

DRVision: A DRV-aware Routability Optimization Framework with Multi-modal Prediction and Vision-Based Routing Guidance

Xu Cheng, Pengcheng Fan, Peng Cao *Member, IEEE*

Abstract—Achieving zero design rule violations (DRVs) is mandatory for tapeout, necessitating precise routability prediction and optimization. However, existing DRV prediction methods neglect inter-layer interconnect dependencies and rely on oversimplified binary DRV hotspot classification. Meanwhile, optimization strategies are hindered by fixed routing constraints, consequently failing to deliver effective DRV reduction or accelerate design convergence. To address these issues, we propose DRVision, a novel DRV-aware routability optimization framework that enhances routability by integrating multi-modal DRV hotspot prediction with vision-based guidance for regional DRV density variations, enabling accurate early prediction and dynamic optimization. We evaluated the proposed DRVision framework on the dedicated routability benchmarks from ISPD’15. Experimental results demonstrate the excellent predictive accuracy of DRVision, achieving an average SSIM of 82.45% and NRMSE of 8.06% for test designs. Furthermore, DRVision enables superior routability optimization, delivering average DRV reductions of 42.17% and 31.40% at the post-routing and post-routing optimization stage respectively, alongside a 19% runtime reduction compared to the baseline flow with design tools.

Index Terms—Routability Prediction & Optimization, Design Rule Violation, Multi-modal Learning.

I. INTRODUCTION

As technology advances, the discrepancy between placement and routing has increased due to complex design rules and higher cell density [1]. Inaccurate routability prediction results in improper routing resource allocation, necessitating additional design iterations to achieve a design rule violation (DRV)-free routing solution. Therefore, how to accurately predict routability and generate optimal routability optimization strategies poses key challenges for accelerating design convergence and reducing turn-around time.

Building upon congestion prediction advances [2]–[5], recent routability prediction studies have focused on employing machine learning (ML) techniques for DRV prediction. Leveraging layout features extracted from placement, various image-to-image translation models have been utilized to predict DRV hotspots, including convolutional neural networks (CNN) [6]–[9], J-Net [10], and U-Net [11], [12]. Utilizing the information from netlists, graph neural networks (GNN) [13] are designed for short violation prediction from placement.

Furthermore, recent studies have also investigated routability optimization based on routability prediction and routing

blockage mechanisms. An explainable artificial intelligence (AI) method [14] is proposed to trace DRV root causes, demonstrating that DRVs can be reduced through routing blockage implementation. Building upon CNN-based DRV hotspot prediction [11] with coarse-grained routing blockage, subsequent work [12] introduces a placement tomography technique and routing blockage optimization method. This establishes layer-wise blockage generation to mitigate post-routing DRVs.

However, existing methodologies face three intertwined limitations: (1) Existing routability prediction models are fundamentally limited by their reliance on purely geometric data, neglecting the crucial topology of routing grid graphs. They flatten the inherently three-dimensional (3D) layout into a compressed two-dimensional (2D) representations, neglecting inter-layer interconnect dependencies and failing to model layer-wise DRV hotspots critical for routability optimization [6]–[9]. (2) They are hindered by an oversimplified binary classification of DRV hotspots, failing to provide fine-grained routability metrics essential for routability optimization [10]–[13]. (3) Current routability optimization techniques are hampered by fixed routing constraints that fail to account for regional DRV density variations, which leads to over-optimization in sparse regions and resource shortages in dense ones, ultimately hindering design convergence [11], [12].

To take the above issues into full consideration, we propose a DRV-aware routability optimization framework that predicts layer-wise DRV hotspots with fine-grained values using fused geometric and topology information, and generates routing guidance for dynamic regional routability optimization. The main contributions are as follows:

- We propose DRVision, a novel DRV-aware routability optimization framework that integrates multi-modal DRV prediction with vision-based routing guidance, enabling early and accurate hotspot prediction and provide dynamic guidance to accelerate design convergence.
- In order to capture both inter- and intra-layer interconnect impacts to routability, a multi-modal model based on heterogeneous graph attention network (HGAT) and multiscale attention U-Net (MAUnet) is proposed to jointly learn layout geometry and routing grid topology information, enabling fine-grained and layer-wise DRV hotspot prediction.
- In order to provide DRV-aware guidance that adapts to regional DRV density variations, a vision-based routing guidance model based on You Only Look Once (YOLO) is developed to generate adaptive routing blockages to optimize routability across different regions, yielding

This research is supported in part by the National Natural Science Foundation of China under Grant 62174031 and in part by the Fundamental Research Funds for the Central Universities.

Xu Cheng, Pengcheng Fan and Peng Cao are with National ASIC System Engineering Technology Research Center, Southeast University, Nanjing, China.

Corresponding author: Peng Cao (caopeng@seu.edu.cn).

fewer DRVs and reduction in design iterations.

The rest of the paper is organized as follows. Section II introduces the preliminaries. Section III shows the details of the proposed DRV-aware routability optimization framework. Section IV presents extensive experimental results and analysis. Finally, Section V concludes the paper.

II. PRELIMINARIES

A. Design Rule Violation

Each technology node in integrated circuit design defines a set of design rules establishing geometric constraints for circuit layouts. Following detailed routing, the routing solution undergoes design rule checking (DRC). In general, the solution that violates any design rules cannot be taped out successfully. The instances of violated design rules in the solution are called design rule violations (DRVs) [15], [16].

B. Routing Grid Graph

Integrated circuits fabricated in advanced technology nodes incorporate numerous metal layers. Efficient global routing (GR) algorithms navigate these multi-layer structures using inter-layer via bridges, enabling accelerated discovery of globally optimal routing solutions [17]. Standard GR implementation partitions the layout into discrete horizontal planes representing metal layers, each subdivided into grid cells (Gcells) through virtual horizontal and vertical lines spanning the layout surface, where the height of each Gcell typically corresponds to the height of standard cells. This constructs a routing grid graph where each Gcell forms a vertex. Vertices within the same layer connect via horizontal/vertical wire edges, while adjacent layers interconnect through vertical via edges.

C. Routing Blockage

Physical implementation tools provide routing blockage mechanisms to resolve congestion and reduce DRVs. Designers specify routing guides that control global routing behavior through routing blockage parameters. Each routing blockage defines a rectangular region on a metal layer with an associated track utilization constraint. For example, a routing blockage value of 100 indicates full default track capacity availability, whereas 70 denotes 30% capacity reduction. During global routing, these blockage directives enforce hard ceilings preventing global routers from exceeding specified resource thresholds in designated regions. Consequently, reserved resources enable detailed routers to resolve congestion and minimize DRVs.

III. METHOD

A. Overview

Fig. 1 highlights the innovations of our framework and the key differences over the native implementation of physical design flow. For a unrouted netlist, different from the baseline flow by design tools without any intelligent guidance for

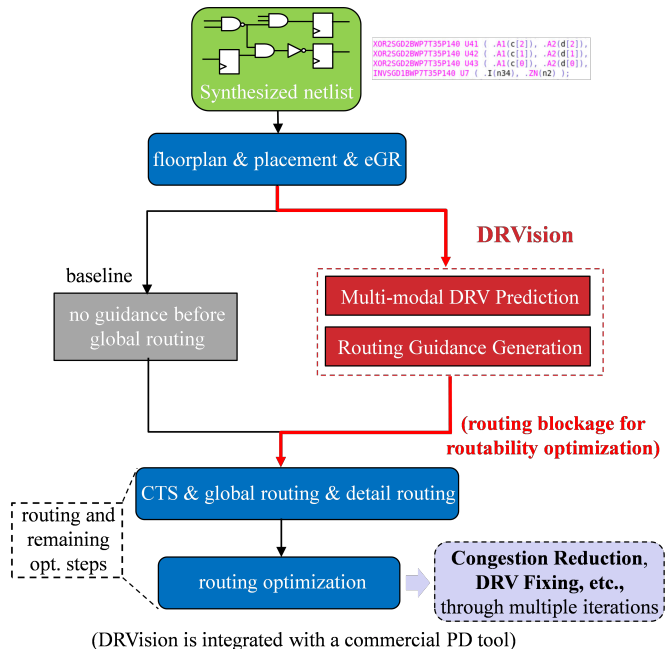


Fig. 1 Baseline physical design flow vs. proposed DRVision flow with routing guidance.

global routing, our proposed DRV-aware routability optimization framework, DRVision, generates routing guidance for commercial tools before global routing.

As shown in Fig. 1, the synthesized netlist is processed through floorplanning, placement, and early global routing (eGR). Subsequently, routing grid graph topology information and layout geometric information are extracted as inputs, which are then embedded via a multi-modal model to predict post-routing layer-wise DRV hotspots. Furthermore, utilizing the predicted hotspots, our DRVision generates routing guidance for dynamic regional optimization, which can be integrated into design tools for routability optimization, presenting significant benefits to routing stage and later tool-driven routing optimization stage including congestion reduction and DRV fixing.

In this work, our proposed DRVision employs multi-modal learning to predict post-routing layer-wise DRV hotspots (detailed in Section III-B), then leverages YOLO model [18] to generate routing guidance (detailed in Section III-C), achieving routability optimization.

B. Multi-modal Learning for Routability Prediction

To address the current inadequacies in modeling inter-layer interconnect dependencies during routability prediction [6]–[13], we propose a multi-modal DRV prediction model based on HGAT and MAUnet [19], [20] that embeds topology information of routing grid and geometric information of layouts. Different from prior approaches that rely solely on planar layout representations, this approach explicitly captures inter-layer interconnect dependencies, enabling fine-grained layer-wise DRV hotspot prediction.

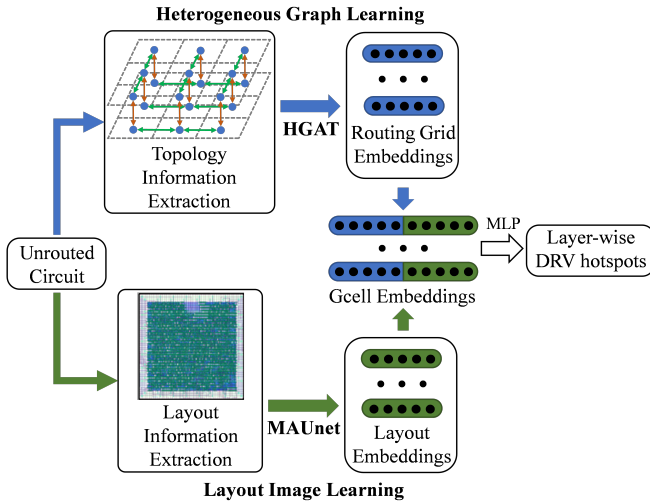


Fig. 2 Our proposed multi-modal DRV prediction model.

The proposed multi-modal DRV prediction model is illustrated in Fig. 2. The topology information of routing grid graph at the eGR stage is firstly represented as a heterogeneous graph, where related physical attributes are extracted as topology features for heterogeneous graph learning (detailed in Section III-B1). Gcell-related physical features are also extracted from the unrouted layout as layout features for layout image learning (detailed in Section III-B2). Then, the topology features are aggregated and updated with different schemes and weights from local view to generate the embeddings of routing grid, while layout features are also transformed to layout embeddings. Finally, the multi-modal Gcell embeddings are obtained by the alignment and fusion with the embedding from both modalities, where the embeddings for each metal layer are processed by a Multi-Layer Perceptron (MLP) to predict DRV counts for each Gcell, enabling fine-grained layer-wise DRV hotspot prediction.

1) *Heterogeneous Graph Learning*: As illustrated in Fig. 3, the heterogeneous graph representation for routing grid graph is composed of one type of Gcell node V_g and two types of bidirectional edges including intra-layer edge E_{intra} and inter-layer edge E_{inter} . Specifically, E_{intra} establishes connections between adjacent GCell nodes within the identical metal layers to model spatial adjacency and routing resource sharing while E_{inter} links Gcell nodes across adjacent metal layers to capture via-dependent physical connections and inter-layer dependencies.

Initial features for each type of nodes and edges are extracted to learn the routing grid topology information, with specific details described below. The Gcell node features are composed of the numbers of usable horizontal and vertical routing tracks for the routing layer in the Gcell, total numbers of used vias in contact with the routing layer in the Gcell and ratio of macro area overlapping with the Gcell to the Gcell area. The intra-layer edge features are composed of the fraction of routing tracks for each layer used in the Gcell by

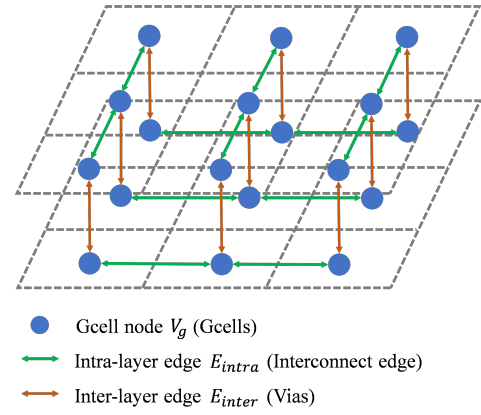


Fig. 3 Illustration of the heterogeneous graph representation for routing grid.

eGR while the inter-layer edge features include the numbers of used vias in via edge. Rich features are necessary not only to learn the physical and topology information but also to understand the local structure, which is critical for accurate DRV hotspot prediction.

2) *Layout Image Learning*: Each metal layer of the circuit layout is divided into $M \times N$ Gcells, and each Gcell is an analogized to a pixel in an image. The metal layer image representation $\mathbf{X} \in \mathbb{R}^{C \times M \times N}$ contains C -channel image data, and the size of each image equals to $M \times N$. Three density feature are extracted from each Gcell including track density, via density, and macro density to model layout density during eGR stage. These multi-channel density maps are processed by MAUnet, which leverages hierarchical feature extraction and attention mechanisms to enhance spatial context modeling for routability prediction.

It is worth noting that compared with the conventional U-Net which suffers from a fixed convolutional kernel receptive field and limited feature representation in the shallow layers, the MAUnet employs multi-scale convolutional blocks and an attention mechanism to enhance both the receptive field and feature representation.

C. Routing Guidance Generation for Routability Optimization

To address the inherent limitations of fixed optimization strategies [11], [12] in routability optimization, we propose a vision-based routing guidance model using the predicted DRV hotspots from our multi-modal DRV prediction model, which can generate adaptive routing blockages by considering regional DRV density variations for dynamic regional routability optimization.

Fig. 4 illustrates the vision-based routing guidance generation process. First, the predicted DRV hotspots of each metal layer from our multi-modal DRV prediction model are combined with the corresponding layer-specific track density and via density maps extracted from each Gcell of layout. They are translated into three-channel images to leverage YOLO's capability in detecting multi-scale irregular objects, enabling precise localization of DRV regions. Second, the images are fed into the *backbone*, which extracts hierarchical

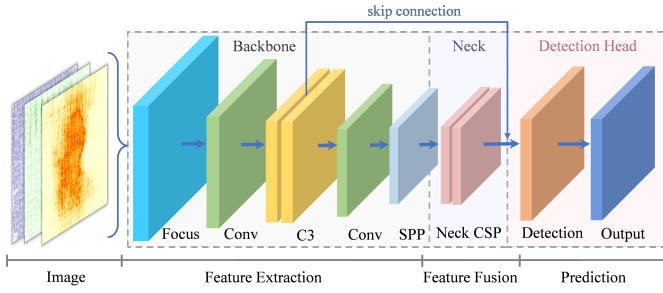


Fig. 4 Routing guidance generation based on YOLO.

feature representations at multiple scales. These features are then processed by the subsequent feature fusion module *neck* to generate three scaled feature maps for target detection. Finally, each *detection head* outputs a feature vector containing the predicted positions for routing blockage bounding boxes and the corresponding track utilization constraints as the final routing guidance for routability optimization.

Algorithm 1 DRV Hotspot Clustering for Label Generation

Require: 1) L : List of metal layers $\{L_1, L_2, \dots, L_n\}$
 2) For each layer L_i :
 Set of Gcells with positions (x, y) and DRV counts
 3) ϵ : Neighborhood radius
 4) $minPts$: Minimum points to form dense region

Ensure: **RouteBlkList**: $(position, util)$

```

1: BBoxList  $\leftarrow \emptyset$ 
2: for each metal layer  $L_i \in L$  do
3:    $P \leftarrow \{(x, y) \mid \text{Gcell in } L_i \text{ has DRV} > 0\}$ 
4:   if  $P = \emptyset$  then
5:     continue
6:   end if
7:    $clusters \leftarrow \text{DBSCAN}(P, \epsilon, minPts)$ 
8:    $valid\_clusters \leftarrow \text{non-noise in } clusters$ 
9:   for each cluster  $C \in valid\_clusters$  do
10:     $points \leftarrow \{p \in P \mid cluster(p) = C\}$ 
11:     $x_{min} \leftarrow \min\{x \mid (x, y) \in points\}$ 
12:     $x_{max} \leftarrow \max\{x \mid (x, y) \in points\}$ 
13:     $y_{min} \leftarrow \min\{y \mid (x, y) \in points\}$ 
14:     $y_{max} \leftarrow \max\{y \mid (x, y) \in points\}$ 
15:     $position \leftarrow (L_i, x_{min}, y_{min}, x_{max}, y_{max})$ 
16:     $area \leftarrow (x_{max} - x_{min}) \times (y_{max} - y_{min})$ 
17:     $density \leftarrow |points|/area$ 
18:     $util \leftarrow \text{ClassifyUtilization}(density)$ 
19:    RouteBlkList.append $((position, util))$ 
20:   end for
21: end for
22: return RouteBlkList

```

During the training of YOLO-based routing guidance generation, the labels are obtained from the real routing results of benchmark circuits by clustering the DRV hotspots with the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [21], whose procedure is presented in

Algorithm 1. For each identified DRV region, routing blockage labels are generated according to target detection bounding boxes and their associated track utilization constraints. The bounding box generation initiates with DRV counts captured per Gcell across all metal layers during routing (Lines 1-6). We then apply DBSCAN with key parameters ϵ (neighborhood radius) and $minPts$ (minimum points to form dense regions) to identify DRV hotspots using its ability to find differently shaped clusters without predefined cluster numbers (Lines 7). For each valid cluster, bounding box positions are computed by deriving coordinate extrema (Lines 8-15), followed by classification of track utilization constraints based on DRV density within bounding boxes (Lines 16-18). Finally, the routing blockage labels are appended to the output list (Line 19) for model training.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experiment Setup

In this work, the dedicated routability benchmarks from ISPD'15 [22] were utilized to evaluate the performance and generalization capability of our proposed DRVision. As summarized in the left part of Table I, the numbers of Gcells and nets range from 18k to 563k and 29k to 159k respectively, indicating the diverse scales of benchmark layout while the numbers of post-routing DRVs vary between 71 and 5586, reflecting the complexity variations of routing congestion among different circuits. For all designs, logic synthesis was performed using Design Compiler [23], followed by placement and routing using Innovous [24], where the command `violationBrowserReport` was employed to generate the golden labels for DRV prediction while the routing blockages for routability optimization were produced using the command `createRouteBlk`. All design tools were executed on an Intel Xeon Platinum 8462Y+ platform. Our DRVision was developed in Python using PyTorch 2.2.0 and Deep Graph Library [25], and was implemented on an NVIDIA A800 GPU paired with an Intel Xeon Gold 6348 CPU.

To validate the effectiveness of DRVision, we conducted comparative evaluations against state-of-the-art methods [9], [12], [13] with the identical features for fair comparison. The works in [9] and [13] propose a hierarchical classification and regression (HCR) model and GNN, respectively. Different from our layer-wise DRV hotspot prediction, both of them simplify 3D layouts into compressed 2D representations for DRV prediction. Since the methods [9], [13] only perform DRV prediction without subsequent routability optimization, its outputs are fed into our YOLO model to validate the critical impact of DRV prediction accuracy on subsequent optimization. For the work proposed in [12], it introduces a U-Net model for layer-wise DRV prediction based on placement tomography technique and routing blockage optimization to mitigate post-routing DRVs. Different from our dynamic regional optimization approach, it employs a fixed routing constraint for routability optimization.

TABLE I Benchmark statistics and prediction accuracy comparison. The comparative inferiority of other methods relative to DRVision are shown in **bold**.

benchmark	Circuit statistics			HCR [9]		GNN [13]		Unet [12]		DRVision(HGAT+MAUnet)		
	#Gcells	#Nets	#DRV	SSIM	NRMSE	SSIM	NRMSE	SSIM	NRMSE	SSIM	NRMSE	mAP
fft_1	18k	33k	3443	68.68%	12.86%	79.58%	9.51%	80.35%	8.99%	86.60%	5.63%	84.2%
matrix_mult_1	76k	159k	4460	72.53%	13.23%	76.43%	10.97%	81.72%	9.78%	88.65%	6.85%	78.6%
des_perf_a	203k	110k	3303	71.95%	12.11%	76.73%	7.82%	76.95%	7.36%	83.78%	2.99%	86.4%
edit_dist_a	160k	131k	4687	68.42%	12.58%	75.93%	9.93%	77.68%	7.04%	82.54%	4.08%	82.7%
fft_a	160k	32k	1582	73.17%	11.00%	78.47%	6.48%	79.94%	6.43%	86.96%	2.03%	85.1%
matrix_mult_a	563k	154k	1759	67.83%	13.69%	73.63%	6.02%	74.26%	5.17%	81.98%	3.56%	79.3%
pci_bridge32_a	40k	30k	1855	73.64%	14.45%	81.93%	7.76%	84.57%	6.08%	90.10%	2.53%	87.5%
pci_bridge32_b	160k	29k	71	71.35%	11.97%	81.12%	7.35%	80.89%	7.22%	86.25%	3.85%	81.9%
matrix_mult_2	77k	159k	3264	69.78%	15.48%	76.93%	10.91%	77.43%	9.72%	83.75%	8.85%	88.6%
matrix_mult_c	563k	152k	2921	70.65%	9.02%	76.38%	8.17%	76.71%	6.18%	85.72%	3.59%	83.4%
Ave. Train	-	-	-	70.80% (14.83%)	12.64% (8.24%)	77.71% (7.92%)	8.49% (4.09%)	79.05% (6.58%)	7.40% (3.00%)	85.63%	4.40%	83.8%
des_perf_1	50k	113k	4891	65.55%	14.58%	70.46%	13.55%	74.52%	12.88%	80.27%	7.00%	83.3%
fft_2	30k	33k	259	67.42%	17.16%	79.42%	15.88%	78.38%	15.34%	84.42%	10.68%	76.8%
des_perf_b	90k	113k	5586	68.32%	15.71%	71.56%	15.99%	74.85%	12.81%	83.50%	8.33%	82.4%
fft_b	160k	32k	2626	64.77%	13.11%	75.00%	11.04%	75.94%	10.24%	81.31%	7.82%	77.1%
matrix_mult_b	563k	152k	4628	65.45%	12.83%	72.82%	12.87%	73.34%	11.53%	82.74%	6.46%	81.7%
Ave. Test	-	-	-	66.30% (16.15%)	14.68% (6.62%)	73.85% (8.60%)	13.87% (5.81%)	75.41% (7.04%)	12.56% (4.50%)	82.45%	8.06%	80.3%

B. Routability Prediction Evaluation

To compare the competitive works [9], [12], [13] with our DRVision on DRV prediction, we employ the commonly used metrics, structural similarity (SSIM) [26] and normalized root mean square error (NRMSE) [27]. In this work, SSIM evaluates the similarity between predicted DRV hotspots and ground-truth post-routing hotspots, while NRMSE quantifies Gcell-wise prediction error by comparing the estimated and actual DRV counts per Gcell.

Table I demonstrates the routability prediction accuracy of DRVision and the competitive works [9], [12], [13]. Owing to multi-modal learning for the layer-wise DRV hotspot predictions, the proposed DRVision achieves excellent prediction accuracy with the average SSIM and NRMSE of 85.63% and 4.40% for train designs as well as 82.45% and 8.06% for test designs. Specifically, in terms of SSIM, it improves over [9], [13] and [12] by 14.83%, 7.92% and 6.58% on the train designs, and by 16.15%, 8.60% and 7.04% on the test designs, respectively. For NRMSE, DRVision reduces the error by 8.24%, 4.09% and 3.00% on train designs, and by 6.62%, 5.81% and 4.50% on test designs compared to [9], [13] and [12], respectively.

The inferior performance of methods [9], [12], [13] compared to our DRVision can be attributed to two main limitations. First, all of them neglect the inter-layer interconnect dependencies and fail to model fine-grained layer-wise DRV hotspots, limiting their prediction accuracy. Second, unlike DRVision’s multi-scale convolutional blocks and attention mechanisms that enhance receptive fields and feature representation, the HCR [9], GNN [13] and U-Net [12] lack advanced feature extraction capabilities to capture complex layout information.

Unlike the methods [9], [13] which perform no subsequent optimization and method [12] which ignores regional DRV

density variations, we deploy a YOLO-based model to predict the adaptive routing blockage for dynamic regional optimization, whose accuracy is evaluated by the mean Average Precision (mAP) in terms of both bounding box position and their associated track utilization constraints. Higher mAP values approaching 1 indicate higher precision in spatial positioning and constraint identification. As shown in the rightmost column of Table I, our DRVision demonstrates excellent prediction accuracy for routing blockage with the average mAP reaching 83.8% for train designs and 80.3% for test designs, which effectively covers the most ground-truth DRV regions while maintaining accurate track utilization constraints, enabling the generation of impactful routing guidance that significantly reduces DRVs.

C. Routability Optimization Evaluation

In order to evaluate design quality improvement with the proposed DRVision, Table II compares it with the baseline flow shown in Fig. 1 and competitive works [9], [12], [13] at both post-routing stage and further post-routing optimization stage, demonstrating that the pre-routing guidance not only improves the initial routing quality but also continues to benefit the post-routing optimization stage. The quality of the above-mentioned methods is evaluated by #DRV, WL, #Vias in Table II to represent the number of design rule violations, total wirelength and the number of vias respectively.

As shown in Table II, owing to accurate and adaptive DRV-aware routing guidance generated by our DRVision, a superior design quality improvement is achieved for all benchmark designs over [9], [12], [13] at both the post-routing stage and the post-routing optimization stage. At the post-routing stage, our DRVision achieves significant average #DRV reductions of 39.71% on train designs and 42.17% on test designs compared to the baseline, while even slightly reducing routing overhead

TABLE II Comparison of routability optimization results, which presents the number of design rule violations (#DRV), decrease of design rule violations ($\Delta\#DRV$), total wirelength (WL), decrease of wire length (ΔWL), the number of vias (#Vias), and decrease of vias ($\Delta\#Vias$) of circuits at post-routing stage and post-routing optimization stage. The unit of #Vias are $\times 10^6$, the unit of WL is m . The positive $\Delta\#DRV/\Delta WL/\Delta\#Vias$ indicates the decreased ratio compared to the baseline.

Benchmark	PD Stage	Baseline			HCR [9]+YOLO			GNN [13]+YOLO			Unet [12]			DRVision (HGAT+MAUnet+YOLO)		
		#DRV	WL	#Vias	$\Delta\#DRV$	ΔWL	$\Delta\#Vias$	$\Delta\#DRV$	ΔWL	$\Delta\#Vias$	$\Delta\#DRV$	ΔWL	$\Delta\#Vias$	$\Delta\#DRV$	ΔWL	$\Delta\#Vias$
fft_1	post-route	3443	0.65	0.20	29.54%	-0.06%	-0.06%	33.48%	-0.06%	-0.04%	35.61%	-0.04%	-0.01%	43.36%	0.11%	0.25%
	post-route opt	443	0.65	0.22	16.93%	-0.08%	3.52%	34.12%	0.02%	0.15%	37.92%	0.02%	0.37%	50.34%	0.01%	0.84%
matrix_mult_1	post-route	4460	2.77	0.84	19.98%	-0.54%	-0.28%	29.55%	0.07%	0.35%	29.96%	0.13%	0.36%	45.40%	0.21%	0.35%
	post-route opt	718	2.77	0.89	22.98%	-0.63%	-0.28%	27.20%	0.07%	0.48%	24.51%	0.06%	0.52%	32.73%	0.12%	0.53%
des_perf_a	post-route	3303	2.24	0.52	29.06%	-0.29%	-0.08%	35.64%	-0.34%	-0.06%	44.93%	-0.32%	-0.04%	60.25%	-0.08%	-0.07%
	post-route opt	845	2.24	0.52	24.85%	-0.27%	0.05%	28.00%	-0.31%	0.05%	28.05%	-0.30%	0.09%	30.06%	-0.06%	0.02%
edit_dist_a	post-route	4687	5.61	1.00	31.49%	1.92%	0.87%	31.84%	0.01%	-0.08%	32.28%	-0.29%	-1.06%	44.29%	0.01%	-0.04%
	post-route opt	688	5.61	1.01	18.31%	0.14%	-1.24%	22.43%	-0.42%	-0.11%	24.13%	-0.39%	-0.97%	40.26%	-0.02%	-0.05%
fft_a	post-route	1582	1.35	0.19	28.95%	0.30%	0.37%	26.75%	-0.05%	0.20%	28.00%	-0.03%	0.25%	34.32%	-0.03%	0.25%
	post-route opt	191	1.35	0.19	3.66%	0.30%	0.41%	7.21%	0.02%	0.24%	6.28%	-0.01%	0.30%	27.23%	-0.01%	0.30%
matrix_mult_a	post-route	1759	4.51	0.77	11.48%	-1.14%	-0.13%	18.76%	-1.06%	-0.12%	22.51%	-0.95%	-0.03%	32.46%	-0.02%	-0.10%
	post-route opt	113	4.51	0.77	29.20%	-1.13%	-0.04%	23.30%	-1.00%	-0.10%	34.51%	-0.94%	0.06%	49.56%	-0.01%	-0.06%
pci_bridge32_a	post-route	1855	0.60	0.13	17.47%	-1.41%	-0.23%	20.46%	-0.03%	-0.09%	21.29%	0.00%	-0.11%	50.19%	0.11%	0.12%
	post-route opt	465	0.60	0.13	18.92%	-1.42%	-0.27%	22.82%	0.02%	-0.04%	24.30%	-0.02%	-0.16%	53.33%	0.29%	0.45%
pci_bridge32_b	post-route	71	0.70	0.13	0.00%	0.00%	0.00%	5.23%	0.00%	0.00%	8.45%	0.00%	0.00%	11.27%	0.00%	-0.01%
	post-route opt	8	0.70	0.13	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	
matrix_mult_2	post-route	3264	2.61	0.82	20.19%	-0.09%	-0.04%	20.31%	-0.07%	-0.05%	21.48%	-0.01%	0.06%	30.91%	0.05%	0.18%
	post-route opt	173	2.62	0.85	30.06%	0.26%	-0.21%	26.58%	-0.07%	-0.06%	34.10%	-0.05%	0.07%	36.99%	0.02%	0.11%
matrix_mult_c	post-route	2921	4.28	0.74	19.72%	-0.18%	-0.10%	26.75%	0.05%	0.07%	31.26%	0.07%	-0.06%	44.64%	0.15%	0.08%
	post-route opt	462	4.28	0.75	39.61%	-0.18%	-0.09%	38.95%	0.04%	0.07%	43.94%	0.06%	-0.08%	54.33%	0.15%	0.06%
Ave. Train	post-route	2735	2.53	0.53	20.79%	-0.15%	0.03%	24.88%	-0.15%	0.02%	27.58%	-0.14%	-0.06%	39.71%	0.05%	0.10%
	post-route opt	411	2.53	0.55	20.45%	-0.30%	0.19%	23.06%	-0.16%	0.07%	25.78%	-0.16%	0.02%	37.48%	0.05%	0.22%
des_perf_1	post-route	4891	2.65	0.62	22.43%	2.02%	-0.06%	30.63%	0.14%	-0.58%	35.96%	0.14%	-0.75%	46.74%	0.15%	-0.27%
	post-route opt	421	2.65	0.63	14.73%	1.87%	1.62%	22.64%	0.13%	-0.15%	25.65%	0.13%	0.52%	37.05%	0.13%	-0.07%
fft_2	post-route	259	1.71	0.52	9.65%	0.02%	0.00%	16.95%	-0.01%	0.00%	22.78%	0.00%	0.00%	52.12%	-0.01%	-0.02%
	post-route opt	0	1.71	0.52	0.00%	0.00%	0.00%	0.00%	0.00%	-0.01%	0.00%	0.56%	0.00%	0.00%	-0.01%	-0.01%
des_perf_b	post-route	5586	0.63	0.18	15.09%	0.48%	0.69%	23.77%	0.16%	0.76%	22.63%	0.22%	0.88%	35.27%	0.47%	0.79%
	post-route opt	126	0.63	0.19	13.49%	0.43%	0.67%	25.80%	0.13%	0.69%	23.81%	0.18%	0.92%	50.00%	0.40%	0.62%
fft_b	post-route	2626	1.69	0.22	24.30%	0.53%	0.21%	27.08%	0.20%	0.42%	27.11%	0.14%	-0.19%	29.36%	0.50%	0.01%
	post-route opt	264	1.69	0.23	21.59%	0.59%	1.26%	24.05%	0.21%	0.33%	23.86%	0.17%	-0.23%	30.30%	0.00%	0.23%
matrix_mult_b	post-route	4628	4.57	0.78	24.18%	-0.79%	0.47%	25.45%	-0.04%	-0.17%	31.16%	0.00%	-0.15%	47.34%	-0.05%	-0.06%
	post-route opt	855	4.57	0.79	17.31%	-0.72%	0.34%	23.30%	-0.70%	-0.20%	27.72%	-0.02%	-0.08%	39.65%	-0.07%	-0.08%
Ave. Test	post-route	3598	2.25	0.47	19.13%	0.45%	0.26%	24.78%	0.09%	0.09%	27.93%	0.10%	-0.04%	42.17%	0.21%	0.09%
	post-route opt	333	2.25	0.47	13.42%	0.44%	0.78%	19.16%	-0.05%	0.13%	20.21%	0.20%	0.23%	31.40%	0.09%	0.14%

for WL and #Vias. At the post-routing optimization stage, it is encouraging to observe that routing guidance generated by DRVision continues to have a positive influence with average #DRV reductions of 37.48% on train designs and 31.40% on test designs.

Compared to the baseline flow, the average #DRV reductions of the competitive methods [9], [12], [13] are significantly inferior to our DRVision with up to 27.93% and 20.21% on train designs at the post-routing and post-routing optimization stages, respectively. The drawbacks of competitive methods stem from two primary factors. First, despite employing the same optimization method as our DRVision, the methods [9], [13] achieve suboptimal optimization results due to its inferior DRV prediction accuracy. Second, the method [12] only applies fixed routing constraints to all DRV regions, neglecting critical regional DRV density variations.

D. Runtime Analysis

The runtime overhead for the proposed DRVision is compared with the baseline and the competitive works [9], [12], [13] as illustrated in Fig. 5, which includes that for the whole physical design flow from floorplan stage to routing stage for all test designs. It can be observed in Fig. 5 that different from the baseline, additional runtime of less than 1% is occupied for this work and [9], [12], [13] to generate routing guidance. In spite of this, the proposed framework achieves 19% runtime reduction, with a 12% decrease in routing time

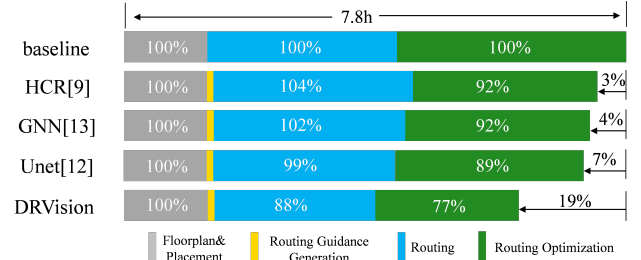


Fig. 5 Illustration of runtime comparison. Yellow part is mainly composed by the time for feature data collection, data processing, and inference.

and a 23% decrease in routing optimization time compared to the corresponding stages in the baseline through fine-grained layer-wise DRV prediction and dynamic regional routability optimization, which enable early identification of critical DRV hotspots to guide effective pre-routing optimization, significantly outperforming competitive works.

V. CONCLUSION

In this work, we propose DRVision, a DRV-aware routability optimization framework that integrates multi-modal prediction with vision-based routing guidance. It accurately predicts fine-grained, layer-wise DRV hotspots and generates adaptive routing blockages. Experimental results demonstrate significant improvements in both prediction accuracy and routability optimization over baseline and existing methods.

REFERENCES

- [1] A. B. Kahng, "Machine learning applications in physical design: Recent results and directions," in *Proceedings of the 2018 international symposium on physical design*, 2018, pp. 68–73.
- [2] Z. Xie, Y.-H. Huang, G.-Q. Fang, H. Ren, S.-Y. Fang, Y. Chen, and J. Hu, "RouteNet: Routability prediction for Mixed-Size Designs Using Convolutional Neural Network," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018, pp. 1–8.
- [3] S. Liu, Q. Sun, P. Liao, Y. Lin, and B. Yu, "Global placement with deep learning-enabled explicit routability optimization," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 1821–1824.
- [4] C. Yu and Z. Zhang, "Painting on placement: Forecasting routing congestion using conditional generative adversarial nets," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–6.
- [5] A. Ghose, V. Zhang, Y. Zhang, D. Li, W. Liu, and M. Coates, "Generalizable cross-graph embedding for GNN-based congestion prediction," in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2021, pp. 1–9.
- [6] W.-T. Hung, J.-Y. Huang, Y.-C. Chou, C.-H. Tsai, and M. Chao, "Transforming global routing report into DRC violation map with convolutional neural network," in *Proceedings of the 2020 International Symposium on Physical Design*, 2020, pp. 57–64.
- [7] R. Liang, H. Xiang, D. Pandey, L. Reddy, S. Ramji, G.-J. Nam, and J. Hu, "DRC Hotspot Prediction at Sub-10nm Process Nodes Using Customized Convolutional Network," in *Proceedings of the 2020 International Symposium on Physical Design*, 2020, pp. 135–142.
- [8] L. Li, Y. Cai, and Q. Zhou, "An Efficient Approach for DRC Hotspot Prediction with Convolutional Neural Network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.
- [9] D. Kim, J. Lee, and S. Kang, "Routability Prediction using Deep Hierarchical Classification and Regression," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023, pp. 1–2.
- [10] R. Liang, H. Xiang, J. Jung, J. Hu, and G.-J. Nam, "A stochastic approach to handle non-determinism in deep learning-based design rule violation predictions," in *2022 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2022, pp. 1–8.
- [11] Z.-H. Lee, C.-H. Lu, H.-H. Pan, T.-C. Wang, P.-Y. Chen, and C.-F. C. Shen, "A Robust Routing Guide Generation Approach for Mixed-Size Designs," in *2023 ACM/IEEE 5th Workshop on Machine Learning for CAD (MLCAD)*, 2023, pp. 1–6.
- [12] A. Kahng, S. Kundu, and D. Yoon, "Placement tomography-based routing blockage generation for drv hotspot mitigation," in *2024 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2024, pp. 1–9.
- [13] X. Chen, Z. Di, W. Wu, Q. Wu, J. Shi, and Q. Feng, "Detailed Routing Short Violation Prediction Using Graph-Based Deep Learning Model," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 2, pp. 564–568, 2022.
- [14] S. Park, D. Kim, S. Kwon, and S. Kang, "Routability Prediction and Optimization Using Explainable AI," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2023, pp. 1–8.
- [15] G. Chen, C.-W. Pui, H. Li, J. Chen, B. Jiang, and E. F. Young, "Detailed routing by sparse grid graph and minimum-area-captured path search," in *2019 24th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2019, pp. 754–760.
- [16] H. Li, G. Chen, B. Jiang, J. Chen, and E. F. Young, "Dr. CU 2.0: A scalable detailed routing framework with correct-by-construction design rule satisfaction," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–7.
- [17] Y. Pan, Z. Zhou, and A. Ivanov, "Routability-driven global routing with 3d congestion estimation using a customized neural network," in *2022 23rd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2022, pp. 1–6.
- [18] Z. Wang, L. Jin, S. Wang, and H. Xu, "Apple stem/calyx real-time recognition using yolo-v5 algorithm for fruit automatic loading system," *Postharvest Biology and Technology*, vol. 185, p. 111808, 2022.
- [19] R. Su, D. Zhang, J. Liu, and C. Cheng, "MSU-Net: Multi-scale U-Net for 2D medical image segmentation," *Frontiers in Genetics*, vol. 12, p. 639930, 2021.
- [20] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.
- [21] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [22] I. S. Bustany, D. Chinnery, J. R. Shinnerl, and V. Yutsis, "ISPD 2015 benchmarks with fence regions and routing blockages for detailed-routing-driven placement," in *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, 2015, pp. 157–164.
- [23] "Synopsys Inc. Design Compiler," <https://www.synopsys.com>.
- [24] "Cadence Inc. Innovus," <https://www.cadence.com>.
- [25] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks," *arXiv preprint arXiv:1909.01315*, 2019.
- [26] S. Zheng, L. Zou, S. Liu, Y. Lin, B. Yu, and M. Wong, "Mitigating distribution shift for congestion optimization in global placement," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2023, pp. 1–6.
- [27] M. B. Alawieh, W. Li, Y. Lin, L. Singhal, M. A. Iyer, and D. Z. Pan, "High-definition routing congestion prediction for large-scale FPGAs," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2020, pp. 26–31.