When Transformer Meets Layout Hotspot: An End-to-End Transformer-based Detector with Prior Lithography

Wenbo Xu*
School of Integrated Circuits, Nanjing University
Suzhou, China
wbxu@smail.nju.edu.cn

Silin Chen*
School of Integrated Circuits, Nanjing University
Suzhou, China
silin.chen@smail.nju.edu.cn

Jiale Li
School of Integrated Circuits, Nanjing University
Suzhou, China
dajiahaowoshiljl@gmail.com

Kangjian Di
School of Integrated Circuits, Nanjing University
Suzhou, China
kangjiandi@smail.nju.edu.cn

Yuxiang Fu
School of Integrated Circuits, Nanjing University
Suzhou, China
yuxiangfu@nju.edu.cn

Ningmu Zou
School of Integrated Circuits, Nanjing University
Suzhou, China
Interdisciplinary Research Center for Future Intelligent Chips (Chip-X), Nanjing University
Suzhou, China
nzou@nju.edu.cn

Abstract

With the advancement of semiconductor technology and the continuous miniaturization of integrated circuit components, detecting the hotspots in layout designs becomes increasingly challenging. This has led to the emergence of numerous deep learning-based hotspot detectors in recent years. However, existing deep learning methods rely heavily on parameters that are closely related to the hotspots defined in the training dataset, which makes the model sensitive to parameters and requires a significant amount of human effort to adjust for different product design parameters. This greatly limits flexibility and generalization. To address these issues, we propose an anchor-free, end-to-end transformer-based hotspot detector that removes the reliance on diverse handcrafted parameters, enabling the model to focus directly on identifying potential hotspot areas. In order to incorporate more prior knowledge into the hotspot detector and improve the interpretability of hotspot detection, we provide query variables to the detector through a pretrained lithography simulator. We also introduce a query initialization module and a feature aggregation module based on the transformer decoder to effectively integrate layout features and lithographic priors. Experimental results validate the effectiveness of our approach, demonstrating superior performance compared to state-of-the-art methods.

CCS Concepts

• **Computing methodologies** → **Object detection**; • **Hardware** → **Electronic design automation**; **Electronic design automation**.

Keywords

Design for manufacturability, anchor-free, hotspot detection, lithography simulation.

ACM Reference Format:
Wenbo Xu, Silin Chen, Jiale Li, Kangjian Di, Yuxiang Fu, and Ningmu Zou. 2025. When Transformer Meets Layout Hotspot: An End-to-End Transformer-based Detector with Prior Lithography. In *Great Lakes Symposium on VLSI 2025 (GLSVLSI '25), June 30–July 02, 2025, New Orleans, LA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3716368.3735160

1 Introduction

With the rapid advancement of semiconductor technology, the shrinking size of integrated circuit components is continually pushing the boundaries of current chip manufacturing processes. As the feature sizes decrease, ensuring the printability of layout designs becomes progressively more difficult, resulting in an increased occurrence of manufacturing defects. This challenge is particularly pronounced in critical regions, known as hotspots, where the risk of defects is heightened. Therefore, it is crucial to accurately and efficiently locate hotspots in the layout.

Currently, there are three main methods for hotspot detection: lithography simulation, pattern matching and machine learning. Conventional lithography simulation is highly accurate but extremely time-consuming, rendering it impractical for large-scale production environments. In contrast, pattern matching methods use a set of predefined hotspot layout patterns to identify potential hotspots in a new design. Although faster than conventional lithography simulation, pattern matching is limited by its inability to detect novel or previously unseen hotspots.

*Both authors contributed equally to this research.
Corresponding author: Ningmu Zou
This work was supported in part by Postgraduate Research Practice Innovation Program of Jiangsu Province (Grant Number: KYCX25_0394) and the National Natural Science Foundation of China (62341408).

GLSVLSI '25, New Orleans, LA, USA

ACM ISBN 979-8-4007-1496-2/25/06
https://doi.org/10.1145/3716368.3735160

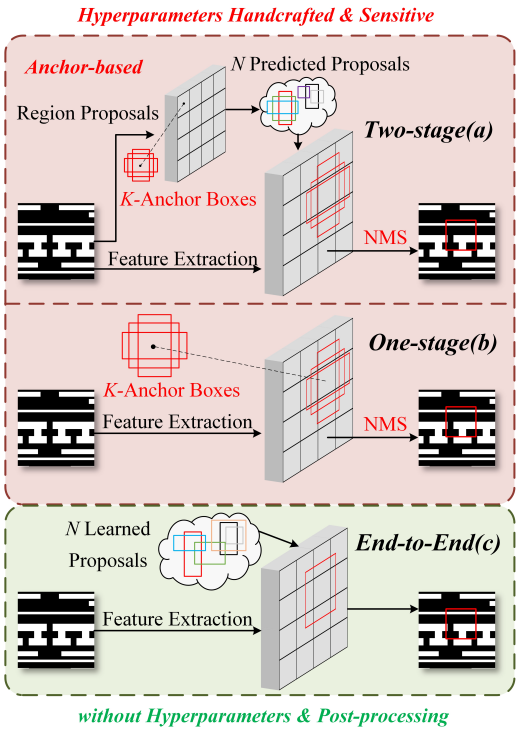

Figure 1: Illustration of the hotspot detection flow. (a) The pipeline of the existing two-stage method. (b) The pipeline of the existing one-stage method. They are both anchor-based methods, which require the manual setting of a large number of sensitive hyperparameters and post-processing. (c) The pipeline of our proposed framework without hyperparameters and post-processing.

Different from traditional pattern matching methods, machine learning-based hotspot detection techniques, particularly deep learning approaches[1–7], have shown exceptional generalization capabilities. By leveraging automated feature learning, these methods have achieved significant improvements in both accuracy and efficiency. For example, as shown in Fig. 1(a), Chen developed a two-stage hotspot detector based on Faster-RCNN[8]. Their model employs an inception-based feature extractor and a clip proposal network to generate region proposals and successfully addresses the issue arising from the imbalanced data distribution[5]. As shown in Fig. 1(b), Zhu proposed a more efficient one-stage detection framework by eliminating the region proposal stage[6]. However, there are still some issues with existing deep learning-based methods. The previous two-stage and one-stage hotspot detectors relied on anchor-based model frameworks to generate candidate boxes, which required the manual setting of a large number of sensitive hyperparameters and post-processing. These hyperparameters include the sizes and aspect ratios of anchor boxes and so on, which are used to define candidate regions for object detection. This approach lacks flexibility in adapting to hotspots of varying scales and shapes. Furthermore, as shown in Fig. 1, existing methods still require NMS (Non-Maximum Suppression), meaning they are not end-to-end and consume additional computational resources.

To address these issues, we propose a transformer-based end-to-end hotspot detector driven by a GPU-accelerated lithography simulator. As shown in Fig. 1(c), it eliminates the need for dataset-related hyperparameters. Furthermore, the prior knowledge from the lithography simulator is integrated into the framework, directing the model to focus on the potential hotspot areas, significantly improving interpretability and generalization. The main contributions of this paper are summarized as follows:

- We propose an anchor-free, end-to-end transformer-based detector for hotspot detection, eliminating the reliance on the diverse handcrafted parameters, which can provide the possibility for other deep learning-based hotspot detectors.
- We integrate prior knowledge from the lithography simulator into the framework, guiding the model to detect potential hotspot regions instead of identifying a hotspot pattern already seen and improving the interpretability and generalization.
- We design a query initialization module and a feature aggregation module based on a transformer decoder, which effectively enhances the model's ability to combine the features of the hotspot in the layout and prior knowledge from the lithography simulator.

2 Preliminary

In the chip manufacturing process, all patterns must be transferred onto the silicon wafer through steps such as photolithography. However, due to various physical variations during the manufacturing process, some patterns may experience deviations, resulting in defects. These defects can cause failures in the circuitry, thereby affecting the chip's functionality. The design areas that are prone to failure due to process variations during photolithography are referred to as hotspots.

In our work, different from [1] and [2], we approach hotspot detection as both a classification and localization task, rather than a simple binary classification problem. In this task, we not only need to care about whether the input image is hotspot or non-hotspot, but also focus on the specific location of the hotspot. Therefore, we do not adopt the original classification evaluation metrics such as Accuracy(Acc), but instead use the more reasonable Average Precision(AP) and Recall as the evaluation metrics for hotspot detection. The following definitions and metrics are used to evaluate the performance of the hotspot detector.

Definition 1 (Average Precision). The ratio of the number of correctly predicted hotspot regions by the hotspot detector to the total number of predicted hotspot regions by the hotspot detector.

Definition 2 (Recall). The ratio of the number of correctly predicted hotspot regions by the hotspot detector to the total number of ground-truth hotspot regions.

As the Average Precision(AP) increases, the number of false alarms(false positives), where non-hotspots are incorrectly predicted as hotspots, decreases. As the Recall increases, the number of missed hotspots (false negatives) decreases.

To match the evaluation metrics above, our hotspot detection problem is formulated as follows.

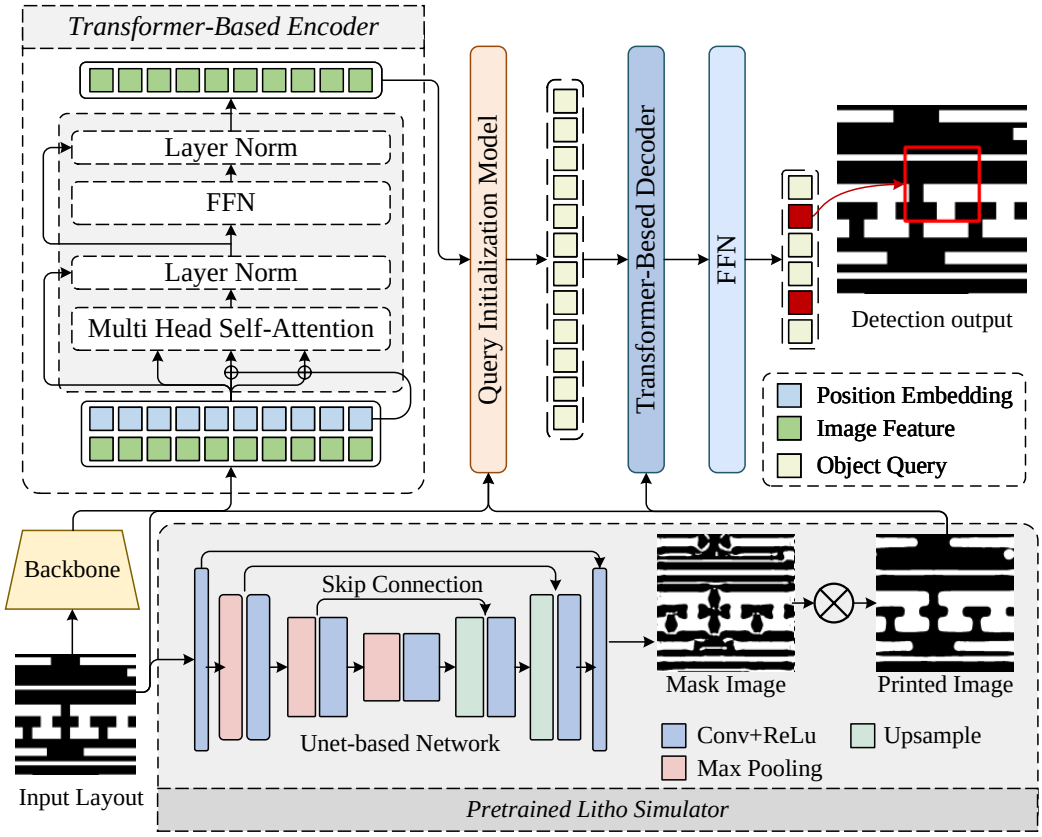

Figure 2: The framework of our proposed model. The framework consists of two main parts: i) a priori lithography simulator. ii) a transformer-based object detector. The detector includes the backbone, transformer encoder, transformer decoder and feed-forward networks (FFN).

Problem 1 (Hotspot Detection). Given a collection of clips containing hotspot layout patterns, the goal of hotspot detection is to train a detector to classify and locate all the hotspots, thereby maximizing the Average Precision and Recall.

3 Methodology

3.1 Overview

We design an anchor-free, transformer-based end-to-end hotspot detector with prior lithography. As shown in Fig. 2, the proposed framework includes three key components: i) a transformer-based object detector, ii) a pretrained lithography simulator, iii) a feature aggregation module based on a transformer decoder. The feature aggregation module enables the model to better combine layout features with prior knowledge obtained by lithography simulation for metrics, which improves the accuracy of hotspot detection by embedding hotspot-prone regions into object queries. At the same time, the model eliminates the need for hyperparameters and post-processing, thus simplifying the architecture and improving scalability.

3.2 Object Detector

3.2.1 Backbone. The backbone of our model is the ResNet50 [9] architecture, which effectively extracts features from the input layout images. It processes the input layout images $x_{img} \in \mathbb{R}^{3\times H_0\times W_0}$ to generate feature maps $f \in \mathbb{R}^{C\times H\times W}$, learning the 2D representations of the input. The extracted features are then flattened and passed to the transformer encoder to capture global dependencies, thereby facilitating accurate hotspot detection.

3.2.2 Transformer-Based Encoder. In our model, the Transformer Encoder plays a critical role in capturing the global dependencies and spatial relationships inherent in layout patterns for hotspot detection. Initially, we apply a 1x1 convolution to the high-level activation map f from the ResNet50 backbone, reducing the channel dimension from C to a smaller dimension d, creating a new feature map $z_0 \in \mathbb{R}^{d\times H\times W}$. Since the encoder expects a sequence as input, we flatten the spatial dimensions of z_0 into a $d \times HW$ sequence, transforming the feature map into a sequence of d-dimensional vectors that can be processed by the transformer.

Each layer of the Transformer Encoder consists of a multi-head self-attention module and a feed-forward network (FFN). The multi-head self-attention mechanism enables the model to capture both local and global context by attending to different parts of the layout

simultaneously, which is essential for identifying hotspots that may not be immediately adjacent but are contextually significant across the entire layout. To handle spatial information, we supplement the encoder with fixed positional encodings, which are added to the input of each attention layer. These positional encodings ensure that spatial relationships are incorporated into the self-attention mechanism, allowing the model to understand the layout's structure and pinpoint potential hotspot regions accurately.

The encoder processes the feature sequence in parallel, enabling efficient computation while preserving the ability to detect complex, multi-scale hotspot patterns. The output from the encoder, enriched with both local features and global context, is then passed to the decoder for the final detection of hotspots.

3.2.3 Transformer Decoder. The feature aggregation module based on the transformer decoder in our model leverages the integration of lithography simulation priors to enhance hotspot detection accuracy. Different from [5], [7], the decoder processes N object queries in parallel at each layer, effectively transforming these learnable positional embeddings into predictions for hotspot locations and classifications. The integration of lithography simulation data ensures that the object queries are guided by domain-specific knowledge, focusing on regions most likely to be hotspots.

As shown in Fig. 3, each decoder layer consists of two key components, including the Self-Attention Module and the Encoder-Decoder Attention Module. The Self-Attention Module allows object queries to reason about pairwise relationships and dependencies among potential hotspots, ensuring that the model captures the spatial and contextual relationships between different regions.

The Encoder-Decoder Attention Module allows the object queries to focus on the global features extracted by the encoder, seamlessly integrating the rich information about the layout and hotspot-prone areas provided by the lithography simulator.

The learnable object queries, enhanced by the lithography simulator, are added to the input of each attention layer, ensuring that the decoder maintains spatial awareness and context. The decoder's output embeddings are independently passed through a feed-forward network (FFN), which predicts the bounding box coordinates and class labels for each hotspot, resulting in N final predictions.

3.2.4 Feed-Forward Network (FFN). In our model, the FFN consists of a 3-layer perceptron with ReLU activation and a hidden dimension of size d, followed by a linear projection layer. The FFN predicts the center coordinates, height, and width of each bounding box, while the linear layer outputs the class label using softmax.

3.3 Prior Lithography Simulator

Hotspot detection is mainly used to identify and detect chip design areas prone to manufacturing problems and operational problems. There are many kinds of hotspots, including those caused by insufficient lithography processes. Traditional lithography simulation methods are too time-consuming. To better enhance the accuracy and interpretability of hotspot detection, we use a GPU-accelerated inverse lithography algorithm, NeuralILT[10]. It provides prior information about potentially sensitive hotspot locations to the detector while also considering computational efficiency. NeuralILT

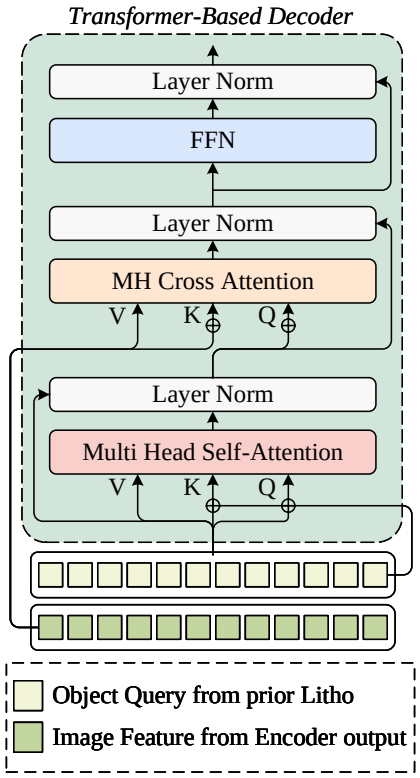

Figure 3: Illustration of the Transformer-Based Decoder.

is an end-to-end mask optimization model with a UNet, which the corresponding mask according to the input layout. Lithography Simulation uses optical projection and photoresist model to convert mask M into photoresist image Z. After obtaining the mask predicted by NeuralILT, the corresponding aerial image I and printed image Z are obtained through lithography Simulation. The former represents the distribution of light intensity on the wafer, and the latter represents the image on the wafer after lithography. The process of lithography simulation can be derived by using Hopkins diffraction theory [11]:

$$I = U(M) = \sum_{k=1}^{K} \mu_k |h_k \otimes M|^2 \tag{1}$$

where h_k is the kth optical kernel function and μ_k is the corresponding weight. The notation $\otimes$ stands for convolution operation, which computes the squared modulus of each element.

$$Z(x,y) = \sigma_z(I(x,y)) = \frac{1}{1 + e^{(-\alpha(I(x,y) - I_{th}))}} \tag{2}$$

where l_{th} is the intensity threshold, α is a constant number that controls the steepness of the function and (x, y) represents a coordinate on the aerial or resist image.

In general, by introducing NeuralILT and lithography simulation, we can get the printed image of the layout. By detecting the deformation of the printed image and the difference between the printed image and the layout, the information on the lithographic defects can be well integrated into our hot spot detection framework.

3.4 Query Initialization Module

The Query Initialization Module is a critical component in integrating lithography simulation priors into the object detector for hotspot detection. By dividing the input layout image and lithographic print image into non-overlapping blocks, the module computes a metric loss for each block to quantify lithographic discrepancies. Specifically, the metric loss is utilized to measure the difference between the two images, producing a loss map that highlights critical regions.

First, the pixel-wise difference for a block $B(i, j)$ is computed as:

$$D(i, j, x, y) = I_{\text{input}}(p) - I_{\text{print}}(p), \quad p \in B(i, j), \tag{3}$$

$$p = (iN + x, jN + y) \tag{4}$$

where N is the size of the block, i and j are the block indices along the height and width of the image, p represents the pixel coordinates, $I_{\text{input}}(p)$ and $I_{\text{print}}(p)$ represent the pixel intensity at position p in the input layout image and lithographic print image, respectively.

Then, the metric loss for each block is computed as:

$$L^{\text{metric}}(i, j) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} D(i, j, x, y)^2 \tag{5}$$

where $L^{\text{metric}}(i, j)$ represents the metric-based loss for the block located at (i, j).

Finally, the loss map is constructed as:

$$L_{\text{map}}^{\text{metric}} = \{L^{\text{metric}}(i, j) \mid 0 \le i < \frac{H}{N}, 0 \le j < \frac{W}{N}\} \tag{6}$$

where H and W are the height and width of the image.

The blocks with the highest losses from $L_{\text{map}}^{\text{metric}}$, typically the top k, are selected as potential hotspots, and their positional information is preserved. This positional prior is subsequently integrated into the transformer decoder, directing the model's attention to regions with pronounced lithographic discrepancies, thereby improving both the precision and efficiency of hotspot detection.

3.5 Loss Function

The total loss function used to train our proposed model consists of three components: the matching loss, the classification loss and the bounding box regression loss. Unlike traditional anchor-based methods, our proposed framework performs one-to-one matching between predicted boxes and ground truth objects, thus eliminating redundant candidate boxes. Specifically, we employ the Hungarian algorithm to find the optimal assignment that minimizes the global matching cost between predicted and ground truth objects.

Let y_i denote the ground truth set, and $\hat{y}_{\sigma(i)}$ denote the predicted set (where σ is the matching permutation). The matching loss is defined as follows[12]:

$$\hat{\sigma} = \arg\min_{\sigma \in S_n} \sum_{i=1}^{n} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) \tag{7}$$

where $\mathcal{L}_{\text{match}}$ represents the matching loss, measuring the similarity between predicted and ground truth boxes, and S_n denotes all possible matching permutations.

For the classification loss, we employ the multi-class cross-entropy loss for hotspot classification. For each predicted bounding box, we compute the probability of the hotspot and supervise it using the ground truth label. The classification loss is defined as:

$$\mathcal{L}_{cls}(y_i, \hat{y}_{\sigma(i)}) = -\sum_{c=1}^{C} p_c \log(\hat{p}_{\sigma(i),c}) \tag{8}$$

where p_c is the one-hot encoding of the ground truth class, $\hat{p}_{\sigma(i),c}$ is the predicted class probability, and C is the total number of classes.

For the bounding box regression, we employ both L1 loss and Generalized IoU (GIoU) loss to ensure tight alignment between the predicted and ground truth boxes[13]:

$$\mathcal{L}_{box}(y_i, \hat{y}_{\sigma(i)}) = \lambda_{L1}\mathcal{L}_{L1}(y_i, \hat{y}_{\sigma(i)}) + \lambda_{GIoU}\mathcal{L}_{GIoU}(y_i, \hat{y}_{\sigma(i)}) \tag{9}$$

$$\mathcal{L}_{L1}(y_i, \hat{y}_{\sigma(i)}) = \sum_{j \in \{x,y,w,h\}} |b_{i,j} - \hat{b}_{\sigma(i),j}| \tag{10}$$

$$\mathcal{L}_{GIoU}(y_i, \hat{y}_{\sigma(i)}) = 1 - \frac{|b_i \cap \hat{b}_{\sigma(i)}|}{|b_i \cup \hat{b}_{\sigma(i)}|} + \frac{|C - (b_i \cup \hat{b}_{\sigma(i)})|}{|C|} \tag{11}$$

where b_i and $\hat{b}_{\sigma(i)}$ represent the ground truth and predicted box coordinates for the respective dimension j, C is the smallest enclosing box containing both b_i and $\hat{b}_{\sigma(i)}$, ensuring that the loss penalizes poor localization even when IoU is the same.

The final total loss in our model is computed as the weighted sum of the classification and bounding box losses, given by:

$$\mathcal{L}_{total} = \sum_{i=1}^{n} \left[\mathcal{L}_{cls}(y_i, \hat{y}_{\sigma(i)}) + \lambda_{box}\mathcal{L}_{box}(y_i, \hat{y}_{\sigma(i)})\right] \tag{12}$$

where λ_{box} is a hyperparameter used to balance the contribution of $\mathcal{L}_{box}$ and $\mathcal{L}_{cls}$ in the total loss function.

4 Experimental Results

Our experimental framework is implemented using PyTorch, with all models trained on 8 NVIDIA GeForce RTX A6000 GPUs (48 GB memory) for accelerated computation. We evaluate our framework using the ICCAD2016 Benchmarks, which include four designs adjusted to comply with EUV metal layer design rules. Based on results obtained from an industrial-grade 7nm EUV lithography simulation, hotspots are accurately detected. Since the first benchmark design contains only a limited number of lithography-identified defects, our experiments focus on the remaining three designs.

Since the size of each layout is very large, we focused only on the areas with hotspots. Small fragments of the hotspots and their surrounding regions are cropped and then fed into our model as part of our dataset. Each clipped sample is of dimension 1024 × 1024, corresponding to a physical size of 1024nm × 1024nm. The ground-truth hotspot area size is 200 × 200, corresponding to a physical size of 200nm × 200nm. In this process, both the train set and the test set are randomly selected, with a ratio of 8:2. More details are shown in Table 1.

In this experiment, we did not adopt the usual evaluation metrics such as Acc used in previous works. Acc is suitable for classification tasks, but in detection tasks like ICCAD2016, where the data has an imbalanced class distribution, it does not reasonably reflect the

Table 1: Benchmark Information

Benchmark	of Source Layouts	Layout size($nm \times nm$)	of Hotspots	of Clips #Train	of Clips #Test	Clip size($nm \times nm$)	Image Size($pixels$)
ICCAD2016-2	1	6950×7000	79	40	10	1024×1024	1024×1024
ICCAD2016-3	1	12915×20144	2821	2211	553	1024×1024	1024×1024
ICCAD2016-4	1	79952×84266	162	118	29	1024×1024	1024×1024

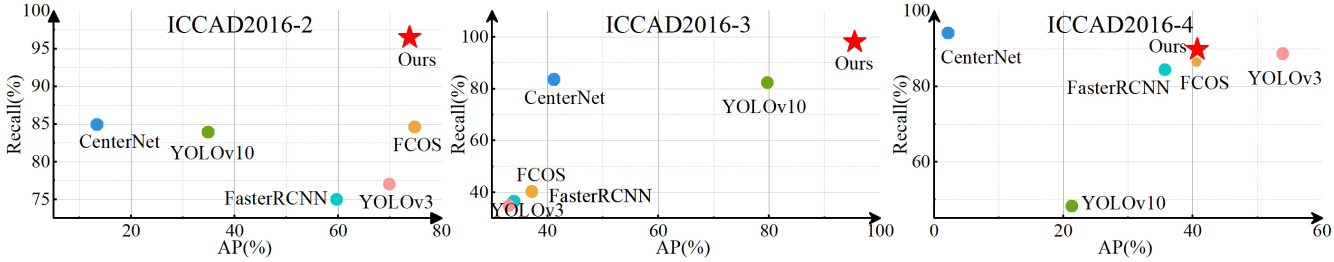

Figure 4: Comparison with Generic Object Detection methods.

Table 2: Comparison with the state-of-the-art methods.

Benchmark	R-HSD[5]		DETR[14]		Ours	
	AP(%)	Recall(%)	AP(%)	Recall(%)	AP(%)	Recall(%)
ICCAD2016-2	61.3	76.8	72.5	95.9	**73.8**	**96.5**
ICCAD2016-3	45.1	57.6	92.6	98.0	**95.4**	**98.1**
ICCAD2016-4	37.7	84.6	35.7	85.2	**40.7**	**89.8**
Avervage	48.00	73.00	66.93	93.03	**69.97**	**94.8**

performance of hotspot detection. To more accurately and comprehensively assess the model's performance, we chose AP and Recall as the primary evaluation metrics.

Figure 4 shows the comparison results between our model and generic object detection models on the ICCAD2016 dataset. Experimental results indicate that anchor-based object detection frameworks, such as Faster R-CNN[8] and YOLOv3[15], involve a substantial number of hyperparameters, which leads to poor performance. In contrast, anchor-free detectors like FCOS[16] and CenterNet[17] eliminate the need for predefined anchor configurations but still necessitate NMS for post-processing, which requires additional parameter tuning. Furthermore, YOLOv10[18] removes the NMS post-processing step in addition to eliminating the need for predefined anchor configurations. However, the lack of prior knowledge integration limits the model's overall performance on hotspot detection.

Table 2 presents the comparison results between our model and previous state-of-the-art hotspot detection models on the ICCAD2016 dataset. R-HSD shows the results of the region-based hotspot detector in [5], which was the first to propose a detector capable of detecting multiple large hotspots in each inference. The results show that our model improves by 3.04% in AP and 1.77% in recall compared to DETR, and by 21.97% in AP and 21.80% in recall compared to R-HSD, proving the advancement of our model.

Compared to both generic detection models and previous state-of-the-art hotspot detection models, both AP and Recall show significant improvements, indicating that our model not only achieves higher detection accuracy but also reduces false alarms in non-hotspot areas and minimizes missed detections in hotspot areas.

Conclusion

In this paper, we propose an anchor-free, end-to-end transformer-based hotspot detector that no longer relies on anchor boxes or region proposals, eliminating the need for the manual setting of numerous sensitive hyperparameters. This allows the model to focus directly on identifying potential hotspot regions. To better align with the hotspots that may appear in real-world processes, we integrate photolithography prior knowledge into the hotspot detector. We also introduce a query initialization module and a transformer decoder-based feature aggregation module to effectively integrate layout features and lithography prior knowledge. We hope this work can provide the possibility for advanced hotspot detectors for manufacturability research in the future.

References

[1] Haoyu Yang, Luyang Luo, Jing Su, Chenxi Lin, and Bei Yu. 2017. Imbalance aware lithography hotspot detection: a deep learning approach. *Journal of Micro/Nanolithography, MEMS, and MOEMS* 16, 3 (2017), 033504–033504.

[2] Yiyang Jiang, Fan Yang, Bei Yu, Dian Zhou, and Xuan Zeng. 2020. Efficient layout hotspot detection via binarized residual neural network ensemble. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 40, 7 (2020), 1476–1488.

[3] Hao Geng, Haoyu Yang, Lu Zhang, Jin Miao, Fan Yang, Xuan Zeng, and Bei Yu. 2020. Hotspot detection via attention-based deep layout metric learning. In *Proceedings of the 39th International Conference on Computer-Aided Design*. 1–8.

[4] Chun Wang, Yi Fang, and Sihai Zhang. 2024. Feature Fusion based Hotspot Detection with R-EfficientNet. In *Proceedings of the Great Lakes Symposium on VLSI 2024*. 446–451.

[5] Ran Chen, Wei Zhong, Haoyu Yang, Hao Geng, Xuan Zeng, and Bei Yu. 2019. Faster region-based hotspot detection. In *Proceedings of the 56th Annual Design Automation Conference 2019*. 1–6.

[6] Binwu Zhu, Ran Chen, Xinyun Zhang, Fan Yang, Xuan Zeng, Bei Yu, and Martin DF Wong. 2021. Hotspot detection via multi-task learning and transformer encoder. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 1–8.

[7] Hao-Chiang Shao, Guan-Yu Chen, Yu-Hsien Lin, Chia-Wen Lin, Shao-Yun Fang, Pin-Yian Tsai, and Yan-Hsiu Liu. 2024. LithoHoD: A Litho Simulator-Powered Framework for IC Layout Hotspot Detection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2024).

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* 39, 6 (2016), 1137–1149.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[10] Bentian Jiang, Lixin Liu, Yuzhe Ma, Hang Zhang, Bei Yu, and Evangeline FY Young. 2020. Neural-ILT: Migrating ILT to neural networks for mask printability and complexity co-optimization. In *Proceedings of the 39th International Conference on Computer-Aided Design*. 1–9.

[11] Harold Horace Hopkins. 1951. The concept of partial coherence in optics. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 208, 1093 (1951), 263–277.

[12] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. 2016. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2325–2333.

[13] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 658–666.

[14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 213–229.

[15] Joseph Redmon. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[16] Zhi Tian, Xiangxiang Chu, Xiaoming Wang, Xiaolin Wei, and Chunhua Shen. 2022. Fully convolutional one-stage 3d object detection on lidar range images. *Advances in Neural Information Processing Systems* 35 (2022), 34899–34911.

[17] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6569–6578.

[18] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. 2025. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems* 37 (2025), 107984–108011.