AnaIR: A Semi-**Ana**lytical Green's Function Inspired Neural Network for Static **IR** Drop Prediction

Shruti Pandey*, Aneeket Yadav†, Smruti R. Sarangi*
*Electrical Engineering, †Computer Science and Engineering
Indian Institute of Technology Delhi, New Delhi, India
shruti.pandey@ee.iitd.ac.in, {cs1221116, srsarangi}@cse.iitd.ac.in

***Abstract*—IR drop estimation is a classical problem in electronic design and automation (EDA), and is an integral part of the design signoff process. Even after decades of research on efficient circuit simulation and power grid modeling, IR drop estimation tools are still slow and often take hours to compute the IR drop profile even for moderate-sized designs. Given that it is hard to parallelize them, and often there is a need to run them iteratively thousands of times for design space exploration, there is a need for a much faster method, albeit with a minor reduction in accuracy. ML-based approaches have proven to be quite effective in this space; they are at least 10-100X faster. Sadly, they are quite inaccurate as of 2024. Given that they are oblivious to the underlying physics, the errors in predicting hotspots are undesirably high. In this paper, we propose a semi-analytical model AnaIR, which uses Green's functions to quickly arrive at an approximate IR-drop profile. It is further refined using a small U-Net based neural network to arrive at a much more precise estimate. It is 3.9× faster than competing work and is 32.6% more accurate.**

***Index Terms*—Green's function, IR drop prediction, Neural network, Electronic design automation, Machine learning**

I. Introduction

IR drop analysis is a vital aspect of an integrated circuit's design and sign-off process. With advancements in technology nodes, the density of transistors on a chip is increasing exponentially, while the supply voltage is decreasing. This combination leads to significant IR drops in circuits. Furthermore, thin metal wires in futuristic technology nodes result in a power delivery network with high resistance, exacerbating the IR drop issue [1]. IR drops are also a serious concern during the testing of an integrated circuit (IC). As a result, it may be necessary to conduct IR drop analyses thousands of times throughout the entire design and testing phases. However, IR drop simulation is computationally intensive and slow. With billions of nodes in a commercial design, computing the voltage at each node can take several hours. Hence, there is a pressing need to accelerate IR drop analysis. In recent studies conducted in the last three years,

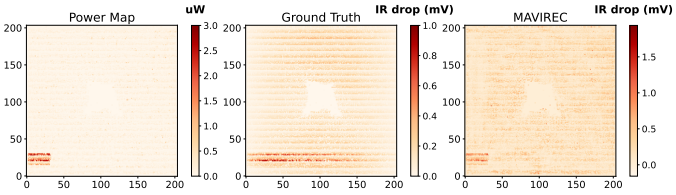

Fig. 1: IR drop prediction (heat map): power map, ground truth, MAVIREC [2] (state-of-the-art, note the inaccuracy)

machine learning (ML) has been used to predict IR drops with notable speedups [2]–[5]. These approaches typically convert IR drop prediction problems to image processing problems, where ML models take the feature maps derived from circuits as inputs and produce the corresponding IR drop maps as outputs. Generative networks such as U-Net [6] have performed well in such tasks. Although the results of these ML-driven methods are encouraging, these models are complex and inaccurate (see Fig. 1). They are entirely based on ML algorithms and are unaware of the underlying physics of IR drop simulation. Hence, these methods perform extensive feature engineering to embed circuit information in their models; for example, Chen et al. [4] extract 124 features per standard cell instance of a design, and Chhabria et al. [2] extract 24 spatial feature maps. Often, these features are correlated, making some of them redundant. Most of these works employ complex neural networks that require large datasets to realize good performance. In contrast, a recent study [5] shows good results with limited data and fewer feature maps. Sadly, their method fails to provide a speedup over traditional solvers. To the best of our knowledge, no prior work leverages the inherent properties of the PDN for faster IR drop prediction.

We shall observe that combining neural networks with an analytical approach based on Green's functions (the impulse response of a unit power source) can significantly speed up ML-based IR drop estimation. It can also make it more accurate. Our proposed tool, AnaIR, eliminates the need for extensive feature engineering and large datasets using such insights. To the best of our knowledge, this is the first work that applies Green's functions to speed up IR drop simulations. Furthermore, unlike competing methods [2], [5], [7], our approach is more explainable. The crux of our approach is to derive the weights of the first layer from the computed Green's function and then use additional layers in the DNN network to account for edge and corner effects, sparsity, orientation of the power grid and to make the method independent of the design size.

Our specific contributions are as follows.

❶ To the best of our knowledge, we are the first to propose a semi-analytical approach, *AnaIR*, for IR drop simulation. The weights of the first few layers is initialized with the Green's function. Our approach is analogous to physics-inspired neural networks (PINNs) [8].

❷ A subsequent fully convolutional encoder-decoder-based network *refines* the solution further.

❸ The Green's function relies on linearity and shift-invariance.

These properties do not hold at the rim of a chip. Hence, we propose a new feature namely the *layout mask* to take care of such effects.

❹ Our novel *AnaIR* scheme outperforms the existing state-of-the-art in terms of both speed and accuracy. AnaIR is **32.6**% more accurate, **3.9X** faster and has **33.3**% better structural similarity than the nearest competing solution MAVIREC [2].

The paper is organized as follows. Section II discusses the relevant background. We present the methodology in Section III followed by experiments and results in Section IV. Finally, we present the related work in Section V and conclude the paper in Section VI.

II. Background

In a digital IC, the Power Delivery Network (PDN) supplies current to the logic gates and memory elements in the design. "IR drop" refers to the voltage drop due to current flowing through the metal wires of the PDN. Excessive voltage drops lead to a slowing down of the circuits; if the gate is on the critical path, this will lead to a timing failure.

Most modern chips use a power grid with a mesh topology. They have supply and ground rings on the periphery of the chip. The PDN has vertical power stripes to transfer current to the horizontal power rails that further deliver power to standard cells. The power grid network consists of resistors and voltage/current sources. Essentially, traditional IR drop simulators solve network equations, Eq. 1, involving n number of nodes and have a time complexity of the order of $O(n^3)$.

$$\nabla \cdot (\sigma \nabla V(x,y)) = -I(x,y) \tag{1}$$

In Equation 1, we are modeling the circuit as a resistive mesh. ∇ is the Gradient operator for the 2-dimensional voltage field that is not time-varying (because we are doing static IR drop estimation), and $\nabla\cdot$ is the divergence. $I(x,y)$ is the current field and σ is a material/circuit dependent constant, which captures the resistivity. After getting the general solution, we can apply different boundary conditions.

A. Green's Functions

The Green's function ($g(x)$) is the impulse response of an inhomogeneous differential equation of the form $L(f(x)) = q(x)$, and is given by $L(g(x)) = \delta(x)$. Here, L is the linear partial differential operator and $\delta(x)$ is the Dirac delta function (refer to [9]). The Green's function can be used to solve any linear differential equation of the form $L(f(x)) = q(x)$. We show this below using the property of the convolution operator $\star$ and Eq. $L(g(x)) = \delta(x)$,

$$L(g(x) \star q(x)) = L(g(x)) \star q(x) = \delta(x) \star q(x) = q(x) \tag{2}$$

Therefore, $g(x) \star q(x)$ gives the solution to equations of the aforementioned form. This motivates us to use the Green's function to solve the IR drop differential equation 1.

III. Methodology

A. The Green's Function for IR Drop Simulation

1) Construction of the Green's Function: The Green's function is the impulse response (IR drop profile) of a unit power source. We compute the Green's function empirically (details in Section IV-A). Given a design, we apply power to a standard cell instance at the center of the chip to simulate an impulse input. We then run IR drop analysis for this impulse input and observe that the resulting IR drop map has a high value at the center with a localized spread across the horizontal power rail. This impulse response is an approximation of the Green's function. Fig. 2 shows the Green's function, where we plot the standard cell instances on a 2D map and represent their IR drop values using their color. The Green's function is robust and design-independent. Computing the Green's function is a one-time effort for a given power grid topology.

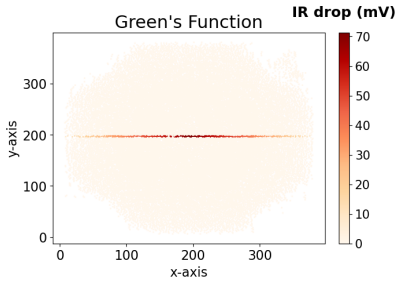

Fig. 2: An illustration of the Green's function

2) Properties of the Green's Function: Let us verify if our empirically derived Green's function follows the two key properties of a classical Green's function: linearity and shift invariance.

Linearity of the Green's Function: We perform experiments to verify the linearity and superposition properties of the Green's function. We apply a power source of 1.11 mW (denoted as x_1) and 2.22 mW (denoted as x_2, a scaled version of x_1) at an instance to obtain the IR drop maps y_1 and y_2, respectively. Finally, we apply a power source of 3.33 mW ($x = x_1 + x_2$) at the same instance to obtain the IR drop map y. We observe that y is equal to the sum of the individual IR drops y_1 and y_2 as shown in Fig. 3. Hence, the Green's function follows the superposition principle, as stated in Eq. 3 and is linear.

$$F(x_1 + x_2) = F(x_1) + F(x_2) \tag{3}$$

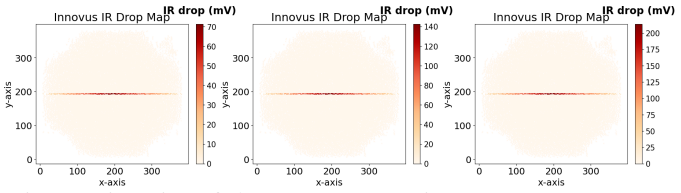

Fig. 3: Linearity of the Green's Function: y_1, y_2, $y = y_1 + y_2$

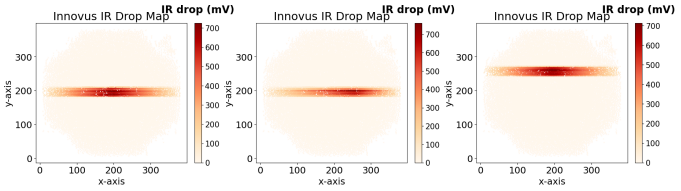

Fig. 4: Shift Invariance of the Green's Function

Shift Invariance of the Green's Function: We translate the input power source across the x and y directions and observe

the corresponding IR drops as shown in Fig. 4. These plots show that when the input power source is shifted along the x and y axes in broadly the central region of the chip, the Green's function remains (mostly) the same. Hence, the Green's function is empirically shift-invariant: if the input is shifted by Δ units, the output remains the same, albeit shifted by Δ units.

Behavior at the Rim: Let us evaluate the Green's function at the rim of the chip. We observe that the maximum IR drop value changes at different locations near the boundary, as shown in Fig. 5. Hence, the Green's function does not follow shift-invariance near the boundary regions of designs. A bespoke solution is needed to address this issue.

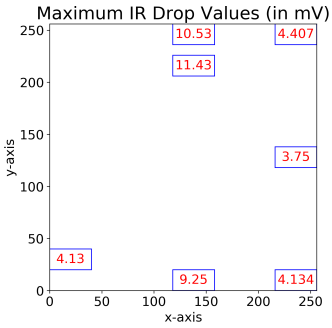

Fig. 5: Variation of the IR Drop values at the rim of the chip

Smearing by the PDN and the Green's Function: The PDN in our work has a mesh topology and the current is drawn from the power rings at the periphery. Since the centre is the farthest from the periphery, the largest IR drop occurs at the center due to the resistance of the long metal wires ($R \propto l$). Additionally, we observe that all standard cell instances placed in the same row draw power from the same horizontal rail. Consequently, a smear pattern in the IR drop is observed along the horizontal rail. Fig. 2 shows this smear pattern in the Green's function, demonstrating that the Green's function captures the intrinsic properties of the power grid through this pattern.

B. Feature Extraction

We pose IR drop simulation as an image-to-image translation problem where both the input and output are 2D images. Our scheme takes a power map image as input and predicts the IR drop map image. We also propose a new feature map, the *layout mask* (described later). We first extract the circuit-level data from the following sources:

- The layout information (x, y coordinates and sizes of the instances) from the post-route DEF file.
- Power values from the power analysis tool.
- IR drop values from static IR drop analysis.

We then parse and process this circuit-level information to create 2D spatial maps as described next. We divide the entire design layout into tiles of size $w_t \times h_t$. Given a design of size $W_d \times H_d$, we can divide it into $\left\lceil \frac{W_d}{w_t} \times \frac{H_d}{h_t} \right\rceil$ number of tiles, where each tile represents a feature value. We define the overlap ratio (O_a) of a standard cell instance with a tile as follows:

$$O_a = \frac{Area_{instance}}{Area_{tile}} \tag{4}$$

here, $Area_{instance}$ is the area of the standard cell instance from the LEF file that falls within the tile, and $Area_{tile}$ is fixed to $w_t \times h_t$.

We compute the contribution of each instance towards the power value of the tile by multiplying its overlap ratio with its power value. We then add the contributions of all the instances in the tile to get the power value of that tile. We follow the same procedure for the IR drop maps. We fix the tile size to $1.5\mu m \times 1.5\mu m$, similar to [3]. Our scheme takes the total power map as an input feature map.

The Layout Mask: We propose the *layout mask* as a feature for IR drop prediction. It captures the layout information of the design. We observed and determined empirically that the tiles where the overlap ratio is below 0.25 do not contribute much to the power and the IR drop map. Hence, we first create a spatial map in which each tile represents the sum of the overlap ratios of the instances within that tile. Next, we convert this spatial map to a Boolean mask (*layout mask*), where the tiles with an overlap ratio less than 0.25 are labeled as zero, and others are labeled as one. This is a layout-dependent activity and needs to be done only once. We shall show the importance of the layout mask in Sections III-D and IV.

Apart from the total power map p_t, we generate maps for internal power p_i, switching power p_s and toggle rate-scaled switching power p_{sca} for implementing the state-of-the-art approaches. We use min-max normalization to bring the values in the power and IR drop maps into the range [0, 1]. Since convolution is distributive over addition, convolving individual power maps with the Green's function is equivalent to convolving the total power map with the Green's function. Therefore, the total power map is sufficient for our approach.

C. Dataset Construction

We took four designs from OpenCores [10] to cover a wide range of functionalities and circuit sizes. We used two cryptographic cores (*aes_256* and *keccak* ($SHA-3$)), one DSP core (*LinkRunCCA*) to perform a "linked-list run-length single-pass connected component analysis" and an arithmetic core (*Cordic*). We used different frequency settings (50, 100, 150 and 200 MHz) to synthesize variations of these designs. We used the Cadence 45nm Generic PDK (supply voltage: 0.9 V) for synthesis and physical design.

TABLE I: Details of the circuits used in our experiments

Circuits	aes_256	LinkRunCCA	Cordic	Keccak	fa32_array
#Instances	3,25,359	54,871	3,479	25,963	27,107
#Pixels	732 × 732	463 × 463	117 × 117	256 × 256	204 × 204

Next, we create a synthetic dataset by creating an array of 10×10 independent 32-bit full adders. Each row of the 2D array is controlled by an *enable* signal; it is possible to generate diverse power maps programmatically. We generate power maps with striped patterns (horizontal and vertical). We vary the stripe widths in the range $[16, 80]$ in steps of 16 and the start location of the stripe in the range [0,151] in steps of 50 rows or columns. We apply power values out of [10, 30, 50, 75, 100] μWs to the instances that lie inside the

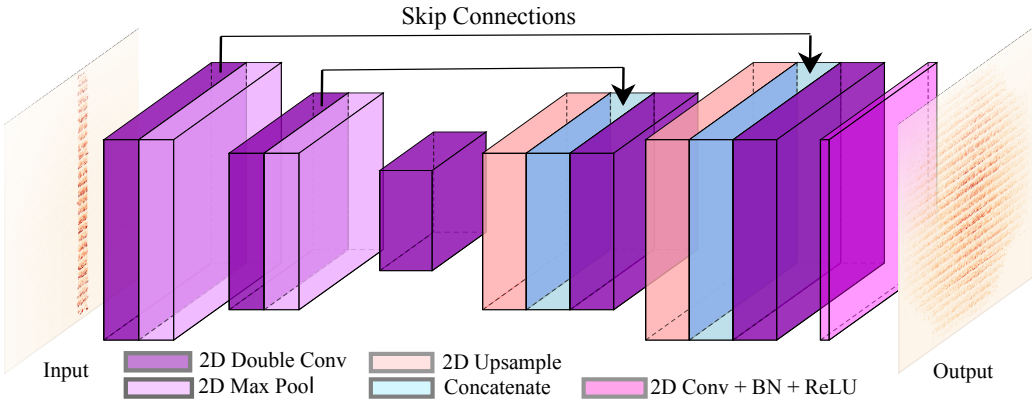

Fig. 6: AnaIR: An encoder-decoder based network embedded with the Green's function

stripes and set the power of other instances to zero. We use our custom design $fa32\_array$ and the $keccak\_150$ circuit to generate the training dataset. We found that for high power values (specifically, 75μW and 100μW), the IR drop values are more than the supply voltage; hence, these samples were discarded from the dataset. We also generated power maps at the edges/rims to study the corresponding IR drop maps. We supply power to small rectangular regions near the rims at various locations (top, bottom, left, right, centre). The difference in IR drop values due to these variations are shown in Fig. 5.

Finally, we get 352 power and IR drop map pairs. These samples were split in the ratio of 80 : 20 (the 20% of samples were varied) to get the training and validation datasets, respectively. We report the final results on the test circuits (different from the train and validation sets) to establish the generalizability of our scheme. Table I shows the number of standard cell instances, and the number of pixels in each design within the dataset.

D. The Green's Function for IR Drop Prediction

The Green's function acts as a rectangular kernel, denoted by f_{grect}, since it smears along the horizontal direction, as shown in Fig. 2. Further, as we show in Section III-A2, the system is linear and shift-invariant (for points not on the rim). For such points, Eq. 5 can be used to obtain an IR drop map (M_{IR}) for a given input power map (M_p),

$$M_{IR} = M_p \star f_{grect} \tag{5}$$

Here, $\star$ is the convolution operator. We compared the obtained M_{IR} map with the golden IR drop map and found that the Green's function provides an accurate approximation. It captures the smeared IR drop pattern of our PDN very well. The success of the Green's function can be attributed to its rectangular kernel, which corroborates the findings of recent work [5] that the selection of rectangular kernels for convolution is effective in the case of IR drop prediction. However, given its empirical nature, additional processing is required to further enhance the accuracy and make the solution work for the rim and sparse regions in the power map.

Furthermore, to remove noisy predictions in such sparse regions, we multiply the convolved map M_{IR} with the layout mask M_d (defined in Section III-B) to obtain the masked convolution map M^*_{IR} (see Eq. 6). This method was quite effective.

$$M^*_{IR} = M_{IR} \cdot M_d \tag{6}$$

In Eq. 6, M_d is a Boolean map that masks the areas of the convolved map with lower IR drop values. This basic IR drop map needs to go through further processing in a DNN for further accuracy enhancement.

E. AnaIR: Fusion of the Green's Function with Machine Learning

We propose a fully convolutional neural network, AnaIR, with initial weights derived from the Green's function. Our novelty lies in mapping the Green's function kernel to the deep learning model. By initializing the model weights with the Green's function, we embed the simulator's behavior into the network and eliminate the need to train the model from scratch.

AnaIR is an encoder-decoder network (as shown in Fig. 6) with weights initialized from the Green's function kernel. It takes the convolved and masked IR drop map (M^*_{IR}) as input, and the golden IR drop map as the ground truth. The latter is generated using a state-of-the-art commercial tool. AnaIR is inspired from the UNet architecture [6]. AnaIR is a *fully convolutional network* (FCN) and thus can handle different input feature sizes. It takes 2D input maps of any size and gives the output of the same size as that of the input.

The primary layer of AnaIR consists of a 2D convolution layer followed by BatchNorm and ReLU layers. Two such primary layers are stacked to form a double convolution layer. In the encoder network, we use two pairs of double convolution and max pooling layers. The symmetrical decoder network contains two pairs of upsample and double convolution layers (refer to Fig. 6). The encoder network learns where features are located in the image by reducing feature dimensions and increasing the number of channels, while the decoder network learns what the image contains by expanding feature maps and reducing the number of channels.

Each layer of the encoder and decoder networks is connected via skip connections, which provide fine-grained details about the input image to the decoder network. Finally, note that

TABLE II: Comparison of AnaIR with ML-based IR Drop Predictors

Metrics/Methods	IRPNet	MAVIREC	XGBoost	Convolution	Convolution + Masking	**AnaIR**
MAPE (%)	409	125.9	348.4	286.73	203.4	**93.25**
SSIM	0.17	0.57	0.38	0.14	0.28	**0.76**
Inference Time (ms)	25.82	161.92	101	2.9	3.06	**41.5**

AnaIR has fewer layers than other state-of-the-art approaches ([2], [5]). AnaIR's smaller size leads to fewer parameters and faster training times.

IV. Evaluations and Results

A. Setup

We use Cadence Genus v19.12 to synthesize the digital circuits and Innovus v20.14 for the physical design (including placement and routing). Cadence Voltus v20.14 runs the power and IR drop analysis. The codebase from the CircuitNet [3] paper was modified to parse the post-route DEF; the power and IR drop reports were generated using Voltus. As a reference, we used the *Keccak_*150 design to derive the Green's function; however, any other design can also be used. We use an Nvidia RTX A6000 GPU and Pytorch v1.11.0 framework to train and infer all the ML models. We found an optimal learning rate for the AdamW optimizer to be 1×10^{-4} through experiments. It is safe to assume that commercial tools are the best-known algorithms in the conventional space. They are our *ground truth*.

B. Evaluation Metrics

We use the structural similarity (SSIM), & mean absolute percentage error (MAPE) as our evaluation metrics for image similarity and IR drop value prediction, respectively. SSIM is computed over tiles of pixels; SSIM between the windows of pixels x and y is defined in Eq. 7. SSIM values lie in the range $[0,1]$, and a higher value is considered better. MAPE is defined in Eq. 8. MAPE is a superior metric compared to mean absolute error (MAE) as it computes the per-pixel relative error; hence, we use it. We use the convolve function of the Scipy library to perform the convolution of the Green's function with the power map.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{7}$$

Here, μ and σ denote the mean and the standard deviation of the pixels in the image window (resp.).

$$MAPE = \frac{1}{n}\sum_{i \in p}\left|\frac{y_{gold_i} - y_{pred_i}}{y_{gold_i}}\right| * 100 \tag{8}$$

Here, y_{gold} and y_{pred} represent the ground truth and predicted IR drop maps, respectively, and p is the set of pixels where y_{gold} is non-zero.

C. Results

We compare our proposed scheme with the most recent and competing state-of-the-art ML-based predictors. We keep the training and test dataset consistent across all three competing methods. We implement their proposed algorithms with the same set of hyperparameters as provided in the respective papers. We implement the IRPNet architecture from the most recent state-of-the-art paper [5]. We fine-tune the pre-trained model provided by Circuitnet [3] for reproducing MAVIREC [2]. We also implement the XGBoost model from [4], where we compare AnaIR with a traditional ML method. We evaluate these models along with our proposed Green's function inspired scheme on the test dataset comprising real circuits (Table I).

Table II shows the comparison of MAPE, SSIM and inference time of all the methods. We show an improvement of **32.6**% in accuracy and **33.3**% in SSIM while being **3.9**$\times$ faster than the nearest competing work [2]. Note that even though IRPNet is the fastest of all methods, it is highly inaccurate. Moreover, masking of the convolved map reduces the error by 83.3% and doubles the SSIM without adding any overhead in terms of execution time, as shown in Table II. Hence, the experiments show that the layout mask is a vital input feature.

We also show a comparison of MAPE and SSIM of all the methods for every test circuit in Fig. 7. Fig. 7 shows that AnaIR outperforms all the existing work through consistent improvement in both the metrics across all the test circuits.

Fig. 8 shows the predicted IR drop maps from AnaIR, masked convolution with the Green's function and results generated by competing works for the *Cordic_200* circuit. We also plot the respective absolute error maps w.r.t. the ground truth maps (generated from Voltus). We observe that AnaIR's IR drop map looks the most similar to the ground truth map, justifying its significant improvement in both MAPE and SSIM. Just masking has a high MAPE; hence, additional neural network layers were required. We experimented with different *convolution neural networks* (CNNs): a CNN with 2-3 layers, Pyramid Scene Parsing Network (PSPN) [11] and UNet [6]. Among all these networks, UNet performed the best (backbone of AnaIR).

V. Related Work

Our primary focus was on recent work (published in the last three years) in the area of ML-based static IR drop prediction. In 2021, Chhabria et al. proposed IREDGe [7], a generative encoder-decoder-based neural network that takes power maps, the power grid, and power pad distributions as inputs to predict static IR drops. Later in the same year, Chhabria et al. [2] introduced a 3D UNet architecture, MAVIREC, which takes per-instance power and effective resistance values, along with their spatial maps, as input features to predict per-instance IR drop values. A year later, Chen et al. [4] proposed using an XGBoost model for per-instance IR drop prediction. This work leverages 72 features, including circuit information, power, and toggle information for each instance. All these works are transferable across designs synthesized in the same technology. A recent work [12] incorporates attention modules in the

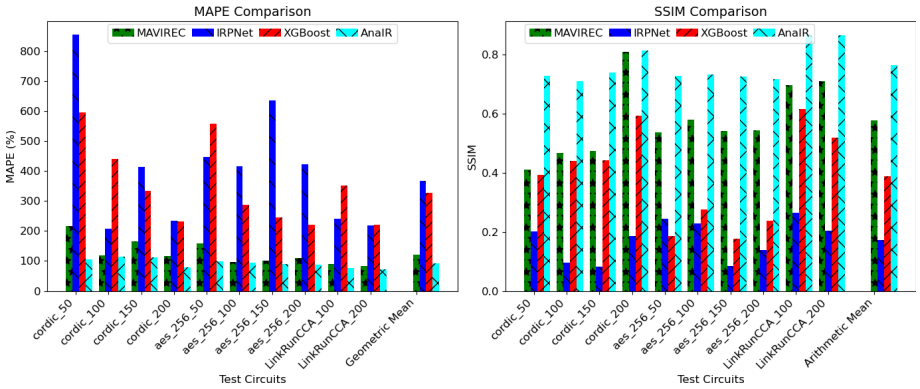

Fig. 7: MAPE and SSIM for the Test Circuits using AnaIR and Different ML-based IR Drop Predictors

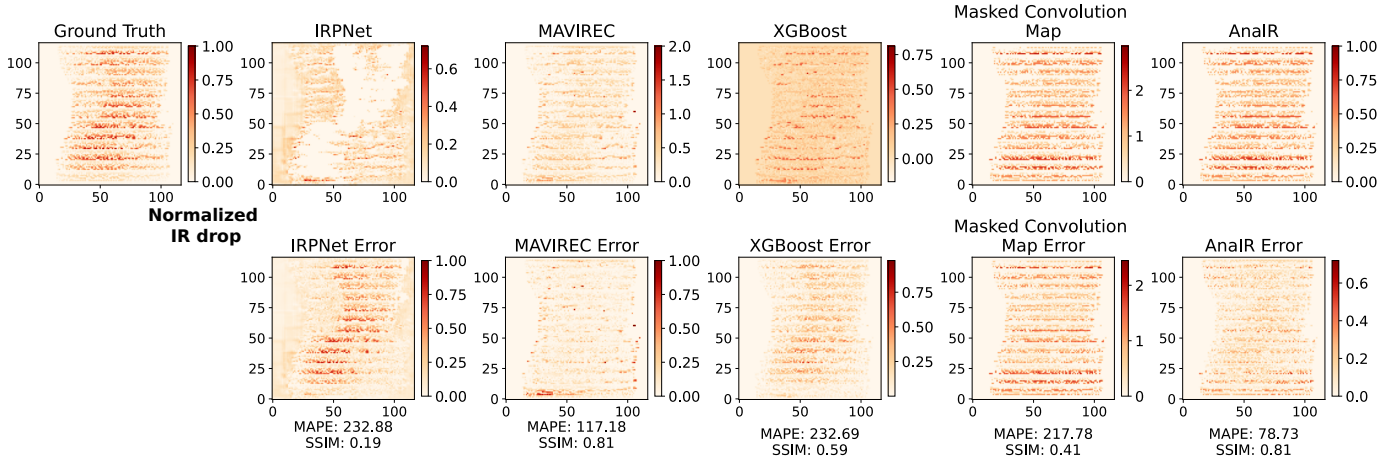

Fig. 8: Comparison of AnaIR's Prediction with the Existing ML-based Predictors for the Cordic Circuit

UNet architecture and performs feature selection. However, incorporating the attention module increases the computational overhead, making it much slower with no or minimal gain in accuracy. Hence, we do not evaluate this work. These ML-based approaches require large datasets and extensive feature engineering, quite unlike AnaIR. Very recently Meng et al. [5] incorporated the Kirchhoff's current law into the loss function to enable neural network training with limited data. This work proposed a pyramid scene parsing network that takes current, PDN density, and via maps as inputs. All these works rely entirely on ML to obtain IR drop values without accounting for the intrinsic properties of the power grid. Thus, there is a need for IR drop physics-aware analytical methods that are more robust, accurate, and explainable.

VI. Conclusion

We propose AnaIR, a semi-analytical Green's function-inspired neural network for static IR drop prediction. Our novelty lies in processing the power map analytically with the Green's function and the layout mask, and then refining it further with a neural network initialized with the Green's function. AnaIR is 32.6% more accurate, has 33.3% better SSIM and is 3.9× faster than the nearest competing proposal [2]. We conclude that using the Green's function for IR drop simulation is more explainable and accurate than other deep learning-inspired approaches.

References

[1] SemiEngineering, "Ir drop," https://semiengineering.com/knowledge_centers/low-power/architectural-power-issues/ir-drop/, 2024.

[2] V. A. Chhabria, Y. Zhang, H. Ren, B. Keller, B. Khailany, and S. S. Sapatnekar, "Mavirec: Ml-aided vectored ir-drop estimation and classification," in *2021 DATE*. IEEE, 2021, pp. 1825–1828.

[3] 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 TCAD*, vol. 42, no. 12, pp. 5034–5047, 2023.

[4] J.-X. Chen *et al.*, "Vector-based dynamic ir-drop prediction using machine learning," in *2022 ASP-DAC*. IEEE, 2022, pp. 202–207.

[5] Y. Meng *et al.*, "Circuits physics constrained predictor of static ir drop with limited data," in *2024 DATE*. IEEE, 2024, pp. 1–2.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI 2015 proceedings, part III 18*. Springer, 2015, pp. 234–241.

[7] 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 *ASP-DAC*, 2021, pp. 690–696.

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations," *arXiv preprint arXiv:1711.10561*, 2017.

[9] H. Sultan and S. R. Sarangi, "Varsim: A fast process variation-aware thermal modeling methodology using green's functions," *Microelectronics Journal*, vol. 142, p. 105995, 2023.

[10] "Opencores," https://opencores.org/, 2024.

[11] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890.

[12] F. Guo, J. Liu, J. Zhai, J. Jia, K. Zhao, and C. Shi, "Pgau: Static ir drop analysis for power grid using attention u-net architecture and label distribution smoothing," in *2024 GLSVLSI*, 2024, pp. 452–458.