

LaRED: Efficient IR Drop Predictor with Layout-Preserving Rebuilder-Encoder-Decoder Architecture

Chengxuan Yu¹, Yanshuang Teng¹, Wenhao Dai¹, Yongjiang Li¹,

Wei W. Xing², Xiao Wu³, Dan Niu⁴ and Zhou Jin¹

1.SSSLab, Dept. of CST, China University of Petroleum-Beijing, Beijing, China

2. SoMaS, The University of Sheffield, Sheffield, United Kingdom

3. Huada Empyrean Software Co. Ltd, Beijing, China

4. School of Automation, Southeast University, Nanjing, China

Email: jinzhou@cup.edu.cn

Abstract—In the realm of integrated circuit verification, IR drop analysis plays a crucial role. Recent advancements in machine learning (ML) significantly enhance its efficiency, yet many current approaches fail to fully leverage the input structure of feature maps and the transmission mechanism of Power Delivery Network (PDN) layouts. To bridge these gaps, we introduce Layout-Preserving Rebuilder-Encoder-Decoder Architecture Predictor (LaRED), which employs a novel Rebuilder-Encoder-Decoder (RED) architecture and utilizes an innovative downsampling approach and upsampling framework to optimize its perception of instances and the transmission of features. LaRED captures information from various regions with asymmetric topological structure while preserving and transferring layout characteristics through deformable convolution, hybrid downsampling, cascaded upsampling, and attentional feature fusion. The rebuilder rebuilds raw input, whereas the encoder ensures comprehensive feature transmission across all instances. The decoder then facilitates seamless transfer of feature information across layers. This approach enables LaRED to integrate chip features of varying topologies and scales, enhancing its representational power. Compared to the current State-Of-The-Art (SOTA), MAUnet, LaRED achieves accuracy improvements of 34.6% to 42.6% in benchmark tests, establishing it as the new standard in static IR drop analysis for integrated circuit design with ML techniques. The code is available at <https://github.com/Todi85/LaRED>.

Index Terms—Static IR drop, RED architecture, Hybrid down-sampling, Cascaded upsampling

I. INTRODUCTION

To ensure that the chip maintains reliable performance up-time, rigorous design and verification of the Power Delivery Network (PDN) are indispensable. The PDN consists of metal rails and stripes that distribute the power supply (VDD and VSS) across the chip, with each metal layer exhibiting a certain resistivity. This leads to an unavoidable IR drop phenomenon when current flows through. The solution to the IR drop can naturally be expressed as a problem of solving a linear system [1]. However, with the continuous development of Very Large Scale Integration (VLSI), the number of transistors on a chip has surged, leading to increasingly large and complex power network structures. This growth has sharply increased computational demands, making verification increasingly challenging [2].

To address this issue, many simulation-based acceleration analysis methods have emerged, such as random-walk-based [3], row-based approaches [4], and multi-grid schemes [5].

TABLE I: Display of related works.

Method	Category	Backbone
Random-Walk-based [3]	Simulation	-
Row-based [4]	Simulation	-
Multigrid-based [5]	Simulation	-
XGBIR [6]	ML / non-ED	XGBoost [7]
PowerNet [8]	ML / non-ED	CNNs [9]
IREDGe [10]	ML / ED	Unet [11]
IRPnet [12]	ML / ED	PSPnet [13]
MAUnet [14]	ML / ED	Unet
LaRED (ours)	ML / RED	-

Although these methods can indeed speed up computation, the extensive need for matrix solution operations remains a barrier to achieve further time savings.

Thus, machine learning (ML) based methods have emerged as a promising solution [15]. As shown in Table I, most ML-based methods adopt Encoder-Decoder (ED) architecture, a few adopt non-ED architecture. For example, XGBIR [6] uses XGBoost as its backbone, but struggles with full-chip analysis for large, complex chips. PowerNet [8] employs Convolutional Neural Networks (CNNs) as its backbone, yet requires tile-by-tile analysis, limiting efficiency. IREDGe [10], utilizing Unet, supports full-chip prediction. However, it unfolds 2D maps into 1D vectors, making it unsuitable for large-size feature maps. IRPnet [12] leverages a Pyramid-Scene-Parsing-Network (PSPnet) backbone and introduces a loss function based on electrical properties. MAUnet [10], also built on Unet, incorporates an attention mechanism to improve prediction performance but tends to over-focus hotspot [16] regions.

Despite ongoing advancements, three key challenges remain prevalent in current ML-based methods, as follows: 1) *Uneven Input Structure*: Before the encoding stage, a common approach in computer vision is applied, i.e., concatenating feature maps along the channel dimension. However, it can lead to an uneven input structure because the feature maps used for IR drop analysis (e.g., power maps vs. PDN density maps) lack the consistency in distribution patterns typically seen in Red-Green-Blue (RGB) channels. 2) *Loss of Feature Information*: During the encoding stage, merely downsampling feature maps through max pooling [17] will lead to only the peak values within the

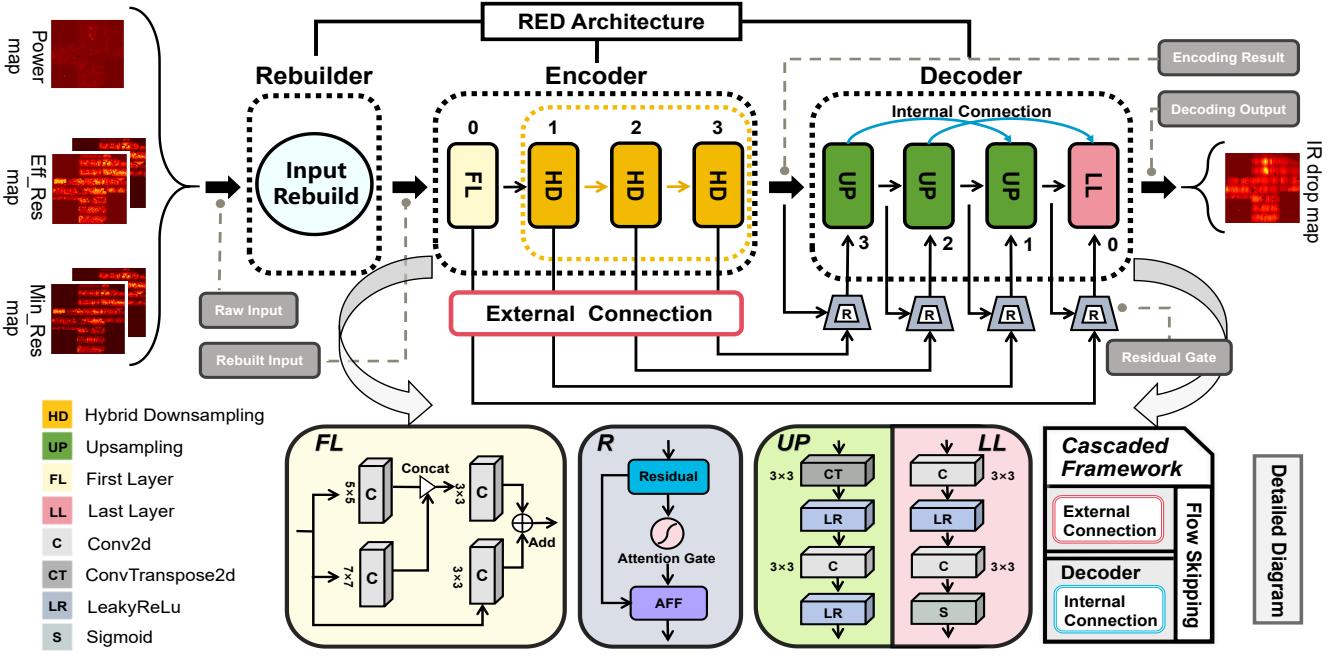


Fig. 1: Schematic illustration of the LaRED architecture. LaRED consists of three parts: (a) Rebuilder: grouping and rebuilding feature maps based on the differences in distribution patterns; (b) Encoder: using a hybrid downsampling approach to fully preserve features during encoding; (c) Decoder: establishing connections with different layers through a cascaded upsampling framework during decoding.

windows being propagated, which results in the loss of a large number of detailed features. 3) *One-sided Transfer Mechanism*: In the decoding stage, a one-sided feature transfer mechanism is often employed, where only the results from the encoder are passed to the decoder, neglecting important feature transfer between bottom and top layers within the decoder, which can also lead to the loss of features.

In this paper, we propose LaRED for IR drop analysis and introduce a novel RED architecture, as shown in Fig. 1. LaRED preserves more layout characteristics by solving the above problems, thus becoming the new State-Of-The-Art (SOTA) for static IR drop prediction. The contributions of this work are as follows:

- **RED Architecture:** In the RED architecture, the unevenness of the raw input is mitigated by grouping and rebuilding it with the rebuilder, making it more suitable for IR drop analysis.
- **Hybrid Downsampling Approach:** This approach reduces the loss of feature information by preserving the features of all instances, instead of focusing exclusively on peak values within the window.
- **Cascaded Upsampling Framework:** This framework extends the one-sided transfer mechanism by establishing connections within the decoder while enhancing the connection to the encoder.
- **Superior Performance and Embeddability:** LaRED improves accuracy by 34.6% to 42.6% compared to the current SOTA MAUnet. Furthermore, the techniques presented in LaRED can be easily embedded to other ML-based methods for optimization.

II. PRELIMINARIES

A. IR Drop Analysis Fundamentals

IR drop analysis in PDN typically employs Modified Nodal Analysis (MNA). For a PDN with n nodes and m voltage sources, MNA formulates the problem as:

$$\begin{bmatrix} Y & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} = \begin{bmatrix} j \\ e \end{bmatrix}. \quad (1)$$

Here, $Y \in \mathbb{R}^{n \times n}$ is the admittance matrix, $A \in \mathbb{R}^{n \times m}$ is the incidence matrix, $v \in \mathbb{R}^n$ represents node voltages, $i \in \mathbb{R}^m$ denotes branch currents, $j \in \mathbb{R}^n$ is the vector of current sources, and $e \in \mathbb{R}^m$ represents voltage sources. Solving (1) yields nodal voltages, revealing IR drop across the PDN. However, modern chips with millions of nodes pose significant computational challenges. Despite advancements simulation-based methods, efficiently solving large-scale MNA systems, remains an open problem in IR drop analysis.

B. Encoder-Decoder Architecture

The architectures of neural networks should be designed to satisfy the task objectives [18], and the ED architecture, commonly used for IR drop analysis, consists of interconnected encoders and decoders. The encoder extracts input data layer by layer, yielding increasingly succinct features. The decoder then reverses this process, reconstructing the encoder's output. Initially designed for language processing, this architecture is now used across domains, including Unet for image processing, which downsamples the feature map in the encoder and upsamples it in the decoder to produce the final result.

C. Deformable Convolution and Attentional Feature Fusion

CNNs have limitations in simulating geometric transformations due to the fixed rectangular area of the convolution kernel. Ref. [19] proposed Deformable Convolution (DC), which adds a two-dimensional offset to the sampling positions of each cell in the regular grid of standard convolutions, allowing the sampling grid to deform freely. Therefore, the convolution kernel of deformable convolution can be adjusted locally and densely according to the characteristics of the feature maps.

The classic feature fusion methods are summation or concatenation, as used in IREDGe and MAUnet. Ref. [20] proposed Attention Feature Fusion (AFF), which is suitable for feature fusion caused by merging different layers or branches in the network, like skip connections. AFF can thus avoid overly complex features hindering the learning ability of the model.

III. METHODOLOGY

LaRED consists of three components: rebuilder, encoder, and decoder. The rebuilder is focused on the rebuilding of the raw inputs to avoid structural unevenness, the encoder is focused on preserving the feature of all instances in the downsampling, and the decoder is focused on the framing and enhancement of the feature transfer mechanism. The feature maps used in our work are focused on the power of the instance, the combined macroscopic effect of the resistive components of the power lines, ground lines, vias, etc., and the resistance on the minimum path where the current density tends to be high, consisting of three types: 1) *Power map*, which represents the total power of each instance, is the sum of leakage power, switching power, and internal power. 2) *Eff_Res map*, which contains the effective resistance values of the instance connected to the power net and the ground net. 3) *Min_Res map*, which includes the minimum path resistance values of the instance connected to the power net and the ground net.

A. Raw Input Rebuilder

In the rebuilder, the key is to reduce the channel unevenness of the raw input structure and to extract feature maps with finer granularity, we propose a method that involves first slicing, then independently extracting, and finally concatenating, as illustrated in Fig. 2.

We input five feature maps into the rebuilder and categorize them into two types based on their distribution patterns after visualization, and then perform channel splitting at the topmost slicing gate according to their categories to obtain two slices. The slicing method can be expressed by

$$\begin{aligned} \mathcal{X}_1 &= SG(a, b) \\ \mathcal{X}_2 &= SG(b, c), \end{aligned} \quad (2)$$

where \mathcal{X}_1 is the first slice, \mathcal{X}_2 is the second slice, SG is the slicing gate, and a, b, c are the slicing positions. The slicing positions we use are $a=0, b=4, c=5$.

After passing through the slicing gate, we process the two slices independently. Considering the disparities in channel dimensions between the two slices, coupled with the commonly irregular arrangement of instances in the chip's dense regions,

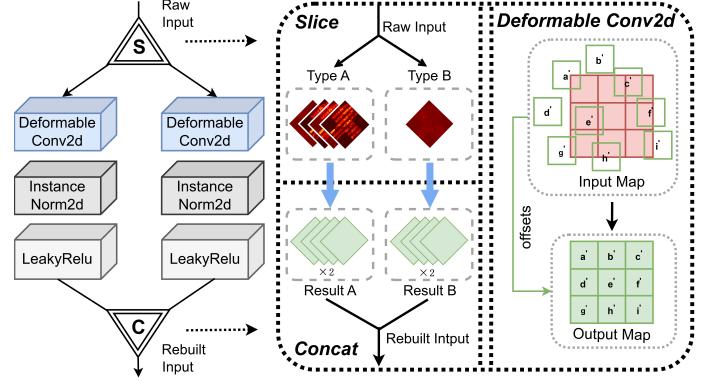


Fig. 2: The working principle of the rebuilder.

and the inherent asymmetry of topological connections within circuits [12], the fixed square convolution kernel of standard convolution is not suitable for finely capturing chip feature maps. Therefore, we employ DC for feature extraction, and in this process, expand the dimensions of the two slices to be equal to ensure input channel evenness.

The calculation principle of DC is as follows:

$$\mathcal{X}_i(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n), \quad (3)$$

where p_0 is each location on the feature map, p_n is the locations in R , w is the weight of the sampled value. As the grid R slides over the slice \mathcal{X}_i , the offset Δp_n , acting as a learnable parameter, enables its receptive field to undergo adaptive deformation based on the topology. This deformability enhances LaRED's perception of complex layouts. Finally, the rebuilder concatenates the two sets of extracted and expanded feature maps at the bottom concatenation gate to obtain the rebuilt input and outputs it to the encoder. Rigorous experimental validation has demonstrated that our method exerts a substantial positive influence on the prediction results.

B. Hybrid Downsampling Encoder

In the encoder, the downsampling operation conducted essentially serves to reduce the size of the map, i.e., compressing it. In the current ML-based methods for IR drop prediction, max pooling is still employed as in conventional image tasks during downsampling. However, this approach obviously considers

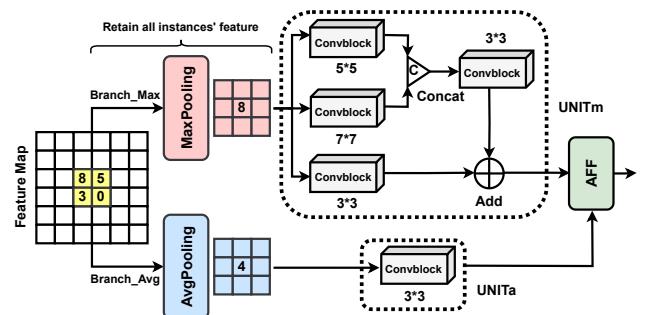


Fig. 3: Illustration of the hybrid downsampling.

only the maximum value within the pooling window, which means that when the window slides across the feature map, information from instances other than the peak value within the local region is completely ignored. This results in feature information being discarded for most instances after subsequent downsampling layers, and this lost information implies that a massive number of instances cannot be represented in the machine learning process. Hence, we propose a hybrid downsampling approach, as shown in Fig. 3.

We use averaged pooling [21] in addition to max pooling for its auxiliary correction during downsampling, and through this hybrid approach ensures that features from all instances continue to be transferred in the network. The whole process uses the strategy of two-branch hybrid pooling followed by multiscale convolutional extraction. Specifically, when the feature map enters the maximum pooling branch, it will simultaneously enter the average pooling branch. Firstly, the peak level and the average level of the features of the instances in each window are retained by the two pooling methods respectively, and then processed in the corresponding multiscale convolutional unit to obtain the results of the two branches. We use the AFF technique to fuse the two branch results, where AFF will adaptively obtain a weight value, and summing the two branch results based on it gives a hybrid downsampling result that preserves all the features of the instances.

This approach achieves a good balance between learning hotspot and non-hotspot regions. Taking pooling stride = 2 as an example, the specific calculation process is as follows:

$$\begin{aligned}\mathcal{S}_{Mij} &= \text{Unit}_{Mi}(\text{Max}(x_{j,1}, x_{j,2}, x_{j,3}, x_{j,4})) \\ \mathcal{S}_{Aij} &= \text{Unit}_{Ai}(\text{Avg}(x_{j,1}, x_{j,2}, x_{j,3}, x_{j,4})) \\ \mathcal{S}_{Hij} &= \Psi_i(\mathcal{S}_{Mij}, \mathcal{S}_{Aij}),\end{aligned}\quad (4)$$

where \mathcal{S}_{Mij} , \mathcal{S}_{Aij} , and \mathcal{S}_{Hij} denote the results of max pooling, average pooling, and hybrid pooling, respectively; Unit_{Mi} and Unit_{Ai} are the convolutional units; Ψ_i refers to AFF, with i indicating the layer and j indicating the pooling window.

C. Cascaded Upsampling Decoder

The decoder's task is to restore the compressed image from the encoder through upsampling. However, both the compression and restoration of feature maps inevitably result in the loss of feature information. Hence, we propose a cascaded upsampling framework for the decoding process, which extends the one-sided transfer mechanism of features. It consists of two parts, as illustrated in Fig. 1:

- External connections of the decoder:** We employ a residual gate to the skip connections from the encoder. The residual gate mitigates the issue of over-focusing on hotspots which is common in traditional attention gates through residual connections and the AFF technique. It allows each upsampling stage to autonomously select the most relevant downsampling features specific to itself. This is beneficial for preserving the layout characteristics of PDN during the transmission of feature.
- Internal connections of the decoder:** We also design an extra skip connection scheme for feature enhancement

within the decoder. This scheme transfers the outputs of the lower-level layers, $up3$ and $up2$, to the upper-level layers, $up1$ and ll , correspondingly, thus breaking the restriction of strict layer-by-layer feature propagation. This enables LaRED to learn information that combines both the raw features of the lower layers and the abstract features of the higher layers.

This framework further ensures that LaRED retains the physical details on the chip at a finer granularity. The specific computational process is as follows:

$$\begin{aligned}\mathcal{R}_{i+1}[c+1, C] &= \Omega_{i+1}(\text{Concat}(\mathcal{R}_i[1, C], \beta_i^k(\mathcal{R}_{i-1}[1, C]))) \\ \mathcal{R}_{i+1}[1, c] &= \Psi_{i+1}(\sigma_{i+1}(\mathcal{R}_i[1, C], \mathcal{D}_{i+1}[1, C]), \mathcal{D}_{i+1}[1, C]) \\ \mathcal{R}_{i+1} &= \beta_i^{k+1}(\mathcal{R}_{i+1}[1, C]) = \beta_i^{k+1}(\text{Concat}(\mathcal{R}_{i+1}[1, c], \\ &\quad \mathcal{R}_{i+1}[c+1, C])),\end{aligned}\quad (5)$$

where \mathcal{R}_i denotes the decoder output, the $[c, C]$ represent the channel-wise input, \mathcal{D}_i is the encoder output, Ω_i is the convolution computation, β_i^k is the k th upsampling computation within the layer, Ψ_i is the AFF, σ_i is the attention gate [14], and i refers to the i th layer.

IV. EXPERIMENTS

A. Experimental Setup

We implement our model using PyTorch library and conduct experiments on an NVIDIA RTX 4060 GPU. The dataset, sourced from the open-source CircuitNet benchmark [22], comprises 3,150 samples representing three 14nm FinFET designs: RISCY, RISCY-FPU, and ZERO-RISCY. We split this dataset into 80% training and 20% testing sets. In preprocessing, we extract features from three designs and convert them into 256×256 maps, which are used for both training and testing LaRED. Subsequently, we tune the model's hyperparameters and the training parameters as detailed in Tab. II.

TABLE II: Parameter settings in LaRED.

Parameters		Settings	
Model	R	DC_1 filters [1,8] DC_2 filters [4,8]	
	E	FL_0 filters [16,32] HD_1 filters [32,64] HD_2 filters [64,128] HD_3 filters [128,256]	
	D	UP_3 filters [256,128] UP_2 filters [256,64] UP_1 filters [128,32] LL_0 filters [64,1]	
	Batch size		16
	Epochs		125
	Max iters		8000
	Optimizer		AdamW
	Betas		[0.9,0.999]
	Learning rate		0.005
	Weight decay		0
Training	Loss function		L1+L2
	Scaler		GradScaler
	Scheduler		CosineAnnealingLR

* R is the Rebuilder, E is the Encoder and D is the Decoder.

TABLE III: Performance comparison of LaRED, MAUnet, MAUnet-RED, and MAUnet-HD across different designs.

METHOD	RISCY				ZERO-RISCY				RISCY-FPU			
	MAPE (%)	MeAE (mV)	MaAE (mV)	CC	MAPE (%)	MeAE (mV)	MaAE (mV)	CC	MAPE (%)	MeAE (mV)	MaAE (mV)	CC
MAUnet [14]	32.5920	1.0186	62.3922	0.8867	38.3244	1.0243	54.1314	0.8636	32.7747	0.9281	67.0332	0.8831
MAUnet-HD	32.1358	0.9904	47.2830	0.8854	31.7730	0.9166	48.1142	0.8873	31.4253	1.0442	78.4432	0.8531
MAUnet-RED	26.7223	0.8586	39.9995	0.9170	27.7150	0.8036	51.2827	0.9025	24.8560	0.8971	55.3726	0.8895
IMP (RED)	18.0%	15.7%	35.8%	3.4%	27.7%	21.5%	5.3%	4.5%	24.2%	3.3%	17.4%	0.7%
LaRED (ours)	21.1852	0.8210	40.8744	0.9247	22.0037	0.7056	47.8176	0.9132	21.4506	0.8233	52.3425	0.9063
IMP	35.0%	19.4%	34.5%	4.3%	42.6%	31.1%	11.7%	5.7%	34.6%	11.3%	21.9%	2.6%

* IMP indicates the improvement of this method compared to the SOTA (MAUnet).

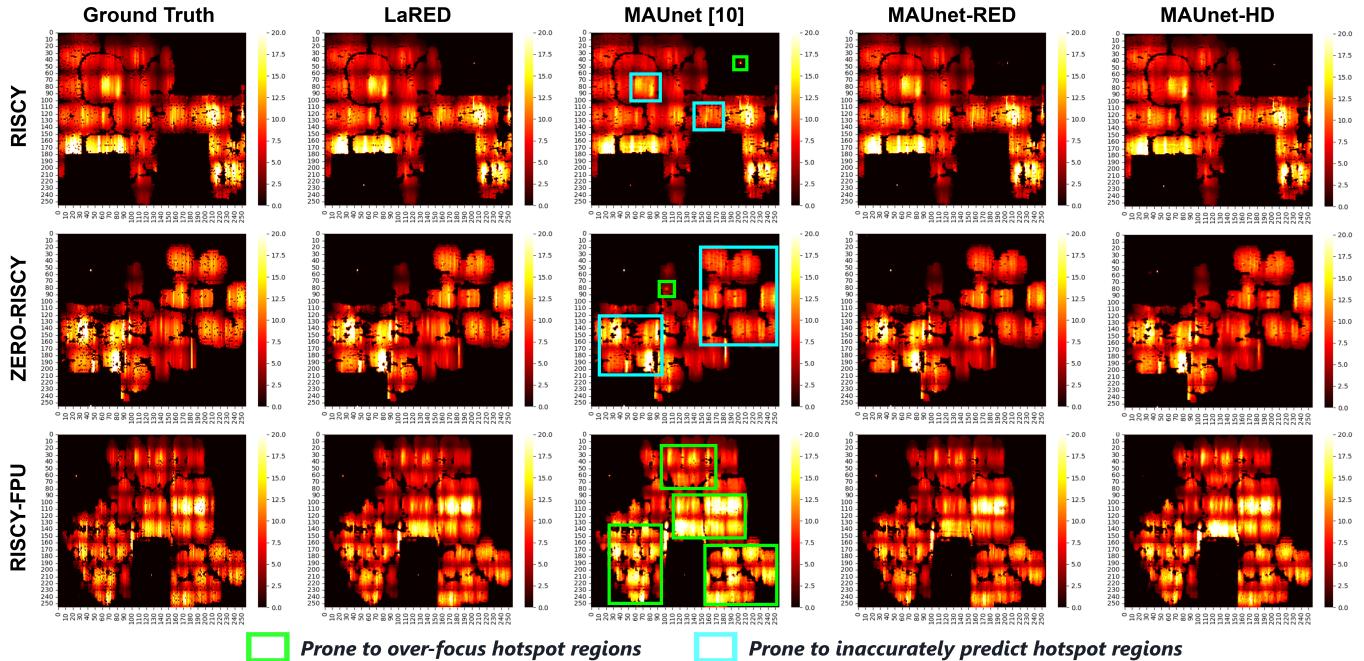


Fig. 4: Comparison of hotspot prediction performance between LaRED, MAUnet, and ground truth across different designs.

To assess LaRED’s performance, we compare it with IREDGe [10] and the current SOTA MAUnet [14]. We also create two optimized versions of MAUnet: MAUnet-RED (incorporating our rebuilder) and MAUnet-HD (using our hybrid downsampling approach). We employ four metrics for evaluation: Mean Absolute Percentage Error (MAPE), Maximum Absolute Error (MaAE), Mean Absolute Error (MeAE), and Correlation Coefficient (CC). These metrics are calculated using standard formulas, as shown in (6)–(9), with y_i representing the predicted IR drop and \hat{y}_i the actual (golden) IR drop.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{\hat{y}_i} \right| \times 100, \quad (6)$$

$$\text{MaAE} = \max_{1 \leq i \leq n} |\hat{y}_i - y_i|, \quad (7)$$

$$\text{MeAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (8)$$

$$\text{CC} = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}}. \quad (9)$$

B. Assessment of Model Convergence

We conduct a comprehensive comparison of the convergence performance among LaRED, IREDGe, and MAUnet. The experiment spans 9000 iterations across three designs. Training loss values are meticulously recorded at 1000 iterations intervals, as illustrated in Fig. 5.

Our results reveal that both LaRED and MAUnet successfully converge by the 8000th iteration. In contrast, IREDGe fails to converge in the early iterations, demonstrating its unsuitability for tasks involving large-size feature maps. This failure results in an inability to obtain a predictive model for IREDGe. Notably, throughout the entire 9000 iterations, LaRED consistently maintains the lowest loss value and exhibits the least frequent fluctuations. These observations provide strong evidence that LaRED possesses a more stable and efficient convergence capability compared to the other methods evaluated.

C. Assessment of Prediction Accuracy

To rigorously validate the effectiveness of LaRED, we conduct a comprehensive comparative evaluation against MAUnet, MAUnet-RED, and MAUnet-HD across three different designs, detailed in Tab. III.

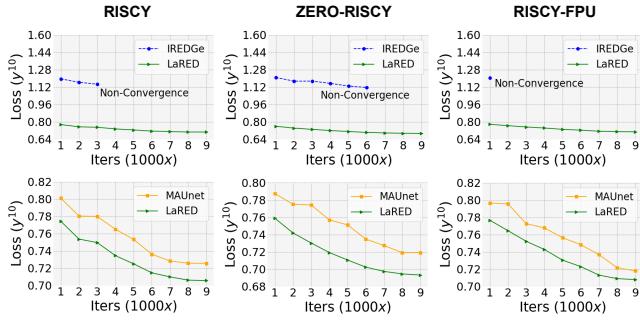


Fig. 5: Comparison of convergence performance between LaRED, MAUnet, IREDGe across different designs.

The analysis results reveals that LaRED achieves the highest accuracy across all three designs. Specifically, LaRED improves MAPE by 34.6% to 42.6%, MaAE by 11.7% to 34.5%, and MeAE by 11.3% to 31.1% compared to the MAUnet. Furthermore, MAUnet-RED, which incorporates our proposed RED architecture, shows substantial improvement across all three designs, with MAPE improving by 18.0% to 27.7%, MaAE by 5.3% to 35.8%, and MeAE by 3.3% to 21.5%. MAUnet-HD, utilizing our hybrid downsampling approach, also demonstrates improvements in two of the three designs.

These results provide strong validation for the effectiveness of our proposed method. Moreover, they demonstrate that both the RED architecture and the hybrid downsampling approach can be successfully embedded into other ML-based methods to achieve significant optimizations in IR drop prediction.

D. Assessment of Hotspot Prediction

We conduct a comparison of the hotspot prediction performance between LaRED and MAUnet, benchmarking both against the ground truth. The results are illustrated in Fig. 4.

Our analysis reveals that LaRED’s predictions for both hotspots and non-hotspots closely align with the real situation. In contrast, MAUnet, despite being an excellent ED architecture method, exhibits two primary issues. The first issue, which we term “Over-Focused Hotspot” (green boxes in Fig. 4), shows MAUnet’s tendency to over-focus on hotspots, mistakenly identifying non-existent hotspot regions or overestimating the intensity of actual hotspots. This issue is particularly severe in the RISCY-FPU example. The second issue, labeled as “Inaccurate Hotspot” (blue boxes in Fig. 4), demonstrates MAUnet’s failure to accurately predict hotspots in certain regions, incorrectly ignoring instances of high IR drop. This problem is evident in both the RISCY and ZERO-RISCY examples.

Notably, LaRED significantly mitigates both of these issues, demonstrating superior performance across all tested samples. The enhanced performance of LaRED can be attributed to its ability to preserve more layout characteristics during feature transmission. This preservation is achieved through the synergistic combination of the rebuilder, hybrid downsampling, and cascaded upsampling framework. As a result, LaRED consistently delivered the most precise prediction of hotspots among all the tested methods, marking a significant advancement in IR drop analysis for integrated circuit design.

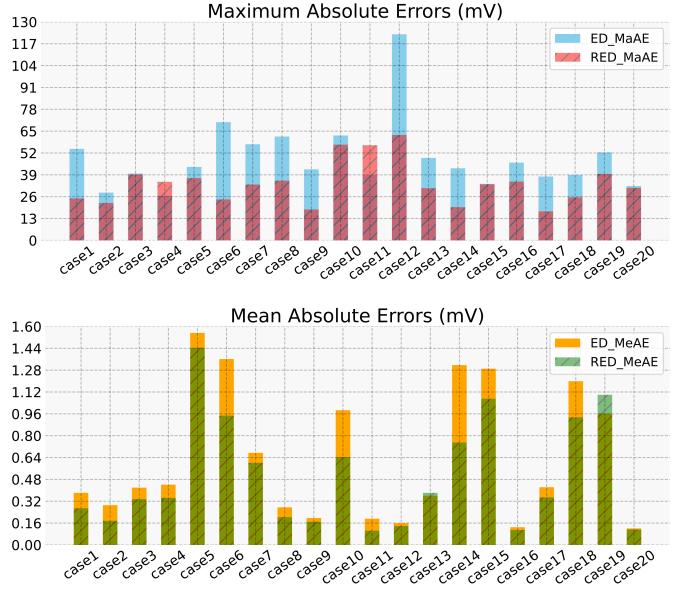


Fig. 6: Comparative analysis of MAUnet-RED (RED architecture) and MAUnet (ED architecture) across 20 test cases.

E. Assessment of RED Architecture

To independently demonstrate the superiority of the RED architecture, we conduct an extensive experiment comparing MAUnet-RED (representing the RED architecture) and MAUnet (representing the ED architecture) across 20 additional test cases (also from CircuitNet). The results are shown in Fig. 6.

Our analysis reveals that the RED architecture effectively reduces both MeAE and MaAE for the majority of the test cases. Particularly noteworthy are the results in cases 1, 6, 9, 12, 14, and 17, where the reduction in MaAE exceeds 50%. These findings illustrate that methods based on the RED architecture consistently achieve higher accuracy and demonstrate more stable and acceptable worst-case prediction scenarios. The substantial improvements observed strongly indicate that rebuilding the raw inputs is a necessary step for accurate IR drop prediction, highlighting the advantages of the RED architecture over traditional ED approaches in this domain.

V. CONCLUSION

We propose a novel RED architecture dedicated to static IR drop prediction. Based on this, we introduce a hybrid downsampling approach and a cascaded upsampling framework to preserve layout characteristics during feature transfer. Additionally, we incorporate DC to extract information from feature maps and AFF to fuse complex feature flows. Experiments on benchmark with different designs demonstrate that LaRED represents a new SOTA.

VI. ACKNOWLEDGMENT

Corresponding author is Zhou Jin. This work was supported by the NSFC under Grant (No. 62234010, 62204265), the Science Foundation of China University of Petroleum, Beijing (No. 2462024BJRC002).

REFERENCES

- [1] C.-W. Ho, A. Ruehli, and P. Brennan, “The modified nodal approach to network analysis,” *IEEE Transactions on circuits and systems*, vol. 22, no. 6, pp. 504–509, 1975.
- [2] M. Zhao, R. V. Panda, S. S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw, “Hierarchical analysis of power distribution networks,” in *Proceedings of the 37th Annual Design Automation Conference*, 2000, pp. 150–155.
- [3] H. Qian, S. R. Nassif, and S. S. Sapatnekar, “Power grid analysis using random walks,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 8, pp. 1204–1224, 2005.
- [4] Y. Zhong and M. D. Wong, “Fast algorithms for ir drop analysis in large power grid,” in *ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, 2005*. IEEE, 2005, pp. 351–357.
- [5] J. Yang, Z. Li, Y. Cai, and Q. Zhou, “Powerrush: A linear simulator for power grid,” in *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2011, pp. 482–487.
- [6] C.-H. Pao, A.-Y. Su, and Y.-M. Lee, “Xgbir: An xgboost-based ir drop predictor for power delivery network,” in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1307–1310.
- [7] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. Association for Computing Machinery, 2016, p. 785–794.
- [8] Z. Xie, H. Ren, B. Khailany, Y. Sheng, S. Santosh, J. Hu, and Y. Chen, “Pownet: Transferable dynamic ir drop estimation via maximum convolutional neural network,” in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2020, pp. 13–18.
- [9] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.
- [10] V. A. Chhabria, V. Ahuja, A. Prabhu, N. Patil, P. Jain, and S. S. Sapatnekar, “Thermal and ir drop analysis using convolutional encoder-decoder networks,” in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 2021, pp. 690–696.
- [11] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
- [12] Y. Meng, R. Lyu, Z. Bi, C. Yan, F. Yang, W. Hu, D. Zhou, and X. Zeng, “Circuits physics constrained predictor of static ir drop with limited data,” in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–2.
- [13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017.
- [14] M. Wang, Y. Cheng, Y. Lin, K. Peng, Y. Shunchuan, Z. Jin, and W. Xing, “Maunet: Multiscale attention u-net for effective ir drop prediction,” in *2024 61st ACM/IEEE Design Automation Conference (DAC)*. ACM/IEEE, 2024, pp. 1–6.
- [15] T. Chen, G. L. Zhang, B. Yu, B. Li, and U. Schlichtmann, “Machine learning in advanced ic design: A methodological survey,” *IEEE Design & Test*, 2022.
- [16] Y. Jiang, J. Song, X. Yin, X. Dong, S. Sun, Y. Lin, Z. Jin, X. Yang, and C. Zhuo, “A parallel simulation framework incorporating machine learning-based hotspot detection for accelerated power grid analysis,” in *MLCAD*, 2024.
- [17] M. a. Ranzato, Y.-l. Boureau, and Y. Cun, “Sparse feature learning for deep belief networks,” in *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20. Curran Associates, Inc., 2007.
- [18] M. Shafique, R. Hafiz, M. U. Javed, S. Abbas, L. Sekanina, Z. Vasicek, and V. Mrazek, “Adaptive and energy-efficient architectures for machine learning: Challenges, opportunities, and research roadmap,” in *ISVLSI*, 2017.
- [19] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.
- [20] Y. Dai, F. Gieseke, S. Oehmeke, Y. Wu, and K. Barnard, “Attentional feature fusion,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3560–3569.
- [21] H. Gholamalinezhad and H. Khosravi, “Pooling methods in deep neural networks, a review,” *arXiv preprint arXiv:2009.07485*, 2020.
- [22] Z. Chai, Y. Zhao, W. Liu, Y. Lin, R. Wang, and R. Huang, “Circuitnet: An open-source dataset for machine learning in vlsi cad applications with improved domain-specific evaluation metric and learning strategies,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.