LITHOGDV: GENERAL VISUAL UNDERSTANDING MEETS DOMAIN-SPECIFIC VISUAL PRIORS FOR ROBUST LITHOGRAPHY HOTSPOT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Lithography hotspot detection (LHD) identifies local integrated circuit layout patterns susceptible to manufacturing defects. As technology nodes shrink, denser layouts and narrower process windows make detection increasingly sensitive to subtle geometric variations. To address this challenge, mainstream LHD methods rasterize polygonal layout clips into images and apply vision models for hotspot detection. These detectors, however, are trained on limited domain data and rely on learned parametric decision boundaries without explicitly reusing validated layout cases. Consequently, they may generalize poorly to unseen designs and miss hotspot patterns that differ only subtly from non-hotspots. We propose LithoGDV, built on the insight that robust LHD requires complementary knowledge: general visual understanding of layout geometry and domain-specific visual priors from comparable cases. LithoGDV fine-tunes a pretrained large vision model to learn fine-grained layout representations and constructs a knowledge base of process-verified cases. Retrieved hotspot and non-hotspot cases provide explicit evidence that complements the model's parametric prediction. On OpenHSD-Bench, LithoGDV achieves state-of-the-art detection accuracy against the leading open-source and non-open-source baselines across full-data, low-resource, and cross-design settings. We will publicly release the complete LithoGDV codebase, including all training and evaluation configurations.

1 INTRODUCTION

Integrated circuit (IC) manufacturing relies on lithography to reproduce the designed circuit layout on silicon wafers. At advanced technology nodes, reduced feature dimensions and spacing increase optical interactions among neighboring structures, making lithographic imaging more sensitive to local geometry and process variation (Zhang et al., 2026; Chen et al., 2024; 2025). Consequently, some local layout patterns are prone to manufacturing defects under given lithographic process conditions. These patterns are known as lithography hotspots (Yang et al., 2019; Geng et al., 2022). As feature dimensions decrease, hotspot discrimination becomes increasingly challenging because hotspot and non-hotspot patterns may differ only in fine geometric details. Lithography hotspot detection (LHD) identifies these patterns during design-for-manufacturability (DFM) verification. The identified hotspot patterns can then be corrected to reduce the risk of circuit failures and yield loss (Sim et al., 2019; Pan et al., 2022; Chen et al., 2025). Figure 1 shows representative lithography hotspots and this detection-and-correction flow.

Existing LHD approaches include lithography simulation (Mack, 2005; Mitra et al., 2005; Kim & Fan, 2003), pattern matching (Wen et al., 2014; Yao et al., 2006; Yu et al., 2012), and learning-based detection (Zhang et al., 2026; Chen et al., 2024; Geng et al., 2022). Lithography simulation offers high fidelity but incurs substantial computational cost, whereas pattern matching efficiently identifies known hotspot templates but has limited coverage of unseen patterns. Learning-based detectors have therefore become mainstream, with individual methods distinguished primarily by their representations of layout geometry. Early deep detectors transform local layouts into spatially ordered feature tensors and apply CNNs with biased-learning objectives (Yang et al., 2019). Subsequent image-based methods learn discriminative representations directly from rasterized layout clips using attention mechanisms and metric-learning objectives (Geng et al., 2022). Graph-based methods ex-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

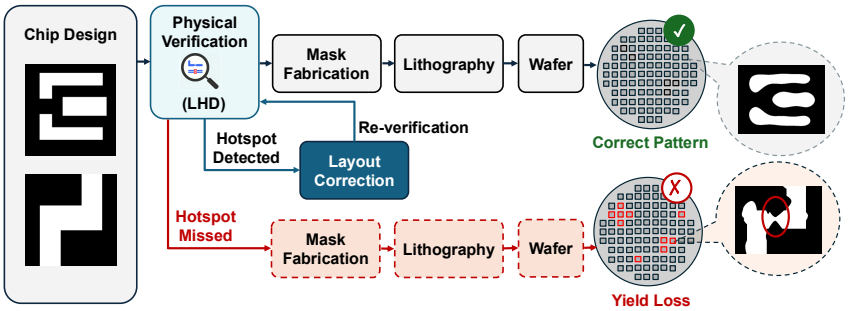


Figure 1: Lithography hotspots and the LHD step.

plicitly model polygons and their geometric relations through structured representations (Sun et al., 2022). GDSII-based methods instead serialize polygon geometry and spatial relations into semantic sequences processed by Transformers (Chen et al., 2024). Recent work further aligns layout images with GDSII semantics through multimodal pretraining and retains the pretrained image encoder for downstream LHD (Zhang et al., 2026). Across these developments in layout representation, image-based detection remains a primary approach to robust LHD, as rasterized layout clips effectively capture polygon geometry and spatial context in a unified visual form.

Current learning-based LHD methods rely mainly on lithography-specific training data and parametric decision boundaries (Chen et al., 2020; Yang et al., 2021; Xiao et al., 2021), leaving transferable visual knowledge and reusable case evidence underused. First, distinguishing hotspots from non-hotspots often hinges on subtle local differences in line ends, corners, spacing, and neighboring polygon configurations (Yang et al., 2019; Geng et al., 2022; Sun et al., 2022). Learning these cues solely from limited labeled layouts restricts the geometric variation available for representation learning. Large-scale visual pretraining offers a broader starting point by exposing the encoder to diverse shapes, contours, and spatial relations before lithography-specific adaptation. However, most image-based detectors still learn their visual representations primarily from task-specific layout data (Yang et al., 2019; Geng et al., 2022; Zhang et al., 2026). Second, process-verified layout cases contain reusable domain evidence. Under fixed process conditions, layouts with similar local geometry and known outcomes provide direct references for assessing a new query. Existing parametric detectors encode such experience only implicitly in model weights and do not retrieve comparable cases at inference time (Yang et al., 2019; Geng et al., 2022). The absence of explicit case evidence may limit robustness to unseen designs, particularly when hotspot and non-hotspot patterns differ only subtly (Jiang et al., 2025b; Zhang et al., 2026). These gaps motivate combining pretrained general visual knowledge with domain-specific priors from comparable layout cases.

In this paper, we propose LithoGDV, an image-based LHD framework that integrates general visual understanding with domain-specific visual priors. LithoGDV fine-tunes a pretrained large vision model (LVM) on rasterized layout clips to obtain fine-grained layout representations. The resulting representations are used to organize validated hotspot and non-hotspot cases into a non-parametric knowledge base. For each input layout clip, LithoGDV retrieves comparable cases and combines their explicit evidence with the fine-tuned LVM's parametric knowledge. Each prediction therefore draws on both transferable visual knowledge and relevant domain experience. On real-world chip layout datasets, LithoGDV achieves state-of-the-art detection accuracy using visual layout information alone, whereas several leading baselines additionally exploit GDSII-derived textual representations. The use of such textual representations is orthogonal to the proposed visual framework. Integrating them may yield further gains and remains a direction for future work.

Our main contributions are summarized as follows:

- We propose LithoGDV, a hybrid framework for image-based LHD that combines parametric knowledge learned by a domain-adapted LVM with non-parametric evidence retrieved from an external layout case base. The LVM integrates general visual knowledge from pretraining with lithography-specific knowledge acquired through fine-tuning, while retrieved cases provide additional domain-specific visual priors for each prediction.
- We design LithoGDV's case retrieval and evidence fusion components using fine-grained layout representations produced by the adapted LVM. We organize validated cases into a

layout case knowledge base and construct a multi-vector graph index over the case representations for efficient retrieval of comparable cases. We further introduce a three-branch classifier that fuses semantic interactions, neighborhood geometry, and retrieval statistics derived from the input layout clip and retrieved cases.

- We validate LithoGDV across three representative open-source LVM backbones on the challenging OpenHSDBench suite of real-world chip layout datasets, covering full-data, low-resource, and cross-design settings. Using visual layout information alone, LithoGDV consistently achieves state-of-the-art detection accuracy against leading open-source and non-open-source baselines. We will publicly release the complete LithoGDV codebase together with all training and evaluation configurations.

2 Related Work

Image-Based Methods. Most learning-based LHD methods rasterize layout clips for vision-based analysis. The resulting images preserve polygon shapes, spacing, and local context, providing direct geometric cues for hotspot detection. A fully convolutional detector predicts multiple hotspot locations in variable-size layout images (Gai et al., 2022), while eADM-HSD uses attention and metric learning to separate hotspot and non-hotspot representations (Geng et al., 2022). Later methods improve image encoding through neural architecture search (Chen et al., 2023), pose-sensitive capsules in PSCaps (Wang et al., 2025a), and multi-scale refinement in MSFRN (Luo et al., 2026). LithoHoD augments layout features with estimated shape deformation (Shao et al., 2025), whereas LithoExp combines the layout image with a predicted defect map for classification (Jiang et al., 2025a). Recent work further incorporates pretraining into image-based LHD through cross-task masked layout modeling (Zhu et al., 2025), multimodal alignment between layout images and GDSII semantics (Zhang et al., 2026), and hybrid spatial–spectral modeling in HyDAS (Chen et al., 2026). However, these methods rely largely on limited lithography-specific data. General visual knowledge from large-scale pretraining may better represent shapes, contours, and spatial relations, yet its potential to capture subtle layout variations remains largely unexplored in LHD.

Other Learning-Based Methods. Learning-based LHD has also been explored using graph, sequence, and multimodal representations of layout geometry. GNN-HSD represents layout rectangles as nodes and their geometric relations as edge features (Sun et al., 2022). Subsequent graph-based methods construct heterogeneous graphs from modified transitive closure graphs and aggregate three-hop neighborhoods (Yan et al., 2025), or encode polygon coordinates and inter-polygon distances (Wang et al., 2025b). LLM-HD serializes layout contours and hierarchy into lossless GDSII semantic sequences processed by a pretrained Transformer (Chen et al., 2024). CAGE-HSD jointly encodes rasterized layouts and GDSII sequences, aligning global image–text representations through contrastive learning and reconstructing polygon coordinates for fine-grained generative alignment (Zhang et al., 2026). These representations provide valuable complementary information by explicitly encoding geometric relations and layout semantics. Nevertheless, the image modality remains central to multimodal methods and continues to serve as a primary source of information in mainstream learning-based LHD. However, existing learning-based LHD methods rarely make explicit use of domain-specific visual priors from validated layout cases during inference.

3 The LithoGDV Framework

3.1 Framework Overview

Given a training set $\mathcal{D}_{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is a rasterized layout clip and $y_i \in \{0, 1\}$ is its binary label (0 for non-hotspot and 1 for hotspot), LithoGDV predicts the hotspot probability of a query layout q. Figure 2 shows the framework's three modules. First, a pretrained large vision model (LVM) is adapted to lithography layouts to produce fine-grained multi-vector representations. Second, validated cases form a layout case knowledge base organized by scalable multi-vector graph indices. Third, LithoGDV retrieves hotspot, non-hotspot, and mixed case neighborhoods for each query and extracts semantic interactions, neighborhood geometry, and retrieval statistics for evidence fusion. The adapted LVM provides parametric visual knowledge, while the external case base supplies query-specific domain evidence that can be updated independently of the encoder.

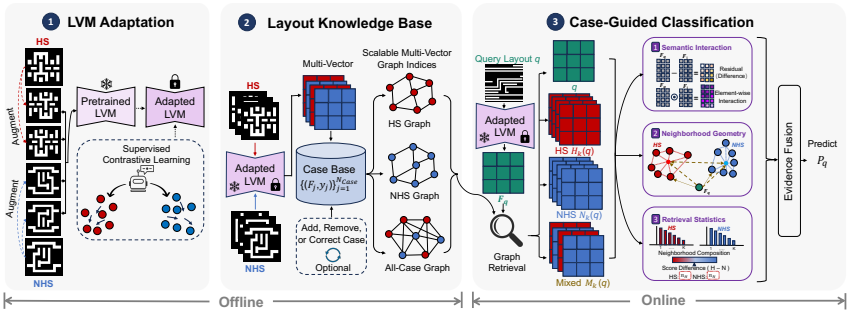


Figure 2: Overview of the LithoGDV framework.

3.2 LVM Adaptation

Patch Encoding. Lithography risk depends on subtle variations in line ends, corners, spacing, and neighboring polygon configurations. Compressing an entire layout into a single global vector may obscure these localized differences (Dosovitskiy et al., 2021). We denote the adapted LVM vision encoder by Φ_θ. For an input layout x, its local visual tokens form the multi-vector representation

$$F_x = \Phi_\theta(x) = \left[f_1^{(x)}, \ldots, f_m^{(x)}\right]^\mathsf{T} \in \mathbb{R}^{m \times d}, \tag{1}$$

where m is the fixed number of visual tokens and $f_j^{(x)} \in \mathbb{R}^d$ represents the j-th local image region. Collectively, these tokens preserve localized geometric evidence while incorporating the surrounding spatial context encoded by the LVM.

Low-Rank Adaptation. Full-parameter fine-tuning on limited lithography data is computationally expensive and may unnecessarily distort the transferable representations from pretraining. We therefore freeze the pretrained weights and introduce low-rank adaptation (LoRA) into selected projection layers (Hu et al., 2022). For a frozen projection matrix W, LoRA parameterizes its update as

$$W' = W + \Delta W, \quad \Delta W = \frac{\lambda}{r} BA. \tag{2}$$

Here, $W, W' \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, with d_{in} and d_{out} denoting the projection's input and output dimensions. Only $A \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}} \times r}$ are trained, where $r \ll \min(d_{\text{in}}, d_{\text{out}})$ is the adaptation rank and λ scales the update. This low-rank parameterization constrains task-specific changes while enabling the encoder to capture lithography-specific geometric variations.

Contrastive Learning. The multi-vector representation supports both case retrieval and downstream classification. We therefore adapt it to separate hotspot from non-hotspot layouts while preserving local correspondences for case comparison. We optimize the LoRA parameters with supervised contrastive learning (Khosla et al., 2020). Each training layout yields two augmented views generated by label-preserving transformations, such as reflection or $180°$ rotation.

We compare two layout representations using directional multi-vector late interaction (Khattab & Zaharia, 2020):

$$S_{\text{sum}}(F_i \to F_j) = \sum_{u=1}^{m} \max_{1 \le v \le m} \left(\widetilde{f}_u^{(i)}\right)^\mathsf{T} \widetilde{f}_v^{(j)}, \tag{3}$$

where $\widetilde{f}_u^{(i)} = f_u^{(i)} / \|f_u^{(i)}\|_2$ denotes the independently ℓ_2-normalized form of $f_u^{(i)}$. In this directional comparison, F_i serves as the anchor representation, whereas F_j serves as the candidate representation. For each anchor vector, the score takes its highest similarity to any vector in F_j and sums these maxima. This preserves local evidence without reducing each layout to one global vector.

Let $s_{ij} = S_{\text{sum}}(F_i \to F_j)$, and let $\mathcal{I}$ denote all augmented views in a mini-batch. For anchor i, the positive set

$$\mathcal{P}(i) = \{p \in \mathcal{I} \setminus \{i\} \mid y_p = y_i\} \tag{4}$$

contains all other views sharing its lithography-risk label. The contrastive objective is

$$\mathcal{L}_{\text{con}} = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \log \frac{\exp(s_{ip}/\tau)}{\sum_{a \in \mathcal{I} \setminus \{i\}} \exp(s_{ia}/\tau)}, \tag{5}$$

where τ is the temperature parameter. We use class-balanced mini-batches to reduce majority-class bias. Label-preserving augmentation promotes view consistency, while the contrastive objective strengthens local alignment for same-class pairs relative to cross-class pairs.

3.3 Layout Case Knowledge Base

Case Representation. After LVM adaptation, we freeze Φ_θ and index each non-hotspot's original view and each hotspot's original plus three label-preserving reflections or rotations:

$$N_{\text{case}} = N_{\text{nhs}} + 4N_{\text{hs}}, \quad \mathcal{K}_{\text{tr}} = \{(F_j, y_j)\}_{j=1}^{N_{\text{case}}}. \tag{6}$$

Each view x_j is encoded once as $F_j = \Phi_\theta(x_j) \in \mathbb{R}^{m\times d}$; y_j inherits the validated label of its source layout. Cases can be added, removed, or relabeled without updating the adapted LVM.

Scalable Case Index. Exhaustive retrieval compares a query with all N_{case} multi-vector cases and becomes increasingly expensive as the case base grows. We therefore organize cases into a bounded-degree proximity graph, where each node represents one layout case. Cases are compared using the same directional multi-vector similarity S defined in Eq. equation 3. This similarity is generally asymmetric because each local vector in the first representation independently selects its strongest correspondence in the second. We therefore use the following symmetric score when constructing graph edges:

$$S_{\text{sym}}(F_i, F_j) = \frac{1}{2}\left[s_{ij} + s_{ji}\right]. \tag{7}$$

The index is built incrementally. For each new case, graph traversal finds a bounded candidate set using S_{sym}. The node links to the most relevant candidates, while edge redirection bounds in- and out-degrees and preserves reachability from multiple entry nodes. Bounded per-node edges yield storage linear in the case count. Applying this procedure separately to hotspot, non-hotspot, and all cases produces $\mathcal{G}\text{hs}$, $\mathcal{G}\text{nhs}$, and $\mathcal{G}_{\text{all}}$, respectively.

3.4 Case-Guided Classification

Case Retrieval. Given a query representation F_q, graph search begins from multiple entry nodes and evaluates visited cases using the directional multi-vector similarity $S(F_q, F_i)$. Traversal expands the highest-scoring unexpanded cases, while a size-L list limits the active search space. Let $\mathcal{V}_L(q;\mathcal{G})$ denote the cases visited for query q in graph $\mathcal{G}$. The retrieved neighborhood is defined as follows

$$\mathcal{R}_K(q;\mathcal{G}) = \underset{i\in\mathcal{V}_L(q;\mathcal{G})}{\text{Top-K}}\ S(F_q, F_i). \tag{8}$$

Each visited case is scored exactly by S; approximation comes only from searching a graph subset, reducing multi-vector comparisons from N_{case} to $|\mathcal{V}_L(q;\mathcal{G})|$. The visited set directly provides the Top-K cases. Searching the hotspot, non-hotspot, and all-case indices yields

$$\begin{gathered}\mathcal{H}_K(q) = \mathcal{R}_K(q;\mathcal{G}_{\text{hs}}), \quad \mathcal{N}_K(q) = \mathcal{R}_K(q;\mathcal{G}_{\text{nhs}}),\\ \mathcal{M}_K(q) = \mathcal{R}_K(q;\mathcal{G}_{\text{all}}).\end{gathered} \tag{9}$$

The hotspot and non-hotspot neighborhoods provide class-specific reference evidence, while the mixed neighborhood reflects the local class composition of retrieved cases.

Semantic Interaction. The semantic branch compares the query representation with class-specific aggregates of the retrieved cases. For each retrieved case i, we obtain a case-level descriptor $e_i = \text{MeanPool}(F_i) \in \mathbb{R}^d$. For a retrieved case set $\mathcal{C}$, its aggregate representation is defined as

$$\text{Agg}(\mathcal{C}) = \left[\underset{i\in\mathcal{C}}{\text{Mean}}\, e_i; \underset{i\in\mathcal{C}}{\text{Max}}\, e_i\right] \in \mathbb{R}^{2d}, \tag{10}$$

where Max denotes the elementwise maximum and $[\cdot\,;\cdot]$ denotes vector concatenation. The hotspot, non-hotspot, and query representations are then summarized as

$$\bar{h} = \text{Agg}(\mathcal{H}_K(q)), \quad \bar{n} = \text{Agg}(\mathcal{N}_K(q)), \quad \bar{q} = [\text{MeanPool}(F_q); \text{MaxPool}(F_q)]. \tag{11}$$

The resulting semantic interaction feature is

$$v_{\text{sem}} = \left[\bar{q}; \bar{q} - \bar{h}; \bar{q} - \bar{n}; \bar{q} \odot \bar{h}; \bar{q} \odot \bar{n}\right] \in \mathbb{R}^{10d}. \tag{12}$$

Difference terms measure query deviation from each class neighborhood, while elementwise products capture feature-wise agreement. The MLP M_{sem} maps v_{sem} to semantic evidence z_{sem}.

Neighborhood Geometry. Semantic aggregates do not encode retrieved-case dispersion or query proximity to each class neighborhood. Let $e_q = \text{MeanPool}(F_q)$ and define the cosine distance between two descriptors as $\delta(a, b) = 1 - \cos(a, b)$. For $K \geq 2$, denote the descriptors of the retrieved hotspot and non-hotspot cases by $\{e_{h_i}\}_{i=1}^{K}$ and $\{e_{n_i}\}_{i=1}^{K}$, respectively. We compute

$$\begin{aligned} d_{hh} &= \frac{2}{K(K-1)} \sum_{1 \leq i < j \leq K} \delta(e_{h_i}, e_{h_j}), \quad d_{nn} = \frac{2}{K(K-1)} \sum_{1 \leq i < j \leq K} \delta(e_{n_i}, e_{n_j}), \\ d_{qh} &= \frac{1}{K} \sum_{i=1}^{K} \delta(e_q, e_{h_i}), \qquad\qquad d_{qn} = \frac{1}{K} \sum_{i=1}^{K} \delta(e_q, e_{n_i}). \end{aligned} \tag{13}$$

The neighborhood-geometry feature is defined as

$$v_{\text{geo}} = \left[d_{hh}, d_{nn}, d_{qh}, d_{qn}, \log \frac{d_{hh} + \epsilon}{d_{nn} + \epsilon} \right]^{\mathsf{T}} \in \mathbb{R}^5, \tag{14}$$

where $\epsilon > 0$ ensures numerical stability. d_{hh} and d_{nn} quantify within-class dispersion; d_{qh} and d_{qn} are the query's mean distances to the class neighborhoods. The log-ratio contrasts their relative dispersion. The MLP M_{geo} maps v_{geo} to geometric evidence z_{geo}.

Retrieval Statistics. The statistical branch models class-specific similarity profiles and mixed-neighborhood composition. Let $s_1^h \geq \cdots \geq s_K^h$ and $s_1^n \geq \cdots \geq s_K^n$ be the ordered directional similarities for hotspot and non-hotspot neighborhoods, respectively. Define $\mathbf{s}_h = [s_1^h, \ldots, s_K^h]^{\mathsf{T}}$ and $\mathbf{s}n = [s_1^n, \ldots, s_K^n]^{\mathsf{T}}$, with means μ_h, μ_n and standard deviations σ_h, σ_n. For the mixed neighborhood $\mathcal{M}K(q)$, let $c\text{hs}$ count hotspot cases and $c\text{src}$ their distinct source layouts. The retrieval-statistics feature is

$$v_{\text{stat}} = \left[\mathbf{s}_h; \mathbf{s}_n; s_1^h - s_1^n; s_1^h - s_K^n; \mu_h - \mu_n; \sigma_h; \sigma_n; c_{\text{hs}}; c_{\text{src}} \right] \in \mathbb{R}^{2K+7}. \tag{15}$$

The ordered scores preserve the similarity profile of each class neighborhood, while the difference terms compare their relative support for the query. The score deviations characterize neighborhood consistency, and the mixed-neighborhood counts capture the prevalence and source diversity of nearby hotspot cases. An MLP M_{stat} maps v_{stat} to the statistical evidence vector z_{stat}.

Evidence Fusion. The branch MLPs map the semantic, geometric, and statistical features to evidence vectors $z_{\text{sem}}, z_{\text{geo}}, z_{\text{stat}} \in \mathbb{R}^{d_z}$. These complementary vectors are fused through elementwise addition, layer normalization, and nonlinear activation:

$$z_q = \text{GELU}(\text{LayerNorm}(z_{\text{sem}} + z_{\text{geo}} + z_{\text{stat}})), \quad \mathbf{p}_q = \text{Softmax}(W_o z_q + b_o). \tag{16}$$

Here, $W_o \in \mathbb{R}^{2 \times d_z}$ and $b_o \in \mathbb{R}^2$ are the output-layer parameters, and $\mathbf{p}_q = [p_{q,0}, p_{q,1}]^{\mathsf{T}}$ contains the predicted non-hotspot and hotspot probabilities. Classifier training freezes the adapted LVM and graph indices, optimizing only the three branch MLPs and output layer. Class imbalance is addressed with a label-smoothed, class-weighted focal objective. (Lin et al., 2017):

$$\widetilde{y}_c = (1 - \varepsilon_{\text{ls}}) \mathbb{I}[c = y] + \frac{\varepsilon_{\text{ls}}}{2}, \quad \mathcal{L}_{\text{cls}} = -\alpha_y (1 - p_{q,y})^{\gamma} \sum_{c \in \{0,1\}} \widetilde{y}_c \log p_{q,c}. \tag{17}$$

Here, α_y is the class weight, with a larger value assigned to hotspot samples; γ downweights well-classified samples; and ε_{ls} controls label smoothing. At inference, LithoGDV encodes the query, retrieves hotspot, non-hotspot, and mixed cases from their graph indices, extracts semantic, geometric, and statistical evidence, and predicts hotspot probability $p_{q,1}$.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate LithoGDV on OpenHSDBench, a standardized benchmark suite (Zhang et al., 2026). The suite covers two benchmark families: ICCAD2012 (Torres, 2012) and the Via benchmarks introduced in the ICCAD-2020 study (Geng et al., 2020). The latter include Via-1–Via-4 and

Table 1: OpenHSDBench statistics (HS: hotspot layouts; NHS: non-hotspot layouts).

Dataset	Train HS	Train NHS	Test HS	Test NHS	Clip size (μm^2)
ICCAD2012	712	16,151	2,524	13,179	4.8×4.8
Via-1	3,429	10,279	2,267	6,877	2.0×2.0
Via-2	1,023	11,231	723	7,465	2.0×2.0
Via-3	586	18,418	422	12,483	2.0×2.0
Via-4	33	13,797	25	12,445	2.0×2.0
Via-Merge	5,071	53,725	3,437	39,270	2.0×2.0

Table 2: Full-data results on OpenHSDBench. Bold marks row-wise best values.

	Non-open-source Methods						**Open-source Methods**						
	eADM-HSD		LLM-HD		CAGE-HSD		BL		BBL		CLIP-ConvNeXt		**LithoGDV**
Dataset	$\mathrm{Acc_{HS}}$ (%)	FA (FAR %)	$\mathrm{Acc_{HS}}$ (%)	FA (FAR %)	$\mathrm{Acc_{HS}}$ (%)	FA (FAR %)	$\mathrm{Acc_{HS}}$ (%)	FA (FAR %)	$\mathrm{Acc_{HS}}$ (%)	FA (FAR %)	$\mathrm{Acc_{HS}}$ (%)	FA (FAR %)	$\mathrm{Acc_{HS}}$ (%)	FA (FAR %)
ICCAD2012	97.39	112 (0.85)	97.90	9 (0.07)	99.41	**4 (0.03)**	84.27	93 (0.71)	82.92	116 (0.88)	98.22	10 (0.08)	**99.68**	59 (0.45)
Via-1	92.41	2,174 (31.61)	94.13	2,721 (39.57)	95.63	**1,972 (28.68)**	79.80	2,742 (39.87)	73.09	4,521 (65.74)	94.35	2,100 (30.54)	**97.44**	3,546 (51.56)
Via-2	88.11	**2,347 (31.44)**	88.65	3,298 (44.18)	91.56	2,404 (32.20)	84.09	5,367 (71.90)	85.33	5,764 (77.21)	88.93	2,626 (35.18)	**93.36**	2,557 (34.25)
Via-3	82.46	2,378 (19.05)	84.60	4,202 (33.66)	84.60	2,108 (16.89)	77.01	8,176 (65.50)	72.98	6,373 (51.05)	82.70	**1,958 (15.69)**	**87.68**	1,985 (15.90)
Via-4	76.00	396 (3.18)	76.00	657 (5.28)	**84.00**	418 (3.36)	68.00	**143 (1.15)**	68.00	245 (1.97)	76.00	345 (2.77)	**84.00**	2,051 (16.48)
Via-Merge	85.57	3,737 (9.52)	87.19	4,290 (10.92)	90.98	**3,460 (8.81)**	70.85	5,467 (13.92)	82.75	8,377 (21.33)	88.45	3,578 (9.11)	**93.80**	5,944 (15.14)
Average	86.99	1,857 (15.94)	88.08	2,530 (22.28)	91.03	**1,728 (14.99)**	77.33	3,664 (32.17)	77.51	4,233 (36.36)	88.10	1,770 (15.56)	**92.66**	2,690 (22.30)

their merged set, Via-Merge. Table 1 summarizes all six. We use OpenHSDBench's fixed splits and build each case base solely from its training split.

Baselines. We compare LithoGDV with six learning-based LHD baselines spanning four methodological categories: frequency-domain learning with Biased Learning (BL) and Batch Biased Learning (BBL) (Yang et al., 2017; 2019); attention-guided metric learning with eADM-HSD (Geng et al., 2020); hierarchical GDSII semantic modeling with LLM-HD (Chen et al., 2024); and multimodal modeling with CLIP-ConvNeXt and CAGE-HSD (Cherti et al., 2023; Zhang et al., 2026). BL, BBL, and CLIP-ConvNeXt are open-source; the remaining baselines are non-open-source.

Metrics. Detection metrics are hotspot accuracy $\mathrm{Acc_{HS}}$, false-alarm count(FA), and rate(FAR):

$$\mathrm{Acc_{HS}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{18}$$

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{19}$$

Here, $\mathrm{FA} = \mathrm{FP}$, and TP, FN, FP, and TN denote true positives, false negatives, false positives, and true negatives, respectively. Higher $\mathrm{Acc_{HS}}$ indicates fewer missed hotspots, while lower FA and FAR indicate fewer false alarms.

To assess representation and retrieval quality, we report Hotspot Hit Rate@K and Non-Hotspot False Hit Rate@K. Let $\mathcal{Q}_{\mathrm{hs}}$ and $\mathcal{Q}_{\mathrm{nhs}}$ denote the hotspot and non-hotspot query sets, respectively, and let $\mathcal{R}_K(q)$ denote the top-K cases retrieved for query q. The metrics are defined as

$$\begin{aligned} \mathrm{HR@K} &= \frac{1}{|\mathcal{Q}_{\mathrm{hs}}|} \sum_{q \in \mathcal{Q}_{\mathrm{hs}}} \mathbb{I}[\exists i \in \mathcal{R}_K(q) : y_i = 1], \\ \mathrm{FHR@K} &= \frac{1}{|\mathcal{Q}_{\mathrm{nhs}}|} \sum_{q \in \mathcal{Q}_{\mathrm{nhs}}} \mathbb{I}[\exists i \in \mathcal{R}_K(q) : y_i = 1]. \end{aligned} \tag{20}$$

Here, $y_i = 1$ denotes a hotspot case. Higher $\mathrm{HR@5}$ and lower $\mathrm{FHR@5}$ indicate better retrieval.

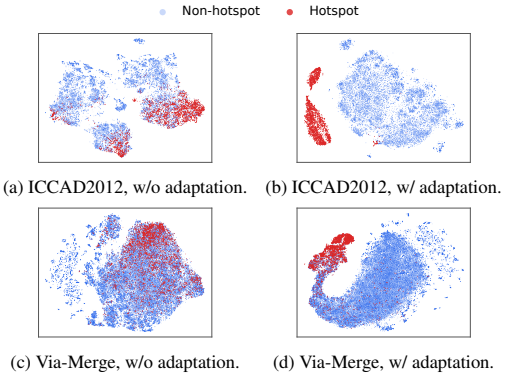
Settings. All runs use one NVIDIA RTX A6000 and default to the `Tomoro-ColQwen3` LVM (Huang & Tan, 2025). LoRA uses rank $r = 4$, scaling $\lambda = 8$, and dropout 0.05. LVM adaptation uses AdamW (learning rate 2×10^{-5}, weight decay 0.01, temperature $\tau = 0.07$, and maximum gradient norm 1.0). The classifier uses AdamW (learning rate 2×10^{-4}, weight decay 10^{-4}, batch size 128, dropout 0.3, $\gamma = 2.5$, and label smoothing 0.05). The hotspot weight is selected from $20, 30, 40, 50, 100, 500, 1000, 2000, 4000$. Retrieval uses $K = 10$, graph degree $R = 32$, construction candidate size $C = 512$, and search-list size $L = 10$. Results are averaged over three independent runs.

Table 3: Low-resource results using 5% data.

Method	$\mathrm{Acc_{HS}}$(%)↑	FA (FAR%)↓
Non-open-source		
eADM-HSD	83.65	5,800 (14.77)
LLM-HD	79.19	6,570 (16.73)
CAGE-HSD	86.38	5,300 (13.50)
Open-source		
BL	62.87	7,466 (19.01)
BBL	64.99	12,433 (31.66)
CLIP-ConvNeXt	85.80	5,700 (14.51)
LithoGDV	**88.07**	**5,135 (13.08)**

Table 4: Cross-design: Via-1 → Via-2/3/4

Method	$\mathrm{Acc_{HS}}$(%)↑	FA (FAR%)↓
Non-open-source		
eADM-HSD	60.35	12,933 (39.92)
LLM-HD	70.02	6,892 (21.27)
CAGE-HSD	73.72	3,534 (10.91)
Open-source		
BL	36.84	6,838 (21.11)
BBL	31.00	5,459 (16.85)
CLIP-ConvNeXt	68.21	7,308 (22.56)
LithoGDV	**82.65**	**2,961 (9.14)**



(a) ICCAD2012, w/o adaptation. (b) ICCAD2012, w/ adaptation.

(c) Via-Merge, w/o adaptation. (d) Via-Merge, w/ adaptation.

Figure 3: t-SNE of layout representations.

Table 5: Effect of LVM adaptation on ICCAD2012.

Configuration	HR@5 (%) ↑	FHR@5 (%) ↓
w/o LVM adaptation	93.19	17.56
w/ LVM adaptation	**98.02**	**0.05**

Table 6: Case-base and evidence-branch effects.

Configuration	$\mathrm{Acc_{HS}}$ (%) ↑	FA (FAR %) ↓
w/o Case Knowledge Base	98.02	6 (0.05)
w/o Semantic Branch	99.60	57 (0.43)
w/o Geometric Branch	99.64	62 (0.47)
w/o Statistical Branch	97.86	6 (0.05)
Full LithoGDV	**99.68**	59 (0.45)

4.2 Experimental Results

Full-Data Performance. Table 2 compares all methods using the complete training splits. LithoGDV achieves the highest $\mathrm{Acc_{HS}}$ across all six datasets. Its average $\mathrm{Acc_{HS}}$ reaches 92.66%, surpassing the best-performing non-open-source and open-source baselines by 1.63 and 4.56 percentage points, respectively. In LHD, $\mathrm{Acc_{HS}}$ is particularly important because undetected hotspots may escape correction and cause manufacturing defects, whereas false alarms primarily increase verification overhead. Notably, LithoGDV achieves these gains using only rasterized layout images, whereas the strongest baseline, CAGE-HSD, requires images and text. LithoGDV can readily incorporate text, potentially improving detection further.

Low-Resource Performance. Table 3 compares all methods on Via-Merge using only 5% of the training split. LithoGDV achieves the highest $\mathrm{Acc_{HS}}$ of 88.07% and the fewest false-alarms, with 5,135 FA and 13.08% FAR. Against the strongest baseline, CAGE-HSD, it gains 1.69 pp in $\mathrm{Acc_{HS}}$ while reducing FA by 165 and FAR by 0.42 pp. These concurrent improvements show that LithoGDV remains effective with limited lithography-specific data, supporting the use of transferable visual representations and retrieved case evidence in low-resource LHD.

Cross-Design Generalization. Table 4 reports cross-design results from Via-1 to Via-2/3/4. LithoGDV achieves 82.65% $\mathrm{Acc_{HS}}$, exceeding the strongest non-open-source and open-source baselines by 8.93 and 14.44 percentage points, respectively. Relative to CAGE-HSD, it also reduces FA by 16.2% and FAR by 1.77 percentage points. These gains demonstrate LithoGDV's strong generalization to unseen layout designs.

Ablation Studies. Tables 5 and 6 summarize the ablations on ICCAD2012. LVM adaptation increases HR@5 from 93.19% to 98.02% and reduces FHR@5 from 17.56% to 0.05%, demonstrating its importance for effective case retrieval. Removing the case knowledge base decreases $\mathrm{Acc_{HS}}$ from 99.68% to 98.02%, corresponding to approximately 42 additional missed hotspots. Among the evidence branches, removing the statistical branch causes the largest accuracy drop of 1.82 percentage points, while removing the semantic and geometric branches reduces accuracy by 0.08 and 0.04 points, respectively. Excluding geometric evidence also increases FA from 59 to 62. Overall, LVM adaptation establishes discriminative layout representations, the case base provides additional layout evidence, and the three branches contribute complementary signals to classification.

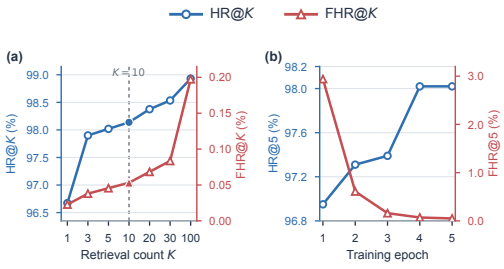


Figure 4: Effects of (a) retrieval count and (b) training epochs.

Table 7: Effect of different LVMs on ICCAD2012.

LVM	HR@5 (%) ↑	FHR@5 (%) ↓
`ColNomic`	96.47	0.24
`Tomoro-ColQwen3` (default)	98.02	**0.05**
`Nemotron-ColEmbed`	**98.18**	**0.05**

Table 8: ICCAD2012 retrieval on $\mathcal{G}_{\text{all}}$.

Method	Time/query (ms) ↓	Visited/query ↓	HR@10 (%) ↑	FHR@10 (%) ↓
w/o Graph Index	2,142.23	18,999	98.14	0.053
w/ Graph Index	113.84	479	98.18	0.053

Effect of Different LVMs. Table 7 compares three LVMs under identical settings. Replacing `Tomoro-ColQwen3` with the stronger `Nemotron-ColEmbed` (Moreira et al., 2026) raises HR@5 from 98.02% to 98.18% while preserving the lowest FHR@5 of 0.05%. The weaker `ColNomic` (Nomic Team, 2025) yields 96.47% HR@5 and 0.24% FHR@5. This confirms LithoGDV is LVM-agnostic and benefits from stronger LVMs.

Representation Space. Figure 3 visualizes the layout representations before and after LVM adaptation using t-SNE (van der Maaten & Hinton, 2008). Before adaptation, hotspot and non-hotspot samples overlap substantially on both ICCAD2012 and Via-Merge. After adaptation, hotspot samples form more compact regions with markedly less overlap with non-hotspots. The consistent change across both datasets indicates that LVM adaptation produces more class-discriminative representations for subsequent case retrieval.

Retrieval Performance. Table 8 evaluates retrieval with and without the graph index on ICCAD2012. Both settings use the same 16-thread CPU environment. The graph index visits only 479 of 18,999 cases per query and reduces the mean query time from 2,142.23 to 113.84 ms, yielding an 18.8× speedup. Meanwhile, HR@10 differs by only 0.04 percentage points, and FHR@10 remains unchanged at 0.053%. The graph index therefore substantially reduces multi-vector matching cost without degrading retrieval quality.

Effect of Retrieval Count. Figure 4(a) evaluates ICCAD2012 retrieval across K. Raising K from 1 to 100 increases HR@K from 96.67% to 98.93%, but FHR@K from 0.023% to 0.197%. At $K = 10$, HR@K is 98.14% and FHR@K is 0.053%. We therefore set $K = 10$ to balance coverage, false hits, and evidence size.

Effect of Training Epochs. Figure 4(b) tracks ICCAD2012 retrieval during LVM adaptation. Over five epochs, HR@5 rises from 96.95% to 98.02%, while FHR@5 falls from 2.94% to 0.05%. Epoch 4 already attains 98.02% HR@5 with 0.07% FHR@5, which epoch 5 reduces to 0.05%. Thus, few epochs achieve strong retrieval, with later gains diminishing toward convergence.

5 Conclusion

In this paper, we presented LithoGDV, an image-based LHD framework that integrates general visual understanding with domain-specific visual priors. LithoGDV adapts a pretrained LVM to encode fine-grained layout geometry, retrieves comparable process-verified cases from a graph-indexed knowledge base, and fuses semantic, geometric, and statistical evidence for classification. On OpenHSDBench, LithoGDV achieves state-of-the-art hotspot detection accuracy under full-data, low-resource, and cross-design settings. Ablation and retrieval analyses further verify the complementary contributions of LVM adaptation and case-based evidence, as well as the efficiency of graph-indexed retrieval. These results demonstrate that the adapted LVM's transferable visual representations and the retrieved cases' domain-specific evidence jointly improve the accuracy and generalization of LHD. Future work will focus on improving the balance between hotspot detection and false-alarm control, incorporating GDSII semantic cues, and extending the case knowledge base across technology nodes and process conditions.

AI USE STATEMENT

Generative AI tools were used in this work solely for language polishing. All AI-assisted revisions were reviewed by the authors. The research ideas, proposed methodology, and experimental design were independently developed by the authors. All reported results were obtained from experiments conducted by the authors. The authors take full responsibility for the final content of this work

REFERENCES

Guojin Chen, Haoyu Yang, Bei Yu, and Haoxing Ren. Intelligent opc engineer assistant for semiconductor manufacturing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 23144–23151, 2025.

Ying Chen, Yibo Lin, Tianyang Gai, Yajuan Su, Yayi Wei, and David Z. Pan. Semisupervised hotspot detection with self-paced multitask learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(7):1511–1523, 2020.

Yuyang Chen, Yiwen Wu, Jingya Wang, Tao Wu, Xuming He, Jingyi Yu, and Hao Geng. LLM-HD: Layout language model for hotspot detection with GDS semantic encoding. In *ACM/IEEE Design Automation Conference (DAC)*, pp. 121:1–121:6, 2024.

Yuyang Chen, Qi Sun, Su Zheng, Xinyun Zhang, Bei Yu, and Hao Geng. HyDAS: Hybrid domain deformed attention for selective hotspot detection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 45(3):1523–1534, 2026.

Zihao Chen, Fan Yang, Li Shang, and Xuan Zeng. Automated and agile design of layout hotspot detector via neural architecture search. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1–6, 2023.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Tianyang Gai, Tong Qu, Shuhan Wang, Xiaojing Su, Renren Xu, Yun Wang, Jing Xue, Yajuan Su, Yayi Wei, and Tian-Chun Ye. Flexible hotspot detection based on fully convolutional network with transfer learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11):4626–4638, 2022.

Hao Geng, Haoyu Yang, Lu Zhang, Jin Miao, Fan Yang, Xuan Zeng, and Bei Yu. Hotspot detection via attention-based deep layout metric learning. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 16:1–16:8, 2020.

Hao Geng, Haoyu Yang, Lu Zhang, Fan Yang, Xuan Zeng, and Bei Yu. Hotspot detection via attention-based deep layout metric learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(8):2685–2698, 2022.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Xin Huang and Kye Min Tan. Beyond text: Unlocking true multimodal, end-to-end rag with tomoro colqwen3, 2025. URL https://tomoro.ai/insights/beyond-text-unlocking-true-multimodal-end-to-end-rag-with-tomoro-colqwen3.

Cong Jiang, Haoyang Sun, Dan Feng, Zhiyao Xie, Benjamin Tan, and Kang Liu. LithoExp: Explainable two-stage CNN-based lithographic hotspot detection with layout defect localization. *ACM Transactions on Design Automation of Electronic Systems*, 30(3):40:1–40:25, 2025a.

Cong Jiang, Yujia Wang, Dan Feng, Haoyu Yang, and Kang Liu. Generalizable lithographic hotspot detection using asynchronous meta-learning with only one shot. In *ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7, 2025b.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 39–48, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 18661–18673, 2020.

Juhwan Kim and Minghui Fan. Hotspot detection on post-opc layout using full-chip simulation-based verification tool: A case study with aerial image simulation. In *BACUS Symposium on Photomask Technology (BACUS)*, pp. 919–925, 2003.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

Binling Luo, Ying Wang, Peng Gao, Beihai Tan, Ziran Zhu, Zihui Zhang, Xiaoming Xiong, and Shuting Cai. Multi-scale feature refinement network for lithography hotspot detection. *ACM Transactions on Design Automation of Electronic Systems*, 2026.

Chris A. Mack. Thirty years of lithography simulation. In *SPIE Optical Microlithography (SPIE)*, pp. 1–12, 2005.

Joydeep Mitra, Peng Yu, and David Z. Pan. RADAR: RET-aware detailed routing using fast lithography simulations. In *ACM/IEEE Design Automation Conference (DAC)*, pp. 369–372, 2005.

Gabriel de Souza P. Moreira, Ronay Ak, Mengyao Xu, Oliver Holworthy, Benedikt Schifferer, Zhiding Yu, Yauhen Babakhin, Radek Osmulski, Jiarui Cai, Ryan Chesler, Bo Liu, and Even Oldridge. Nemotron colembed v2: Top-performing late interaction embedding models for visual document retrieval, 2026. URL https://arxiv.org/abs/2602.03992.

Nomic Team. Nomic embed multimodal: Interleaved text, image, and screenshots for visual document retrieval, 2025. URL https://nomic.ai/blog/posts/nomic-embed-multimodal.

Jingyu Pan, Chen-Chia Chang, Zhiyao Xie, Jiang Hu, and Yiran Chen. Robustify ml-based lithography hotspot detectors. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 134:1–134:7, 2022.

Hao-Chiang Shao, Guan-Yu Chen, Yu-Hsien Lin, Chia-Wen Lin, Shao-Yun Fang, Pin-Yian Tsai, and Yan-Hsiu Liu. LithoHoD: A litho simulator-powered framework for IC layout hotspot detection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 44(3):1098–1111, 2025.

Woojoo Sim, Kibok Lee, Dingdong Yang, Jaeseung Jeong, Ji-Suk Hong, Sooryong Lee, and Honglak Lee. Automatic correction of lithography hotspots with a deep generative model. In *SPIE Optical Microlithography (SPIE)*, pp. 1096105, 2019.

Shuyuan Sun, Yiyang Jiang, Fan Yang, Bei Yu, and Xuan Zeng. Efficient hotspot detection via graph neural network. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1233–1238, 2022.

J. Andres Torres. ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 349–350, 2012.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Ying Wang, Haopeng Yan, Yiwen Zhang, Peng Gao, Fei Yu, Xiaoming Xiong, and Shuting Cai. PSCaps: High-performance pose-sensitive layout hotspot detector based on CapsNet. *ACM Transactions on Design Automation of Electronic Systems*, 30(4):58:1–58:21, 2025a.

Yujia Wang, Jiaxing Wang, Dan Feng, Yuzhe Ma, and Kang Liu. Location is all you need: Efficient lithographic hotspot detection using only polygon locations. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1–7, 2025b.

Wan-Yu Wen, Jin-Cheng Li, Sheng-Yuan Lin, Jing-Yi Chen, and Shih-Chieh Chang. A fuzzy-matching model with grid reduction for lithography hotspot detection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(11):1671–1680, 2014.

Yifeng Xiao, Miaodi Su, Haoyu Yang, Jianli Chen, Jun Yu, and Bei Yu. Low-cost lithography hotspot detection with active entropy sampling and model calibration. In *ACM/IEEE Design Automation Conference (DAC)*, pp. 907–912, 2021.

Haopeng Yan, Ying Wang, Peng Gao, Fei Yu, Yuzhe Ma, Xiaoming Xiong, and Shuting Cai. A lightweight heterogeneous graph embedding framework for hotspot detection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 44(9):3479–3489, 2025.

Haoyu Yang, Jing Su, Yi Zou, Bei Yu, and Evangeline F. Y. Young. Layout hotspot detection with feature tensor generation and deep biased learning. In *ACM/IEEE Design Automation Conference (DAC)*, pp. 62:1–62:6, 2017.

Haoyu Yang, Jing Su, Yi Zou, Yuzhe Ma, Bei Yu, and Evangeline Fung Yu Young. Layout hotspot detection with feature tensor generation and deep biased learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(6):1175–1187, 2019.

Haoyu Yang, Shuhe Li, Cyrus Tabery, Bingqing Lin, and Bei Yu. Bridging the gap between layout pattern sampling and hotspot detection via batch active learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(7):1464–1475, 2021.

Hailong Yao, Subarna Sinha, Charles Chiang, Xianlong Hong, and Yici Cai. Efficient process-hotspot detection using range pattern matching. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 625–632, 2006.

Yen-Ting Yu, Ya-Chung Chan, Subarna Sinha, Iris Hui-Ru Jiang, and Charles C. Chiang. Accurate process-hotspot detection using critical design rule extraction. In *ACM/IEEE Design Automation Conference (DAC)*, pp. 1167–1172, 2012.

Xinyun Zhang, Yuyang Chen, Yiwen Wu, Su Zheng, Ran Chen, Min Li, Hao Geng, Binwu Zhu, Mingxuan Yuan, and Bei Yu. From contrastive to generative alignment: Large-scale hierarchical multi-modal pre-training for hotspot detection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2026.

Binwu Zhu, Su Zheng, Yuzhe Ma, Bei Yu, and Martin D. F. Wong. Bridging hotspot detection and mask optimization via domain-crossing masked layout modeling. *ACM Transactions on Design Automation of Electronic Systems*, 30(4):54:1–54:20, 2025.