Low-Cost Lithography Hotspot Detection with Active Entropy Sampling and Model Calibration

Yifeng Xiao[1], Miaodi Su[2], Haoyu Yang[3], Jianli Chen[1,4], Jun Yu[4], Bei Yu[3]
[1]Department of Microelectronics, Fudan University, Shanghai, 201203, China
[2]College of Mathematics and Computer Science, Fuzhou University, Fuzhou, 350108, China
[3]Department of Computer Science and Engineering, The Chinese University of Hong Kong, NT, Hong Kong, China
[4]State Key Lab of ASIC & System, Fudan University, Shanghai, 200433, China
yfxiao16@fudan.edu.cn; chenjianli@fudan.edu.cn

***Abstract*—With feature size scaling and complexity increase of circuit designs, hotspot detection has become a significant challenge in the very-large-scale-integration (VLSI) industry. Traditional detection methods, such as pattern matching and machine learning, have been made a remarkable progress. However, the performance of classifiers relies heavily on reference layout libraries, leading to the high cost of lithography simulation. Querying and sampling qualified candidates from raw datasets make active learning-based strategies serve as an effective solution in this field, but existing relevant studies fail to take sufficient sampling criteria into account. In this paper, embedded in pattern sampling and hotspot detection framework, an entropy-based batch mode sampling strategy is proposed in terms of calibrated model uncertainty and data diversity to handle the hotspot detection problem. Redundant patterns can be effectively avoided, and the classifier can converge with high celerity. Experiment results show that our method outperforms previous works in both `ICCAD2012` and `ICCAD2016` Contest benchmarks, achieving satisfactory detection accuracy and significantly reduced lithography simulation overhead.**

I. Introduction

Along with the rapid scaling of transistor feature size, the VLSI industry has been seriously challenged by increasing complexity and manufacturability issues. In chip design, though equipped with diverse resolution enhancement technologies, mask layouts still suffer from process variations, causing manufacturing defects. These defects, also known as hotspots, should be detected at early stages in chip design to ensure high fidelity.

A conventional hotspot detection scheme, illustrated in Fig. 1, consists of training set generation and hotspot detection. Training set sampling aims to set up the labeled layout patterns and fed them into hotspot detectors for further layout printability estimation. State-of-the-art hotspot detectors are mainly based on pattern matching and machine learning techniques. On one hand, pattern matching-based approaches [1], [2] are profoundly dependent on the identified hotspot patterns in the pattern library to identify matched patterns in new designs. Similar or same patterns in the entire layout will be sampled in this approach, resulting in high accuracy for known patterns but failing to predict unseen patterns. On the other hand, the superiority of machine learning-based approaches [3], [4] lies in the robust learning ability and the capability of detecting unseen patterns, while false alarm issues should be carefully treated.

Currently, the convolutional neural network (CNN) plays an increasingly significant role in hotspot detection [5]–[7] The first hotspot detection technique based on CNN in [5] is used to calibrate the model by generating virtual samples, [6] proposes a CNN-based framework for learning representative features, and [7] designs a CNN-based hotspot detection framework with double inception modules. However, it is high-priced to obtain labeled data during the design and manufacturing process of chips. For example, in the DUV era, it is necessary to build a lithography simulation model for full-chip detection and finding out hotspots [8], and in the EUV era, tapout and SEM images inspection are inevitable to localize the label hotspots.

This work was supported by the National Science Foundation of China under Grant 61977017 and the Fujian Science Fund for Distinguished Young Scholars under Grant 2019J06010.

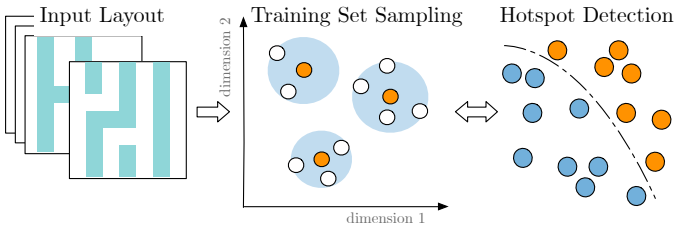


Fig. 1 A traditional process of training set sampling and hotspot detection.

To alleviate the above concerns, active learning-based frameworks are able to mitigate the tension of labeling. In the active learning, the learning engine is continuously fine-tuned by extending the training set with new labeled samples. A significant step of active learning is sampling qualified candidates with the most productive information into the training set, which also will mainly determine the labeling cost of the whole process. Two main concerns of data sampling in previous studies are the model uncertainty [9] and the data diversity [10], [11]. Specifically, a model is uncertain on an input instance when the instances are located near the hyper-plane in the raw data space, and data diversity can avoid redundancy of data sampling effectively via selecting the most representative instance. Also, a number of strategies take both uncertainty and diversity into consideration [12], [13]. For example, Ash *et al.* [13] takes the gradient of the loss function as an uncertainty metric and adopts the k-means++ seeding algorithm to ensure high diversity of the samples.

In the hotspot detection problem, Yang *et al.* [14] proposes an active learning framework combining pattern sampling and hotspot detection. The detection framework, however, is still flawed as detailed below. Firstly, in the sampling process, the model uncertainty is not actually considered with a poorly calibrated model [15] and the diversity metric with a convex optimization equation is time-consuming. Secondly, informative patterns may lose when samples are discarded in each iteration. Also, the error introduced from the relaxing integer constraints of the quadratic programming (QP) formulation will always cause diversity loss, and the loss is inevitable even if the two-step number of batch mode sampling is adjusted appropriately to improve the diversity.

To address these concerns, we develop a new batch mode selection method with model calibration for pattern sampling and hotspot detection that effectively coordinate uncertainty and diversity for a better performance. In the designed framework, model generality increases

with weights adjusted according to new labeled instances in the training set. The main contributions of this paper are listed as follows:

- We conduct a novel uncertainty score with calibration error addressed, selecting hotspot-like samples with higher uncertainty and improving the reliability of model confidences.
- We apply a new diversity metric with high efficiency in the sampling process, reducing computational cost and error rate compared with the QP method in [14].
- We present an entropy-based sampling strategy via applying the entropy weighting method, weighting the metrics according to information quantity.
- Experimental results show that the proposed strategy increases hotspot detection accuracy while significantly minimizing lithography simulation overhead and runtime. Specifically, 100% accuracy is achieved in `ICCAD16` benchmark.

The rest of this paper is organized as follows. Section II introduces some terminologies and definitions that are used throughout this paper. Section III provides details of the pattern sampling and hotspot detection flow. Section IV lists experimental settings and conduct the result comparisons, followed by conclusion in Section V.

II. Preliminaries

In this section, we will introduce some of the terminologies and essential contents related to hotspot detection. We use the following metrics to evaluate the performance of this framework.

Definition 1 (**Hit** [14]). *A hit is defined as when the detector reports hotspot on a clip of which at least one defect occurs at the core region. We also denote the ratio between the number of hits and the number of total hotspot clips as detection accuracy.*

Definition 2 (**Extra** [14]). *An extra is defined as when the detector reports hotspot on a clip of which no defect occurs at the core region.*

Definition 3 (**Litho-clip** [14]). *A litho-clip is a clip in the training set or an extra that is labeled hotspot or non-hotspot based on results of lithography simulation. The count of litho-clips reflects the lithography simulation overhead.*

According to the evaluation metrics above, we define the problem of layout pattern sampling and hotspot detection (PSHD) as follows.

Problem 1 (**PSHD** [14]). *Given a layout design, the objective of PSHD is sampling the representative clips that will generalize the hotspot pattern space and maximize the machine learning model generality, i.e., maximizing the detection accuracy while minimizing the number of litho-clips.*

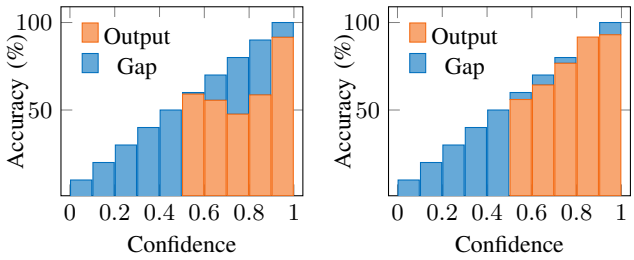
It can be concluded from the PSHD that the proposed framework requires full chip layout designs as input, and outputs a labeled dataset which consists of a training set and a validation set, an unlabeled dataset and a machine learning model. The evaluation metrics of this detection process consist of accuracy, defined in Equation (1),

$$Acc = \frac{\#HS_{Train} + \#HS_{Val} + \#Hits}{\#HS_{Total}}, \tag{1}$$

where $\#HS_{Train}$, $\#HS_{Val}$ and $\#HS_{Total}$ are the number of hotspots in the training set, validation set, and total data set, $Hits$ denotes the number of hotspots that are correctly predicted in the unlabeled set, and lithography simulation overhead, as shown in Equation (2),

$$Litho = \#Tr + \#Val + \#FA, \tag{2}$$

where $\#Tr$,$\#Val$ are the number of instances in the training set and validation set, respectively. $\#FA$ is the false alarm.



(a) Original (b) Calibrated

Fig. 2 Reliability diagrams of confidence v.s. Accuracy. Blue bars represent the calibration gap between confidence and accuracy, and orange bars represent average accuracy in different bins.

Also, throughout the paper, scalers are written as lowercase letters (e.g. x), vectors are denoted as bold lowercase letters (e.g. $\mathbf{x}$) and metrics are bold uppercase letters (e.g. $\mathbf{X}$).

III. The Algorithm

In this section, we will discuss details of our proposed pattern sampling strategy, including an entropy-based sampling algorithm and a summary of the entire sampling framework.

A. Entropy-based Sampling

In active learning, model uncertainty and pattern diversity are both key factors to ensure the model generality. Hence, we propose an entropy-based sampling according the uncertainty and diversity metrics, selecting the most useful instances into the training set.

1) *Calibrated Uncertainty Metric*

Uncertainty is the most common metric in active learning, such as the entropy of prediction probabilities [9], [12] and the Best-versus-Second-Best (BvSB) [10]. Generally, a classification model is more uncertain on an instance if the prediction probability is closer to 0.5, which also means the instance is near the separating hyper-plane. For a given unlabeled layout clip x_i from a set of n instances, the related binary BvSB can be denoted as follows,

$$u_i = 1 - |\sigma(z_i^{(0)}) - \sigma(z_i^{(1)})|, \tag{3}$$

where z_i (i = 1,...,n) denotes the output of the machine learning model, $\sigma(z_i^{(c)})$ is the output of the softmax function, signifying the prediction probability that x_i belongs to the c-th class (c = 0,1 in this case), as represented in Equation (4). u_i is the uncertainty score of sample x_i, and it is positively correlated with the model uncertainty of x_i.

$$\sigma(z_i)^{(c)} = \frac{\exp(z_i^{(c)})}{\sum_{j=1}^{C} \exp(z_i^{(j)})}. \tag{4}$$

However, the uncertainty in Equation (3) comes with disadvantages to be addressed for a learning engine based on poorly calibrated neural networks. Model calibration refers to the problem of predicting probability estimates representative of the true correctness likelihood [15]. We visualize it by partitioning predictions of our neural network into 10 equally-spaced bins and computing weighted averages of accuracy and confidence in different bins, as shown in Fig. 2(a), where blue bars represent the calibration gap between confidence and accuracy. Also, considering that biased datasets include limited hotspot instances, more importance should be attached to hotspot-like instances in the sampling process. In other words, we expect to sample instances both near the decision boundary and in hotspots regions. So we propose a novel hotspot-aware uncertainty score equipped with model calibration.

$$\sigma(z_i)^{(c)} = \frac{\exp(\frac{z_i^{(c)}}{T})}{\sum_{j=1}^{C} \exp(\frac{z_i^{(j)}}{T})}, \tag{5}$$

$$u_i = \begin{cases} \sigma(z_i)^{(0)} + h, & \text{if } \sigma(z_i)^{(1)} > h; \\ \sigma(z_i)^{(1)}, & \text{if } \sigma(z_i)^{(1)} < h. \end{cases} \tag{6}$$

As shown in Equation (5), we first calibrate the neural network by a post-processing method, temperature scaling [15], modifying the softmax function with parameter T. The results of the calibrated model are visualized in Fig. 2(b), where the prediction probabilities approach the true correctness likelihood. T ($T > 0$) is obtained by optimizing negative log likelihood, i.e. cross-entropy loss, on the validation set. It should be noted that the modified softmax function only affects the probabilities of instances rather than the prediction results. Symbolizing ground truth correctness likelihood, the calibrated confidences also indicate the reliability of prediction results, which is valuable when it comes to the real-world classifying task.

We also convert the uncertainty score with a parameter h, which is the decision boundary between 0 and 1. With the preference of hotspot-like instances and uncertain samples, the uncertainty score of hotspots should be higher than that of non-hotspots, and the probability close to h should result in a higher score, as shown in Equation (6). We fix $h = 0.4$ throughout our experiments since the datasets are imbalanced.

To sum up, *compared with the method proposed in [14], not only do we regard instances near decision boundary and in hotspot regions as more important, but address the calibration error in practice,* making the model prediction more accurate and adaptive in hotspot detection tasks.

2) ***Diversity Metric***

If only sampling on account of uncertainty metric, the selected instances will not have adequate information since samples with high uncertainty scores may be located in similar regions of the data space, possibly causing redundant instances in the training set. Various methods of diversity have been described in previous studies, such as the clustering method [11] and QP method [10], [14]. Observing the drawbacks of QP-based diversity sampling in [14], we propose a new metric that simply evaluates the diversity score of a single instance, as shown in Equation (7),

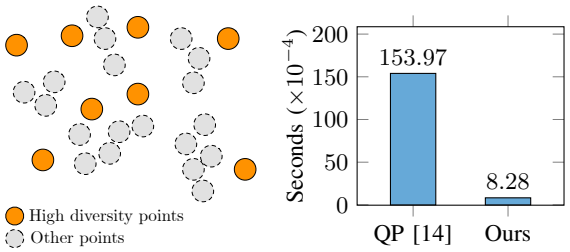
$$d_i = \min_{\mathbf{x} \in \mathcal{Q} \setminus \mathbf{x}_i} \mathsf{dist}(\mathbf{x}_i, \mathbf{x}), \tag{7}$$

where $\mathbf{x}_i$ is the features obtained from the fully-connected layer, dist is a distance function, $\mathcal{Q}$ represents the query set with n unlabeled instances, and d_i is the diversity score of a sample x_i. It is worth noticing that instances with the highest diversity scores tend to be sampled.

The design of dist function will be demonstrated below. A general method to measure the distance between two unlabeled instances is using the Gaussian kernel function, however, features are auto-learned from deep neural networks which attain much express power than Gaussian Kernel, so we use the normalized inner product of features for representing the similarity of two instances. We can further express the difference between two instances as Equation (8).

$$D_{ij} = 1 - \mathbf{x}_i^\top \cdot \mathbf{x}_j. \tag{8}$$

Specifically, $D_{ij} = 1$ represents the upper bound of diversity assessment and $D_{ij} = 0$ denotes the lower bound. Then we can obtain the diversity matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ or the distance matrix, whose entries are defined by Equation (8). Consequently, the minimum distance to neighbor feature points is taken as the diversity score, calculated by Equation (7), which implies the lowest value of each row other than the diagonal in $\mathbf{D}$, and instances with higher diversity scores d_i tend to be sampled from the query set. *Compared with the batch mode sampling method in [14], where data diversity is measured via addressing a QP problem, our method can achieve high efficiency with low computational cost and less error.*



(a) diversity visualization (b) runtime comparison

Fig. 3 Visualization of layout patterns diversity metric and runtime comparison.

We also visualize the diversity metric in Fig. 3(a). The instances with higher diversity scores are labeled in orange in the query set, and points away from clusters or near the boundary of groups tend to be selected with high possibility. Besides, we compare the runtime of the diversity metric in this paper with that in [14], as shown in Fig. 3(b). It can be seen that our diversity metric is less time-consuming.

3) ***Entropy-based Score***

To balancing the influences of two metrics, we design the entropy-based score to synthetically evaluate the contributions of uncertainty and diversity via normalizing and weighting two scores, as shown in as Equation (9),

$$s_i = \omega_1 \times u_i + \omega_2 \times d_i, \tag{9}$$

where ω_j ($j = 1, 2$) are the weights of the scores, satisfying $0 \leq \omega_j \leq 1$ and $\omega_1 + \omega_2 = 1$. Uncertainty score and diversity score need to be normalized in the dataset, while they are still expressed as u_i and d_i for convenience. We try to pick k entries with the highest entropy-based scores from the query set into the training set.

Next, we will focus on the determination of ω_j. Distributions of two indices (uncertainty and diversity) change constantly along with the query process, bringing various information quantity and representing different contributions in each iteration. In other words, weights should be determined dynamically in different iterations to avoid information loss. Specifically, if an indicator distributes evenly in the query set, the indicator will be less discrete with higher entropy. In this case, no matter how much weight is assigned to the indicator, it has no impact on the sampling results based on the entropy-based score, thus a weight of 0 should be given since the order of clips in the query set is totally determined by the other indicator. Consequently, to balance the influences of indicators and achieve better performance, we adopt the entropy weighting method to determine the weights of two scores according to the dispersion degree of indicators in each iteration.

Supposing that there are n samples with two indices, the entropy weighting method is described in details as follows: Firstly, n scores of two indices should be normalized, as shown in Equation (10),

$$r_{ij} = \frac{a_{ij} - \min\{a_{sj}\}}{\max\{a_{sj}\} - \min\{a_{sj}\}}, s = 1, \ldots, n, \tag{10}$$

where $a_{ij} (i = 1, 2, ..., n$ and $j = 1, 2)$ is the scores obtained from uncertainty metric and diversity metric, and r_{ij} is the normalized score.

We then calculate the proportion of i-th score among the total of j-th index, i.e. the disorder of each example, as shown in Equation (11).

The entropy of each index can be further calculated by Equation (12).

$$q_{ij} = \frac{r_{ij}}{\sum_{i=1}^{n} r_{ij}}, \tag{11}$$

$$E_j = -b \sum_{i=1}^{n} q_{ij} \ln q_{ij}. \tag{12}$$

Let $b = 1/\ln n$, so that the value of E_j can be guaranteed between 0 and 1. By analyzing the equation above, we can infer that if $q_{ij}(i = 1, \ldots, n)$ incline to be the same, the value of E_j approaches 1. Under such circumstances, the role of the indicator in linear combination can be weakened due to larger entropy and less information contained in the related scores. Consequently, the weights of the indices in the linear combination are determined by the consistency of q_{ij}, varying in different iterations. The dynamic weights are defined as follows:

$$\omega_j = \frac{1 - E_j}{2 - \sum_{j=1}^{2} E_j}. \tag{13}$$

Algorithm 1 describes the process of entropy-based sampling specifically. The algorithm requires a query set $\mathcal{Q}$, the number of patterns to be sampled k, and a parameter T used for temperature scaling. Firstly, to prevent the neural network from making inaccurate predictions, we adjust the softmax function for calibrated confidence according to Equation (5), which is used to obtain the uncertainty score $\mathcal{F}$ of the query set $\mathcal{Q}$ based on the binary BvSB strategy as Equation (6) (line 1); besides, we design the diversity score $\mathcal{D}$ in view of Equation (7), and the distance between features is provided by the diversity matrix, ensuring low time-consuming and sufficient information in selected samples (line 2); then the entropy-based score $\mathcal{S}$ is obtained to bring two metrics into full play by weighting the uncertainty score $\mathcal{F}$ and diversity score $\mathcal{D}$, where weights are determined by the entropy weighting method and Norm denotes the normalization function as Equation (10) (lines 3–4); finally, clips with k the highest entropy-based scores $\mathcal{S}$ in the query set $\mathcal{Q}$ will be sampled to form a set $\mathcal{B}$ (line 5).

Algorithm 1 $\mathsf{EntropySampling}(\mathcal{Q}, k, T)$

Require: $\mathcal{Q}, k, T$.
Ensure: $\mathcal{B}$.
1: $\mathcal{F} \leftarrow$ Compute uncertainty scores of $\mathcal{Q}$ by Equation (6) and Equation (5);
2: $\mathcal{D} \leftarrow$ Compute diversity scores of $\mathcal{Q}$ by Equation (7);
3: $\omega_j \leftarrow \mathsf{EntropyWeight}(\mathcal{F}, \mathcal{D})$, $j = 1, 2$;
4: $\mathcal{S} \leftarrow \omega_1 \times \mathsf{Norm}(\mathcal{F}) + \omega_2 \times \mathsf{Norm}(\mathcal{D})$;
5: $\mathcal{B} \leftarrow$ Sample k instances with the highest entropy-based scores ($\mathcal{S}$) from $\mathcal{Q}$;
6: **return** $\mathcal{B}$.

B. Overall Sampling Flow

Algorithm 2 describes the proposed overall pattern sampling flow. The algorithm requires the pre-processed features set $\mathcal{X}$, the number of patterns to be sampled in two steps n and k, and a standard variation σ used to initiate the sampling model. We first calculate posterior probabilities $\mathcal{P}$ of features based on Gaussian Mixture Model and split $\mathcal{X}$ into three sets: an original training set with labeled data $\mathcal{L}_0$, an unlabeled data set $\mathcal{U}_0$ to sample instances from, and a validation set $\mathcal{V}_0$ for temperature scaling (lines 1–2), where the samples with the lowest posterior probability, that is, hotspot-like samples, are taken as the original training set; we then initialize the model and train it based on $\mathcal{L}_0$ (lines 3–5); further, the iterative sampling procedure is implemented until the termination condition is satisfied. In each iteration, firstly, n instances are sampled in the light of $\mathcal{P}$ to form a query set $\mathcal{Q}$ (line 7) ; afterwards, the parameter T is optimized on $\mathcal{V}_0$

Algorithm 2 Overall Sampling Framework

Require: $\mathcal{X}, n, k, \sigma$.
Ensure: $\mathbf{w}$.
1: $\mathcal{P} \leftarrow$ Calculate posterior probabilities of the unlabeled dataset;
2: $\mathcal{U}_0, \mathcal{L}_0, \mathcal{V}_0 \leftarrow$ Split $\mathcal{X}$ based on $\mathcal{P}$;
3: Initialize $\mathbf{w} \sim \mathcal{N}(0, \sigma)$;
4: $\mathcal{U} \leftarrow \mathcal{U}_0, \mathcal{L} \leftarrow \mathcal{L}_0$;
5: $\mathbf{w} \leftarrow$Train the machine learning model based on $\mathcal{L}$.
6: **for** $i = 1 : N$ **do**
7: $\quad \mathcal{Q} \leftarrow$ Sample n instances with the lowest posterior probabilities ($\mathcal{P}$) from $\mathcal{U}$;
8: $\quad T \leftarrow$ get a modified parameter is optimized on $\mathcal{V}_0$;
9: $\quad \mathcal{B} \leftarrow \mathsf{EntropySampling}(\mathcal{Q}, k, T)$;
10: $\quad \mathcal{U} \leftarrow \mathcal{U} \backslash \mathcal{B}$;
11: $\quad \mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{B}$;
12: $\quad \mathbf{w} \leftarrow$Update machine learning model based on $\mathcal{L}$;
13: **end for**
14: **return** $\mathbf{w}$.

as one of the inputs of the entropy-based sampling algorithm, which outputs a set $\mathcal{B}$ consisting of k instances with the highest information as the second step of the sampling process (lines 8–9); labeled via lithography simulation, instances in $\mathcal{B}$ will be transferred from the unlabeled set $\mathcal{U}$ to the training set $\mathcal{L}$, and the machine learning model will be updated by adjusting the weights of each layer (lines 10–12). The trained model returns after N times iteration when the termination condition is satisfied. *It should be noted that in each iteration, we do not discard unselected samples in the query set to avoid loss of informative patterns.* Finally, the full chip detection will be applied to the rest of the unlabeled clips by the trained model. Prediction confidences of the model are trustworthy since they are well-calibrated by temperature scaling.

IV. Experimental Results

Our entropy-based sampling framework is implemented in Python with the TensorFlow library [16], and evaluated on two industrial benchmark sets: `ICCAD12` [17] and `ICCAD16` [18], as shown in TABLE I. Clips of `ICCAD16` are obtained from the given layouts and labeled using EUV lithography models as [14]. We can notice that `ICCAD16-1`, lacking hotspots, can be ignored in the following experiments.

A. Comparison with PSHD Methods

We first compare the sampling results of the proposed batch mode strategy with the exact/fuzzy matching methods and two sampling strategies on `ICCAD12` and `ICCAD16`, as listed in TABLE II. Columns "PM-exact", "PM-a95", "PM-a90", "PM-e2" correspond to the results acquired from pattern matching [2] respectively, where "PM-exact" denotes the exact pattern matching, while the others are three fuzzy matching methods under certain conditions. Column "TS" lists the results of a batch sampling method only based on calibrated confidence after the temperature scaling (TS) method, and column "QP" represents the method in [14], which applies the diversity metric via addressing a quadratic programming (QP) problem. Column "Acc(%)" denotes the accuracy of each method, and column "Litho#" lists the number of

TABLE I Statistics of benchmarks.

Benchmarks	HS #	NHS #	Tech (nm)
`ICCAD12`	3728	159672	28
`ICCAD16-1`	0	63	7
`ICCAD16-2`	56	967	7
`ICCAD16-3`	1100	3916	7
`ICCAD16-4`	157	1678	7

TABLE II Full chip pattern sampling and hotspot detection on ICCAD12/16 benchmarks.

Benchmarks	PM_exact [2]		PM_a95 [2]		PM_a90 [2]		PM_e2 [2]		TS		QP [14]		Ours	
	Acc(%)	Litho#	Acc(%)	Litho#	Acc(%)	Litho#	Acc(%)	Litho#	Acc(%)	Litho#	Acc(%)	Litho#	Acc(%)	Litho#
ICCAD12	100.00	127746	96.83†	38879†	73.38†	15923†	100.00	124320	98.44	12728	98.25	12280	98.25	9717
ICCAD16-2	100.00	1022	92.86	717	48.21	328	100.00	1022	96.42	752	100.00	881	100.00	640
ICCAD16-3	100.00	4838	99.64	4420	96.73	3717	99.91	4777	99.27	4225	99.74	3589	100.00	3693
ICCAD16-4	95.54	1134	2.55	65	1.91	20	78.34	842	93.63	1429	98.09	1608	100.00	1631
Average	98.89	33685	52.22	11020	55.06	4997	94.56	32740	96.94	4784	99.02	4590	99.56	3920
Ratio	0.993	8.593	0.525	2.811	0.553	1.275	0.989	8.352	0.974	1.220	0.996	1.171	**1.000**	**1.000**

†Experiments are conducted on the center 600 × 600 region of each clip.

clips being labeled via lithography simulation, including clips in the final labeled set and false alarms.

It can be seen that our approach can achieve 100% accuracy with less overhead on average on each case of ICCAD16, which is the highest accuracy among other previous works. Specifically, 98.25% of accuracy is achieved on ICCAD12 via the proposed method, with over 13 times fewer instances that need to be labeled than "PM-exact". Additionally, as for three different settings of the fuzzy matching method, results show that they fail to extract proper instances, obtaining a quite low accuracy value in some cases. Moreover, compared to QP, another batch mode sampling method of active learning, the litho-clips cost can be reduced by 17% in our method with higher accuracy. TS fails to achieve satisfactory accuracy, as listed in "TS", performing only 96.44% on average.

Comparisons of the results of sampling strategies cannot fully demonstrate the superiority of our method. On one hand, PSHD problem aims at maximizing detection accuracy while minimizing lithography overhead, but a higher accuracy always corresponds to increased overhead. On the other hand, since our workflow adopts CNN as the machine learning model, uncertain behavior can be introduced by weights initialization and batch sampling process. Thus another experiment is designed for trade-off analysis and model stability, as shown in Fig. 4. QP and TS are also taken into account.

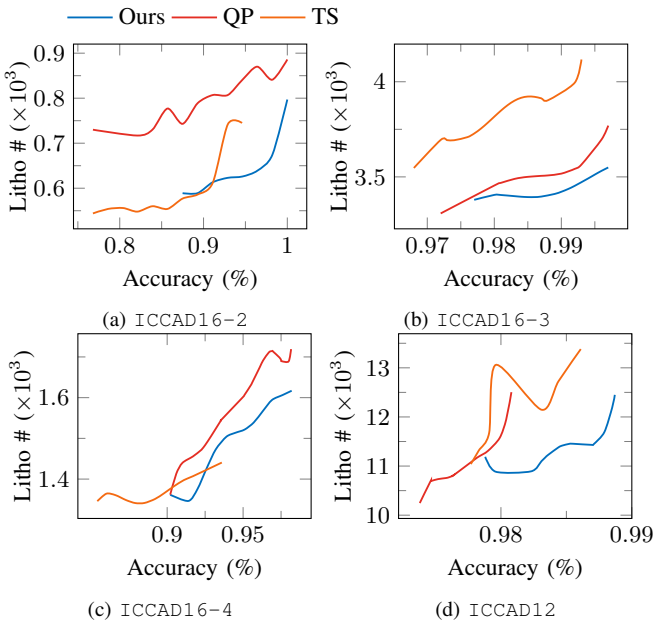


Fig. 4 Trade-off of different batch selection strategies. The x-axis represents various detection accuracy, and the y-axis denotes the average lithography simulation overhead of multiple processes.

TABLE III Components effectiveness verification of the entropy-based method on ICCAD12/16 benchmarks.

Benchmarks	w/o.E		w/o.D		w/o.U		Full	
	Acc(%)	Litho#	Acc(%)	Litho#	Acc(%)	Litho#	Acc(%)	Litho#
ICCAD12	98.12	11577	98.31	11648	98.27	11014	98.25	9717
ICCAD16-2	98.33	816	100.00	897	98.21	894	100.00	640
ICCAD16-3	99.83	4010	99.64	3788	99.82	3751	100.00	3693
ICCAD16-4	99.42	1801	99.36	1699	100.00	1747	100.00	1631
Average	98.93	4551	99.33	4507	99.08	4352	99.56	3920
Ratio	0.994	1.161	0.998	1.150	0.995	1.110	**1.000**	**1.000**

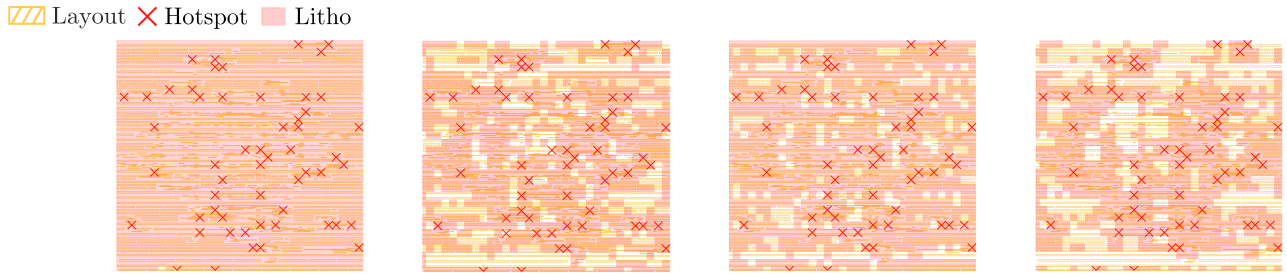
We implement the methods on ICCAD12 and ICCAD16 for a hundred times with alternative parameters. Then we average the lithography simulation overhead corresponding to different accuracy values, where the x-axis represents the detection accuracy, and the y-axis denotes the average lithography simulation overhead. We drop several low accuracy values and restrict the x-axis bound to better perform the trade-off of different batch selection methods. As shown in Fig. 4, the proposed framework obtains satisfactory results in both accuracy and lithography simulation overhead, while QP results in higher overhead, and TS fails to gain sufficient accuracy value. Results of ICCAD16-2, for instance, achieve 100 percent accuracy with 797 average lithography overhead, while QP yields 886 overhead; TS achieves the lowest simulation overhead but has a gap on accuracy. Additionally, the proposed method is further stable since the blue curve lies in a narrow range of the x-axis.

Fig. 5 visualizes outputs of different frameworks on the layout of ICCAD12-2, including exact pattern matching (PM-exact), TS, QP, and the proposed method.

B. Ablation Study

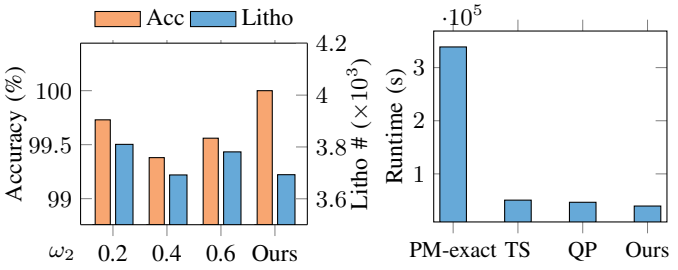
We then study the effect of different configurations of our framework. TABLE III illustrates the contribution of the entropy-based method. Results in column "w/o.E" derive from the framework without the entropy-based score. "w/o.D" stands for the framework without the diversity metric, "w/o.U" represents framework without the uncertainty metric, and "Full" is the entire framework. The results show that the framework with entire strategies achieves the best performance, in other words, uncertainty, diversity, and the entropy-based method are essential in the sampling framework.

We also test the effectiveness of the entropy weighting method. In the experiment, the entropy-based score with dynamic weights is compared with different fixed ω_2: 0.2, 0.4, and 0.6. The experiment results of detection on ICCAD16-3 are shown in Fig. 6(a), where the x-axis denotes various weights of the diversity score, and the y-axis represents prediction accuracy and lithography simulation overhead. Obviously, dynamic weights outperform fixed weights in both criteria, so the effectiveness of the entropy weighting method is verified.



(a) PM-exact [2] (b) TS (c) QP [14] (d) Ours

Fig. 5 Hotspots distribution and sampled clips of different methods on `ICCAD16-2` layout. The red forks denote positions of real hotspots, and the shadow area represents the overall lithography simulation overhead.



(a) weight comparison (b) runtime comparison

Fig. 6 Comparison of fixed weight and dynamic weights and runtime comparison among different solutions.

C. Runtime Comparison

Runtime is also the main improvement of our entropy-based sampling framework. On one hand, the diversity score we used avoids solving the convex optimization equation with intensive computation; on the other hand, our framework can reduce the lithography simulation overhead, which is the main source of real backend layout verification flow. The overall runtime is evaluated by the summation of 10s of penalty on each litho-clip [14] and PSHD overhead, as shown in Fig. 6(b). It can be seen that the proposed framework is more efficient with performance optimization.

V. Conclusion

A novel batch mode selection strategy has been proposed for hotspot detection in chip design, consisting of a calibrated hotspot-aware uncertainty metric and an efficient diversity metric. Embedded in a pattern sampling and hotspot detection framework, a query set formed according to posterior probability firstly, and then the entropy-based score was applied for further sampling, where the entropy weighting method is conducted to determine the weights. Compared with the state-of-the-art sampling strategies, the experimental results have shown that the proposed batch sampling strategy, along with learning model optimization, can not only effectively benefit model generality, but also reduce the cost in both lithography simulation overhead and runtime. We hope this study can offer a new perspective in hotspot detection with active learning techniques, formulating novel sampling strategies in terms of uncertainty and diversity metrics from different methods.

References

[1] F. Yang, S. Sinha, C. C. Chiang, X. Zeng, and D. Zhou, "Improved tangent space-based distance metric for lithographic hotspot classification," *IEEE TCAD*, vol. 36, no. 9, pp. 1545–1556, 2016.

[2] K. Chen, Y. Chuang, B. Yu, and S. Fang, "Minimizing cluster number with clip shifting in hotspot pattern classification," in *Proc. DAC*, 2017, pp. 1–6.

[3] 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 TCAD*, vol. 34, no. 3, pp. 460–470, 2015.

[4] 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 *Proc. SPIE*, vol. 9427. International Society for Optics and Photonics, 2015, p. 94270S.

[5] M. Shin and J. Lee, "Accurate lithography hotspot detection using deep convolutional neural networks," *JM3*, vol. 15, no. 4, p. 043507, 2016.

[6] H. Yang, L. Luo, J. Su, C. Lin, and B. Yu, "Imbalance aware lithography hotspot detection: a deep learning approach," *JM3*, vol. 16, no. 3, p. 033504, 2017.

[7] J. Chen, Y. Lin, Y. Guo, M. Zhang, M. Alawieh, and D. Pan, "Lithography hotspot detection using a double inception module architecture," *JM3*, vol. 18, no. 1, p. 013507, 2019.

[8] E. Roseboom, M. Rossman, F. Chang, and P. Hurat, "Automated full-chip hotspot detection and removal flow for interconnect layers of cell-based designs," in *Proc. SPIE*, vol. 6521. International Society for Optics and Photonics, 2007, p. 65210C.

[9] J. Zhu, H. Wang, B. Tsou, and M. Ma, "Active learning with sampling by uncertainty and density for data annotations," *IEEE ASL*, vol. 18, no. 6, pp. 1323–1331, 2009.

[10] G. Wang, J. Hwang, C. Rose, and F. Wallace, "Uncertainty sampling based active learning with diversity constraint by sparse selection," in *MMSP*. IEEE, 2017, pp. 1–6.

[11] R. Zhang and A. Rudnicky, "A large scale clustering scheme for kernel k-means," in *Object recognition supported by user interaction for service robots*, vol. 4, 2002, pp. 289–292.

[12] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *IJCV*, vol. 113, no. 2, pp. 113–127, 2015.

[13] J. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds," *arXiv:1906.03671*, 2019.

[14] 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 TCAD*, 2020.

[15] C. Guo, G. Pleiss, Y. Sun, and K. Weinberger, "On calibration of modern neural networks," in *Proc. ICML*, vol. 70. JMLR. org, 2017, pp. 1321–1330.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, 2016, pp. 265–283.

[17] J. Torres, "ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite," in *Proc. ICCAD*, 2012, pp. 349–350.

[18] R. Topaloglu, "ICCAD-2016 CAD contest in pattern classification for integrated circuit design space analysis and benchmark suite," in *Proc. ICCAD*, 2016, pp. 1–4.