Litho-GPA: Gaussian Process Assurance for Lithography Hotspot Detection

Wei Ye, Mohamed Baker Alawieh, Meng Li, Yibo Lin, and David Z. Pan

ECE Department, University of Texas at Austin, Austin, TX, USA

***Abstract*—Lithography hotspot detection is one of the fundamental steps in physical verification. Due to the increasingly complicated design patterns, early and quick feedback for lithography hotspots is desired to guide design closure in early stages. Machine learning approaches have been successfully applied to hotspot detection while demonstrating a remarkable capability of generalization to unseen hotspot patterns. However, most of the proposed machine learning approaches are not yet able to answer one critical question: how much a hotspot predicted from a trained model can be trusted? In this work, we present Litho-GPA, a lithography hotspot detection framework, with Gaussian Process assurance to provide confidence in each prediction. The framework also incorporates a data selection scheme with a sequence of weak classifiers to sample representative data and eventually reduce the amount of training data and lithography simulations needed. Experimental results demonstrate that our Litho-GPA is able to achieve the state-of-the-art accuracy while obtaining on average 28% reduction in false alarms.**

I. Introduction

Lithography hotspot detection plays a pivotal role in physical verification. It detects potential manufacturing hotspots in layout patterns such that early fixing is possible before shipping the designs. Efficient and accurate lithography hotspot detection is critical for layout finishing and design closure towards yield improvement in the physical verification stage. Lithography hotspots can be accurately detected through full-chip lithography simulations which compute the aerial images and contours of printed patterns [1], [2]; though, at a tremendous computational cost [3].

Pattern matching and machine learning based techniques have been proposed for early and quick detection of lithography hotspots during physical verification [4]. Pattern matching is a direct and fast method for hotspot detection [5], [6]. However, pattern matching, including fuzzy pattern matching [7], [8], is still insufficient to handle never-before-seen hotspot patterns. On the other hand, machine learning approaches have demonstrated good generalization capability to recognize unseen hotspot patterns [9]–[17]. In these approaches, a labeled dataset is used to train a machine learning model capable of detecting hotspots in new layout patterns with high accuracy. The primary objective is to achieve high accuracy while minimizing false alarms. Practically, accuracy is given the highest priority; hence, a moderate number of false alarms is typically tolerated for the sake of achieving better accuracy. This is due to the fact that missing any hotspot may result in significant yield degradation. Recently, deep learning techniques have been actively explored to improve the accuracy of hotspot detection [18]–[23].

Nonetheless, most of the proposed machine learning approaches are not yet able to answer one critical question: how much a hotspot predicted from a machine learning model can be trusted? With efforts mainly tailored towards achieving better accuracy, little attention has been given to this confidence issue. In practice, addressing this concern requires machine learning models to provide confidence guarantees alongside the label predictions. For example, in a deep learning model, the results of the softmax are usually interpreted as probability estimates. However, it has been shown that these probability estimates do not match the correct likelihood [24]; in fact, networks are often too confident about their predictions. In other words, in a classification problem, the output of the softmax can lead to correct labeling of samples; however, the values of the softmax is not a good uncertainty measure.

Bayesian-based methods are the typical option when confidence estimation is needed. In this work, we adopt a Gaussian Process (GP) based classification that can provide a confidence metric for each predicted instance. In practice, a GP prediction is used as a final label only when the confidence level matches a user-defined metric, otherwise, the prediction is marked as untrusted and lithography simulation can be used to further verify the results.

On the other hand, learning based approaches usually require a large amount of training data to obtain models with good generalization, especially for imbalanced datasets, as in the case of the hotspot detection task. This imbalance increases the cost of data preparation and slows down the design closure. This is mainly because each training data sample requires lithography simulation to obtain its label and hotspot samples appear much less often than non-hotspot ones. Therefore, we also propose an active learning scheme with a sequence of weak classifiers to reduce the turnaround time and the cost of data preparation. The combination of GP and active learning scheme is not only able to achieve high accuracy, but also provides a confidence estimation for predictions, with a small amount of training data.

Our main contributions in the proposed Litho-GPA framework can be summarized as follows:

- A Gaussian Process based hotspot detection technique is proposed with the capability of providing the confidence level associated with the predicted label.
- A novel hotspot detection flow is proposed in which the accuracy of prediction can be improved by examining the confidence of prediction.
- An active learning selection scheme based on weak classifiers is developed to reduce the cost of data preparation.
- Experimental results demonstrate Litho-GPA is able to

978-3-9819263-2-3/DATE19/©2019 EDAA

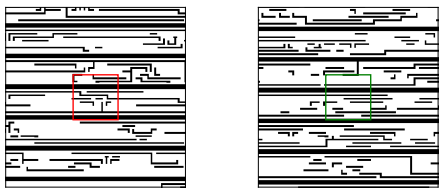
achieve comparable accuracy to the state-of-the-art deep learning approaches while obtaining on average 28% reduction in false alarms.

The rest of this paper is organized as follows. Section II reviews the challenges in hotspot detection and gives the problem formulation. Section III provides a detailed explanation of the proposed approach. Section IV demonstrates the effectiveness of our approaches with comprehensive results, followed by the conclusion in Section V.

II. Preliminaries

Figure 1 shows an example of two clips where one encompasses lithography hotspot marked by the red rectangle region (a) while the other does not (b). The hotspot detection task to be solved by machine learning techniques can be formulated as a two-class image classification problem; a classical problem which has been studied extensively in literature. However, the problem at hand has its unique characteristics that should be taken into account. First, despite the fact that the lithography defects are critical, their relative number is significantly small across the whole chip. This poses a major challenge when formulating the task as a learning problem because the two classes are highly imbalanced which necessitates a proper handling to remove the inherent bias in the data.

Second, with such imbalanced data, the number of false alarms is usually comparable to, or even higher than, the number of true hotspots. In practice, the number of false alarms is among the most important metrics to evaluate hotspot detection methods [20]. **Accuracy** (i.e., true positive rate [25]) and the number of **false alarms** (i.e., false positives) are the two prevailing metrics used for detection evaluation. However, to make use of these models, a new criterion should be considered which is *trust*. Among the questions we address in this work is *should we trust all predictions from a highly accurate model?*



(a) Hotspot (b) Non-hotspot

Fig. 1: An example of lithography hotspot clip (a) and non-hotspot clip (b).

III. Lithgraphy Hotspot Detection

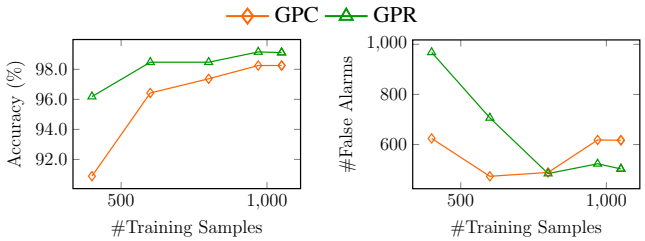
In this section, we explain the details of the proposed Litho-GPA framework for lithography hotspot detection. It consists of two key components: Gaussian Process for hotspot detection and active learning for data preparation.

A. Hotspot Detection using Gaussian Process

Gaussian Process (GP) classification falls under the category of probabilistic classification where test predictions take the form of class probabilities; this contrasts with methods which provide a class label only [26]. Since generalization to test cases inherently involves some level of uncertainty, it is natural to attempt to make predictions in a way that reflects these uncertainties. For hotspot detection, GP can provide, alongside the label, a confidence measure about the label which can help judge the trustworthiness of the obtained classification decision.

In literature, different schemes have been proposed for binary GP classification. Among the most commonly used are those based on logistic or probit mapping where Laplace Approximation is used to estimate the posterior distribution [27]. In other approaches, the binary classification is cast as a regression problem where the objective is to predict a continuous label that can be mapped through thresholding to binary labels. In theory, GP classification with Laplace Approximation (GPC) uses a Bernoulli likelihood in the Bayesian inference, thus incorporating the binary labels into the inference. While such likelihood is an accurate representation of the binary data, it is not conjugate with GP prior; hence it makes the inference intractable and requires approximating the posterior distribution. On the other hand, using a regression based GP for classification (GPR) moves the binary mapping outside the inference; hence, preserving the conjugacy that results in a closed form posterior distribution. In the hotspot detection task, hotspots are assigned to +1 and non-hotspots are assigned to -1; 0 can be a decision boundary which maps the continuous quantity output of GPR to the two discrete classes.

The comparison of GPR and GPC is shown in Figure 2. Examining the figure, one can notice that, with the same number of samples, GPR is always achieving higher prediction accuracy and the number of false alarms resulting from GPR is lower than that from GPC when it converges. Moreover, the fact that the posterior distribution can be obtained with no approximation in GPR is reflected in the computational cost where GPC is significantly more expensive computationally than GPR. Based on this comparison, GPR was adopted in this work.



(a) Accuracy (b) False alarm

Fig. 2: Comparison between GPC and GPR on `Layout2` where accuracy (a) and the number of false alarms (b) are shown.

For the hotspot detection task, the class label y is assumed to be a continuous noisy version of an underlying GP $f(\mathbf{x})$ with a Radial Basis Function (RBF) kernel. Based on the prior distribution of $f(\mathbf{x})$, the joint distribution of the observed outputs $\mathbf{y}$, and the GP function values for the test outputs, the predictive posterior for the images in the test data $p(\mathbf{f}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*)$ is given by [27]:

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*) &\sim \mathrm{N}(\mu, \Sigma), \\ \mu &= K(\mathbf{X}_*, \mathbf{X})\big[K(\mathbf{X}, \mathbf{X}) + \sigma^2 I\big]^{-1}\mathbf{y}, \\ \Sigma &= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})\big[K(\mathbf{X}, \mathbf{X}) + \sigma^2 I\big]^{-1}K(\mathbf{X}, \mathbf{X}_*), \end{aligned} \tag{1}$$

where $\mathbf{X}$ and $\mathbf{X}_*$ are matrices containing the training and testing clip data respectively. σ^2 represents the level of noise in the

data, $\mathbf{y}$ is a vector representing the labels for training data, $\mathbf{f}_*$ is a vector representing the function prediction for the testing clip data, and the matrices $K(\cdot,\cdot)$ represent the covariance matrices obtained by evaluating the RBF kernel.

In the GPR approach, the posterior distribution in Equation (1) represents the distribution of the class label in the continuous domain. To get a point estimate of the class label, proper thresholding scheme (usually at 0) is used for the mean of the posterior distribution. However, we are interested in a predictive distribution that can provide a confidence to judge upon prediction. This can be achieved by leveraging all information provided by the posterior distribution; i.e., the distribution of the continuous class label. To elaborate on this, we consider the example shown in Figure 3 where the posterior distributions for two samples $\mathbf{x}_1$ and $\mathbf{x}_2$ are shown. By looking at the point estimate, both samples have a mean value greater than 0; hence, they will be mapped to label 1. However, it is clear that the uncertainty associated with $\mathbf{x}_2$ is much higher compared to that associated with $\mathbf{x}_1$. In other words, there is a higher probability for the label of $\mathbf{x}_2$ to be less than 0.

Therefore, for a sample with mean greater than 0, a confidence metric can be defined based on the probability that the predicted label is higher than 0. In such case, sample $\mathbf{x}_1$ has a probability of 98% compared to 65% for $\mathbf{x}_2$, which implied higher confidence around the prediction of $\mathbf{x}_1$. To utilize this information, a confidence metric α can be defined to judge upon the validity of the predictions obtained from GPR.

However, while 0 is the intuitive choice for a boundary between the two labels $\{-1,+1\}$ in a classification task, the value of the threshold boundary can be tuned for problems with special characteristics such as class imbalance in the hotspot detection task. Hence, the compromise between accuracy and false alarms can be controlled using a threshold different from zero. In other words, such thresholding scheme can provide control over how *conservative* the model is.

In this hotspot detection task, missing a hotspot can have much more significant consequences when compared to having additional false alarms. With this risk assessment in mind, the threshold can be set to a value κ, where $\kappa < 0$, to bias the prediction towards the hotspot class. Therefore, the labeling processes can be performed according to the following scheme:

$$\hat{y}_i = \begin{cases} +1, & \text{if } p(f_i > \kappa) > \alpha \\ -1, & \text{if } p(f_i < \kappa) > \alpha \\ \text{untrusted}, & \text{otherwise.} \end{cases} \quad (2)$$

In Equation (2), a class label is given to a particular sample if it meets the user-defined confidence metric α. Otherwise, the prediction for the particular sample is set to untrusted reflecting the low confidence in the model prediction and requiring an actual simulation run to validate this sample. For example, considering the two samples in Figure 3 with $\alpha = 0.7$ and $\kappa = 0$, sample $\mathbf{x}_1$ will get a label of $+1$ while $\mathbf{x}_2$ will not be assigned a label, and a lithography simulation is needed to get the right label.

B. Active Learning for Data Selection

When formulating the lithography hotspot detection problem as a learning based classification problem, class imbalance comes forth among the major challenges characterizing the learning

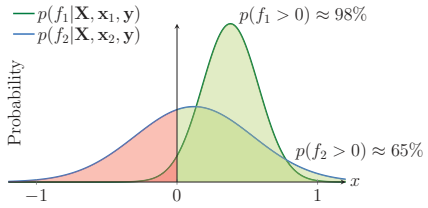


Fig. 3: The posterior distributions obtained through GPR for two samples are shown. The distributions show higher confidence in the prediction of f_1 compared to that of f_2.

task. In practice, there is an abundance in the non-hotspot data on one end and scarceness in the hotspot data on the other. With such setup, a large number of samples is needed to guarantee enough hotspot samples for building accurate classification models. This translates to an enormous computational cost associated with running a large number of lithography simulations. The main reason to endure this cost is based on the fact that given a set of un-simulated data samples, one cannot tell beforehand which ones are hotspots. Hence, the trivial way of collecting data is to randomly select samples for simulation until enough hotspot samples are available.

To address this issue, we propose an active learning framework with the objective of selecting samples that are *likely* to be hotspots and simulating them to get the actual labels. This way, a balanced training dataset, adequate for model training, can be constructed with minimal simulation cost. The main idea is to iteratively select hotspot candidates for simulation based on labels obtained using trained weak classifiers. As a first step, a relatively small set of randomly selected samples, for which simulations are performed and labels are available, is used to build a weak classifier that can point out tentative hotspot samples among the un-simulated ones. These selected samples are then simulated and added to the available training dataset to help improve the performance of the classifier in the next iteration. A weak classifier is adequate here because its training cost is cheaper and the accuracy requirement at this stage is not high.

Here, a weak classifier is one that relies on a simple model; hence, it does not require a large number of samples to train. Although such a classifier may not have a high true positive rate, it can help guide the sampling scheme, especially that the nature of the data will result in a relatively high precision value even with a low true positive rate. Among the possible options, Support Vector Machine (SVM) is used as the weak classifier in this active sampling scheme mainly because of its relatively superior performance and fast training [25]. The details of the active learning method are summarized in Algorithm 1.

Algorithm 1 takes a pool of unlabeled data samples $\mathcal{P}$ and the maximum allowable size n of final dataset $\mathcal{S}$ as input. An initial training set $\mathcal{S}$ is generated by randomly sampling from the pool, followed by label queries through lithography simulations (line 1). Next, the algorithm builds a sequence of weak classifiers to seek more hotspots with the knowledge of previous sampled and simulated hotspots (lines 3 – 8). Each weak classifier is trained with the obtained labeled dataset $\mathcal{S}$ so far (line 5) and is applied

Algorithm 1 Active Learning for Data Selection

Require: A pool of unlabeled data samples $\mathcal{P}$, n
Ensure: Labeled training dataset $\mathcal{S}$
1: $\mathcal{S} \leftarrow$ Select m_0 samples randomly from $\mathcal{P}$ and obtain their labels through simulation;
2: $k \leftarrow 0$;
3: **repeat**
4: $k \leftarrow k + 1$;
5: Train the SVM model with the training dataset $\mathcal{S}$;
6: $S_k \leftarrow$ Select m_k samples from $\mathcal{P} \backslash \mathcal{S}$ with highest hotspot probability by SVM and simulate the labels;
7: $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_k$;
8: **until** No hotspot in $\mathcal{S}_k$ or $|\mathcal{S}| \geq n$
9: **return** $\mathcal{S}$.

to the remaining unlabeled samples in the pool $\mathcal{P}$, i.e., $\mathcal{P} \backslash \mathcal{S}$. The top m_k samples with the high probability of being hotspot are chosen from $\mathcal{P} \setminus \mathcal{S}$, and their labels are obtained by lithography simulations (line 6). Then, the selected set $\mathcal{S}_k$ is added to the labeled dataset $\mathcal{S}$. Note that the algorithm will return early if no actual hotspot is detected among the m_k samples (line 8). That is because, after several iterations, the trained SVM model is expected to have good accuracy, hence, if no hotspot can be detected by the latest classifier, it is likely that none are still present in the pool. Besides, the algorithm can exit from the iterations of weak classifier building if there are enough samples for the Gaussian Process in Section III-A (line 8).

C. Overall Flow

The proposed Litho-GPA framework is illustrated in Figure 4. We first leverage the iterative weak classifier-based sampling scheme to prepare a training set containing enough hotspots (Section III-B). A GPR model is trained with the selected data samples. We then apply the GPR model to make predictions with confidence estimation on the testing set (Section III-A). If GPR gives the predicted label with high confidence, the result is trusted; otherwise, the unsure testing samples will be verified with lithography simulations.

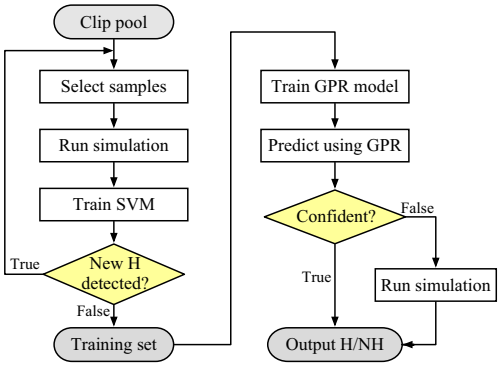


Fig. 4: Overall flow including data preparation with active sampling and hotspot detection with Gaussian process.

IV. Experimental Results

Our Litho-GPA framework is implemented in Python with the scikit-learn library [28] and validated on the ICCAD 2012 CAD contest benchmark set [29]. `Layout1` is not used because it contains only a few clips and has a different technology node from the rest of the four benchmarks. `Layout5` has a small number of hotspots, and hence we merge it with `Layout4`. Table I summarizes the benchmark information, the number of all the clips (#All) and the number of hotspot clips (#H) in the training set (Train) and testing set (Test). The input image is downsized to 128×128 by a nearest-neighbor reduction to improve SVM and GPR training time. We run ten experimental trials for each evaluation, each with a different random seed, and report the average results. It is important to note that, although all samples in the training sets are already labeled in these benchmarks, to validate our framework we assume that they are not labeled at the beginning and obtain the labels through simulations in the framework.

TABLE I: ICCAD 2012 contest benchmark statistics [21].

Design	Train		Test	
	#All	#H	#All	#H
`Layout2`	5,459	174	41,796	498
`Layout3`	5,552	909	48,141	1,808
`Layout4&5`	7,289	121	51,435	218

A. Active Learning for Data Selection

The purpose of the proposed active sampling approach in Section III-B is to balance the dataset by selectively choosing tentative hotspots to be included in the training set. Here, we compare random data selection and the proposed data selection scheme. In the experiments, we set m_0 to 300 and m_k to 100 in Algorithm 1. SVM takes 22.3s at each iteration on average. Table II displays the number of total sampled data (columns "#All") and the number of hotspots (columns "#H") selected by the two schemes when setting 1400 as the maximum allowable size of training samples for both schemes. It is observed that the active learning scheme converges before reaching the size limit for `Layout2` and `Layout4&5`.

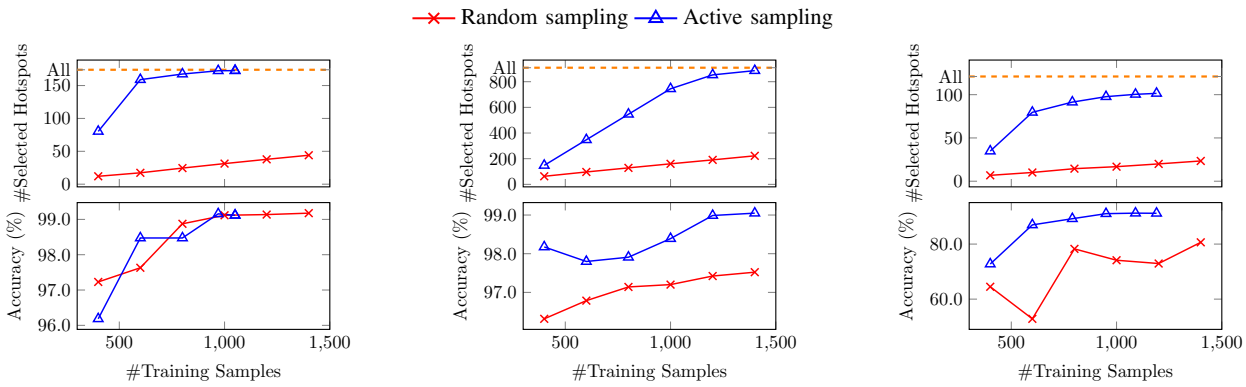
TABLE II: Comparison of different sampling strategies.

Design	Random				Active			
	#All	(%)	#H	(%)	#All	(%)	#H	(%)
`Layout2`	1,400.0	25.6	44.0	25.3	1,050.0	19.2	172.7	99.3
`Layout3`	1,400.0	25.2	222.3	24.5	1,400.0	25.2	886.3	97.5
`Layout4&5`	1,400.0	19.2	23.4	19.3	1,190.0	16.3	101.5	83.9

Varying the maximum training set size n in Algorithm 1, the comparison of the two sampling schemes is shown in Figure 5. The figure shows that, with the same number of training samples, the proposed approach can achieve higher accuracy compared to the random sampling. Note that the accuracy is based on the GPR direct prediction results without lithography simulations. This is in fact due to the higher number of hotspots available in the training data when using the active sampling scheme compared to the random sampling strategy as demonstrated also in Figure 5. Moreover, one can easily notice that the iterative SVM evaluations are capable of detecting most of the hotspots in the dataset within a few iterations.

B. Validation of Gaussian Process

We demonstrate the effectiveness of the proposed GPR with validation simulations for hotspot detection. Table III shows the comparison between the state-of-the-art method [21] and our method, in terms of accuracy (ACC) and the number of false alarms (#FA). In this table, "All" denotes model training uses



(a) Layout 2 (b) Layout 3 (c) Layout 4&5

Fig. 5: The number of selected hotspots and testing accuracy (without validation simulations) for different sampling techniques are shown. "All" represents the total number of hotspots in the entire training set.

TABLE III: Comparison of different flows in terms of accuracy and false alarms. The results are averaged over ten runs.

Design	All + [21]		Random + [21]		Active + [21]		Active + [21] + VS		Active + GPR		Active + GPR + VS		#Sim
	#FA	ACC (%)	#FA	ACC (%)	#FA	ACC (%)	#FA	ACC (%)	#FA	ACC (%)	#FA	ACC (%)	(%)
Layout2	234.1	97.4	370.9	91.3	1,030.7	99.4	733.3	99.6	502.8	99.1	71.4	99.4	16.4
Layout3	3,064.1	98.3	3,333.4	97.7	6,716.3	99.1	5,189.7	99.5	4,443.2	98.3	2,463.4	99.0	17.2
Layout4&5	443.4	91.7	512.5	64.2	1,598.4	96.3	1,162.3	98.9	1,130.2	91.2	177.5	99.1	26.8
Average	1,247.2	95.8	1,405.6	84.4	3,115.1	98.2	2,361.8	99.3	2,025.4	96.2	904.1	99.2	20.1
Ratio	1.0	1.0	—	—	—	—	1.89	1.04	—	—	0.72	1.04	—

all the training samples in the benchmark, while "Random" and "Active" denote the training data obtained from random sampling and the proposed active sampling scheme in Table II. For the method [21], we strictly use its DCT representation and CNN structure for the comparison. To further demonstrate that softmax output of CNN is not a good uncertainty measure, we compare the performance of CNN and GPR after performing the same number of validation simulations (VS). For GPR, threshold κ in Equation (2) is set to -0.2; the confidence metric α is set to 0.682, which is equivalent to one standard deviation confidence interval for a Gaussian distribution. According to this criterion, any untrusted sample needs to be further verified through lithography simulation. Since there is no well-defined metric to quantify confidence interval for CNN, to ensure fairness, we perform the same number of validation simulations to the test samples which has nearly the same softmax probability of being hotspot/non-hotspot and then compare the accuracy and the number of false alarms; that is, we choose the samples which minimize $|\text{softmax(NH)} - \text{softmax(H)}|$. Column "#Sim" gives the ratio of the number of validation simulations to the testing data size.

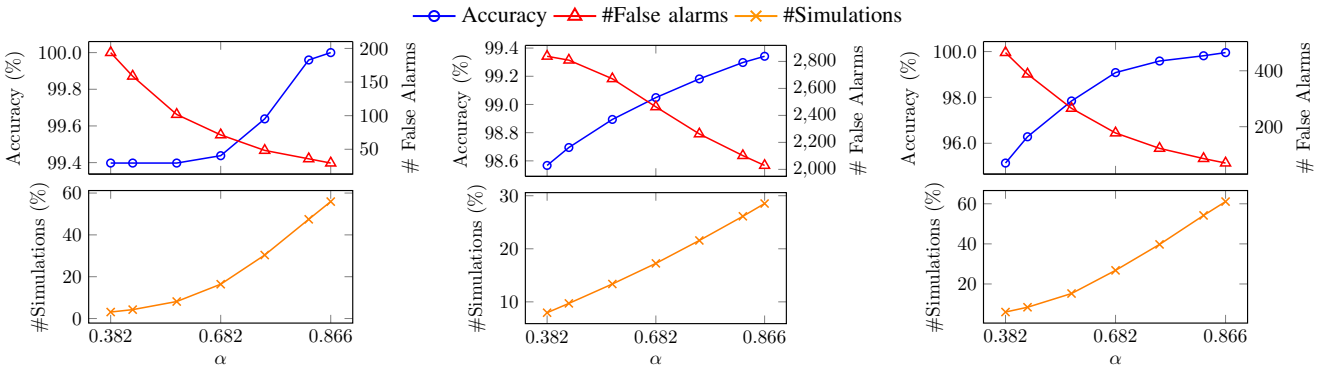
Table III shows that the state-of-the-art work [21] using all the training dataset (All + [21]) achieves 95.8% accuracy on average. Our proposed active learning data selection further improves the accuracy of its model to 98.2% (Active + [21]). However, the average number of false alarms of this flow increases from 1247.2 to 3115.1. Active data selection together with our GPR method (Active + GPR) gives a similar accuracy (96.2%) as the state-of-the-art result. Moreover, given the strength of providing confidence of GPR, the accuracy (Active + GPR + VS) is improved to 99.2% after performing validation simulations, and meanwhile, it reduces the number of false alarms by 28% compared with the All + [21] flow. Compared with the Active + [21] + VS flow, the Active + GPR + VS flow obtains comparable accuracy and 2.6× false alarm reduction, which demonstrates the effectiveness of employing confidence measure provided by GPR. In the experiments, GPR training takes 296.6s, 1490.5s and 235.4s on average for the three benchmarks while testing takes 579.4s, 1342.2s and 586.7s.

C. Control of Prediction Confidence

Lastly, we explore the effect of α to control the desired prediction confidence. Figure 6 plots the testing accuracy after validation simulations and the percentage of simulated testing samples using different values of α. The accuracy reflects that of the trusted GPR predictions in addition to the instances validated through simulation. As one would expect, larger α values translate to better results in terms of accuracy and false alarms at the expense of higher simulation cost. It is important to note that the choice of α gives the user the flexibility to control the trade-off between the overall detection quality and the number of simulations needed.

V. Conclusion

In this work, we present Litho-GPA, a hotspot detection framework with Gaussian Process assurance to provide confidence in classifier prediction. The prediction accuracy is improved by exploring both the mean and confidence of prediction. Besides, an active data selection scheme based on weak classifiers is developed to reduce the computational cost in data preparation. Experimental results demonstrate Litho-GPA can achieve comparable accuracy to the state-of-the-art deep learning approaches while obtaining on average 28% reduction in false alarms.



(a) Layout 2 (b) Layout 3 (c) Layout 4&5

Fig. 6: The testing accuracy, number of false alarms and percentage of simulated testing samples for different α are shown.

ACKNOWLEDGMENT

This work is supported in part by NSF under award No. 1718570 and Toshiba Memory Corporation.

REFERENCES

[1] J. Kim and M. Fan, "Hotspot detection on Post-OPC layout using full chip simulation based verification tool: A case study with aerial image simulation," in *Proceedings of SPIE*, vol. 5256, 2003.

[2] E. Roseboom, M. Rossman, F.-C. Chang, and P. Hurat, "Automated full-chip hotspot detection and removal flow for interconnect layers of cell-based designs," in *Proceedings of SPIE*, vol. 6521, 2007.

[3] C. A. Mack, "Thirty years of lithography simulation," in *Optical Microlithography XVIII*, vol. 5754. International Society for Optics and Photonics, 2004, pp. 1–13.

[4] Y. Lin, M. B. Alawieh, W. Ye, and D. Z. Pan, "Machine learning for yield learning and optimization," in *IEEE International Test Conference (ITC)*, 2018.

[5] J. Xu, S. Sinha, and C. C. Chiang, "Accurate detection for process-hotspots with vias and incomplete specification," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2007, pp. 839–846.

[6] Y.-T. Yu, Y.-C. Chan, S. Sinha, I. H.-R. Jiang, and C. Chiang, "Accurate process-hotspot detection using critical design rule extraction," in *ACM/IEEE Design Automation Conference (DAC)*, 2012, pp. 1167–1172.

[7] S.-Y. Lin, J.-Y. Chen, J.-C. Li, W.-Y. Wen, and S.-C. Chang, "A novel fuzzy matching model for lithography hotspot detection," in *ACM/IEEE Design Automation Conference (DAC)*, 2013, pp. 68:1–68:6.

[8] W.-Y. Wen, J.-C. Li, S.-Y. Lin, J.-Y. Chen, and S.-C. Chang, "A fuzzy-matching model with grid reduction for lithography hotspot detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 33, no. 11, pp. 1671–1680, 2014.

[9] D. G. Drmanac, F. Liu, and L.-C. Wang, "Predicting variability in nanoscale lithography processes," in *ACM/IEEE Design Automation Conference (DAC)*, 2009, pp. 545–550.

[10] D. Ding, J. A. Torres, and D. Z. Pan, "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 30, no. 11, pp. 1621–1634, 2011.

[11] D. Ding, B. Yu, J. Ghosh, and D. Z. Pan, "EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2012, pp. 263–270.

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

[13] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, and C. Chiang, "Machine-learning-based hotspot detection using topological classification and critical feature extraction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, no. 3, pp. 460–470, 2015.

[14] H. Zhang, B. Yu, and E. F. Y. 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. 47:1–47:8.

[15] Y. Tomioka, T. Matsunawa, C. Kodama, and S. Nojima, "Lithography hotspot detection by two-stage cascade classifier using histogram of oriented light propagation," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2017, pp. 81–86.

[16] H. Zhang, F. Zhu, H. Li, E. F. Y. Young, and B. Yu, "Bilinear lithography hotspot detection," in *ACM International Symposium on Physical Design (ISPD)*, 2017, pp. 7–14.

[17] J. W. Park, A. Torres, and X. Song, "Litho-aware machine learning for hotspot detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 37, no. 7, pp. 1510–1514, 2018.

[18] T. Matsunawa, S. Nojima, and T. Kotani, "Automatic layout feature extraction for lithography hotspot detection based on deep neural network," in *SPIE Advanced Lithography*, vol. 9781, 2016.

[19] M. Shin and J.-H. Lee, "Accurate lithography hotspot detection using deep convolutional neural networks," *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol. 15, no. 4, p. 043507, 2016.

[20] 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.

[21] H. Yang, J. Su, Y. Zou, B. Yu, and E. F. Y. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," in *ACM/IEEE Design Automation Conference (DAC)*, 2017, pp. 62:1–62:6.

[22] W. Ye, Y. Lin, M. Li, Q. Liu, and D. Z. Pan, "LithoROC: Lithography hotspot detection with explicit ROC optimization," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2019.

[23] Y. Chen, Y. Lin, T. Gai, Y. Su, Y. Wei, and D. Z. Pan, "Semi-supervised hotspot detection with self-paced multi-task learning," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2019.

[24] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *International Conference on Learning Representations (ICLR)*, 2018.

[25] C. M. Bishop *et al.*, *Pattern Recognition and Machine Learning*. Springer New York, 2006, vol. 4, no. 4.

[26] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[27] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press, 2006.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct., pp. 2825–2830, 2011.

[29] A. J. 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)*, 2012.