional approaches.

2. RELATED WORK

2.1 Computational Lithography Using Deep Learning

There have been several studies on the application of deep learning to computational lithography, including lithography modeling, mask correction, and hotspot detection. In case of modeling, the regression error rate was considerably reduced [1] by the application of CNN. In addition, effective generation of modeling data using residual neural networks and transfer learning has been explored [3]. Since mask correction using deep learning was initiated by [4], correction runtime has been reduced by more than half using hierarchical Bayes model combined with model-based correction [5], and polar basis has been used for efficient feature extraction [6]. Moreover, pattern correction considering etching process has been explored using deep learning, where pattern densities and polygon biases are mapped to each other as in mask correction, resulting in the improvements in the biasing accuracy [17]. For the generation of assist features, which is a part of mask correction, when neural network with binary classification (good and bad placement of the assist features) was applied, the generation runtime has been substantially reduced [7], maintaining the assistance performance. Also, assist features were optimized in Fourier space of the mask function using CNN to reduce the degree of freedom in grayscale image data [8,9]. More recently, GAN has been also applied to mask correction [18]. Finally, for detecting lithography hotspots, deep neural networks have been successfully applied to layout data [10–15]. They mainly focus on the

[2]Conventional hotspot correction takes at least a few days even when the lithography model is prepared before the correction. When model is not prepared, it takes more than two or three weeks. In contrast, the correction using our method (i.e., performing both training and testing) can be done in a few hours.



(a) Hotspot correction by a neural network (our method)

(b) Conventional hotspot correction workflow

Figure 1: Comparison of hotspot correction methods: a neural network (left) and conventional correction workflow (right). There are many sub-procedures including the generation of a lithography model, mask correction and verification using the conventional model, and the manual iterations of those procedures are necessary.

detection of hotspots by the binary classification of layout data into hotspot images and coldspot images, where CNN plays key role to effectively capture the hotspot properties. In addition, efficient data augmentation method to balance hotspot and coldspot domain [12], and image tensor extraction using discrete cosine transform have been studied to improve the network performance [13].

However, those methods only focus on the detection of hotspots, not on the correction of the hotspots. Conventionally, the correction of hotspots is much more complicated and time consuming, because mask correction and verification need to be added to the series of sub-processes and moreover the iterations of the whole process are inevitable to improve the correction accuracy. (See Figure 1 for the whole hotspot correction process.) In contrast, our proposed framework performs the correction in a single process, where a target layout including hotspots is directly transformed to the one without hotspots. Moreover, our method does not require separate detection or segmentation of hotspots as pre-processing steps prior to hotspot correction.

2.2 Deep Generative Model

Generative Adversarial Networks (GANs) [19] have achieved remarkable results in many computer vision tasks such as image generation [20–22], image-to-image translation [16, 23], and text-to-image synthesis [24, 25]. The key idea of GAN is *adversarial learning* of two models. A generator learns to generate realistic images by mapping a simple Gaussian distribution to a complex image distribution, while a discriminator learns to distinguish real images and images generated from the generator. In this competitive learning scenario, the generator evolves to fake the discriminator, which turns out to be an effective learning strategy for deep generative models.

CycleGAN [16] is a deep generative model based on GAN for unpaired image-to-image translation. It learns to translate an image in a domain to another domain and vice versa. The key idea is to enforce the *cycle consistency* by training two mappings together with an additional cycle consistency loss to guarantee that two mappings are an inverse mapping of each other. It has been observed that learning GANs with the cycle consistency loss incurs training more stable and avoids the mode collapse issue. An advantage of training with the cycleGAN objective is that training can be done in an unpaired manner: an image in one domain does not have to be matched with a corresponding image in another domain.

Since the correction of lithography hotspots can be considered as a translation from hotspot to coldspot, the idea of cycleGAN is well suited to the problem. Similar to the idea of the original cycleGAN, a deep generative model for correction of lithography hotspots learns to translate hotspots to coldspots and vice versa.

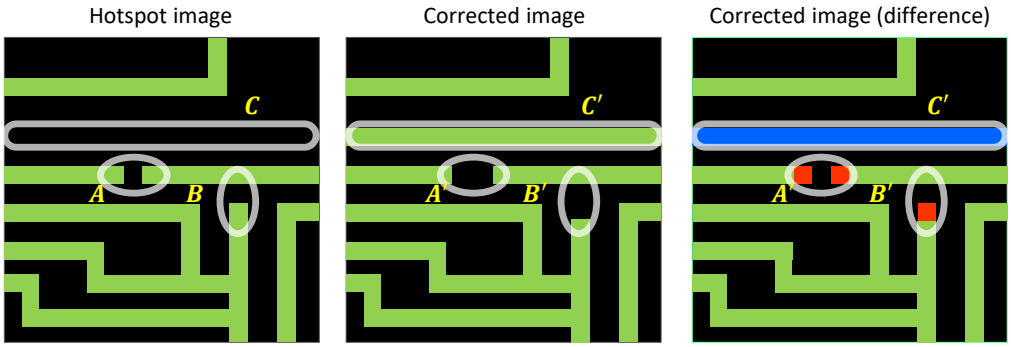


Figure 2: Hotspot correction as an image-to-image translation. Polygons in green represent original patterns, and pixels in red and blue represent deleted and added polygons after correction, respectively.

3. FRAMEWORK

We define our problem in Section 3.1 and introduce the cycleGAN and how to adapt it to hotspot correction in Section 3.2. Finally, we present a method of preparing data applicable to real world scenarios in Section 3.3.

3.1 Overview

Lithography hotspots are identified according to the geometries of the patterns associated with lithographic proximity in some regions of an image. However, except for the region of hotspots, the corresponding coldspot image is mostly the same with the hotspot image. For example, Figure 2 (left) shows hotspots of small tip-to-tip (A) or tip-to-bar (B) spaces. Then, the corresponding coldspot image where those hotspots are corrected is shown in Figure 2 (middle), where the tip-to-tip or tip-to-bar space has been increased; this correction is visualized as color-coded image in Figure 2 (right). Note that some hotspots would be determined by non-local lithographic pattern properties. For example, in Figure 2, the low density of the polygons near the tip-to-tip (C) causes hotspots; note that such hotspots are formed because of non-local interaction. Thus, in the mappings between hotspots and coldspots, the properties extracted are not only the local geometries but also non-local properties like as topologies of patterns and densities.

Mathematically, as an image-to-image translation problem, we denote the domain of hotspot images and coldspot images by X and Y, and their data distributions by $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$, respectively. Then, our goal is to learn a mapping $G : X \to Y$. Since it is difficult to obtain a paired dataset in real world cases, the mapping cannot be learned with a naive autoencoder approach. For example, the following loss function is only applicable when data from X and Y are paired, i.e., $(x, y) \sim p_{data}(x, y)$ such that

$$\mathcal{L}(G, X, Y) = \mathbb{E}_{(x,y)\sim p_{data}(x,y)}[\|y - G(x)\|_1]$$

To overcome this issue, we introduce our formulation based on the cycleGAN framework in Section 3.2.

3.2 Hotspot Correction with Image-to-Image Translation by CycleGAN

When there are two data domains X and Y with data distributions $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$, respectively, the cycleGAN learns bi-directional mappings between them. The networks that transform the data from one domain to the other are denoted as $G : X \to Y$ and $F : Y \to X$. D_X and D_Y are discriminator networks that try to differentiate the images in one domain (real images) from the images translated from the other domain (fake images). Those generators and discriminators are optimized in an adversarial manner with cycle consistency. The loss function of cycleGAN consists of three terms as follows:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F). \quad (1)$$

Here the first and second terms are GAN losses of image-to-image translation from domain X to Y and from Y to X, respectively. The first term is defined as follows (the second term is defined in a similar way):

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x\sim p_{data}(x)}[\log(1 - D_Y(G(x))]. \quad (2)$$

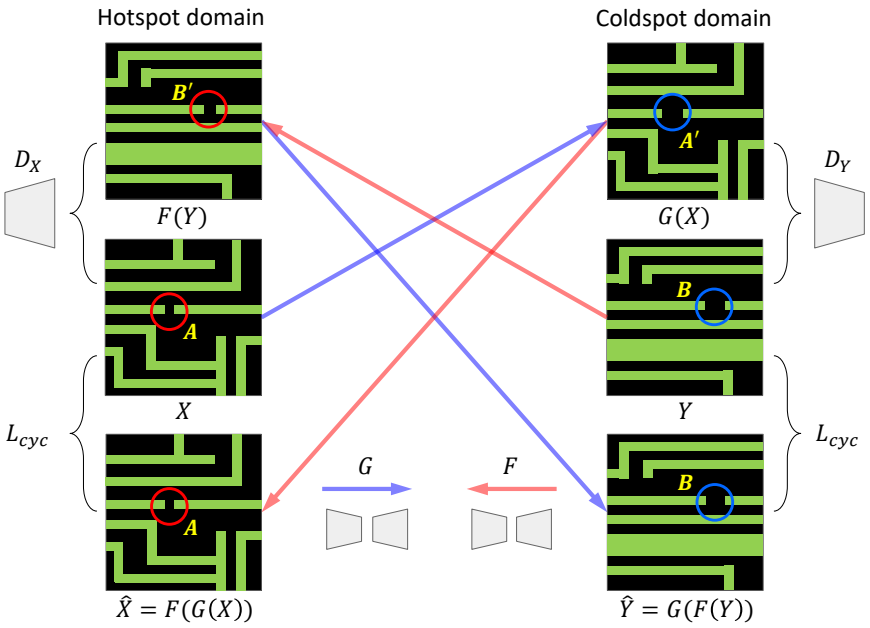


Figure 3: CycleGAN for hotspot correction. Tip-to-tip shapes that are common to both hotspot and coldspot domains (A and B respectively) are extracted and their sizes are corrected when the images of each domain are mapped to the other (A to A' and B to B'), whereas the background of the images are maintained. In this example, X and $\widehat{X}$ (Y and $\widehat{Y}$) should be identical in order to minimize L_{cyc}, and D_X (or D_Y) learns to discriminate X and $F(Y)$ (or Y and $G(X)$) to learn F (or G) in an adversarial manner, respectively.

When mini-max game is applied to this loss, the discriminator D tries to differentiate images in one domain (y) from the translated images from the other domain ($G(x)$) using the generator G, while G tries to make $G(x)$ non-distinguishable from y. Note that in Eq. (1), the two GAN loss terms are symmetrically placed, which induces unpaired image-to-image translations when they are combined with the third term of the equation, which is the cyclic consistency loss:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1]. \tag{3}$$

In the loss, the difference between original images (x) and cyclically generated images ($F(G(x))$) indicates that some features of the original images might be kept under the translation. Now, when all the terms are combined and the mini-max optimization is applied to the total loss:

$$G^*, F^* = \arg\min_{G,F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y), \tag{4}$$

This image-to-image translation formulation encourages the generator G to correct only the regions that are associated with hotspots, while keeping the rest of the background non-hotspot regions unchanged. Specifically, there are properties that hotspots and coldspots have in common, e.g., the overall shapes of patterns, while there are differences in details, e.g., the size or the placement of the shapes. Therefore, when cycleGAN is applied to the hotspot correction, the common properties and differences in details are extracted from both domains, and the domain-specific patterns are transformed from one to the other while the common properties remain the same. In addition, the background is maintained as well in terms of their shapes and geometries. This is the key property of cycleGAN that makes it an effective solution for hotspot correction with unpaired data.

Figure 3 illustrates an example of our framework with the tip-to-tip hotspots and its corresponding coldspots denoted by A and B, respectively. While the geometrical shapes representing tip-to-tip's are common in both hotspot and coldspot domain, the size of the tip-to-tip's differ between the two domains. Then, by the cycleGAN, common tip-to-tip shapes are extracted from both domains and only their sizes are corrected, A to A' and B

to B', with the translation of hotspot images into coldspot images and vice versa. At the same time, the backgrounds as well as the overall shape of the tip-to-tip are maintained during the translations. Note that, in the training of cycleGAN, coldspot images are translated to hotspot images as well by the symmetric nature of the cycleGAN formulation. For example, in Figure 3 coldspot tip-to-tip B is changed into hotspot B'. However, we only utilize the generator G for correction of hotspots into coldspots in practice, e.g., A to A'. This is because the correction of coldspots into hotspots are not only unnecessary but also non-deterministic, as there are many ways of making a hotspot image from a coldspot image.

In summary, there are several aspects that cycleGAN benefits the hotspot correction. First, by the cyclic nature of its loss function, the images from the two domains are not needed to be paired in training. In addition, prior knowledge of the hotspots or coldspots are not required. In other words, preprocessing like classification or segmentation of hotspots is not necessary for our method. In particular, from experiments in Section 4.3 and Section 4.4, we observe that even when there are multiple hotspot types, our proposed method is able to correct them properly. We note that translating multiple types of objects is not discussed in the original cycleGAN paper, but our experimental results show that it is possible.

3.3 Synthetic data generation for training

Although the cycleGAN does not require a paired training dataset, collecting a training dataset for various cases of hotspots is still nontrivial. For such cases, we propose to train our model with a synthesized dataset to deal with various hotspot patterns in the real world. During training, we do not provide information about the types and locations of hotspots. This makes the training more difficult but gives flexibility in the real world application: whenever a new type of hotspot is found, we can update the model with the corresponding hotspot images. To make the synthesized dataset similar to the one extracted from the real circuit layouts, we only take a small subset from a huge variety of hotspot patterns for training (i.e., varying size of polygons that cause hotspots as well as their locations in the image). In addition, we also consider rare but crucial cases when multiple hotspots appear in a small region, which is discussed in Section 4.3.

4. EXPERIMENTAL RESULTS

We experiment and evaluate our proposed model in three different settings together with a baseline described in Section 4.1. In the first two experiments, single and multiple types of hotspots appear in a random background, respectively. In the case of multiple types of hotspots, to make it more realistic, we mix the images including different hotspot types. Moreover, we construct hard-to-detect hotspot types using the difference in the dimensions or densities of neighboring patterns. In the last experiment, more diverse types of hotspots are experimented, but non-hotspot backgrounds in the images are fixed for ease of generation of variety of pattern types.

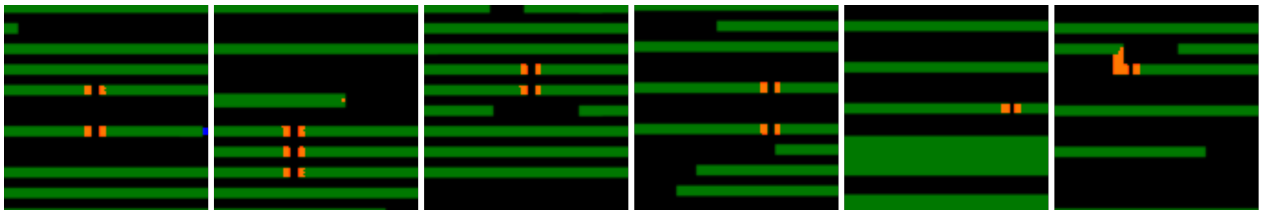
4.1 Experimental Setup

Baseline. As a baseline, we propose a hotspot correction method based on a classification model. Intuitively, given a pre-trained classification model, we can modify the hotspot image with gradient descent in the pixel space such that the prediction of the model becomes coldspot. This idea is relevant to prior work on adversarial perturbations [26] and visualizing neural networks [27, 28].[3] For example, suppose $Z \triangleq X \cup Y$, where X and Y are hotspot and coldspot domain respectively, and $D : Z \to [0, 1]$ is a classification model where $D(z)$ learns to be close to 1 if $z \in X$ and to 0 if $z \in Y$, respectively. Once the classification model is trained by minimizing a classification loss $\mathcal{L}_{cls}(D, Z)$, then one step of the correction can be described as follows:

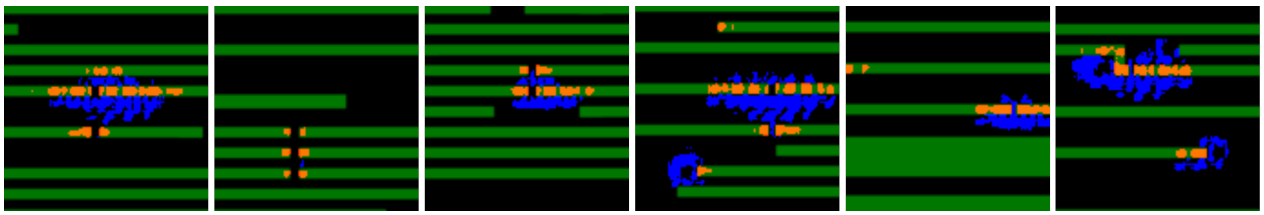
$$x \leftarrow x + \alpha \frac{\partial}{\partial x} \mathcal{L}_{cls}(D, x), \tag{5}$$

where α is a hyperparameter. By applying this update multiple times, the classification loss grows larger, such that the updated image will be eventually classified as a coldspot image.

[3] Specifically, [26] describes a gradient-based method for generating adversarial perturbations. [27] presents a method for visualizing neural network by optimizing a hidden layer activation (or classifier output) via gradient ascent in pixel space. [28] focused on visually localized explanation on which areas of the image contribute to the prediction of the model.

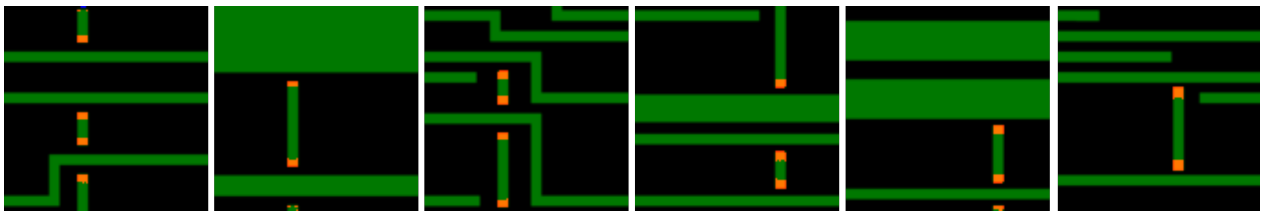


(a) Correction of tip-to-tip types by cycleGAN

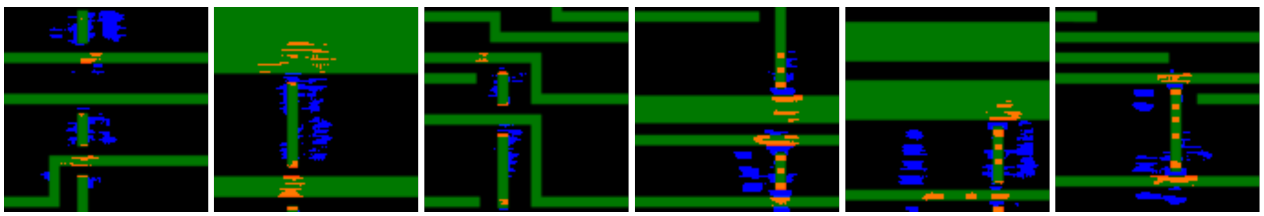


(b) Correction of tip-to-tip types by baseline method

Figure 4: Correction of tip-to-tip hotspot types in random backgrounds. By the cycleGAN, the small tip-to-tips are properly changed to larger ones with the least amount of modifications for correction into coldspots (a). In contrast, the corrections by baseline are excessive and unclear (b).



(a) Correction of tip-to-bar types by cycleGAN



(b) Correction of tip-to-bar types by baseline method

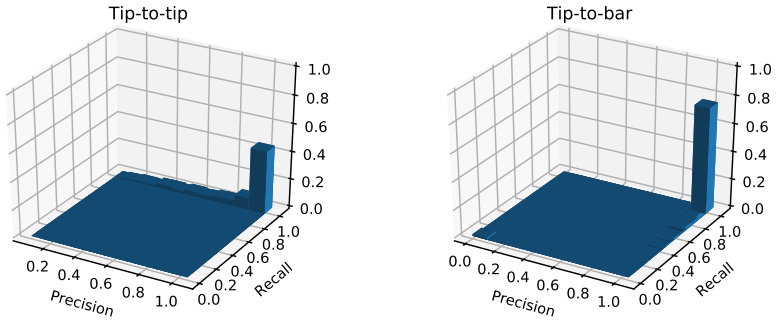
Figure 5: Correction of tip-to-bar hotspot types in random backgrounds. Tip-to-bar spaces are extended properly by cycleGAN (a), whereas the corrected regions by baseline are not clearly focused (b).

Network architecture. Our cycleGAN has the following structure. The generator network consists of encoding and decoding CNNs of three layers with varying kernel size optimized to the geometries of hotspot types, and seven residual layers connecting those two CNNs. The discriminator network is composed of CNNs of four layers with varying kernel size as well as generator CNNs, and with channels whose numbers are optimized to the variety of the patterns in the images including both hotspot regions and backgrounds. For both generator and discriminator, we used Adam optimizer with an initial learning rate 0.001 and 80 iterations.

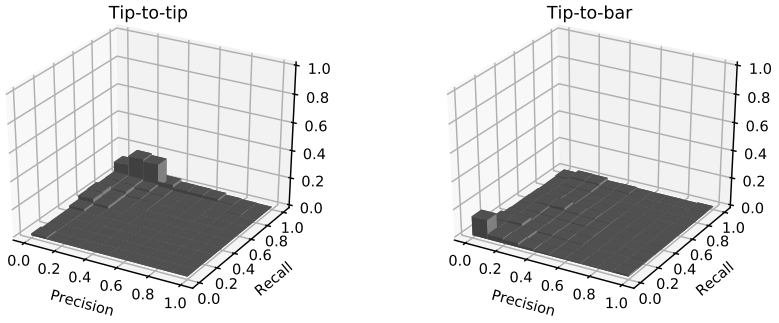
Evaluation metric. For quantitative evaluation, we propose to measure precision and recall for each corrected image. Note that the output of the hotspot correction model is real-valued, thus we map the output image to binary values with a threshold. For comparison with the ground truth, we generate two masks for each hotspot:

Table 1: Test result of representative hotspot types: mean precision and recall

Hotspot type	CycleGAN		Baseline	
	Mean precision	Mean recall	Mean precision	Mean recall
Tip-to-tip	0.810	0.998	0.195	0.850
Tip-to-bar	0.986	0.927	0.250	0.374
Tip-to-tip + Tip-to-bar	0.976	0.956	0.588	0.699
Tip-to-tip + Width of neighboring lines	0.873	0.797	0.295	0.207



(a) Single-type hotspots



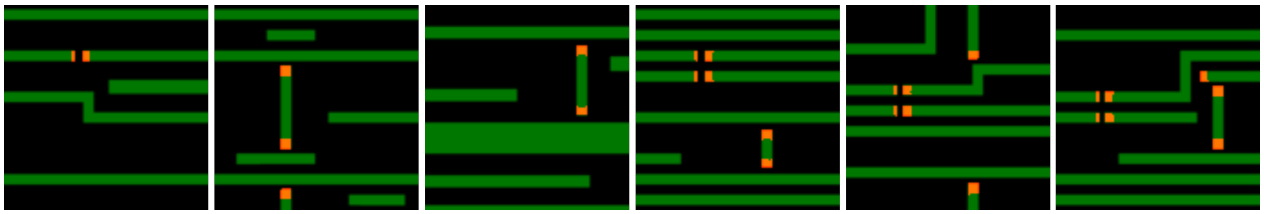
(b) Single-type hotspots: baseline method

Figure 6: Histograms of precisions and recalls for the correction of single-type hotspots. While the cycleGAN results have large number of images with high precisions and recalls, images of low precision and recalls are dominant in the baseline result.

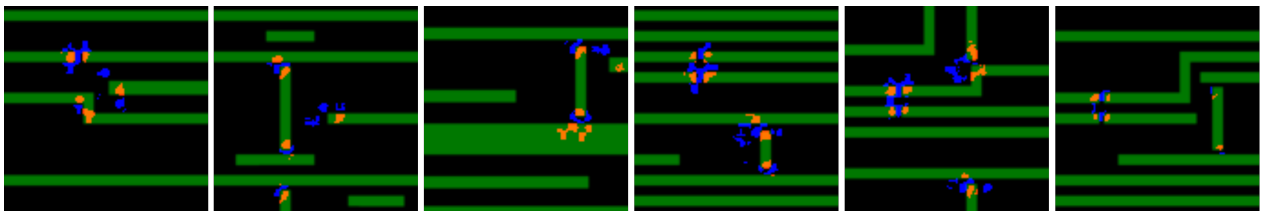
one covers a region that must be corrected, and the other covers a larger region that is a buffer, for which correction is not required but acceptable. With these masks, precision is defined by the ratio of the number of pixels modified (i.e., flipped binary pixel values) in the large mask and the total number of pixels modified. On the other hand, recall is defined by the ratio of the number of pixels modified in the small mask and the number of pixels in the small mask. We compute precision and recall per image and average them over all test images to evaluate the overall performance.

4.2 Single-type Hotspots in Random Background

Figure 4 and 5 show the correction results when tip-to-tip and tip-to-bar hotspot types are given, respectively. As shown in Figure 4a and 5a, small tip-to-tip and tip-to-bar spaces causing hotspots are extended such that they are accepted as coldspots using our proposed cycleGAN model. However, in Figure 4b and 5b, although the baseline detects the locations of hotspots successfully (because the classification model knows which regions

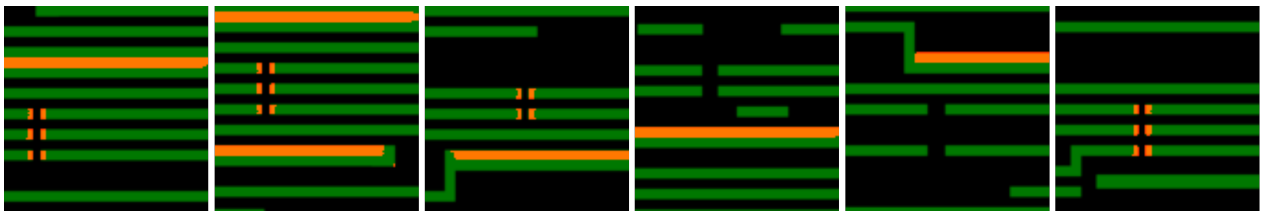


(a) Correction of tip-to-tip + tip-to-bar types by cycleGAN

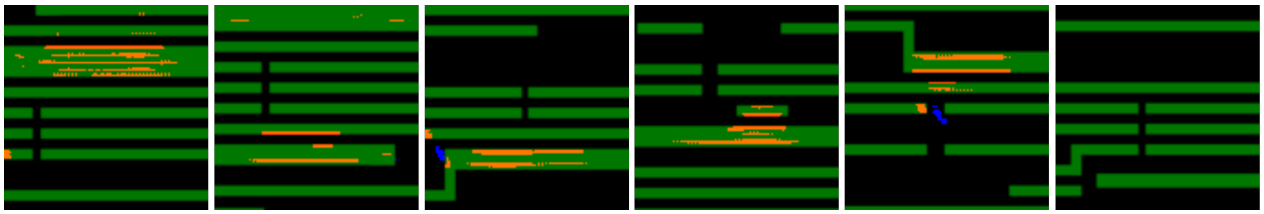


(b) Correction of tip-to-tip + tip-to-bar types by baseline method

Figure 7: Correction of tip-to-tip + tip-to-bar hotspot types in random backgrounds. By cycleGAN (a), both single hotspots (left three images) and multiple hotspots (right three images) are corrected minimizing the change of images. Whereas, by the baseline (b), some hotspot regions are not corrected and inappropriate regions are corrected.



(a) Correction of tip-to-tip with the width of neighboring lines by cycleGAN

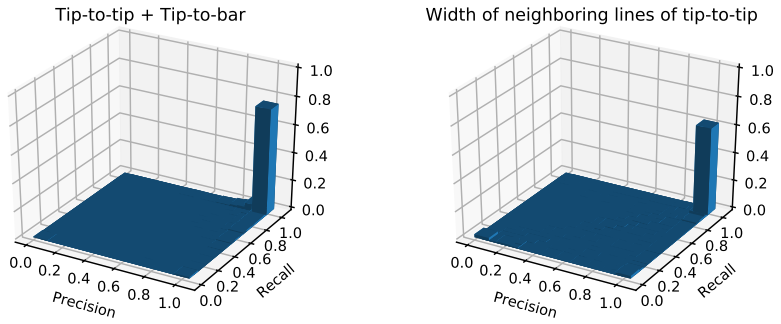


(b) Correction of tip-to-tip with the width of neighboring lines by baseline method

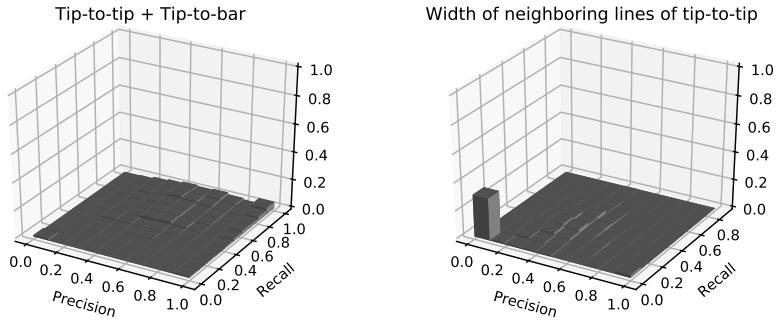
Figure 8: Correction of tip-to-tip + width of neighboring line types in random backgrounds. The disparity of performance between cycleGAN and the baseline method is the most apparent for this case. While the thick lines near tip-to-tip are clearly corrected as well as small tip-to-tips by cycleGAN (a), there are almost no corrections for both tip-to-tips and thick lines in the baseline (b).

contribute to the hotspot properties), it fails to correct them properly. The corrected images by the baseline method have many dots, which is unrealistic in real circuit layouts, and it often adds unnecessary polygons around hotspots. Since the classification model is trained with a discriminative objective, the corrected images would not necessarily reflect the distribution of real circuit layouts. Such issues are addressed by our proposed cycleGAN model because it learns with a generative objective such that the model can generate realistic images.

The first two rows in Table 1 show the quantitative evaluation results in single-type hotspot cases. Our proposed cycleGAN approach outperforms the baseline in all cases with a significant margin. Among the cycleGAN result, the mean precision of tip-to-tip is relatively low. This is because some patterns in vertical directions are



(a) Multi-type and complicated hotspots



(b) Multi-type and complicated hotspots: baseline method

Figure 9: Histograms of precision and recall for the correction of multi-type and complicated hotspots. As in the case of single-type hotspots, majority of the images reside in high precision and recall regions for cycleGAN result (a), whereas there are no such behaviour for the baseline (b).

deleted as shown in the rightmost example in Figure 4a; since vertical patterns are rare in the training dataset, they are not expected to be in any domain. These images are represented by the small bins along the top horizontal axis in the histograms of precision and recall in Figure 6.

4.3 Multi-type and Complicated Hotspots in Random Background

In more realistic cases, there could be multiple hotspot types in a dataset, such that the images including only one of those types and images including more than one type are mixed in the dataset. For example, as depicted in Figure 7, while some images have only one of the tip-to-tip or the tip-to-bar hotspots, others have both types of hotspots. In some cases, it can be difficult to detect the hotspot region or to analyze the cause of hotspot. Figure 8 shows such a case, where what determines whether the tip-to-tip is hotspot or not is the width of neighboring (next-next to the tip-to-tip) lines. That is, width of neighboring line in Figure 8 is a more non-local hotspot pattern, where the tip-to-tip is hotspot only if the width of the line in the two-hop next is thick. We note that this type of hotspots is hard to be captured by the conventional hotspot correction approaches, because the lithography has many unknown factors coming from materials and complicated processes. For both cases, we see that our cycleGAN model performs proper corrections. In Figure 7, both tip-to-tip and tip-to-bar spaces are enlarged, regardless that either tip-to-tip or tip-to-bar exists in a image or both of them exist in a image. For the hard-to-detect cases in Figure 8, all the imaged are corrected to the ones where both small tip-to-tip spaces and thick neighboring lines are suppressed. In contrast, as shown in Figure 7b and 8b, the baseline cannot correct the hotspots properly, and furthermore, it often fails to detect the hotspots.

The last two rows in Table 1 and the histograms in Figure 9 show the quantitative evaluation results in multi-type hotspot cases. Similar to the results of single-type hotspots, our proposed cycleGAN approach outperforms the baseline in all cases with a significant margin. Among the cycleGAN result, the mean precision of the tip-to-tip + width of neighboring lines is relatively low. This is because the correction on width is difficult to evaluate

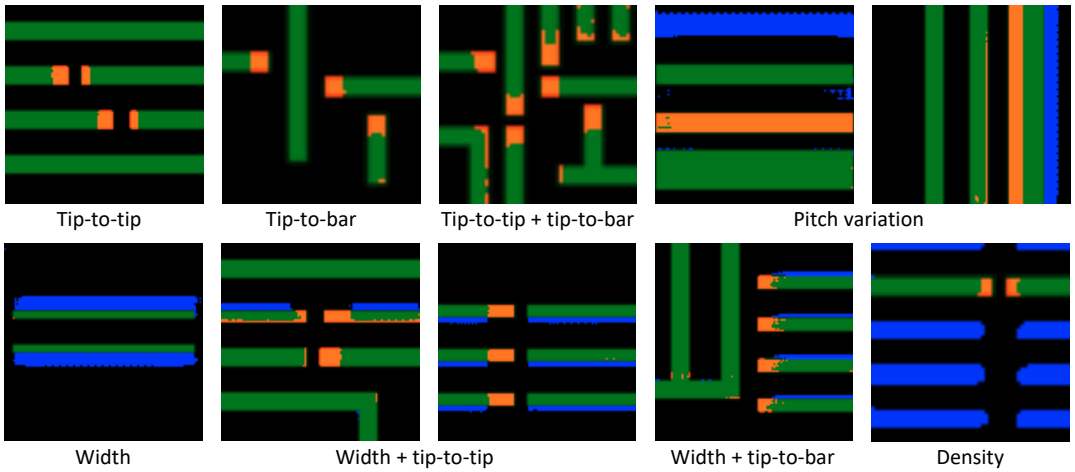


Figure 10: Correction of various hotspot types in fixed background with CycleGAN. Not only the single types but also combined multiple-types are corrected into coldspots by the minimum changes of the corresponding polygons.

quantitatively. In some cases, the model would split the line to two as in the first example in Figure 8a, but it would also shrink the width of the line as in the forth example in Figure 8a.

4.4 Diverse Hotspot Types

Figure 10 shows the correction results when the hotspot types are more diverse: tip-to-tip, tip-to-bar, line or space width, pitch variation, density of neighboring patterns and their combinations. Similar to Section 4.3, multiple types of hotspots are detected and corrected properly by our proposed method, though the resulting images are sometimes noisy because the hotspot patterns are complicated. The overall evaluation results show that hotspots of all types are corrected with high accuracy, which is comparable to the result of conventional correction process which takes much longer processing time as depicted in Fig 1.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel hotspot correction method based on a deep generative model. Our method is built on the cycle-consistent generative adversarial network, which allows for effective training without paired data or separate preprocessing steps. Our method learns to detect and correct various types of hotspots, achieving comparable correction performance as conventional iterative processes with significantly reduced processing time.

As future work, for the large-scale circuit layouts in real world cases, our framework can be applied multiple times in different resolutions in a hierarchical manner: if we have large-scale images with hotspots but do not know their location, a model can be first trained with the large-scale images to detect the region of hotspots. After then, we zoom in the detected region of the hotspots and train another model with the zoomed images, such that the model can correct the hotspots with high precision. Thus, our work provides the base of a systematic learning strategy that can be applied to circuit layouts in the real world.

REFERENCES

[1] Watanabe, Y., Kimura, T., Matsunawa, T., and Nojima, S., "Accurate lithography simulation model based on convolutional neural networks," in [*SPIE Advanced Lithography*], *Proc. SPIE* (2017).

[2] Shim, S., Choi, S., and Shin, Y., "Machine learning-based 3d resist model," in [*SPIE Advanced Lithography*], *Proc. SPIE* **10147** (2017).

[3] Lin, Y., Watanabe, Y., Kimura, T., Matsunawa, T., Nojima, S., Li, M., and Pan, D. Z., "Data efficient lithography modeling with residual neural networks and transfer learning," *Proc. ISPD*, 82–89 (2018).

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

[5] Matsunawa, T., Yu, B., and Pan, D. Z., "Optical proximity correction with hierarchical bayes model," *Journal of Micro/Nanolithography, MEMS, and MOEMS* **15**, 021009 (2016).

[6] Choi, S., Shim, S., and Shin, Y., "Machine learning (ML)-guided opc using basis functions of polar fourier transform," in [*Optical Microlithography XXIX*], *Proc. SPIE* **9780** (2016).

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

[8] Wang, S., Su, J., Zhang, Q., Fong, W., Sun, D., Baron, S., Zhang, C., Lin, C., Chen, B., Howell, R., Hsu, S., Luo, L., Zou, Y., Lu, Y., and Cao, Y., "Machine learning assisted sraf placement for full chip," in [*SPIE Photomask Technology and EUV Lithography*], *Proc. SPIE* **10451** (2017).

[9] Wang, S., Baron, S., Kachwala, N., Kallingal, C., Sun, D., Shu, V., Fong, W., Li, Z., Elsaid, A., Gao, J., Su, J., Ser, J., Zhang, Q., Chen, B., Howell, R., Hsu, S., Luo, L., Zou, Y., Zhang, G., Lu, Y., and Cao, Y., "Efficient full-chip sraf placement using machine learning for best accuracy and improved consistency," in [*SPIE Advanced Lithography*], *Proc. SPIE* **10587** (2018).

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

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

[12] Yang, H., Luo, L., Su, J., Lin, C., and Yu, B., "Imbalance aware lithography hotspot detection: a deep learning approach," *JM3* **16**, 033504 (2017).

[13] Yang, H., Su, J., Zou, Y., Yu, B., and Young, E. F. Y., "Layout hotspot detection with feature tensor generation and deep biased learning," *Proc. DAC* **62**, 1–6 (2017).

[14] Shin, M. and Lee, J., "Cnn based lithography hotspot detection," *International Journal of Fuzzy Logic and Intelligent Systems* **16**, 208–215 (2016).

[15] V. Borisov, J. S., "Research on data augmentation for lithography hotspot detection using deep learning," in [*34th European Mask and Lithography Conference*], Behringer, U. and Finders, J., eds., *Proc. SPIE* **10775** (2018).

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

[17] Shim, S. and Shin, Y., "Etch proximity correction through machine-learning-driven etch bias model," in [*SPIE Advanced Lithography*], *Proc. SPIE* **9782** (2016).

[18] Yang, H., Li, S., Ma, Y., Yu, B., and Young, E. F. Y., "Gan-opc: Mask optimization with lithography-guided generative adversarial nets," in [*55th ACM/ESDA/IEEE Design Automation Conference (DAC)*], *ACM*, 131 (2018).

[19] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," in [*NIPS*], (2014).

[20] Radford, A., Metz, L., and Chintala, S., "Unsupervised representation learning with deep convolutional generative adversarial networks," in [*ICLR*], (2016).

[21] Arjovsky, M., Chintala, S., and Bottou, L., "Wasserstein generative adversarial networks," in [*ICML*], (2017).

[22] Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C., "Veegan: Reducing mode collapse in gans using implicit variational learning," in [*NIPS*], (2017).

[23] Yi, Z., Zhang, H. R., Tan, P., and Gong, M., "Dualgan: Unsupervised dual learning for image-to-image translation.," in [*ICCV*], (2017).

[24] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H., "Generative adversarial text to image synthesis," in [*ICML*], (2016).

[25] Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., and Metaxas, D., "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in [*ICCV*], (2017).

[26] Goodfellow, I. J., Shlens, J., and Szegedy, C., "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572* (2014).

[27] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H., "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579* (2015).

[28] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D., "Grad-cam: Visual explanations from deep networks via gradient-based localization," in [*ICCV*], (2017).