Integration

Semi-supervised lithography hotspot detection based on feature fusion and residual attention

--Manuscript Draft--

Manuscript Number:	VLSIJ-D-24-00517
Article Type:	Research Paper
Keywords:	lithography hotspot detection; integrated circuits; semi-supervised learning; feature fusion
Abstract:	Given that traditional lithography hotspot detection methods based on semi-supervised learning struggle to meet the detection accuracy requirements of advanced integrated circuit (IC) manufacturing and to address the accuracy loss caused by dataset imbalance, a semi-supervised detection method based on feature fusion and residual attention is proposed in this paper. This approach incorporates the inception module into the feature extraction network as a multi-scale feature fusion module, allowing the model to capture and combine features at various scales. By incorporating the convolutional block attention module (CBAM) and residual blocks into a joint multitask network structure, attention toward layout areas is enhanced. The introduction of a weighted cross-entropy loss function effectively mitigates dataset imbalance issues. This method effectively leverages a large number of unlabeled data for training, improving the accuracy of lithography hotspot detection in the case of insufficient labeled data. The experimental results show that compared with the existing semi-supervised lithography hotspot detection methods, the proposed method has improved accuracy, false alarm, F1 score, and overall detection simulation time using 10%~50% of training data on the ICCAD 2012 contest benchmarks by 3.48%, 22.03%,12.76% and 20.26%, respectively.

Highlights

- A multi-scale feature fusion module is introduced to enhance the receptive field of the model, enabling it to capture broader context information and local details of layouts.
- A classification and clustering network based on residual attention is developed, and the network is constructed using residual blocks with the convolutional block attention module.
- The classification and clustering network with joint learning can fully utilize a large amount of unlabeled data to enhance the performance of the hotspot detector.
- A weighted cross-entropy loss function is developed, which assigns different weights to hotspots and nonhotspots in the dataset, aiming to alleviate the challenges posed by imbalanced dataset.

Semi-supervised lithography hotspot detection based on feature fusion and residual attention

Hui Xu[a], Xinzhong Xiao[a], Wenxin Huang[a], Ruijun Ma[a,*],Fuxin Tang[b], Pan Qi[b], Ye Yuan[a], Huaguo Liang[c],

[a]School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China

[b]School of Artificial Intelligence, Anhui University of Science and Technology, Huainan 232001, China

[c]School of Microelectronics, Hefei University of Technology, Hefei 230009, China

Abstract

Given that traditional lithography hotspot detection methods based on semi-supervised learning struggle to meet the detection accuracy requirements of advanced integrated circuit (IC) manufacturing and to address the accuracy loss caused by dataset imbalance, a semi-supervised detection method based on feature fusion and residual attention is proposed in this paper. This approach incorporates the inception module into the feature extraction network as a multi-scale feature fusion module, allowing the model to capture and combine features at various scales. By incorporating the convolutional block attention module (CBAM) and residual blocks into a joint multitask network structure, attention toward layout areas is enhanced. The introduction of a weighted cross-entropy loss function effectively mitigates dataset imbalance issues. This method effectively leverages a large number of unlabeled data for training, improving the accuracy of lithography hotspot detection in the case of insufficient labeled data. The experimental results show that compared with the existing semi-supervised lithography hotspot detection methods, the proposed method has improved accuracy, false alarm, F1 score, and overall detection simulation time using 10%~50% of training data on the ICCAD 2012 contest benchmarks by 3.48%, 22.03%,12.76% and 20.26%, respectively.

Keywords: lithography hotspot detection, integrated circuits, semi-supervised learning, feature fusion

1 Introduction

As the feature sizes continue to shrink on integrated circuits, the manufacturability of the associated designs faces significant limitations due to technological bottlenecks, especially in the lithography process. The critical size of the nanostructure in the lithography patterns has dropped below the wavelength of the illumination light. Due to the influence of the optical proximity effect (OPE) [1], the diffraction effect of the light hinders the exposure system from achieving the best imaging of the mask pattern. Optical proximity correction (OPC) [2] technology is widely used in the industry to correct exposure errors in silicon patterns. Due to the influence of the layout design, even after the application of OPC, lithography patterns will still exist defects like short circuits and open circuits, called lithography hotspots. Figure 1 shows examples of hotspot layout clips, and the red regions indicate known hotspots.

★This work was supported in part by the Special Fund for Research on National Major Research Instruments of China under grant 62027815, the National Natural Science Foundation of China under grants 61834006.

*Correspondence author

Email: ruijun_ma@foxmail.com (Ruijun Ma)

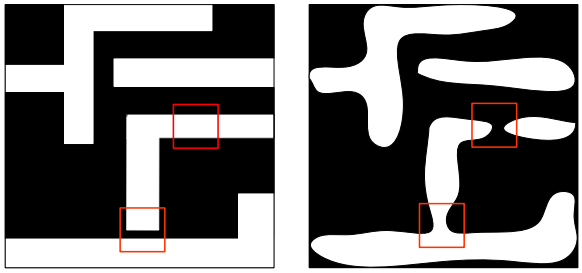

Fig. 1 (a) Mask layout and (b) corresponding hotspots in the layout clips.

The existing lithography hotspot detection methods mainly include lithography simulation-based [3] [4], pattern matching-based [5] [6], and machine learning-based [7] [8] lithography hotspot detection methods. However, these methods have inherent limitations, including significant computational overhead, limited robustness, and a cumbersome feature extraction process. As technology advances, deep learning-based lithography hotspot detection techniques have emerged as a promising alternative.

Deep learning based approaches automatically learn complex feature representations from data by constructing and training multi-layer neural networks and finally obtaining detection results through recognition detection. Shin et al. [9] applied the convolutional neural network (CNN) algorithm to lithography hotspot detection, and transformed the hotspot detection problem into a typical image classification problem. Yang et al. [10] proposed a deep convolutional neural network to improve the lithography hotspot detector accuracy, a bias learning technique to solve the problem of class imbalance in the dataset and the discrete cosine transform (DCT) to preprocess the layout data. Jiang et al. [11] simplified the layout clips into a binary image, and used a binary neural network to construct a hotspot detection model to improve the detection efficiency. The above methods are all supervised learning methods, which depend entirely on labeled samples for training. Its performance can suffer significantly when the amount of labeled data is limited. In this case, semi-supervised learning, which can make full use of unlabeled samples for training, becomes an attractive alternative.

This paper proposes a semi-supervised lithography hotspot detection method, which realizes the gradual introduction of unlabeled data into training and reduces the dependence of the model on a large amount of labeled data. The method predicts unlabeled data with a classification flow to obtain pseudo-labels, and then selects unlabeled data based on clustering confidence for the next training round, reducing the impact of incorrect pseudo-labels.

The main contributions in this paper are summarized as follows:

- A multi-scale feature fusion module is introduced into our hotspot detector to enhance the receptive field of the model, enabling it to capture broader context information and local details of layouts.
- A classification and clustering network based on residual attention is developed in our hotspot detector. The network is constructed using residual blocks with CBAM attention mechanism, deepening the overall network while avoiding issues like gradient vanishing.
- A weighted cross-entropy loss function is developed in the classification flow of our hotspot detector. This function assigns different weights to hotspots and nonhotspots in the dataset, aiming to alleviate the challenges posed by imbalanced dataset.
- The experimental results show that the framework outperforms the state-of-the-art works, our hotspot detection method achieves 3.48% better accuracy, the false alarm is reduced by 22.03%, the F1 score is increased by 12.76%, and the overall detection simulation time (ODST) is reduced by 20.26% using 10%~50% of training data on the ICCAD 2012 contest benchmarks.

The remainder of this paper is organized as follows. Section 2 presents metrics and problem formulation on hotspot detection. Section 3 describes the proposed method. Section 4 reports the experimental results of the proposed approach. Finally, Sect.5 concludes this paper.

2. Metrics and problem formulation

The relevant terms and evaluation metrics in hotspot detection are defined as follows: TP (FN) is the case of predicting the actual hotspot as a hotspot (nonhotspot), and TN (FP) is the case of predicting the actual nonhotspot as a nonhotspot (hotspot).

Definition 1 (*Accuracy*). The ratio between the number of correctly predicted hotspots and the total number of actual hotspots [12]:

$$Accuracy = \frac{\sum TP}{\sum (TP + FN)}. \quad (1)$$

Definition 2 (*Precision*). The ratio between the number of correctly predicted hotspots and the total number of predicted hotspots:

$$Precision = \frac{\sum TP}{\sum (TP + FP)}. \quad (2)$$

Definition 3 (*False Alarm* (*FA*)). The number of actual nonhotspots that are predicted as hotspots by the model [12]:

$$False\ Alarm = \sum FP. \quad (3)$$

Definition 4 (*F1 Score*). Indicators that take into account both precision and accuracy:

$$F1\ \ Score = \frac{2 \times Precision \times Accuracy}{Precision + Accuracy}. \quad (4)$$

Definition 5 (*Overall Detection and Simulation Time* (*ODST*)). The total lithographic simulation time for predicted hotspots, and the time for model evaluation [12]:

$$ODST = t_{ev} + t_{ls} \times FA, \quad (5)$$

where t_{ev} and t_{ls} are the evaluation time and the lithography simulation time on each layout clip, respectively. We assumed t_{ls} is 10 seconds for similar comparison with previous works.

Problem 1 (*Semi-supervised Lithography Hotspot Detection Problem*). Use a set of labeled and unlabeled hotspot and nonhotspot samples to train a model to maximize the accuracy with minimal false alarm over the test samples.

Problem 2 (*Dataset Category Imbalance*). The ICCAD 2012 introduced an open-source dataset for evaluating hotspot detection methods, which has since become the most widely used benchmark suite [13], consisting of five different sets of layout clips as shown in Table 1. The proportion of categories in 'ICCAD-2', 'ICCAD-4', and 'ICCAD-5' is seriously imbalanced, causing a serious imbalance problem in the lithography hotspot detector, resulting in the classifier neglecting the minority class of hotspots or overfitting to the majority class of nonhotspots [14].

Table 1 The statistics of benchmark datasets.

Benchmark	Training		Testing	
	HS#	NHS#	HS#	NHS#
ICCAD-1	99	340	226	4679
ICCAD-2	174	5285	498	41298
ICCAD-3	909	4643	1808	46333
ICCAD-4	95	4452	177	31890
ICCAD-5	26	2716	41	19327

3 Semi-supervised lithography hotspot detection network

3.1 Multi-scale feature fusion module

The formation of lithography hotspots is primarily attributed to light diffraction, making the spatial correlation of local area features within layout clips critically important. To enhance the hotspot detector's ability to capture layout features effectively, we concatenate two inceptions [15] as the multi-scale feature fusion module (MFF) of the backbone network, allowing the model to learn features at different scales, as shown in Figure 2.

This module is used for feature extraction, and the whole extraction process is regarded as a process of splitting, extraction and fusion. Firstly, 1×1, 3×3, and 5×5 convolution kernels are used to convolve the input at different scales in parallel to obtain feature information at different scales. Then, 3×3, 5×5 convolutions are decomposed into 1×3, 3×1 convolutions and 1×3, 3×1, 1×3, 3×1 convolutions by using separable convolution. The use of separable convolution can significantly reduce the model parameters and computational complexity and

increase the nonlinear ability of the model. Finally, the above branches are concatenated on the channel dimension to obtain the output of the whole feature extraction module.

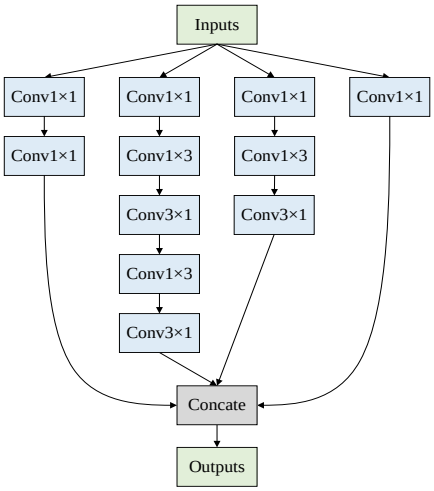

Fig. 2 Multi-scale feature fusion module(MFF)

The multi-scale feature fusion module presented in this paper enhances the width of convolutional layers and incorporates kernels of various scales within a single layer. The model can learn more fine-grained features at deeper levels of representation by stacking two inception modules. As a result, the hotspot detector can effectively learn the features of layout clips, leading to improved detection accuracy.

3.2 Residual attention module

With the increase of network depth, not only did the learning performance get better, but there were two problems in the actual experiment. Firstly, gradient disappearance and gradient explosion. Secondly, the degradation problem, which is the prediction performance is getting worse with the increase in the number of layers. This paper introduces a residual attention module (RA) based on ResNet [16] to address these issues, which deepens the network while retaining the advantages of shallow networks. This effectively avoids network degradation and improves optimization performance.

RA module based on convolutional block attention module (CBAM) is used [17], as shown in Fig. 3(a). By introducing CBAM attention into the basic residual block, the model pays more attention to the important features of space and channel. CBAM enhances the representation of learned features in the model by adaptively applying attentional weighting to feature maps in both channel and spatial dimensions. This weighted mechanism enables the network to learn and express features more effectively, enhancing the model's representational capacity.

The neck network named RANeck consists of three RA modules, as depicted in Fig. 3(b). When is_basic is true, the network performs downsampling to reduce the dimensions of the input tensor and feature maps. In contrast, when is_basic is false, no downsampling occurs, and the network maintains a straightforward residual connection by adding the output of the second convolutional block directly to the original input tensor. This approach preserves feature integrity by passing feature information without modifying spatial dimensions or channel configurations.

In summary, we have utilized RANeck to build a multi-task network for joint classification and clustering. We applied residual blocks in the backbone network, allowing the direct addition of original layout clips features to subsequent layer outputs. This approach ensures that the network retains more original layout features throughout the learning process. Furthermore, we integrated the CBAM module within the residual blocks. This implementation enables the model to effectively select and emphasize important feature channels, suppressing irrelevant or noisy channels and enhancing the model's representational capacity and generalization ability.

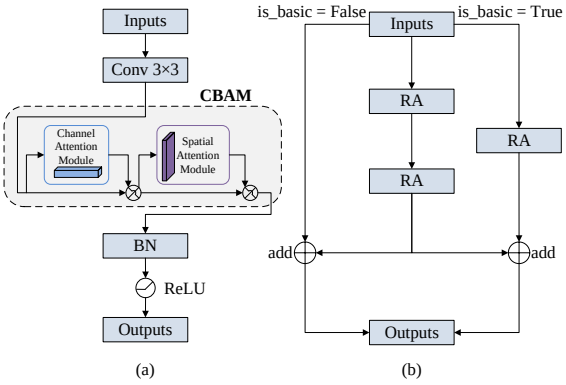

Fig. 3 (a) The structure of RA module and (b) the RANeck composed of RA

3.3 Joint multi-task network structure for hotspot detection

The challenges of semi-supervised learning include the decrease in accuracy due to the scarcity of labeled data and the errors in the model's prediction of unlabeled data. In this paper, a joint multi-task network [18] is used to alleviate these problems through joint learning classification and clustering models. The structure of the network is shown in Fig. 4.

The hotspot detector has two branches of the shared feature extraction layer, which are the classification flow that performs pseudo-label prediction and classification tasks, and the clustering flow that provides pseudo-label confidence. Although the two branches of the network share the feature extraction layer and have the same network structure, the loss function is different, and the weight of the model is also learned separately. The training loss function of classification flow adopts weighted cross-entropy loss, in which the weight in the loss function is calculated by the number of hotspots and nonhotspots in the training data. The purpose is to alleviate the problem caused by the imbalance of datasets. The loss function is formulated as follows:

$$L_{class} = -\sum_{i=1}^{N} w_i^1 (\sum_{k=1}^{K} w_i^2 y_i^k \log \hat{y}_i^k), \tag{6}$$

where N is the number of input samples, K is the number of classes and K=2 in lithography hotspot detection. The vectors y_i and $\hat{y}_i$ are the one-hot encoding [19] corresponding to the true label of the sample and the probability output of the *i*th sample, respectively. The weight w_i^1 is the confidence of the *i*th sample, where the weight of the labeled sample is 1, and the weight of the unlabeled sample is calculated according to the loss provided by the clustering flow. The weight w_i^2 is the category weight in the weighted cross-entropy loss function. When the number of samples in the two categories is quite different, the model will be biased towards the majority category in the training process, thus introducing the category weight to alleviate the problem. w_i^2 is a tensor containing two elements. The first element is the weight of the hotspots, and the second element is the weight of the nonhotspots. Suppose that the number of hotspots is a and the number of nonhotspots is b, the weight in the loss function is calculated by then $w_i^2 = [\frac{b}{a+b}, \frac{a}{a+b}]$.

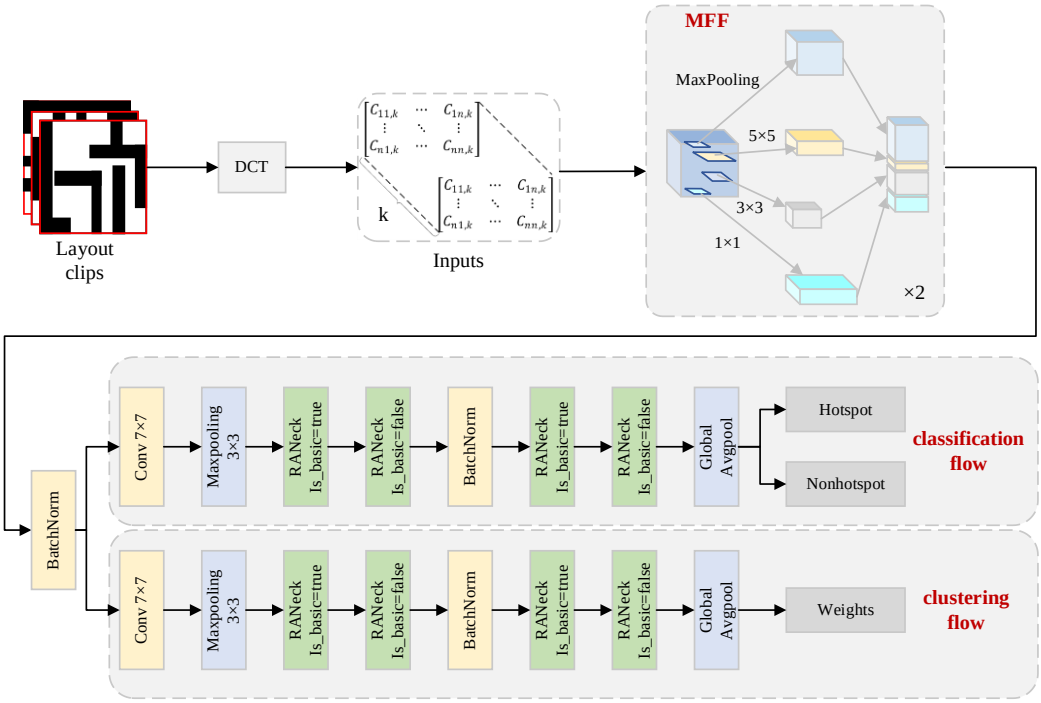

Fig. 4 Lithography hotspot detection network structure based on feature fusion and residual attention

The clustering flow assigns weights to each sample based on the distance from unlabeled samples to the predicted clusters. The distance from the unlabeled sample to the predicted cluster is generated by the Kullback-Leibler (KL) divergence [20], which is an indicator to measure the similarity of the two probability distributions. Its formula and loss function are defined as follows:

$$KL(p \,\|\, q) = \sum_{i=1}^{N} p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right), \tag{7}$$

$$L_{cluster} = \sum_{i,j=1}^{N} KL(x_i \,||\, x_j), \tag{8}$$

where $p(x_i)$ and $q(x_i)$ are the softmax layer outputs of the pth and qth samples, respectively. N represents the number of classes, with N=2 in our problem.

The pseudo-labels of unlabeled samples closer to other samples in the prediction cluster should have higher confidence. Therefore, the confidence of each sample is determined by the average KL divergence from itself to other samples within the same cluster. The average distance is defined as follows:

$$d_i = \frac{\sum_{j=1}^{N} KL(x_i \,\|\, x_j) P(x_i, x_j)}{\sum_{j=1}^{N} P(x_i, x_j)}, \tag{9}$$

where $P(x_i, x_j)$ =1 if (x_i, x_j) is a similar pair, otherwise 0. The weight of the ith sample can be obtained by normalizing d_i.

This method effectively utilizes unlabeled samples that are easy to obtain to improve model performance and solve the challenges posed by a limited number of labeled samples. This method of gradually introducing unlabeled samples into training has improved the performance of the lithography hotspot detection model, promoting a more stable and accurate learning process.

4 Experimental Results

4.1 Experimental setup

This section focuses on showcasing the experimental results that validate the effectiveness of our proposed method. The model is evaluated on 28-nm industrial benchmarks from ICCAD 2012 CAD contest, as described in Table 1. These benchmarks, derived from Ref. [10], utilize datasets processed through DCT from the ICCAD 2012 contest data. Among them, ICCAD-1 is omitted because the number of samples is too small, and it is the 32-nm process. The columns “Training HS#” and “Training NHS#” indicate the total number of hotspots and nonhotspots in the training set, respectively. Similarly, the columns “Testing HS#” and “Testing NHS#” show the total number of hotspots and nonhotspots in the testing set.

This multi-task network is implemented in Python with Tensorflow 2.4.0 [21] with an 8-core 3.20-GHz CPU, an Nvidia RTX 3060 GPU, and 6-GB memory.

To evaluate the performance of the hotspot detector with varying amounts of labeled data, we trained the model using different ratios of labeled data: {0.1, 0.3, 0.5, 0.7, 0.9, 1.0}. Table 2 contains the relevant configuration information for the model training in this paper. The model starts training with a learning rate of 0.001, then gradually decreases by 0.65 times after 3200 training steps to ensure smaller updates as it approaches the optimal solution. Additionally, to prevent overfitting, a random dropout ratio of 0.5 is applied at the classifier layer following global average pooling.

Table 2 Training configurations.

Training configurations	Value
Optimizer	Adam[22]
Initial learning rate	0.001
Learning rate decay	0.65
step	3200
Batch size	32
Dropout ratio	0.5
Training round	4

4.2 Performance evaluation

We evaluate the accuracy and false alarm on the testing dataset, as shown in Table 3. “DAC” represents a lithography hotspot detection method using a biased learning algorithm [10]. “SSL” denotes to the original

self-paced semi-supervised learning algorithm [18]. "ISSL" denotes an imbalance-aware self-paced semi-supervised learning algorithm [18]. "RSSL" denotes to the self-paced multi-task network based on feature fusion and residual attention that is applied in this paper.

At a ratio of 0.1, RSSL improves both accuracy and false alarm by an average of 6.91% (68.72% to 75.63%) and 21.28% (1001 to 788), respectively. At a ratio of 0.3, RSSL achieves comparable hotspot detection accuracy to ISSL while exhibiting significantly fewer false alarms and the false alarm is improved on an average of 25.54% (744 versus 554). At higher ratios, such as 0.9 and 1.0, the average accuracy of the two models shows no significant difference, given the ample amount of labeled training data available. However, the number of false alarm is significantly reduced by 6.42% and 20.47%, respectively. High-accuracy models often pay more attention to the judgment of subtle differences, which may lead to misjudgment in some cases. Simultaneously, it is observed that the performance of RSSL is suboptimal when utilizing low-rate labeled data with a ratio of 0.1 or 0.3 in benchmark 5. This is attributed to benchmark 5 being highly biased, with very few hotspots, resulting in too few hotspots in the initially divided dataset. This leads to incorrect pseudo-label predictions for unlabeled data by the model, thereby compromising the detection performance of the model.

Figure 5 shows the trend of accuracy change of datasets with different benchmarks under different ratios. In Fig. 5(a), the accuracy of DAC shows a rising trend with the increase of the ratio and its variation range is relatively large. Meanwhile, RSSL maintains a relatively stable trend, with consistently high accuracy that fluctuates only within a small range. In particular, in Fig. 5(b), the performance of RSSL is not the best because this paper achieves the lowest false alarm in this accuracy, that is, a better compromise is selected. In addition, in Fig. 5(c) and Fig. 5(d), due to the smaller amount of data in benchmark 4 and benchmark 5, there is a serious data category imbalance problem. With the increasing proportion of labeled samples, RSSL achieves a more stable accuracy when the accuracy of all models fluctuates greatly, which indicates that RSSL has better robustness.

Table 3 Accuracy and false alarm comparison for different amount of labeled training data.

Benchmark	Models	0.1		0.3		0.5		0.7		0.9		1.0	
		Accu(%)	#FA	Accu(%)	#FA	Accu(%)	#FA	Accu(%)	#FA	Accu(%)	#FA	Accu(%)	#FA
b2	DAC[10]	89.44	700	93.33	383	96.51	297	97.11	294	97.79	287	97.19	239
	SSL[18]	97.99	1643	98.11	643	97.67	425	97.87	265	97.51	211	97.75	231
	ISSL[18]	96.91	968	96.95	305	97.99	307	97.51	227	97.47	212	97.79	207
	RSSL	97.21	696	96.81	160	98.20	141	98.40	118	98.80	252	99.00	163
b3	DAC	97.94	4288	98.34	3569	98.04	3098	98.17	3001	98.22	2780	98.22	2878
	SSL	98.47	5130	98.43	3593	98.26	3083	98.15	2740	98.24	2665	98.27	2854
	ISSL	97.77	2524	97.83	2345	97.83	2513	97.95	2701	98.25	2618	98.16	2938
	RSSL	97.66	2026	97.55	1716	97.93	2070	97.93	2093	97.82	2351	97.98	2395
b4	DAC	35.14	230	65.99	315	78.19	359	77.85	261	90.73	387	91.75	309
	SSL	52.66	536	73.45	342	81.69	379	84.29	261	88.81	317	90.62	306
	ISSL	54.35	454	72.54	282	81.69	309	81.24	256	82.82	272	89.60	343
	RSSL	79.03	367	82.80	285	81.18	335	85.48	75	91.40	312	90.32	243
b5	DAC	8.29	3	27.32	39	75.12	86	90.73	72	93.66	79	95.61	90
	SSL	14.63	11	40.00	73	84.88	104	96.59	141	94.71	100	95.12	94
	ISSL	25.85	59	40.49	43	88.29	107	93.17	97	95.12	75	93.17	87
	RSSL	28.60	64	42.56	58	90.70	69	97.67	50	95.34	60	97.67	43
Average	DAC	57.70	1305	71.24	1076	86.97	960	90.97	907	95.10	883	95.69	879
	SSL	65.94	1830	77.50	1163	90.63	998	94.23	852	94.82	823	95.44	871
	ISSL	68.72	1001	76.95	744	91.45	809	92.47	820	93.42	794	94.68	894
	RSSL	75.63	788	79.93	554	92.00	653	94.87	584	95.84	743	96.24	711

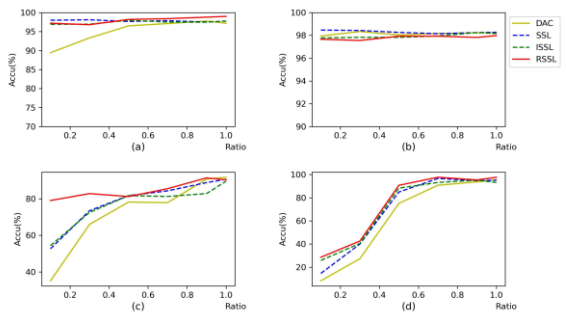

Fig. 5 Comparison of testing accuracy versus ratio of training dataset for (a) benchmark 2, (b) benchmark 3, (c) benchmark 4, and (d) benchmark 5, respectively.

Correspondingly, Fig. 6 shows the trend of false alarm change of datasets with different benchmarks under different ratios. On the one hand, the advantages of RSSL over some previous models can be seen in Fig. 6(a) and Fig. 6(b), which greatly reduces the number of false alarm while improving the accuracy. In Fig. 6(c) and Fig. 6(d), due to the small amount of data in benchmark 4 and benchmark 5, the RSSL model achieves a higher detection accuracy at the expense of certain false alarms. It is stricter and more accurate in classifying samples when the model's accuracy is high. This means that the model may also mark samples that are otherwise marginal or close to the edge, resulting in an increase in the number of false alarm. As the selected ratio increases, the prediction accuracy of the RSSL model is improved while the false alarm is further reduced.

As shown in Table 4, the F1 score metric is used to measure the comprehensive performance of DAC, SSL, ISSL and RSSL on ICCAD 2012 and compare them. According to this table, in comparison to ISSL, the F1 score of RSSL shows improvement by 8.11% on benchmark 2, 8.77% on benchmark 3, 18.2% on benchmark 4, and 20.5% on benchmark 5. Therefore, RSSL exhibited superior overall performance compared to ISSL.

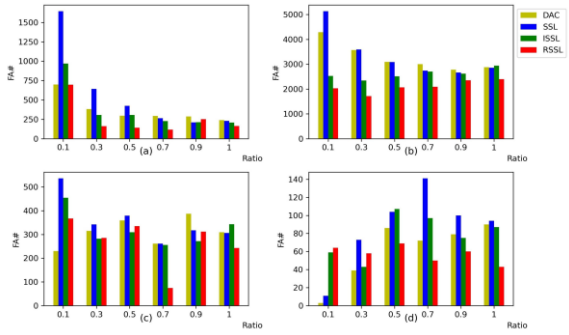

Fig. 6 Comparison of testing false alarm versus ratio of training dataset for (a) benchmark 2, (b) benchmark 3, (c) benchmark 4, and (d) benchmark 5, respectively.

Table 5 presents the mean ODST of DAC, SSL, ISSL, and RSSL across various ratios of labeled data. From this table, compared with ISSL, the ODST of RSSL is 30.34% better on benchmark 2, 18.99% better on benchmark 3, 15.82% better on benchmark 4 and 30.44% better on benchmark 5. Therefore, RSSL demonstrated superior performance over ISSL in ODST.

Table 4 The F1 score of DAC, SSL, ISSL and RSSL under different amount of data. The F1 score of DAC, SSL, ISSL and RSSL under different amount of data.

Benchmark	Models	0.1	0.3	0.5	0.7	0.9	1	Average
b2	DAC[10]	0.54	0.69	0.75	0.76	0.77	0.79	0.72
	SSL[18]	0.37	0.60	0.69	0.78	0.81	0.80	0.68
	ISSL[18]	0.50	0.75	0.75	0.80	0.81	0.82	0.74
	RSSL	0.58	0.85	0.87	0.89	0.79	0.85	0.80
b3	DAC	0.45	0.50	0.53	0.54	0.56	0.55	0.52
	SSL	0.41	0.50	0.53	0.56	0.57	0.55	0.52
	ISSL	0.58	0.60	0.58	0.56	0.57	0.54	0.57
	RSSL	0.63	0.67	0.63	0.62	0.60	0.59	0.62
b4	DAC	0.26	0.38	0.41	0.48	0.44	0.50	0.41
	SSL	0.23	0.40	0.41	0.51	0.48	0.50	0.42
	ISSL	0.26	0.44	0.46	0.50	0.49	0.47	0.44
	RSSL	0.41	0.48	0.44	0.75	0.50	0.55	0.52
b5	DAC	0.13	0.24	0.39	0.49	0.48	0.46	0.37
	SSL	0.21	0.25	0.39	0.36	0.43	0.45	0.35
	ISSL	0.20	0.34	0.39	0.43	0.50	0.46	0.39
	RSSL	0.21	0.29	0.50	0.61	0.56	0.65	0.47

Table 5 The average ODST(s) of DAC, SSL, ISSL and RSSL under different amount of data.

Benchmark	DAC[10]	SSL[18]	ISSL[18]	RSSL
b2	4847	5817	3830	2668
b3	38946	33562	26185	21212
b4	4758	3688	3313	2789
b5	1007	992	900	626

4.3 Ablation studies

4.3.1 Best Number of Inception Modules

We evaluated the impact of the number of inception modules on the performance of the proposed lithography hotspot detector. Specifically, we evaluate the performance of networks with one, two, and three inception modules, and the experimental results under labeled data proportions of 0.1, 0.5, and 1.0 are presented in Table 6. All results shown in the table are the mean detection outcomes across benchmark2-5. It can be observed that using one inception module achieves the highest accuracy at the cost of increased false alarms. Compared to using a single inception module, the accuracy with two inception modules decreases by 0.19%, but the number of false alarms significantly reduces by approximately 12.70%. Utilizing three inception modules results in a comparable number of false alarms to that of using two inception modules, while the detection accuracy is 0.70% lower than with two inception modules. On the one hand, the F1 score is highest when using two inception modules; on the other hand, fewer inception modules have a lesser impact on model training speed. Therefore, we introduced two inception modules within the MFF module.

Table 6 Experimental results using different numbers of inception modules

Number of inceptions	0.1					0.5					1					Average				
	Accu(%)	#FA	F1	cpu(s)	ODST	Accu(%)	#FA	F1	cpu(s)	ODST	Accu(%)	#FA	F1	cpu(s)	ODST	Accu(%)	#FA	F1	cpu(s)	ODST
one	76.21	911	0.43	93	9203	92.18	788	0.58	89	7969	96.05	764	0.6	90	7730	88.15	821	0.54	91	8301
two	75.63	788	0.48	95	7975	92	653	0.61	94	6624	96.24	711	0.66	96	7206	87.96	717	0.58	95	7268
three	75.03	769	0.5	98	7788	91.55	640	0.6	97	6497	95.2	695	0.62	98	7048	87.26	701	0.57	98	7111

4.3.2 Validation of Method Effectiveness

To further validate the method's effectiveness, ablation experiments are proposed to test the effects of different improvement measures. The detailed results of the model with labeled data proportions of 0.1, 0.5, and 1.0 are shown in Table 7. All results presented in the table are the mean detection results across benchmark2-5. ISPL represents a semi-supervised network using imbalance-aware self-paced multi-task learning, RA represents a method using residual attention modules, MFF represents a multi-scale feature fusion method, and Wloss represents a method using a weighted cross-entropy loss function.

Table 7 Ablation experimental results with a labeled data ratio of 10%, 50% and 100%

Method				**0.1**					**0.5**					**1**				
ISPL	RA	MFF	Wloss	Accu(%)	#FA	F1	cpu(s)	ODST	Accu(%)	#FA	F1	cpu(s)	ODST	Accu(%)	#FA	F1	cpu(s)	ODST
✓				68.72	1001	0.38	120	10130	91.45	809	0.55	120	8210	94.68	894	0.57	120	9060
✓	✓			59.18	830	0.41	92	8392	84.88	702	0.63	89	7109	94.20	662	0.68	100	6720
✓		✓		62.31	1064	0.32	84	10724	91.96	1120	0.55	92	11292	95.80	989	0.57	103	9993
✓			✓	56.10	860	0.45	97	8697	87.83	684	0.69	95	6935	92.14	544	0.62	99	5539
✓		✓	✓	67.85	889	0.46	105	8995	90.23	797	0.65	112	8082	96.20	862	0.64	114	8734
✓	✓	✓	✓	75.63	788	0.48	95	7975	92.00	653	0.61	94	6624	96.24	711	0.66	96	7206

According to the results of Table 7, after the application of RA, the accuracy have been declined to varying degrees, but other indicators have been improved to varying degrees. In particular, the false alarms are improved by 17%, 13% and 26%, respectively, which reduces the probability of target missed events. The MFF module improves the CPU of the model in different proportions of labeled data by about 22%, and the false alarm and ODST have been increased slightly. As the ratio of labeled data increases, the impact of the module on false alarms gradually decreases. The Wloss method uses a weighted cross-entropy loss function to alleviate the impact of class imbalance in the dataset, which reduces the false alarm of the model in different labeled data by about 23% on average. At the same time, it can be seen that other indicators, except the accuracy, have improved to varying degrees. The introduction of the MFF module on the basis of ISPL+Wloss can further improve the accuracy of the model when the ratio of labeled data is low, but it will introduce more false alarms than using only Wloss.

In fact, the detection performance of the model can be improved to the best state by adding RA, MFF, Wloss in turn to the original ISPL network model, which shows that the accuracy is increased by 3.0%, the false alarm is reduced by 20.0%, the F1-score increased by 20.0%, the CPU decreased by 21.0%, and the ODST decreased by 21.0%. It can be seen that the improvement measures adopted in this paper can effectively improve the overall performance of the traditional ISPL network model, and the improved network model can give full play to the advantages of each improved module to achieve the purpose of improving the comprehensive detection performance of the network model.

5 Conclusion

An improved semi-supervised lithography hotspot detection method based on feature fusion and residual attention is proposed. Through joint learning using a classification and clustering network, the model leverages a large number of unlabeled samples for training, resulting in significant detection performance, which holds practical research significance. This method introduces a multi-scale feature fusion module capable of extracting features at different scales and integrating them to enhance the model's ability to perceive and represent features across scales. Additionally, a classification and clustering network based on residual attention is introduced to deepen the overall network architecture. By reusing modules, this approach alleviates gradient vanishing issues and accelerates network convergence. Furthermore, improvements to the loss function include using a weighted cross-entropy loss function to address dataset class imbalance. Experimental results demonstrate that our method outperforms existing semi-supervised hotspot detection methods, presenting a novel framework for lithography hotspot detection.

References

[1] H. Carre, R. H. Doxtator, and M. C. Duffy. Semiconductor manufacturing technology at IBM. IBM Journal of Research and Development. 26(5) (1982) 528-531.

[2] A. Chen, Y. M. Foong, T. Thaler, et al. Aerial image metrology for OPC modeling and mask qualification. Proceedings of the 33rd European Mask and Lithography Conference. 2017.

[3] J. Mitra, P. Yu, and D. Z. Pan. RADAR: RET-aware detailed routing using fast lithography simulations. Proceedings of the 42nd Annual Design Automation Conference. 2005.

[4] P. Gupta, A. B. Kahng, S. Nakagawa, et al. Lithography Simulation-Based Full-Chip Design Analyses. Proceedings of the Design and Process Integration for Microelectronic Manufacturing. 2006.

[5] W. Wen, J. Li, Lin S., et al. A fuzzy-matching model with grid reduction for lithography hotspot detection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 11(33) (2014) 1671-1680.

[6] F. Yang, S. Sinha, C. C. Chiang, et al. Improved Tangent Space-Based Distance Metric for Lithographic Hotspot Classification. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 9(36) (2017) 1545-1556.

[7] T. Matsunawa, J R. Gao, B. Yu, et al. A new lithography

hotspot detection framework based on AdaBoost classifier and simplified feature extraction, 2015. Proceedings of the Design-Process-Technology Cooptimization for Manufacturability. 2015.

[8] D. Ding, X. Wu, J. Ghosh, et al. Machine Learning Based Lithographic Hotspot Detectionwith Critical-Feature Extraction and Classification. Proceedings of the IEEE International Conference on IC Design and Technology. 2009.

[9] M. Shin and J. H. Lee. Accurate lithography hotspot detection using deep convolutional neural networks. Journal of Micro/Nanolithography, MEMS, and MOEMS. 4(15) (2016) 043507-043507.

[10] H. Yang, J. Su, Y. Zou, et al. Layout hotspot detection with feature tensor generation and deep biased learning. IEEE Trans. on CAD of Integrated Circuits and Systems, 6(38) (2019) 1175-1187.

[11] Y. Y. Jiang, F. Yang, B. Yu, et al. Efficient layout hotspot detection via binarized residual neural network ensemble. IEEE Trans. on CAD of Integrated Circuits and Systems. 7(40) (2021) 1476-1488.

[12] Q. Zhang, Y. Zhang, J. Li, et al. Litho-NeuralODE 2.0: Improving hotspot detection accuracy with advanced data augmentation, DCT-based features, and neural ordinary differential equations. Integration 85 (2022): 10-19.

[13] J. A. Torres. ICCAD-2012 CAD contest in fuzzy pattern matching for physical verificationand benchmark suite. Proceedings of the International Conference on Computer-Aided Design. 2012.

[14] M. S. Santos, J. P. Soares, P. H. Abreu, et al. Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches. IEEE Computational Intelligence Magazine. 2018.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, et al. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[16] K. He, X. Zhang, Ren S., et al. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[17] S. Woo, J. Park, J. Y. Lee, et al. Cbam: Convolutional block attention module. Proceedings of the European conference on computer vision. 2018.

[18] Y. Chen, Y. Lin, T. Gai, et al. Semi-supervised hotspot detection with self-paced multi-task learning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 7(39) (2020) 1151-1523.

[19] L. Jie, C. Jiahao, Z. Xueqin, et al. One-hot encoding and convolutional neural network based anomaly detection. Journal of Tsinghua University (Science and Technology). 7(59) (2019) 523-529.

[20] J. Shlens. Notes on kullback-leibler divergence and likelihood. Mathematics, 2014.

[21] M. Abadi, A. Agarwal, P. Barham, et al. Tensorflow: Large-scale machine learning on heterogeneous systems. 2015.

[22] P. D. Kingma and J. Ba. Adam: A method for stochastic optimization. Computer Science, 2014.