Circuits Physics Constrained Predictor of Static IR Drop with Limited Data

Yuan Meng[1,†], Ruiyu Lyu[1,†], Zhaori Bi[1,*], Changhao Yan[1], Fan Yang[1],
Wenchuang Hu[2], Dian Zhou[1,3] and Xuan Zeng[1,*]
[1] State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai, China
[2] Precision Medicine Center, West China Hospital, Sichuan University, Chengdu, 610041, China
[3] Department of Electrical Engineering, University of Texas at Dallas, USA

***Abstract*—We propose a pyramid scene parsing network (PSPN) with skip-connection architecture to effectively utilize physical information that characterizes IR drop distribution, including current source locations, via locations, and asymmetric topological connections, achieving highly accurate IR drop prediction for power delivery networks (PDN) of varying scales, even with a limited dataset. Skip-connection architecture preserves the positional information of current sources, which often correlates with large IR drop, facilitating the identification of hotspots. We incorporate via locations into the model to effectively describe the topological connection distance between voltage sources and different nodes in the multi-layer PDN, while the traditional method only considers the horizontal distance between nodes and voltage sources, which is invalid for prediction. To capture asymmetric connection features within the PDN efficiently, we introduce a shape-adaptive convolutional kernel to solve the problem of inadequate extraction of feature information in a traditional method. Finally, we propose a loss function with Kirchhoff's law constraints to ensure the model's prediction aligns with the electrical characteristics of the circuit, which can't be guaranteed by traditional machine learning-based methods only taking the prediction accuracy into consideration. Our results, based on training with only 100 synthetic circuits, demonstrate the superiority of our method over the state-of-the-art prediction technique. Across evaluations on 10 real circuits, our approach consistently delivers a 50% improvement in precision.**

***Index Terms*—Static IR drop prediction, Physical information embedding, Limited data**

I. Introduction

The design of the Power Delivery Network (PDN) is important to ensure the proper functioning of individual circuit instances while minimizing the overall IR drop. However, with the ongoing advancement of Very-Large-Scale Integrated Circuits (VLSI), the voltage supplied to chips continues to diminish while transistor density steadily escalates, thus the importance of PDN design has experienced a substantial upsurge. Unfortunately, the PDN design is a formidable bottleneck for the integrated circuit design framework, primarily attributable to the ensuing challenges:1) Simulating the complex IR drop distribution incurs significant costs. Empirical studies have revealed that within advanced 7nm fabrication processes, the quantity of PDN nodes within a System-on-Chip (SoC) can surpass 10 billion [1]. Accurately predicting IR drop distribution costs hours and demands extensive memory resources, making single-simulation analysis difficult. 2) Designing a PDN often necessitates multiple iterations to resolve design-rule-related issues. Given that PDN design typically performs during the physical design stage, the time and effort invested in these iterations become considerably substantial. Henceforth, there exists an imperative necessity for the development of a tool that is capable of providing precise and expeditious IR drop predictions to streamline the chip design cycle. This need has garnered scholarly attention, resulting in the proposition of diverse methodologies, broadly classified into simulation-based and machine learning-based approaches.

Simulation-based techniques involve the solving of the linear equation denoted as $GV = J$, wherein G is the conductance network, V is the node voltage, and J is the branch current. Consequently, strategies to accelerate the simulation-based methodologies are primarily towards three key objectives: the reduction in the dimensions of the aforementioned equation as discussed in prior works [2]–[4], augmentation of the computational speed associated with LU decomposition [5], and the enhancement of the convergence speed within iterative methods [6]. Nonetheless, it is crucial to acknowledge that the pursuit of computational expediency in these methodologies may give rise to a commensurate decrease in predictive accuracy.

The machine learning-based approach utilizes a dataset containing PDN design parameters for model training and prediction of IR Drop distributions, which avoids the requirement of matrix calculations and significantly reduces computational time. IncPIRD [7] and XGBIR [8] employ the XGBoost model as their core computational framework. With regard to feature selection, these approaches extract pertinent information encompassing the relative spatial relationships of nodes in proximity to the current sources, equivalent resistances, and the overall topology of the PDN. It is important to note that IncPIRD exclusively focuses on predicting incremental IR drop. Furthermore, both XGBIR and IncPIRD adopt a partitioned chip-based prediction approach, precluding their capacity for global predictions. PowerNet [9] employs a CNN model for full-chip dynamic IR drop prediction. Compared to XGBoost, CNNs can capture two-dimensional spatial information, resulting in higher prediction accuracy. IREDGe [10] proposed a prediction model based on CNNs with automatic window size adjustment, using the full-chip power map as a feature, which further reduces the average error.

Nonetheless, CNNs employ kernels with an inherent square shape, which hinders their ability to effectively capture the

†Equal contributions.
*Corresponding authors: {zhaori_bi, xzeng}@fudan.edu.cn.

979-8-3503-4859-0/DATE24/© 2024 EDAA

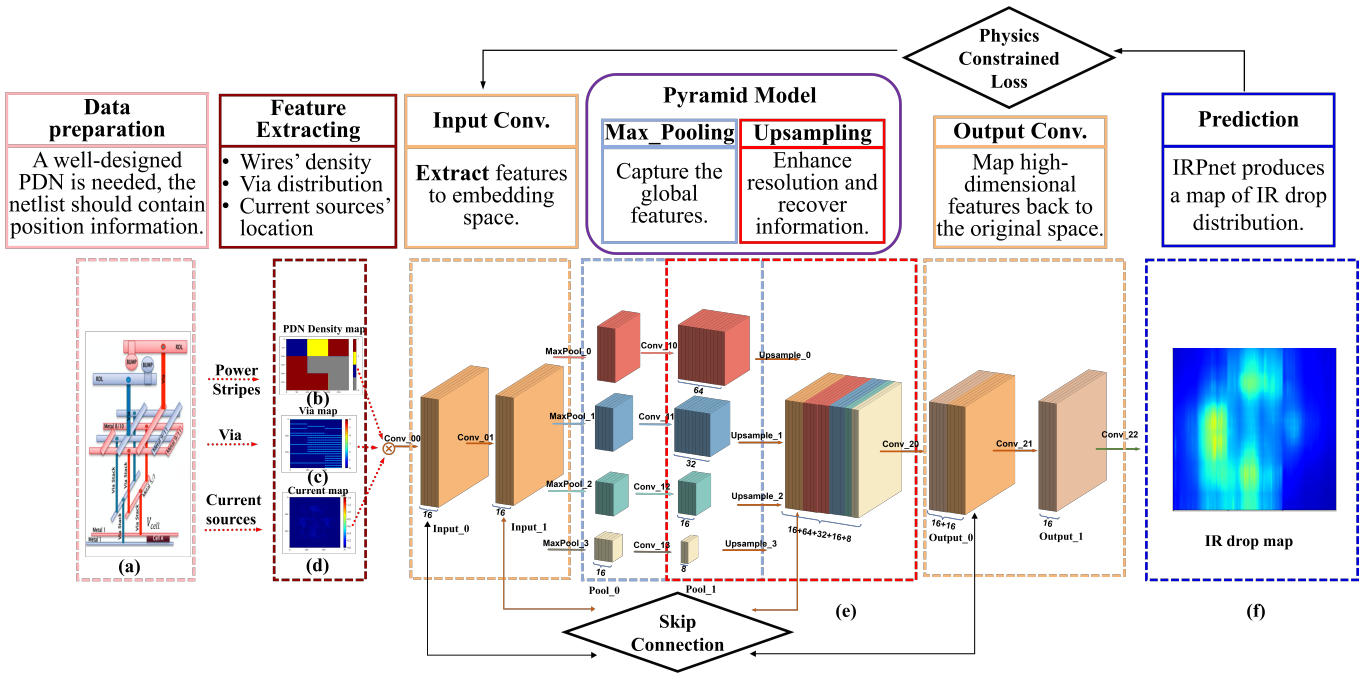

Fig. 1: The architecture and workflow of the proposed IRPnet. **(a)**Raw PDN data [11]. **(b)** A PDN density map, which represents the PDN wire density in different regions of a chip across all metal layers. **(c)** A via map represents the locations of vias. **(d)** A current map represents the locations of current sources. **(e)** The architecture of IRPnet comprises an input/output convolution layer and a pyramid pooling model.**(f)** The IR drop map, generated through predictions by IRPnet, visually represents the distribution of IR drop at the lowest layer of a chip.

intricate and asymmetrical adjacency relationships intrinsic to circuit networks. The feature engineering aspect of prior methodologies also exhibits certain deficiencies. For instance, previous approaches predominantly focus on the horizontal distance between nodes and voltage sources, neglecting to account for the actual physical distance between voltage sources and nodes via connection points. This oversight results in an incomplete representation of the true spatial relationships within the circuit. Furthermore, the loss function utilized in previous methodologies lacks consideration for the genuine physical attributes of the circuit network. This deficiency hinders the ability to ensure that the predictions align with the electrical characteristics inherent to the circuit, potentially compromising the accuracy and fidelity of the model's outputs.

To tackle the aforementioned challenges, we propose an IRPnet model based on a Pyramid Scene Parsing Network (PSPN) featuring a skip-connection architecture, which is devised to achieve precise predictions of IR drop within PDNs of diverse scales. Notably, the proposed IRPnet model was trained on a considerably small dataset, comprising merely 100 instances, yet it attains a prediction accuracy that surpasses IREDGe by a factor of two. The contributions of this work are as follows.

- We propose an IRPnet model incorporating a skip-connection architecture, specifically designed to better capture the features of PDN.
- We propose a shape-adaptive convolutional kernel to capture voltage interactions between asymmetric horizontal and vertical connector components in the underlying network. Moreover, we incorporate via position information as model input, effectively depicting the topological connection distances between different nodes in multi-layer circuit networks influenced by voltage sources.
- We propose a novel loss function with Kirchhoff's law constraints to ensure that the model's predictions align with the electrical characteristics of the circuit. By enforcing the constraints, we not only ensure the convergence of the model during training but also enhance the accuracy of its predictions.

The rest of this article is organized as follows. Section II introduces the implementation details of the proposed IRPnet model. Section III gives the experimental results, and Section IV summarizes the paper.

II. Proposed Method

The architecture of IRPnet comprises input convolutional layers, a pyramid pooling model, output convolutional layers and skip connections. As depicted in Fig. 1, the initial raw input PDN data undergoes a process of feature extraction to generate feature maps, which are subsequently fed into IRPnet for the prediction of the PDN IR drop distributions.

A. Feature Maps

From the raw PDN data, we extract three key features, namely the current map, the PDN density map, and the via map, which collectively serve as the input data for IRPnet. All feature maps have the same size as the output IR drop map. (a) Current maps, which represent the locations and values of all current sources in PDN. For nodes without current sources, their values are explicitly set to zero, aligning with

the approach described in [10]. (b) PDN density maps, which represent the spacing between power stripes within each chip region, are derived by extracting the average PDN pitch for individual regions across all metal layers. The values in PDN density maps can be categorized into four distinct levels: 0, 1, 2, and 3, corresponding to very low, low, medium, and high densities, respectively. This categorization aligns with the approach detailed in [10]. (c) Via maps, which represent the locations of vias in the circuit. The position with a via has a value of 1, and the position without a via has a value of 0. The position near a via means that its topological connection distance is close to a voltage source. This feature allows us to model the impact of the distribution of voltage source locations on IR drop. [10] only considers the horizontal distance between nodes and voltage sources, which does not reflect the actual distance between voltage sources and nodes through via connections, as shown in Fig. 2(a).

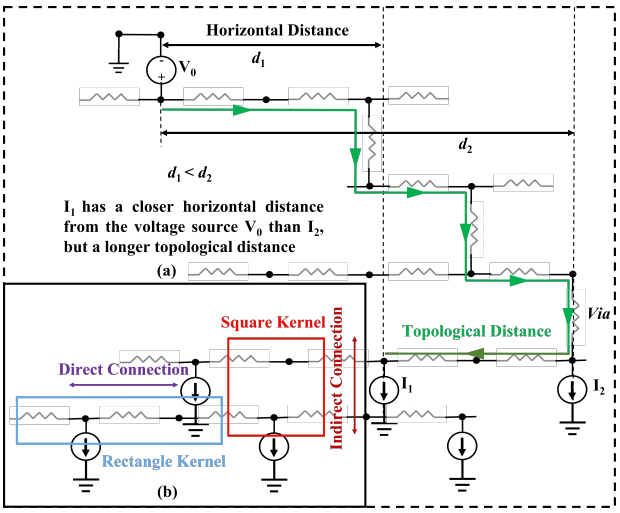

Fig. 2: PDN structure. **(a)** The measurement of the distance between the voltage source V_0 and the current sources I_1 and I_2 using horizontal distance proves to be inadequate, as evidenced by the scenario depicted in the figure where d_1 is less than d_2. In contrast, the measurement of distance based on topology, considering the vias, more accurately reflects the actual physical distance, as illustrated by the green line.**(b)**The conventional red square convolutional kernel tends to erroneously group indirectly connected devices together. In contrast, the proposed blue adjustable rectangular convolutional kernel is specifically designed to capture directly connected devices.

B. IRPnet Model

The feature maps are fed into the IRPnet to predict the IR drop distribution. To accommodate chips of varying sizes, i.e., differing matrix dimensions of feature maps, the IRPnet model exclusively employs convolutional and pooling layers. Skip connections are introduced to have more parameters participate in the training process.

(a) Input/Output convolutional layer. The convolutional layer [12] performs a weighted sum on a sliding window, extracts key information from input features, and at last decodes embedding features to an IR drop map. For an input feature map I, a convolutional kernel K, and an output map O, the convolutional operation is represented as Eq. (1).

$$O_{m,n} = (I * K)_{m,n} = \sum_{i=m}^{m+W_K-1} \sum_{j=n}^{n+L_K-1} I_{i,j} \cdot K_{i-m\, j-n}, \quad (1)$$

where $*$ represents the convolutional operation, and m, n are the index of the row and column in the output feature map, respectively. i, j are the index of the row and column in the input feature map, respectively. L and W are the length and width of the kernel respectively.

1) Shape-adaptive convolutional kernel: Due to the topological connections in the circuit are often not symmetrical, we propose a shape-adaptive convolutional kernel. As shown in Fig. 2(b), the blue rectangular convolutional kernel extracts more information from directly connected devices, while the red square kernel treats two directions equally, resulting in low extraction efficiency.

2) Skip connection: Due to its multi-layer structure, IRPnet is prone to encountering the 'vanishing gradient' problem during training, i.e., the gradients of the loss function with respect to the network's parameters become extremely small, approaching zero, as they are back-propagated through the layers of the network. This leads to the inability to capture features from the layers closer to the input, which keeps the most information. To mitigate this effect, as shown in Fig.1(e), we concatenate the input convolutional layer to the output convolutional layer. For instance, during backpropagation, when updating the *Output_0* layer, it becomes possible to update the parameters of the *Input_0* layer, thus avoiding the issue where the gradients of certain parameters in *Input_0* vanish after multiple applications of the chain rule. Skip connections enable more parameters to participate in the training process, preserving feature information to a greater extent, such as the locations of current sources and via positions, thereby improving the model's ability to capture features.

(b) Pyramid pooling model. We incorporate the Pyramid pooling model to enhance the IRPnet's receptive field. During the pooling phase, the pyramid pooling model reduces the dimension of the same feature in four different scales as shown in Fig. 1(e) red, blue, green, and yellow layers, generating compressed representations of the input data. This process is akin to isolating and processing the most salient aspects of the original information. In the upsampling phase, the upsampling layers restore the features to their original dimensions by replicating adjacent rows and columns.

Empirical findings, as demonstrated in [13], indicate that the pyramid pooling model effectively enlarges the receptive field, enabling the network to comprehensively capture background information. In the context of PDN, this background information can be interpreted as the number of instances surrounding a predicted node. For example, a node located close to numerous current sources is likely to exhibit a larger IR drop. Leveraging this background information proves instrumental in enhancing prediction accuracy.

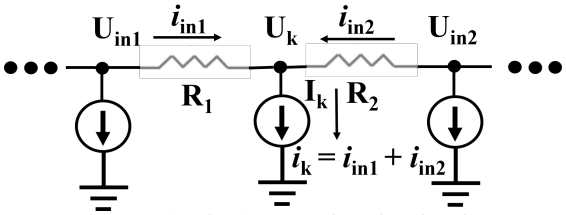

Fig. 3: Physical constraints in circuits.

C. Physics constrained loss function

Traditional machine learning-based methods treat the IR drop prediction problem as a regression task, with the overarching aim of optimizing the prediction accuracy for all PDN nodes. However, given the physics constraints inherent to circuits, neglecting these constraints can lead to a scenario where, despite achieving an overall small error, there may be substantial errors at individual critical points. This undermines the reliability of the prediction results. Hence, we opt to incorporate physical information into the loss function.

In accordance with Kirchhoff's law, the total current flowing into a node equals the total current flowing out of that node. We assign higher importance to nodes connected to current sources but not directly linked to vias, as hotspots tend to emerge at these specific nodes. As illustrated in Fig. 3, for a set of current sources not directly connected to vias $I = \{I_0, I_1, ..., I_k\}$, there is a constraint $\mathcal{C}_k$ as shown in Eq. (2):

$$\sum_s \hat{i}_s^{in} - i_k^{cs} = \sum_s \frac{\hat{U}_s^{in} - \hat{U}_k}{R_s} - i_k^{cs} = 0, \tag{2}$$

where i_k^{cs} represents the current value of the current source.

To make the model have a more accurate prediction of hotspots, we define a constrained loss $\mathcal{L}_{f1}$ in Eq. (3):

$$\begin{aligned} min\ \mathcal{L}_{f1} &= \tfrac{1}{WL}\textstyle\sum_{i=1}^{W}\sum_{j=1}^{L} rate_{i,j}(y_{i,j}^* - y_{i,j})^2 \\ s.t. \textstyle\sum_s \hat{i}_s^{in} - i_k &= \textstyle\sum_s \frac{\hat{U}_s^{in} - \hat{U}_k}{R_s} - i_k = 0 \\ rate_{ij} &= \begin{cases} 1,\ y_{i,j}^* < thread \\ \dfrac{y_{i,j}^*}{thread},\ y_{i,j}^* \geqslant thread \end{cases} \\ thread &= 0.85 \cdot max(y^*), \end{aligned} \tag{3}$$

where W and L represent the width and length of the IR drop map respectively. y^* and y represent a real IR drop map and a predicted map respectively. $\mathcal{L}_{f1}$ imposes greater weight on places with larger y^* to increase the prediction accuracy of the model for hotspots.

To simplify the training, we change the constraint to the form of a loss function $\mathcal{L}_{pi}$. Then we obtain the total loss as Eq. (4):

$$\mathcal{L}_{all} = \mathcal{L}_{pi} + \mathcal{L}_{f1} = \frac{1}{K}\sum_k^K (\sum_s \frac{\hat{U}_s^{in} - \hat{U}_k}{R_s} - i_k) + \mathcal{L}_{f1}. \tag{4}$$

TABLE I: Testcase Information

Testcase	1	2	3	4	5
Nodes	21, 672	77, 095	208, 025	208, 587	209, 981
Shape	298*298	298*298	930*930	930*930	641*641
Testcase	6	7	8	9	10
Nodes	209, 981	21, 762	21, 805	76, 908	77, 095
Shape	641*641	204*204	204*204	566*566	566*566

*Shape is measured in pixels

III. Experiments

A. Experimental Setup

We conduct experiments using the dataset provided by the ICCAD 2023 CAD Contest Problem C [11], which comprises 100 synthetic circuits generated using BeGAN [14]. The corresponding images of these circuits vary in size, ranging from 200x200 to 1000x1000 pixels, with each circuit having a different size. Additionally, there are 10 real circuits generated using OpenROAD, varying in size within the same range. The positions of nodes in the matrix are determined by dividing the positions specified in the netlist by 2000. The matrices contained in the .csv files are provided at a granularity of 1 μm. The details of the real circuits are summarized in TABLE I, with corresponding chip sizes spanning from $298 \times 298 \mu m^2$ to $930 \times 930 \mu m^2$.

To assess the quality of prediction results, we first define the absolute error $E_{IR}(n) = |IR_{pre}(n) - IR_{true}(n)|$ for node in the lowest layer of circuit. Subsequently, we calculate the mean error $E_{mean} = \frac{1}{N}\sum_{n=1}^{N} E_{IR}(n)$ and maximum error $E_{\max} = MAX(E_{IR}(n)), n \in N$ for all nodes, where n represents the node index, and N is the total number of nodes. Additionally, to evaluate the prediction accuracy of nodes with significant IR drop, we adapt the definition of the $F1$ as described in Eq. (5) [11]. We categorize each benchmark's nodes at the bottom layer by determining whether their IR drops are greater than 90% of the maximum IR drop in the ground truth. If it does, the node is labeled as $P(Positive)$; otherwise, it is labeled as $N(Negative)$. If the prediction is accurate, it is marked as $T(True)$; otherwise, it is marked as $F(False)$. Therefore, all nodes can be classified into four categories: $TP(TruePositive)$, $TN(TrueNegative)$, $FP(FalsePositive)$, and $FN(FalseNegative)$. By counting the number of nodes in each category and substituting these values into the formula, we can obtain the $F1$. From its definition, it becomes evident that the more large IR drop points the model predicts correctly, the higher the $F1$.

$$\begin{aligned} F1 &= 2 * \frac{Precision\ *Recall}{Precision\ +\ Recall}, \\ Precision &= \frac{TP}{TP\ +\ FP}, \\ Recall &= \frac{TP}{TP\ +\ FN}. \end{aligned} \tag{5}$$

B. Model Training

The proposed IRPnet comprises 492, 673 trainable parameters, and the hyperparameters employed in the model training process are outlined in TABLE II. We conduct the training using 100 synthetic circuits, with the training parameters detailed in TABLE III. The input data for each channel is normalized by

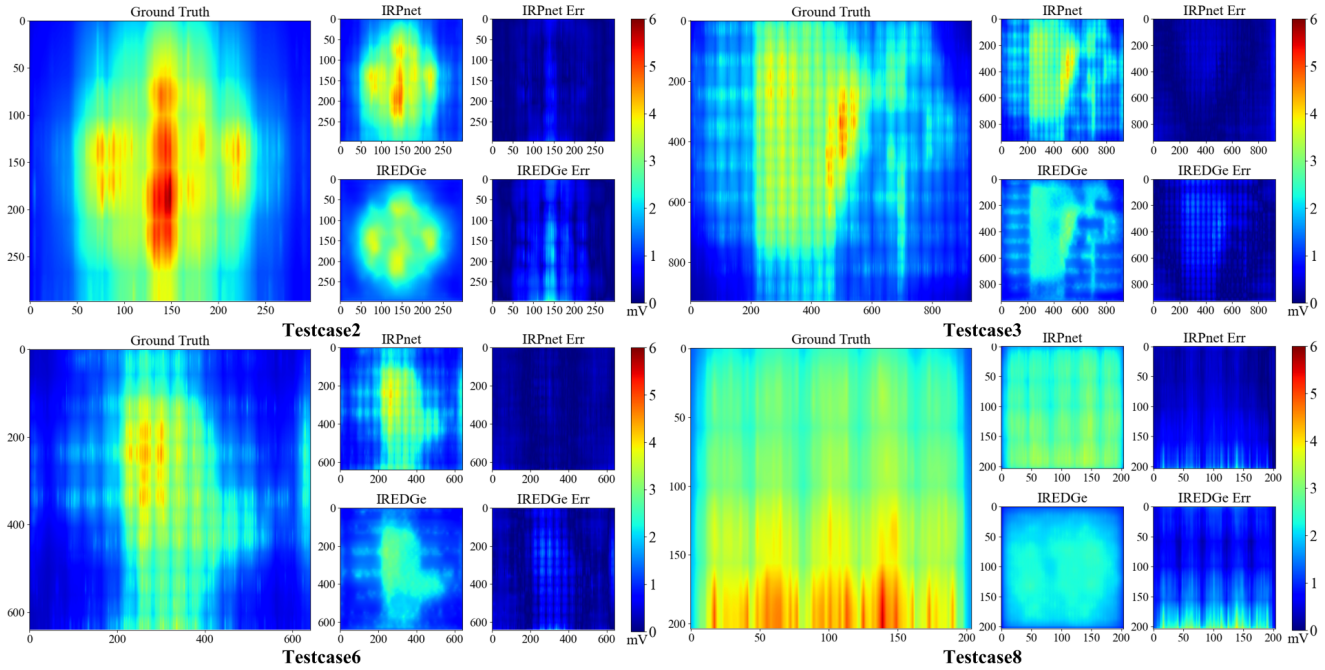

Fig. 4: Comparison of prediction results between IRPnet, IREDGe, and the ground truth. We compute the absolute errors (Err) for both IRPnet and IREDGe by subtracting their predicted values from the corresponding ground truth values for each pixel. Lower error values are depicted in deeper blue. It is evident that the prediction results generated by IRPnet closely align with the ground truth, exhibiting significantly lower errors when compared to IREDGe.

dividing by the respective maximum value. The optimization process is carried out using the Adam optimizer [15], and the loss function proposed earlier is utilized. Rectified Linear Unit (ReLU) serves as the activation function applied after each convolutional layer to introduce non-linearity. Max pooling is employed for dimensionality reduction while copying the nearest rows and columns is used for dimensionality expansion in upsampling. The training process is executed on an NVIDIA RTX A6000 GPU, utilizing PyTorch 2.0.1. After approximately 3 hours of training, the IR drop distributions can be predicted in just a few hundred milliseconds. The trained model is versatile and can be applied to IR drop predictions for circuits of varying sizes, provided they maintain the same resolution. Although under the current cases, the time consumed by traditional solvers is comparable to that of our method, theoretically, the solution time of traditional solvers tends to be $O(n^3)$ with the change in the scale of the circuit, that is, the number of nodes n, while our method is $O(n)$.

TABLE II: Model Hyperparameters

Layer Name	Kernal	Filters	Layer Name	Kernal
Conv_00	19*5	16	MaxPool_0	2*2
Conv_01	19*5	16	MaxPool_1	4*4
Conv_10	19*5	64	MaxPool_2	8*8
Conv_11	19*5	32	MaxPool_3	16*16
Conv_12	19*5	16	Upsample_0	2*2
Conv_13	19*5	8	Upsample_1	4*4
Conv_20	19*5	16	Upsample_2	8*8
Conv_21	19*5	16	Upsample_3	16*16
Conv_22	19*5	1		

C. Experimental results

We trained 100 synthetic circuits using the parameters shown in TABLE III, and the prediction results on 10 real circuits are shown in Fig. 5. The mean of the average error in the 10 real circuits is 0.182 mV, and the mean of the maximum error is 1.187 mV. The maximum error is only 0.079% of the supply voltage 1.5 V.

TABLE III: Training Parameters

Optimizer	Loss function	Epochs	Reg rate
Adam	$\mathcal{L}_{pi} + \mathcal{L}_{f1}$	1500	0.00009
Scheduler	Learning rate	Step size	Gamma
StepLR	0.0001	1000	0.90

We reproduce IREDGe [10] and adjust the convolutional kernel sizes to adapt to the change in image resolution. The comparison results are shown in Fig. 5, we have a 23.90% - 62.96% mean accuracy improvement, and the prediction accuracy of hotspots has been greatly improved from 0% to 21.61%. Fig. 4 shows IR drop maps predicted by two models. The prediction results of our model are very close to the real situation. In some cases the $F1$ of IRPnet remains at 0% because the maximum IR drop predicted is less than 90% of the maximum value in the ground truth, failing to predict any hotspots.

The comparison results of different models are shown in TABLE IV. *IREDGe - eff* sets the values of the *eff_distance* dimension to 1, which is equivalent to removing the information in this dimension. It can be observed that when considering the effective distance (eff_distance), the prediction accuracy of IREDGe deteriorates, which also validates our argument in SectionII-A. *IRPnet - via* sets the values of the via map dimension to 1. The result shows that the mean error is reduced by 0.078 mV and the maximum error is reduced by 0.25 mV by using via maps as input and $F1$ merely loses 3.85%. *IRPnet + square kernel* replaces the rectangle convolutional kernels

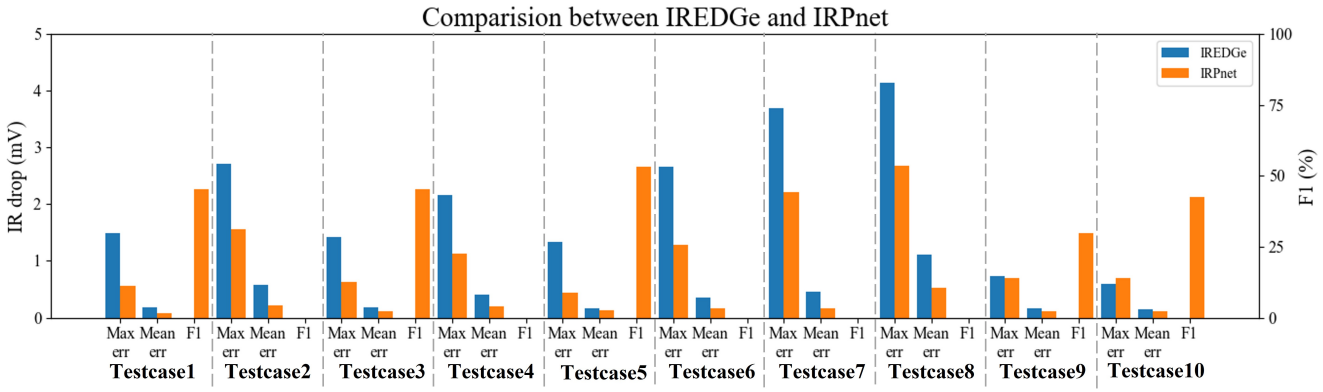

Fig. 5: Comparison of prediction results between IREDGe and IRPnet in regard to maximum error (max error), mean error (mean error), and $F1$ score.

TABLE IV: Comparison of different models

Model	IREDGe	IREDGe - eff	Im	IRPnet - via	Im	IRPnet + square kernel	Im	**IRPnet**	**Im**
Mean err (mV)	0.371	0.354	4.58%	0.260	29.92%	0.220	40.70%	**0.182**	**50.94%**
Max err (mV)	2.096	1.907	9.02%	1.437	31.44%	1.380	34.16%	**1.187**	**43.37%**
$F1$ (%)	0	2.30	2.30%	25.46	25.46%	24.40	24.40%	**21.61**	**21.61%**

*Im refers to the improvement over IREDGe. IREDGe - eff means IREDGe removes the eff_distance dimension. IRPnet - via refers to IRPnet removes the via map dimension. IRPnet + square kernel means IRPnet changes the rectangle convolution kernels to square kernels.

with square kernels. Using rectangular convolutional kernels reduces the mean error by 0.038 mV and the maximum error by 0.193 mV, and $F1$ merely loses 2.79%. These comparative experiments prove that our selection of via maps as the input feature and the selection of rectangle convolution kernels are effective.

IV. Conclusion

This paper proposes a neural network model for static IR drop prediction. With a limited dataset, we fully apply the topological structure information of PDN into the neural network, effectively improving the prediction accuracy. We also apply physics constraints to the loss function, so that the predicted value can meet the actual physical laws. After training on a very small data set, the average error and maximum error of our prediction on 10 real circuits are 0.182 mV, 1.187 mV, and the accuracy of hot spot prediction is 21.61%, which are respectively 50.94%, 43.37% and 21.61% better than the previous work. In future work, we will consider adding more physics information to further improve the performance.

V. Acknowledge

This research is supported partially by National Key R&D Program of China 2020YFA0711900, 2020YFA0711901, partly by National Natural Science Foundation of China (NSFC) research projects 62141407, 62304052, 61974032, 61929102, 61674042, 62090025.

References

[1] A. Krishnaswamy and R. Deokar, "Accelerate power integrity closure with redhawk™ fusion on the latest armv8-a processors," 2019, arm TechCon 2019.

[2] R. Jiang, T.-H. Chen, and C.-P. Chen, "PODEA: Power delivery efficient analysis with realizable model reduction," in *2003 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4. IEEE, 2003, pp. IV–IV.

[3] S. R. Nassif and J. N. Kozhaya, "Fast power grid simulation," in *Proceedings of the 37th Annual Design Automation Conference*, 2000, pp. 156–161.

[4] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 10, pp. 1148–1160, 2002.

[5] T.-H. Chen and C. C.-P. Chen, "Efficient large-scale power grid analysis based on preconditioned krylov-subspace iterative methods," in *Proceedings of the 38th annual Design Automation Conference*, 2001, pp. 559–562.

[6] 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.

[7] C.-T. Ho and A. B. Kahng, "IncPIRD: Fast learning-based prediction of incremental ir drop," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8.

[8] 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.

[9] Z. Xie, H. Ren, B. Khailany, Y. Sheng, S. Santosh, J. Hu, and Y. Chen, "PowerNet: 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.

[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] V. A. C. Gana Surya Prakash Kadagala, "Problem C: Static IR Drop Estimation Using Machine Learning." 2023. [Online]. Available: http://iccad-contest.org/Document/Problems/Problem_C_%20Static_IR%20Drop_Estimation_Using_Machine_Learning_v5.pdf

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[14] V. A. Chhabria, K. Kunal, M. Zabihi, and S. S. Sapatnekar, "BeGAN: Power grid benchmark generation using a process-portable GAN-based methodology," in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2021, pp. 1–8.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.