

GIRD: A Green IR-Drop Estimation Method

Chee-An Yu¹, Yu-Tung Liu², Yu-Hao Cheng², Shao-Yu Wu², Hung-Ming Chen², C.-C. Jay Kuo¹

¹ University of Southern California, Los Angeles, USA

² National Yang Ming Chiao Tung University, Hsinchu, Taiwan

Abstract—An energy-efficient high-performance static IR-drop estimation method based on green learning called GIRD (Green IR Drop) is proposed in this work. GIRD processes the IC design input in three steps. First, the input netlist data are converted to multi-channel maps. Their joint spatial-spectral representations are determined with PixelHop. Next, discriminant features are selected using the relevant feature test (RFT). Finally, the selected features are fed to the XGBoost (eXtreme Gradient Boosting trees) regressor. Both PixelHop and RFT are green learning tools. GIRD yields a low carbon footprint due to its smaller model sizes and lower computational complexity. Besides, its performance scales well with small training datasets. Experiments on synthetic and real circuits are given to demonstrate the superior performance of GIRD. The model size and the complexity, measured by the Floating Point Operations (FLOPs) of GIRD, are only 10^{-3} and 10^{-2} of deep-learning methods, respectively.

Index Terms—IR-drop Estimation, Green Learning, Machine Learning, EDA

I. INTRODUCTION

The IR (or voltage) drop problem occurs as integrated circuits (ICs) become more complex and compact. The power delivery network (PDN) cannot deliver the target voltage (or current) level to each cell due to the physical limitation of narrower wire widths. Significant IR drops can hurt the performance and reliability of chips. Analyzing PDN and IR drop is essential in modern complex IC design. Designers can model PDN as an interconnected system of resistors and current/voltage sources to examine the IR drop phenomenon using Kirchhoff's current and voltage laws (i.e., KCL and KVL). Solving thousands of linear equations is computationally challenging and time-consuming as the circuits become more complicated. The use of machine learning to estimate the IR drop offers an alternative. We face several significant challenges in this direction. First, since the intellectual properties (IP) of ICs are valuable in practical applications, the number of training data is typically small. Second, state-of-the-art machine-learning tools are primarily built upon deep neural networks (DNNs) called deep learning. Deep learning solutions demand large model sizes, high computational resources, and substantial training datasets. They yield a high carbon footprint. Besides, they are mathematically opaque.

A new machine learning methodology called green learning [1]–[3] has been proposed to address the shortcomings of deep learning in recent years. Green learning has been successfully applied to several tasks in image processing [4], computer vision [5], and natural language processing [6]. This is the first time green learning has been applied to computer-aided IC design. Besides energy efficiency, it scales well with a small training dataset. Since our work is based on the green learning

paradigm, we name it the green IR-drop (GIRD) estimation method.

Our GIRD method processes the IC design input in three steps. The input netlist data are first converted to multi-channel maps. We derive their joint spatial-spectral representations with PixelHop. Afterward, we select discriminant features automatically using the relevant feature test (RFT) [7]. Finally, we feed the features chosen to the XGBoost (eXtreme Gradient Boosting trees) regressor. PixelHop and RFT are tools developed in green learning literature. Our solution yields a low carbon footprint because of smaller model sizes and lower computational complexity. Besides, its performance scales well with small training datasets. Experiments on synthetic and real circuits are given to demonstrate the superior performance of the proposed solution. The model size and the complexity, measured by the Floating Point Operations (FLOPs), of our method are only 10^{-3} and 10^{-2} of deep-learning methods, respectively.

The rest of this paper is organized as follows. Section 2 reviews related previous work on IR drop estimation and gives a brief overview of green learning. Section 3 briefly overviews the green learning system and its modules. Section 4 presents the green-learning-based IR drop estimation method. Experimental results are shown in Section 5. Finally, Section 6 offers concluding remarks and points out possible future extensions.

II. REVIEW OF RELATED WORK

Generally, IR drop problems can be categorized into two types: dynamic and static IR drop. Dynamic IR drop arises from transient currents when transistors switch. As more transistors switch simultaneously, dynamic IR drop becomes severe. Static IR drop is caused by the wire's impedance, demonstrating PDN's weakness. Static IR drop can be mitigated by improving the PDN structure. We can enhance the PDN structure more efficiently through IR drop estimation analysis. It is our main focus. We will review the related work on static IR drop estimation below.

A. Classical IR Drop Estimation

Traditionally, people have proposed several different metrics to estimate IR drop value. For example, weighted switching activity (WSA) [8], [9], switching cycle average power (SCAP) [10], and flip-flop toggle count(FFTC) [11]. Although these metrics demonstrate good correlations with IR drop values, finding a transfer function to evaluate the actual IR drop values is difficult. Then, people use a linear model [12] or

the modified nodal analysis (MNA) [13] to model PDN and IR drop for each cell node. PDN comprises resistors and current and voltage sources. One can derive a linear system of equations using KCL and KVL:

$$\mathbf{G}\mathbf{x} = \mathbf{J} \quad (1)$$

where \mathbf{G} is an $N \times N$ matrix to model the conductance of PDN, and \mathbf{x} and \mathbf{J} are $N \times 1$ vectors denoting cells' voltage and current value. By solving (1), we can find the voltage value of each node. As the IC design becomes more complex and compact, N can reach the order of millions. The solution of (1) is computationally expensive with millions of PDN nodes [14]. Machine learning techniques have recently received attention in the electronic design automation (EDA) field [15]–[18]. People turn to data-driven optimization as an alternative to accelerate the EDA process.

B. Learning-based IR Drop Estimation

Researchers have applied machine learning to static and dynamic IR drop problems [19]–[31]. Based on domain knowledge, people choose feature vectors manually and feed them to regressors, such as XGBoost, artificial neural networks (ANN), convolutional neural networks (CNN), and support vector machine (SVM) to estimate static or dynamic IR drop. For example, Ye *et al.* [29] applied a support vector machine to predict IR drop. They only used input patterns without feature extraction to estimate the IR drop value. Therefore, it is not scalable to large circuit designs due to the high dimensional input vector. Lin *et al.* [20] used ANN to predict static IR drop before engineering change order (ECO). They chose 7D feature vectors, including cell power, toggle rate, type, location, neighbor toggle rate, count, and distance-to-via. Ho *et al.* [21] designed features such as the width and length of the Chip/Block, pitch information from the layout geometry, and pulldown and pullup components of nodes and their neighbors. Then, they employed XGBoost as a regressor to predict IR drop results. Xie *et al.* [25] represented selected features as spatial maps and proposed a CNN-based encoder followed by a fully connected layer to estimate IR drop. Viday *et al.* [30] proposed an encoder and decoder architecture to formulate the IR drop estimation problem as an Image-to-image translation problem. They applied U-Net architecture as the encoder and decoder to extract spatial information and predict IR drop and temperature contour information. There is work on dynamic IR drop. For example, dynamic IR drop values were predicted using the CNN encoder/decoder in [25], [26], [32]. The electromigration-induced dynamic IR drop was analyzed in [22], [33]. Although learning-based methods offer good results, they demand powerful computational platforms like GPU. Furthermore, the end-to-end training adopted by deep learning makes the system a black box, and its high carbon footprint makes DL models environmentally unfriendly. Here, we aim at a green and interpretable learning solution for static IR drop estimation.

III. GREEN LEARNING

Green learning is a new paradigm with a low carbon footprint and mathematical transparency [1]. It features a small model size, low computational complexity, logical transparency, and model scalability. It provides an energy-efficient solution for cloud servers and mobile/edge devices. Green learning models differ from deep learning models entirely as they have no neurons or networks. They are modularized and trained in a feedforward manner without backpropagation. Green learning has been successfully applied to various applications such as image classification [7], [34], [35], texture analysis [36], [37], and fake image detection [38]–[41]. This work tackles the static IR drop estimation problem with green learning. The application of green learning to EDA is new.

The green learning paradigm comprises three modules: 1) representation learning, 2) feature learning, and 3) decision learning. All intermediate results are explicit and explainable. Fig. 1 shows the three modules of the entire pipeline. Each module is briefly explained below. For more details, we refer to [1]



Fig. 1. An overview of the green learning system.

A. Representation Learning via PixelHop and Saab Transform

The Saab transform is an unsupervised representation learning method to decompose source data into DC and AC subspaces. It is achieved through principle component analysis (PCA). The kernel weights are automatically determined from training data, the primary difference between handcrafted features in traditional pattern recognition. The channel-wise Saab transform (c/w Saab) is a variation of the Saab Transform. Due to the orthogonality of DC and AC subspace, the correlation between channels becomes very weak. Compared to the Saab transform, the c/w Saab transform is more computationally efficient and has fewer parameters. Through Saab or c/w Saab, we acquire more concise representations of DC and AC components. We provide more mathematic derivation and comparisons in Appendix B.

PixelHop unit (PU) applies saab and c/w saab with maximum pooling in a cascade way to consider spatial information in different resolutions. The local and global information are captured from various levels of the PU. The upper PUs offer more local information, while the lower PUs provide global information. A threshold is chosen to classify channels into two subgroups: leaf nodes and intermediate nodes. If the energy of nodes is larger than the threshold, these nodes, called intermediate nodes, are passed to the next level of PU. Otherwise, the remaining nodes, called leaf nodes, stayed at the current level. The illustration of PUs with different spatial resolution is shown in Fig. 2.

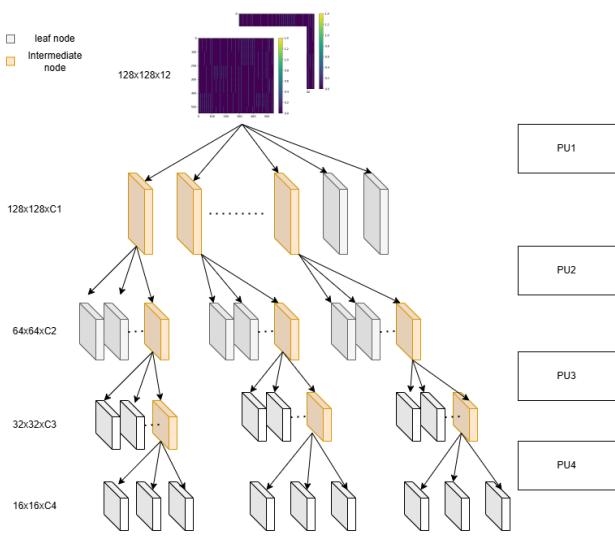


Fig. 2. Illustration of the c/w Saab Transform and PixelHop Unit.

B. Feature Learning via RFT

A regressor maps input features to a target scalar function. The relevant feature test (RFT) [7] exploits the supervision labels to select more relevant features from the candidate feature set. Inspired by binary decision trees, RFT divides a feature dimension into left and right intervals and assesses the total mean square error (MSE) resulting from this division. We apply this approximation error as an RFT loss function, where features with a lower RFT value indicate more influential ones. We can identify more influential feature dimensions for regression tasks through the RFT. The overview of RFT is shown as figure 3.



Fig. 3. Illustration of the relevant feature test (RFT) procedure [7]

C. Decision Learning via XGBoost

Extreme Gradient Boosting Tree (XGBoost) builds an ensemble tree sequentially. Unlike traditional ensemble methods that combine weak learners to form a strong learner, gradient gradient-boosting tree enhances the model by iteratively adding new trees to correct errors made by previous ones. The process begins with an initial prediction with the mean value for regression tasks or the log(odds) for classification tasks. XGboost calculates the current predictions concerning the loss function's gradients or residual errors in each iteration. A new decision tree is then trained to minimize the residual errors. The predictions are updated by adding weighted predictions from the new tree to the existing ensemble results. Some parameters, such as the depths of trees and the number of trees, determine the model size and the computational complexity.

IV. PROPOSED GIRD METHOD

An overview of the proposed GIRD system is illustrated in Fig. 4. It consists of four modules: 1) input pre-processing, 2) representation learning, 3) feature learning, and 4) decision learning. Module 1 is tailored to the static IR drop estimation problem, while Modules 2-4 are building blocks of green learning. The GIRD system has a fine-to-coarse architecture for representation learning and a coarse-to-fine architecture for decision learning. At first glance, the GIRD structure resembles the U-Net architecture in the neural net literature [42]. However, we emphasize that GIRD does not have neurons or backpropagation-based training. All parameters in each processing unit are determined in a feedforward fashion. They are mathematically transparent, and the training can be conducted efficiently.

A. Module 1: Input Pre-processing

The netlist data files serve as the input source. We parse them into multi-channel 2D maps of four types: 1) current source map (i.e., locations of current sources), 2) voltage distance map (the effective distance to the voltage source), 3) PDN density map, and 4) multi-layer resistor maps. Our study has 12 maps: one current source map, one voltage distance map, one PDN density map, and nine layered resistor maps. All maps are resized to the same spatial resolution of 128×128 pixels. All pixel values are re-scaled to an integer between [0,255]. Section A of Appendices provides the visualization of the 12 input maps.

B. Module 2: Representation Learning

The representation learning module converts each spatial map to joint spatial-spectral representations of different trade-offs. Four spatial-spectral representations are derived using four cascaded PixelHop units (PU), as shown in Fig. 4. The input and output spatial resolutions of each PU are denoted by D_{in} and D_{out} , respectively.

- **PU1.** $D_{in} = 128 \times 128$, $D_{out} = 64 \times 64$.
- **PU2.** $D_{in} = 64 \times 64$, $D_{out} = 32 \times 32$.
- **PU3.** $D_{in} = 32 \times 32$, $D_{out} = 16 \times 16$.
- **PU4.** $D_{in} = 16 \times 16$, $D_{out} = 16 \times 16$.

Three channel-wise (c/w) Saab transforms [43] of kernel sizes (3×3 , 5×5 , 7×7) with stride equal to one are applied to PUs of different spatial resolutions to derive a rich set of joint spatial-spectral representations from the input. There is a (2×2) -to- (1×1) maximum absolute pooling operation after PU1, PU2, and PU3. The c/w Saab transforms produce c/w Saab coefficients, offering rich spatial-spectral representations. We review the c/w Saab transform in Section B of Appendices.

C. Module 3: Feature Learning

By leveraging supervising labels, we select effective representations from the whole c/w Saab coefficients as features. This process is called the relevant feature test (RFT) [7]. We use the RFT curve to rank the discriminant power of all representation dimensions. The lower the RFT loss value, the better the representation dimension. We sort the RFT loss

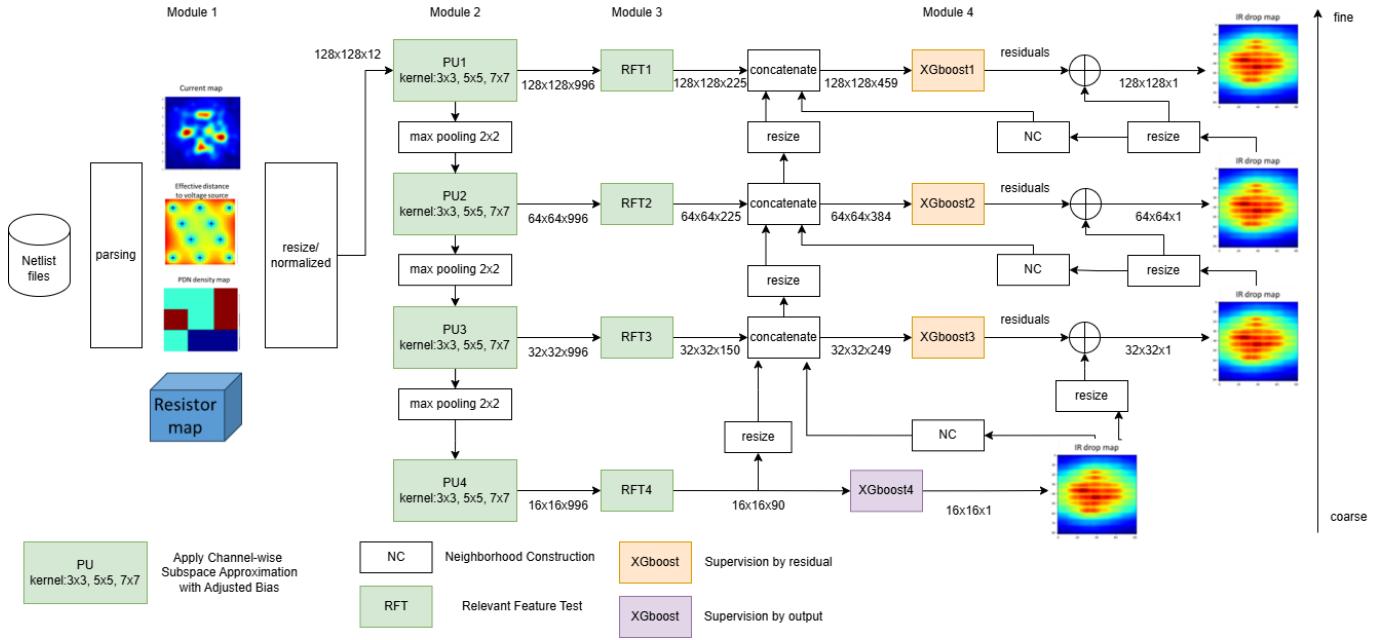


Fig. 4. The diagram of the green IR drop estimation system. First, PixelHop utilizes the c/w Saab transform of three kernel sizes to extract rich spatial-spectral representations from the input. Next, RFT selects discriminant features. The result from the previous coarse resolution with NC is concatenated as a feature to guide XGBoost in predicting residuals. The IR drop ground truth supervises the purple-boxed XGBoost. IR drop residuals supervise the orange-boxed XGBoost units.

value from the lowest to the highest and observe an elbow point of the RFT curve. Those dimensions before the elbow point are selected as the discriminant features. The result of the RFT training is shown in Figure 5. The MSE value of each feature from different kernel sizes is sorted in ascending order. They are used to predict the IR drop more efficiently and accurately. We will review the RFT process in detail in Section C of Appendices.

D. Module 4: Decision Learning

After feature extraction by Saab and RFT, we apply XGBoost [44], which is a tree structure, to predict the final result. In this work, we supervise XGBoost in two ways. We directly supervise XGBoost 4 by IR drop value for the coarse resolution. However, we supervise the XGBoost 1, XGBoost 2, and XGBoost 3 by residuals for the middle and high resolution. The residual **Res** can be calculated by IR drop ground truth at current resolution $\mathbf{V}_{\text{drop_cur_res}}$ and the interpolation result of IR drop prediction at the previous resolution $\hat{\mathbf{V}}_{\text{drop_prev_res}}$ as follows:

$$\mathbf{Res} = \mathbf{V}_{\text{drop_cur_res}} - \text{Interpolation}(\hat{\mathbf{V}}_{\text{drop_prev_res}}) \quad (2)$$

Therefore, the IR drop estimation $\hat{\mathbf{V}}_{\text{drop_cur_res}}$:

$$\hat{\mathbf{V}}_{\text{drop_cur_res}} = \text{Interpolation}(\hat{\mathbf{V}}_{\text{drop_prev_res}}) + \hat{\mathbf{Res}}_{\text{cur_res}} \quad (3)$$

The result from the previous coarse resolution with neighborhood construction (NC) is concatenated as a feature to guide XGBoost in predicting residuals. The IR drop ground truth supervises the purple-boxed XGBoost unit, while IR drop residuals supervise the orange-boxed XGBoost units.

The idea behind predicting residual instead of ground truth is model architecture's flexibility. With more refining residual

predictions, adding high-frequency components can gradually boost the performance.

To boost the performance of residual predictions, we designed NC to encode more spatial information based on the previous prediction as guidance. For each pixel in the last prediction, we select its k nearest neighbors ($k=3$, for example) to describe the spatial information. After resizing, we acquire $32 \times 32 \times 9$, $64 \times 64 \times 9$ and $128 \times 128 \times 9$ NC feature maps for XGBoost 3, XGBoost 2, and XGBoost 1, respectively. The example of NC is illustrated in Figure 6. The NC features are concatenated with current RFT features and previous resized RFT features. For instance, the input dimension of XGBoost 3 is $32 \times 32 \times (150 + 90 + 9)$. The details of the input dimension of each XGBoost are summarized in Table I.

TABLE I
SUMMARY OF INPUT DIMENSION FOR EACH XGBOOST

XGBoost	spatial resolution	RFT	NC	Input dim
XGBoost 1	128×128	225	9	$128 \times 128 \times 459$
XGBoost 2	64×64	225	9	$64 \times 64 \times 384$
XGBoost 3	32×32	150	9	$32 \times 32 \times 249$
XGBoost 4	16×16	90	-	$16 \times 16 \times 90$

V. EXPERIMENTS

The experiments were run on the AMD EPYC 7543 32-core CPU and Nvidia RTX A6000 GPU with Ubuntu 22.04 and CUDA 12.04.

Datasets. The BeGAN dataset [45] and the data released by the 2023 ICCAD competition [46] were used for training and testing in our experiments. BeGAN [45] is a public PDN benchmark dataset in SPICE netlist form using three technologies. GANs generate their current maps, and OpenROAD and

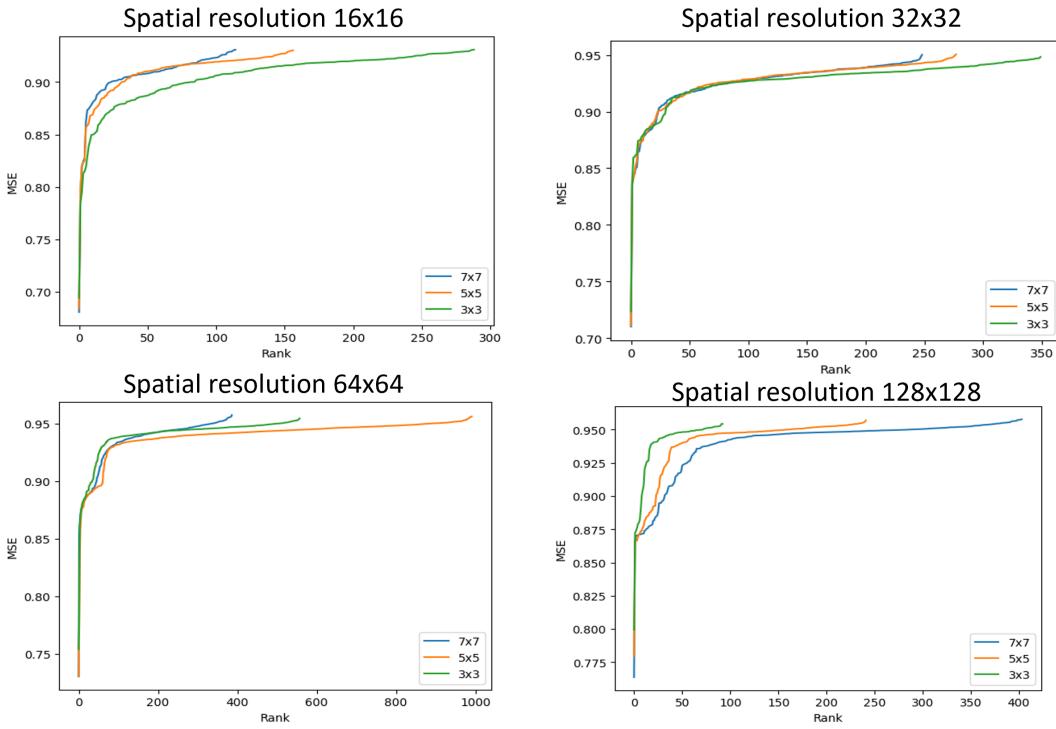


Fig. 5. Illustration of RFT curves at different spatial resolutions, where three kernel sizes are applied to combine multiple information from different receptive fields. Dimensional reduction is achieved by selecting top-rank features with lower MSE.

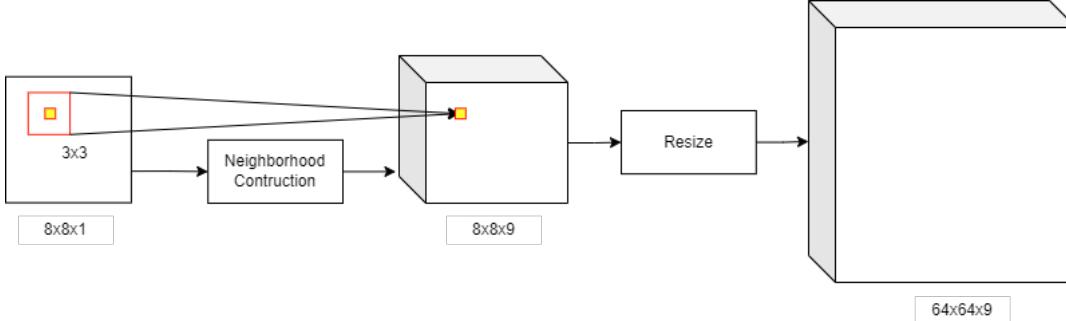


Fig. 6. Illustration of the neighborhood construction (NC) process.

OpeNPDN generate their PDNs. BeGan dataset in FreePDK45 has two sets, Set1 and Set2, with the Open Cell Library (NanGate 45nm). Both are synthetic datasets consisting of 1000 designs. Set1 has no macros, while Set2 has macros. For the synthetic data, we combine Set1 and Set2 synthetic designs from BeGAN (NanGate 45nm) and the released data from the ICCAD 2023 competition. We partition the combined dataset with 80% for training and 20% for testing. Also, we utilize ten real circuit designs and ten real hidden testing data released by the ICCAD 2023 competition to validate our method. The sizes of the datasets are summarized in Table II. The configuration of different amounts of training data is summarized in Table III.

Prediction Performance We measure the predicted IR drop error in six metrics: 1) the mean absolute error (E_{mean}) (4), 2) the maximum absolute error (E_{max}) (5), 3) the mean ratio between predicted and ideal IR values (R_{mean}) (6), 4) the

TABLE II
SIZE OF DATASETS FROM THREE SOURCES

name	ICCAD 2023	BeGAN set1	BeGAN set2
size	100	1000	1000

TABLE III
DATASET CONFIGURATION FOR DIFFERENT AMOUNTS OF TRAINING DATA

# training set	100	200	400	800
ICCAD 2023	50	100	100	100
BeGAN set1	25	50	150	350
BeGAN set2	25	50	150	350

ratio of the maximum IR value (R_{max}) (7), 5) the correlation coefficient (CC) (8) between prediction and ideal IR value, and 6) Normalized Root Mean Square Error (NRMSSE) (10). To demonstrate the robustness of GIRD against smaller training data sizes, we report the five metrics on synthetic and real

circuit designs in Table IV and Table V, respectively. We demonstrate that there is no significant performance drop as the amount of training data decreases, which shows the robustness of the proposed method in the small data.

$$E_{mean} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4)$$

$$E_{max} = \max_{i=1,\dots,n} |y_i - \hat{y}_i| \quad (5)$$

$$R_{mean} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n y_i} \quad (6)$$

$$R_{max} = \frac{\hat{y}_i}{\max_{i=1,\dots,n} y_i} \quad (7)$$

$$CC = \frac{\sum_{i=1}^N [y_i - \text{mean}(y)][\hat{y}_i - \text{mean}(\hat{y})]}{\sqrt{\sum_{i=1}^N [y_i - \text{mean}(y)]^2} \sqrt{\sum_{i=1}^N [\hat{y}_i - \text{mean}(\hat{y})]^2}} \quad (8)$$

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

$$NRMSE = \frac{RMSE}{\text{mean}(y)} \quad (10)$$

We see the prediction performance boosts as the spatial resolution increases. The training and validation curves of XGBoost 1 with different amounts of data are shown in Fig. 7. We can see that the training MAE and validation MAE are very close and converged, which means there is no overfitting or underfitting between the training and validation set. Even though the amounts of data decrease from 800 to 100, the training set and validation set curves still approach each other, showing the robustness of the small amount of data.

In Table VI, we benchmarked ten real circuit designs in the ICCAD 2023 competition on static IR drop estimation [47]. All participants (25 teams) ran their algorithms on the same datasets, and the performance of the top 3 results is shown on the website. We compared our mean absolute error (MAE) results to validate our method's performance. Since the proposed GIRD method is modular, we can adopt other classical machine learning methods in its decision-learning module. For example, we replace the XGBoost regressor in GIRD with two other regressors: the support vector machine (SVM) and the Random Forest (RF) regressors. We used a radial basis function kernel for SVM with an epsilon value of 0.1. For RF, we use 2,000 trees with a maximum depth of 3. The MAE value is in the unit of 10^{-2} (mV). For each test case, the lowest MAE is in bold, while the second and the third ones are underlined. Generally, our green learning method can achieve performance in the top 3.

Although the top three performances for individual test cases were released, the exact techniques and their training sets were not disclosed. The performance of GIRD with various regressors is among the top three except for Test Cases 9 and 19. For the worst case, Test 19, there is still a performance gap between the top 3. We examined the input and training

data to understand the current limitations. We observed that the PDN map changed frequently in the spatial domain in Test Case 19. However, this high-frequency and complicated PDN structure is out of the distribution of our training sets. To address this issue, we should increase the variety of samples in our training data, including more complicated PDN structures in our training sets. Then, our green learning method could achieve lower MAE results. The visualization of each test case is shown in Fig. 8. The visualization results from coarse to fine spatial resolutions of the real and synthetic data are shown in Fig. 9.

TABLE IV
EVALUATION INDEXES ON SYNTHETIC DESIGNS

# training set	100	200	400	800
E_{mean} (mV)	0.1574	0.1672	0.1377	0.1438
R_{mean}	15.4%	16.2%	13.3%	13.4%
E_{max} (mV)	1.64	1.69	1.15	1.27
R_{max}	17.8%	17.8%	12.3%	15.4%
CC	0.82	0.81	0.86	0.84
MSE (10^{-8})	6.4891	5.6098	4.9326	4.6702
R^2	0.5807	0.7338	0.7441	0.7333

TABLE V
EVALUATION INDEXES ON REAL DESIGNS

# training set	100	200	400	800
E_{mean} (mV)	0.2143	0.2282	0.2108	0.2007
R_{mean}	22.0%	23.1%	21.3%	20.9%
E_{max} (mV)	1.38	1.38	1.21	1.19
R_{max}	14.3%	12.7%	18.7%	14.1%
CC	0.70	0.71	0.73	0.75
MSE (10^{-8})	7.3592	8.6551	7.0698	6.2672
R^2	0.6104	0.5297	0.5690	0.5989

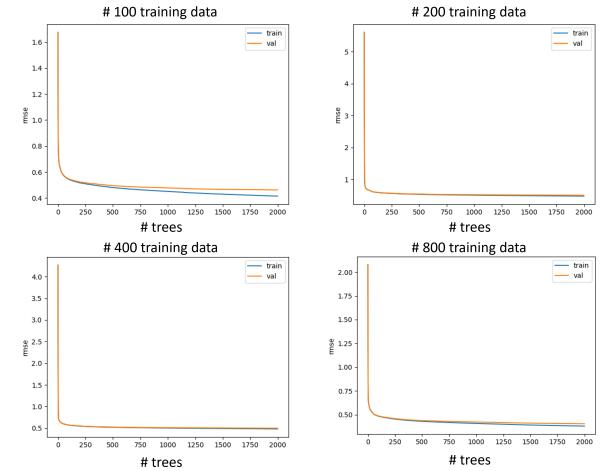


Fig. 7. Training curves analysis for different amounts of training data. The number of training data is 100, 200, 400, 800.

Model Size and Computational Complexity The concept of greening learning is pursuing low computational complexity, small model size, and mathematically transparent. First, the low complexity means fewer operations in the unit of time, indicating the computational efficiency of the model. Second, small model sizes show fewer memory requirements,

TABLE VI
COMPARISON OF MAE RESULTS OF REAL CIRCUIT DESIGNS ($10^{-2} mV$)

Rank	test 7	test 8	test 9	test 10	test 13	test 14	test 15	test 16	test 19	test 20
1	3.82	5.98	3.76	6.65	9.19	13.26	9.77	17.78	2.07	2.75
2	<u>4.06</u>	<u>6.01</u>	<u>4.29</u>	<u>7.69</u>	<u>16.38</u>	35.04	<u>11.02</u>	18.41	<u>6.75</u>	9.31
3	13.86	20.91	<u>10.63</u>	20.83	18.87	58.91	22.16	65.58	<u>7.40</u>	14.86
GIRD (SVM)	13.70	<u>13.49</u>	12.57	21.22	15.39	21.20	18.66	30.48	26.43	<u>12.53</u>
GIRD (RF)	23.24	16.47	<u>15.58</u>	<u>12.38</u>	20.75	<u>12.68</u>	<u>14.04</u>	12.87	14.49	35.79
GIRD (XGBoost)	12.49	19.17	15.51	25.31	<u>14.29</u>	34.87	16.33	<u>17.34</u>	26.07	14.80

The lowest MAE is in bold, while the second and the third ones are underlined for each test case.

TABLE VII
MODEL SIZE AND FLOPS NUMBER ANALYSIS

	# trees	Depths	Spatial resolution	# Parameters	# FLOPs	#Parameters/pixel	#FLOPs/pixel
XGBoost 4	600	3	16×16	13,200	614,400	0.81	37.5
XGBoost 3	2,000	3	32×32	44,000	8,192,000	2.68	500
XGBoost 2	2,000	3	64×64	44,000	32,768,000	2.68	2000
XGBoost 1	2,000	3	128×128	44,000	131,072,000	2.68	8000
PixelHop 4	-	-	16×16	13,012	6,503,424	0.79	396.9
PixelHop 3	-	-	32×32	24,692	49,582,080	1.51	3,026.3
PixelHop 2	-	-	64×64	46,758	375,349,248	2.85	22,909.5
PixelHop 1	-	-	128×128	27,100	875,479,040	1.65	53,435
Total	6,600	-	128×128	256,762	1,479,560,192	15.67	90,305.2

demonstrating the storage efficiency of the model. Third, the mathematical transparency provides good interpretability of the model. The measurement of exact carbon emissions is out of the scope of our topic. However, we can estimate the carbon footprint based on the FLOPs number.

$$\text{Power (W)} = \text{Total FLOPs (FLOP/s)} \times \text{Hardware Efficiency (Joules/FLOP)} \quad (11)$$

Then, we can convert total energy consumption to carbon emissions from the U.S. Environmental Protection Agency's (EPA) Emission Factors [48]. Therefore, we realize the concept of green by measuring the FLOPs and size of the model, while the exact power and carbon emission can be specified given the operated hardware. The following analyzes the FLOPs and model size for the components in the green learning pipeline. First, we compute the model size and computational complexity of a c/w Saab transform.

- Calculation of the number of parameters of a c/w Saab transform.

Without loss of generalizability, we use PU1 as an example. It has 12 input channels, each having a spatial resolution of 128×128 . For the c/w Saab with a kernel size of 7×7 , the number of parameters is calculated by $(7 \times 7) \times \text{numbers of channels}$. As a result, the total parameters of PU1 are $(7 \times 7) \times 121 + (5 \times 5) \times 162 + (3 \times 3) \times 337 = 13,012$.

- Calculation of the FLOPs per pixel of a c/w Saab transform.

For the FLOPs number per pixel, each n by n kernel requires $n \times n$ multiplications and $(n \times n - 1)$ addition. Therefore, the FLOPs/pixel can be calculated by $(\text{numbers of multiplication} + \text{numbers of additions}) \times (\text{numbers of kernels})$. The total FLOPs number for PU1 is $((49+48) \times 49 \times 12 + (25+24) \times 25 \times 12 + (9+8) \times 9 \times 12) \times 16 \times 16 = 6,503,424$.

Next, we analyze the model size and computational complexity of an XGBoost regressor. They can be calculated below.

- Calculation of the number of parameters of an XGBoost:

$$(\text{number of trees}) \times [2 \times (2^{\text{depth}} - 1) + 2^{\text{depth}}]. \quad (12)$$

- Calculation of the FLOPs per pixel of an XGBoost:

$$(\text{number of trees}) \times (\text{depth} + 1). \quad (13)$$

For example, XGBoost 1 has 600 trees with a maximum depth of 3. It demands $600 \times [2 \times (2^3 - 1) + 2^3] = 13,200$ parameters, and its FLOP number is equal to $600 \times (3 + 1) \times 16 \times 16 = 614,400$. Table VII reports the model size and FLOPs per pixel.

In Table VII, we detail the number of parameters and the FLOPs number for each c/w Saab feature extraction and XGBoost model. The comparison with the Deep Learning-based (DL-based) model is shown in Table VIII. Due to the difference in the input resolution and scale of dataset value, we compare the parameters per pixel, FLOPs per pixel, and NRMSE. We show our average NRMSE value of the ten cases in the ICCAD 2023 real hidden dataset. Our model size and FLOPs number are on the scale of 10^{-3} and 10^{-2} , respectively, compared with the DL-based model.

TABLE VIII
COMPARISON WITH DL MODEL

	#Params/pixel	#FLOPs/pixel	NRMSE (%)
CNN [19]	51,865.6 (3309x)	17,142.1K(189.8x)	26.0
Ours	15.67 (1x)	90.3K (1x)	24.1

VI. CONCLUSION AND FUTURE WORK

This work addressed the static IR drop problem by following the Green Learning methodology. Our GIRD method offers an alternative solution for solving complicated systems of linear

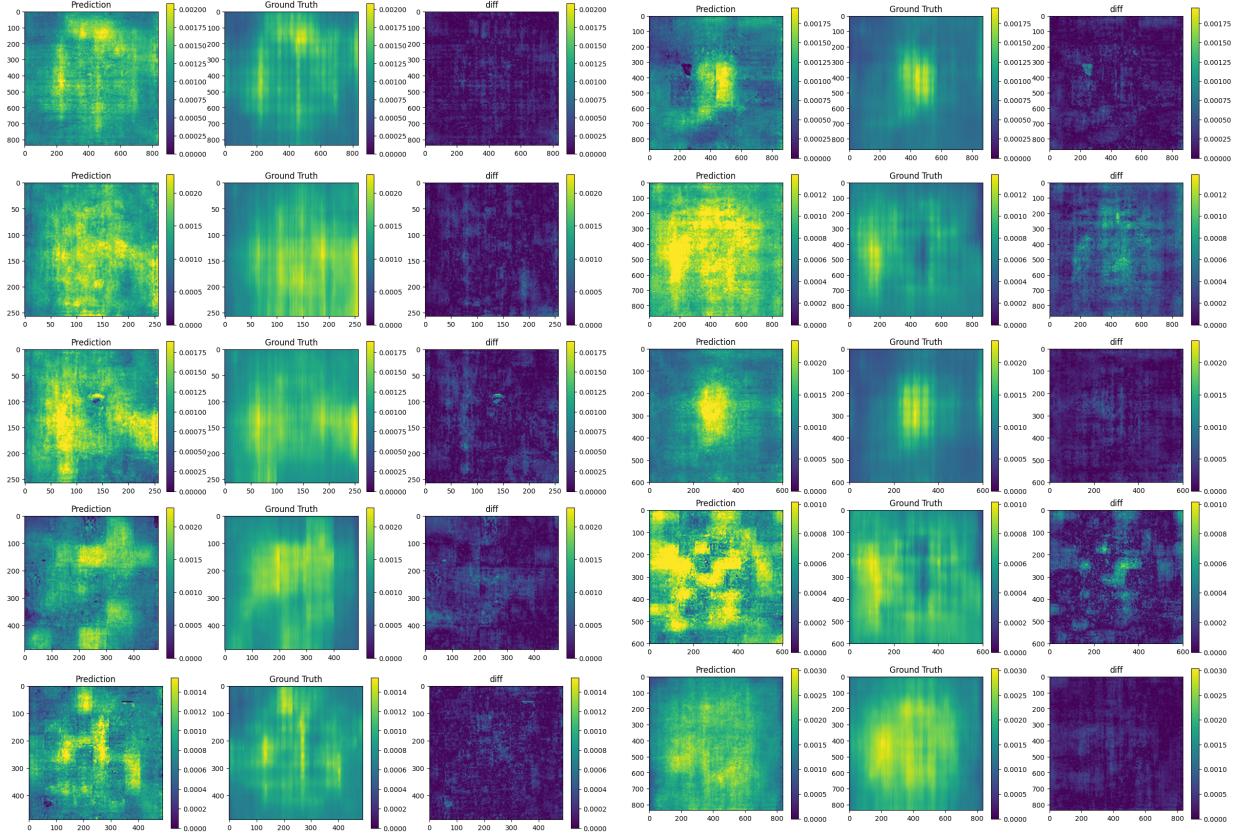


Fig. 8. Visualization of the MAE results of the real circuit designs in the hidden dataset.

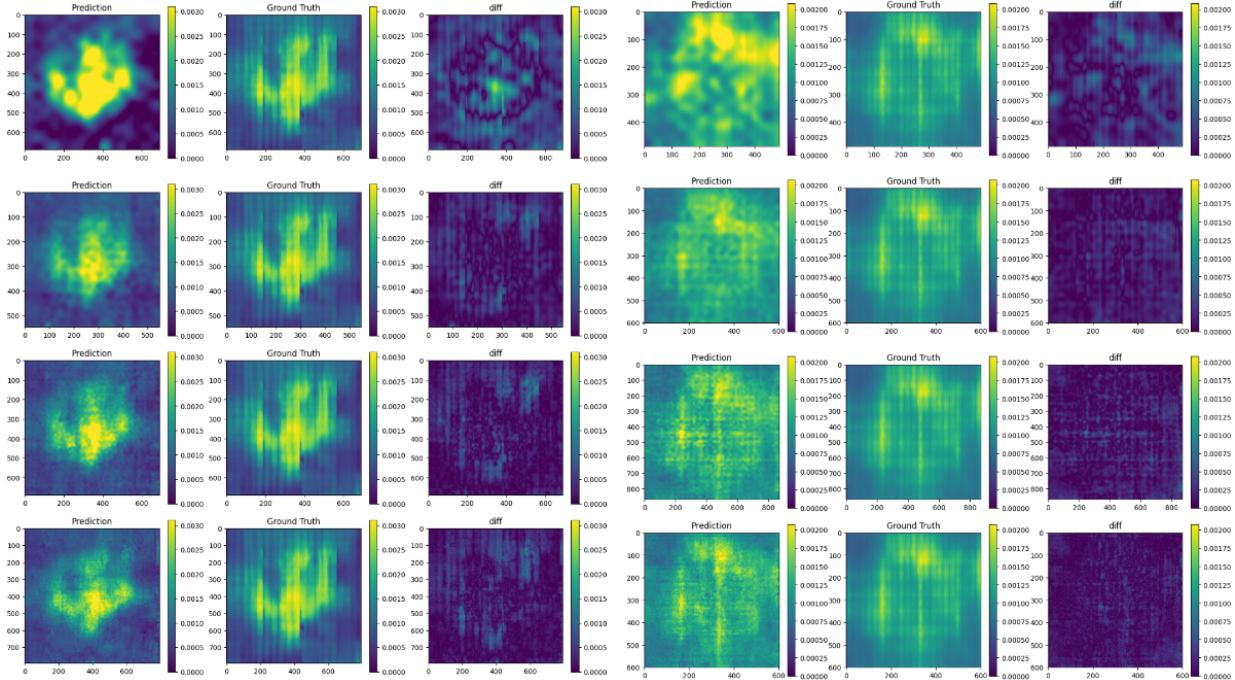


Fig. 9. Visualization of the MAE results on the real and synthetic circuit designs. The three columns on the left are the results of synthetic circuit designs. The three columns on the right are the results of real circuit designs. From top to bottom: the coarsest to highest spatial resolutions.

equations. GIRD is mathematically transparent, unlike black-box DL models using end-to-end optimization via backpropagation. It adopts a non-parametric modeling approach by exploiting input and output statistics. We demonstrated the efficiency and efficacy of GIRD in several experiments. Its MAE value for synthetic and real circuit design is on a scale of 10^{-3} (mV). GIRD's model size and FLOPs/pixel are only 1.2×10^{-4} and 2.52×10^{-2} of those of the DL model in [49], respectively. Thus, GIRD is ideal for fast IR drop estimation in terminal devices.

As a future extension, it is desired to address the dynamic IR drop estimation problem with additional timing switch information. Chen et al. [33] proposed using raw features to handle the dynamic IR-drop problem. Also, Chhabria et al. [31] proposed a list of features with U-net to address the dynamic IR-drop issue. Inspired by these two works, we may toggle several quantities (e.g., toggle input, toggle output, toggle internal connection, toggle arrival time, etc.), measure the associated waveform files, and use them as input channels to the GIRD system. Moreover, we may apply the green learning methodology to heat source analysis and shed light on other learning-based EDA problems.

ACKNOWLEDGMENT

The authors acknowledge the Center for Advanced Research Computing (CARC) at the University of Southern California for providing computing resources that contributed to the research results reported in this publication. URL: <https://carc.usc.edu>. We would like to thank Dr. Joe Wang and Chee-En Yu for their valuable feedback and proofreading.

APPENDICES

A. Visualization of Input Channels

Fig. 10 shows 12 input maps. The three from left to right in the first row are the current sources map, PDN density map, and voltage distance map. The nine from the second to fourth rows are multi-layer resistor maps.

B. Saab Transform and Channel-wise Saab Transform

The Saab transform is a statistic-based method to determine the filter weights in PUs.

They are a variation of the principle components analysis (PCA). PCA can decorrelate the input vectors and generate a compact description. For example, if input \mathbf{x} is an n -dimensional vector, we can construct a $n \times n$ covariance matrix based on N samples. Through PCA, we can find n eigenvalues with corresponding eigenvectors. Typically, we can select $k < n$ based on eigenvalue λ_k in a decreasing order to achieve a lossy approximation. If $k = n$, this results in a lossless Saab Transform.

Furthermore, the input vector \mathbf{x} should be a mean-removed vector for PCA. We can remove the mean of the input in the training, but we can not guarantee the same mean value in the testing set. Therefore, Saab transforms introduce a DC kernel whose elements have the same value. The DC kernel can yield the mean value of a local patch. Then, we can remove the local

mean of the input and apply PCA to yield AC kernels. For example, if the kernel size is 3×3 , we can acquire 1 DC kernel and 8 AC kernels. Lastly, every filter within a layer applies an identical bias, ensuring all filter responses are positive to achieve the same effect of nonlinear activation. This approach to filter weight design enhances computational efficiency and facilitates a straightforward interpretation of the convolutional layer's function.

Channel-wise, Saab transform (c/w Saab) is the improved version of Saab transform. Saab coefficients exhibit weak correlations in the spectral domain due to the utilization of PCA. Due to the weak correlation between channels, we can facilitate the transformation by decomposing a 3D tensor (2D spatial and 1D spectral dimensional) with dimensional $S_i \times S_i \times K_i$ into K_i spatial tensors of neighborhood size $S_i \times S_i$. That is, instead of applying 3D Saab transform of $S_i \times S_i \times K_i$, c/w Saab transform operates K_i times to 2D $S_i \times S_i$ tensor. There are two advantages for c/w Saab Transform. Firstly, for the lossless Saab transform, the model size of the c/w Saab transform is smaller than that of the standard 3D Saab transform. Secondly, for the lossy Saab transform, the reduction in model size achieved by the c/w Saab transform can be even more significant. For the lossless case, the input of the Saab transform and the c/w Saab transform are $n_{3D} = S_i^2 K_i$ and $n_{c/w} = S_i^2$ respectively. Therefore, their model sizes are:

$$\text{Saab : } n_{3D}^2 = S_i^4 K_i^2 \quad (14)$$

$$\text{c/w Saab : } K_i n_{c/w}^2 = S_i^4 K_i \quad (15)$$

To illustrate the lossy case, we present a multi-stage c/w Saab transform for the entire 2D array (128×128) in Figure 2. For high-frequency components, the spatial correlation of pixels in a local patch is so weak that it is trivial to conduct the Saab transform afterward. These channels are marked as gray and called leaf nodes. On the other hand, the spatial correlation is still significant enough for the low-frequency components. Therefore, c/w Saab is applied to these channels. They are colored orange and called intermediate nodes.

C. Relevant Feature Test

After acquiring data representation from PU, the relevant feature test (RFT) aims to reduce dimension by selecting discriminative features with the guidance of labels. The three steps of RFT are detailed as follows.

First, we search for the optimal threshold f_{opt}^i between $[f_{min}^i, f_{max}^i]$ and partition training samples into two subsets S_L^i and S_R^i for the i th feature f^i . To limit the search space of f_{opt}^i we partition the entire feature range $[f_{min}^i, f_{max}^i]$ into B uniform segments and search the optimal threshold among the following $B - 1$ candidates:

$$f_b^i = f_{min}^i + \frac{b}{B} [f_{min}^i, f_{max}^i], \quad b = 1, \dots, B - 1 \quad (16)$$

Second, we calculate the RFT loss measured by estimated regression MSE. We denote the regression target value as y . For the i th feature dimension, we partition the sample space

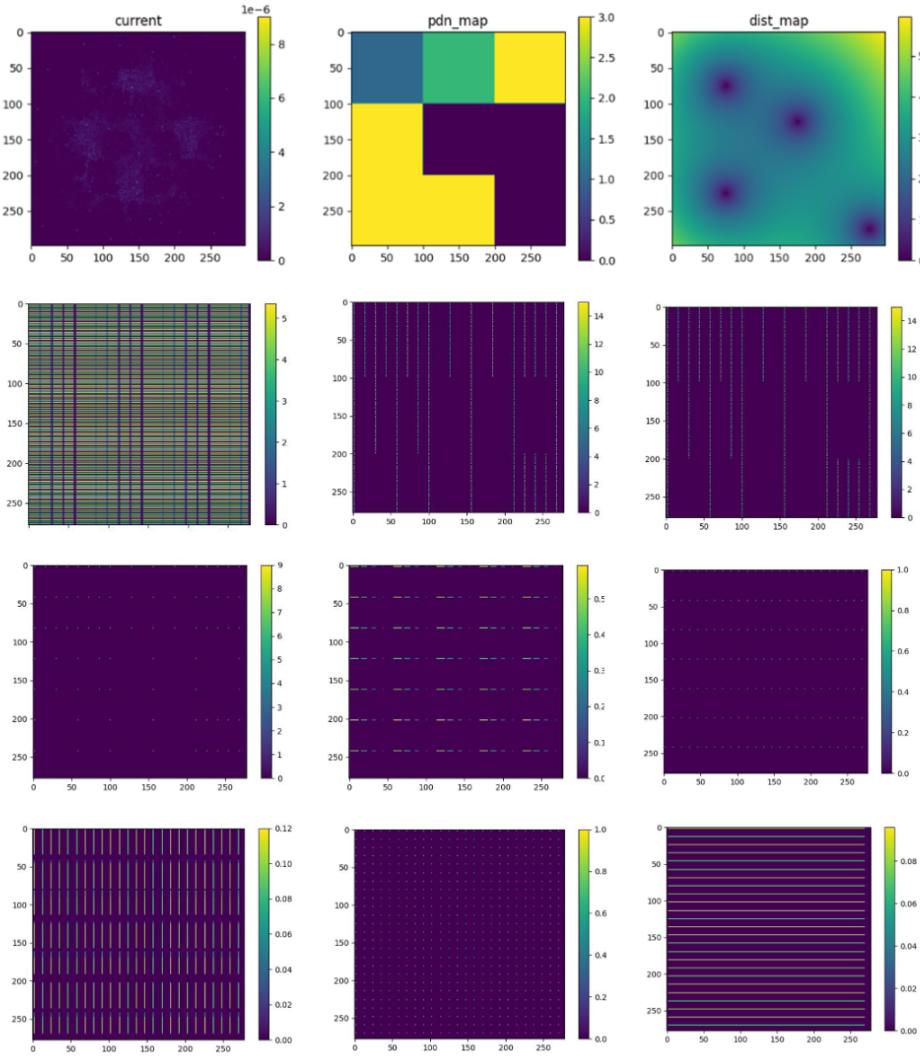


Fig. 10. Visualization of input data comprising 12 maps: one current source map, one voltage distance map, one PDN density map, and nine layered resistor maps. All maps are resized to the same spatial resolution of 128×128 pixels.

into two disjoint spaces: S_L^i and S_R^i . Let y_L^i and y_R^i be the mean value of S_L^i and S_R^i . Then, we can use y_L^i and y_R^i as the estimated regression value of all samples in S_L^i and S_R^i , respectively. The RFT loss is the sum of estimated regression MSEs of S_L^i and S_R^i .

$$R_t^i = \frac{N_{L,t}^i R_{L,t}^i + N_{R,t}^i R_{R,t}^i}{N}, \quad (17)$$

where $N = N_{L,t} + N_{R,t}$ is the total number of samples, $N_{L,t}$ and $N_{R,t}$ are the sample number in the S_L^i and S_R^i according to threshold t ; R_t^i is the estimated MSEs of the i th feature and threshold t . Hence, feature f^i is characterized by its optimized estimated regression MSE of the threshold set T in the partition points:

$$R_{opt}^i = \min_{t \in T} R_t^i \quad (18)$$

Last, we sort the optimized estimated regression MSE value, R_{opt}^i , for all feature dimensions f^i , $1 \leq i \leq P$, in ascending order. The lower the R_{opt}^i value, the more relevant the i th-dimensional feature, f^i . Afterward, we select the top

K features with the lowest estimated regression MSE values as relevant features based on the elbow point of the curve.

REFERENCES

- [1] C-C Jay Kuo and Azad M Madni. Green learning: Introduction, examples and outlook. *Journal of Visual Communication and Image Representation*, 90:103685, 2023.
- [2] C-C Jay Kuo. Understanding convolutional neural networks with a mathematical model. *Journal of Visual Communication and Image Representation*, 41:406–413, 2016.
- [3] C-C Jay Kuo, Min Zhang, Siyang Li, Jiali Duan, and Yueru Chen. Interpretable convolutional neural networks via feedforward design. *Journal of Visual Communication and Image Representation*, 60:346–359, 2019.
- [4] Zhanxuan Mei, Yun-Cheng Wang, C-C Jay Kuo, et al. Lightweight high-performance blind image quality assessment. *APSIPA Transactions on Signal and Information Processing*, 13(1), 2024.
- [5] Yueru Chen and C-C Jay Kuo. Pixelhop: A successive subspace learning (SSL) method for object recognition. *Journal of Visual Communication and Image Representation*, 70:102749, 2020.
- [6] Chengwei Wei, Runqi Pang, and C-C Jay Kuo. A green learning approach to spoofed speech detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12956–12960. IEEE, 2024.

- [7] Yijing Yang, Wei Wang, Hongyu Fu, C-C Jay Kuo, et al. On supervised feature selection from high dimensional feature spaces. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.
- [8] Jeremy Lee, Sumit Narayan, Mike Kapralos, and Mohammad Tehranipoor. Layout-aware, ir-drop tolerant transition fault pattern generation. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1172–1177, 2008.
- [9] Junxia Ma, Jeremy Lee, and Mohammad Tehranipoor. Layout-aware pattern generation for maximizing supply noise effects on critical paths. In *2009 27th IEEE VLSI Test Symposium*, pages 221–226. IEEE, 2009.
- [10] Nisar Ahmed, Mohammad Tehranipoor, and Vinay Jayaram. Transition delay fault test pattern generation considering supply voltage noise in a soc design. In *Proceedings of the 44th annual Design Automation Conference*, pages 533–538, 2007.
- [11] Xiaojing Wen, Yoshiyuki Yamashita, Shohei Morishima, Seiji Kajihara, Laung-Terng Wang, Kewal K Saluja, and Kozo Kinoshita. Low-capture-power test generation for scan-based at-speed testing. In *IEEE International Conference on Test, 2005.*, pages 10–pp. IEEE, 2005.
- [12] Yuta Yamato, Tomokazu Yoneda, Kazumi Hatayama, and Michiko Inoue. A fast and accurate per-cell dynamic ir-drop estimation method for at-speed scan test pattern validation. In *2012 IEEE International Test Conference*, pages 1–8. IEEE, 2012.
- [13] Farid N Najm. *Circuit simulation*. John Wiley & Sons, 2010.
- [14] Gana Surya Prakash Kadagala and Vidya A Chhabria. 2023 iccad cad contest problem c: Static ir drop estimation using machine learning. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 1–5. IEEE, 2023.
- [15] Guyue Huang, Jingbo Hu, Yifan He, Jialong Liu, Mingyuan Ma, Zhaoyang Shen, Juejian Wu, Yuanfan Xu, Hengrui Zhang, Kai Zhong, et al. Machine learning for electronic design automation: A survey. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 26(5):1–46, 2021.
- [16] Kevin Immanuel Gubbi, Sayed Aresh Beheshti-Shirazi, Tyler Sheaves, Soheil Salehi, Sai Manoj PD, Setareh Rafatirad, Avesta Sasan, and Houman Homayoun. Survey of machine learning for electronic design automation. In *Proceedings of the Great Lakes Symposium on VLSI 2022*, pages 513–518, 2022.
- [17] Deepthi Amuru, Andleeb Zahra, Harsha V Vudumula, Pavan K Cherupally, Sushanth R Gurram, Amir Ahmad, and Zia Abbas. Ai/ml algorithms and applications in vlsi design and technology. *Integration*, 93:102048, 2023.
- [18] Yufei Chen, Haojie Pei, Xiao Dong, Zhou Jin, and Cheng Zhuo. Application of deep learning in back-end simulation: Challenges and opportunities. In *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 641–646. IEEE, 2022.
- [19] Yen-Chun Fang, Heng-Yi Lin, Min-Yan Sui, Chien-Mo Li, and Eric Jia-Wei Fang. Machine-learning-based dynamic ir drop prediction for eco. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–7. IEEE, 2018.
- [20] Shih-Yao Lin, Yen-Chun Fang, Yu-Cheng Li, Yu-Cheng Liu, Tsung-Shan Yang, Shang-Chien Lin, Chien-Mo Li, and Eric Jia-Wei Fang. Ir drop prediction of eco-revised circuits using machine learning. In *2018 IEEE 36th VLSI Test Symposium (VTS)*, pages 1–6. IEEE, 2018.
- [21] Chia-Tung Ho and Andrew B Kahng. Incpird: Fast learning-based prediction of incremental ir drop. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8. IEEE, 2019.
- [22] Han Zhou, Wentian Jin, and Sheldon X-D Tan. Gridnet: Fast data-driven em-induced ir drop prediction and localized fixing for on-chip power grid networks. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pages 1–9, 2020.
- [23] Heng-Yi Lin, Yen-Chun Fang, Shi-Tang Liu, Jia-Xian Chen, Chien-Mo Li, and Eric Jia-Wei Fang. Automatic ir-drop eco using machine learning. In *2020 IEEE International Test Conference in Asia (ITC-Asia)*, pages 7–12. IEEE, 2020.
- [24] Zhiyao Xie, Haoxing Ren, Brucek Khailany, Ye Sheng, Santosh Santosh, Jiang Hu, and Yiran Chen. Powernet: Transferable dynamic ir drop estimation via maximum convolutional neural network. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 13–18. IEEE, 2020.
- [25] Zhiyao Xie, Hai Li, Xiaoqing Xu, Jiang Hu, and Yiran Chen. Fast ir drop estimation with machine learning. In *Proceedings of the 39th international conference on computer-aided design*, pages 1–8, 2020.
- [26] Vidya A Chhabria, Kishor Kunal, Masoud Zabih, and Sachin 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)*, pages 1–8. IEEE, 2021.
- [27] Vidya A Chhabria, Vipul Ahuja, Ashwath Prabhu, Nikhil Patil, Palkesh Jain, and Sachin S Sapatnekar. Encoder-decoder networks for analyzing thermal and power delivery networks. *ACM Transactions on Design Automation of Electronic Systems*, 28(1):1–27, 2022.
- [28] Santanu Kundu, Manoranjan Prasad, Sashank Nishad, Sandeep Nachireddy, and K Harikrishnan. Mlir: Machine learning based ir drop prediction on eco revised design for faster convergence. In *2022 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID)*, pages 68–73. IEEE, 2022.
- [29] Fangming Ye, Farshad Firouzi, Yang Yang, Krishnendu Chakrabarty, and Mehdi B Tahoori. On-chip voltage-droop prediction using support-vector machines. In *2014 IEEE 32nd VLSI Test Symposium (VTS)*, pages 1–6. IEEE, 2014.
- [30] Vidya A Chhabria, Vipul Ahuja, Ashwath Prabhu, Nikhil Patil, Palkesh Jain, and Sachin S Sapatnekar. Thermal and ir drop analysis using convolutional encoder-decoder networks. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, pages 690–696, 2021.
- [31] Vidya A Chhabria, Yanqing Zhang, Haoxing Ren, Ben Keller, Brucek Khailany, and Sachin S Sapatnekar. Mavirec: ML-aided vectored ir-drop estimation and classification. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1825–1828. IEEE, 2021.
- [32] Yonghwi Kwon, Giyoon Jung, Daijoon Hyun, and Youngsoo Shin. Dynamic ir drop prediction using image-to-image translation neural network. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2021.
- [33] Jia-Xian Chen, Shi-Tang Liu, Yu-Tsung Wu, Mu-Ting Wu, Chieo-Mo Li, Norman Chang, Ying-Shiu Li, and Wen-Tze Chuang. Vector-based dynamic ir-drop prediction using machine learning. In *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 202–207. IEEE, 2022.
- [34] Hongyu Fu, Yijing Yang, Vinod K Mishra, and C-C Jay Kuo. Classification via subspace learning machine (slm): Methodology and performance evaluation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.
- [35] Yijing Yang, Hongyu Fu, and C-C Jay Kuo. Design of supervision-scalable learning systems: Methodology and performance benchmarking. *Journal of Visual Communication and Image Representation*, 96:103925, 2023.
- [36] Kaitai Zhang, Hong-Shuo Chen, Xinfeng Zhang, Ye Wang, and C-C Jay Kuo. A data-centric approach to unsupervised texture segmentation using principle representative patterns. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1912–1916. IEEE, 2019.
- [37] Kaitai Zhang, Hong-Shuo Chen, Ye Wang, Xiangyang Ji, and C-C Jay Kuo. Texture analysis via hierarchical spatial-spectral correlation (hssc). In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4419–4423. IEEE, 2019.
- [38] Hong-Shuo Chen, Mozhdeh Rouhsedaghat, Hamza Ghani, Shuowen Hu, Suya You, and C-C Jay Kuo. Defakehop: A light-weight high-performance deepfake detector. In *2021 IEEE International conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.
- [39] Hong-Shuo Chen, Shuowen Hu, Suya You, C-C Jay Kuo, et al. Defakehop++: An enhanced lightweight deepfake detector. *APSIPA Transactions on Signal and Information Processing*, 11(2), 2022.
- [40] Yao Zhu, Xinyu Wang, Ronald Salloum, Hong-Shuo Chen, C-C Jay Kuo, et al. Rgjid: A robust and green gan-fake image detector. *APSIPA Transactions on Signal and Information Processing*, 11(2), 2022.
- [41] Yao Zhu, Xinyu Wang, Hong-Shuo Chen, Ronald Salloum, and C-C Jay Kuo. A-pixelhop: A green, robust and explainable fake-image detector. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8947–8951. IEEE, 2022.
- [42] Olaf Ronneberger, Philipp Fischer, and Thomas 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*, pages 234–241. Springer, 2015.
- [43] Yueru Chen, Mozhdeh Rouhsedaghat, Suya You, Raghuvir Rao, and C-C Jay Kuo. Pixelhop++: A small successive-subspace-learning-based (ssl-based) model for image classification. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3294–3298. IEEE, 2020.
- [44] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794, 2016.

- [45] Vidya A. Chhabria, Kishor Kunal, Masoud Zabihi, and Sachin S. Sapatnekar. Began: Power grid benchmark generation using a process-portable gan-based methodology. In *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–8, 2021.
- [46] Gana Surya Prakash Kadagala and Vidya A Chhabria. 2023 iccad cad contest problem c: Static ir drop estimation using machine learning. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 1–5. IEEE, 2023.
- [47] 2023 cad contest @ iccad. <https://www.iccad-contest.org/2023/Problems.html>, 2023. Accessed: 2024-09-07.
- [48] Environmental protection agency's (epa) emission factors. <https://www.epa.gov/egrid/data-explorer>, 2023. Accessed: 2024-11-10.
- [49] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023.



Hung-Ming Chen received the B.S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, and the M.S. and Ph.D. degrees in computer sciences from the University of Texas at Austin.

Dr. Chen is currently a Professor at the Institute of Electronics at National Yang Ming Chiao Tung University, Hsinchu, Taiwan. He has served as the chair of IEEE CEDA Taipei Chapter and the technical program committee members, including ACM/IEEE DAC, ASP-DAC, IEEE/ACM ICCAD, and ACM ISPD. He also served as EDA track co-chair in the IEEE VLSI-DAT and placement track co-chair in ICCAD. He has supervised a team to win first place at the 2014 ISPD Placement Contest and 2023 CAD Contest at ICCAD. His research interests include design automation in digital and analog circuits, beyond-die integration, and VLSI design methodologies.



Chee-An Yu (Student Member, IEEE) received his B.S. degree in mechanical engineering and M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2018 and 2020, respectively. He is pursuing a Ph.D. in electrical engineering at the University of Southern California, Los Angeles, USA. His research interests include signal processing, machine learning, and electronic design automation.



Yu-Tung Liu (Student Member, IEEE) received his B.S. degree in electronics and electrical engineering from National Yang Ming Chiao Tung University, Hsinchu, Taiwan. His research interests focus on applying machine learning to electronic designs.



C.-C. Jay Kuo C.-C. Jay Kuo (Life Fellow, IEEE) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1980 and the M.S. and PhD degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively.

He holds the Ming Hsieh Chair Professorship, is a Distinguished Professor of Electrical and Computer Engineering and Computer Science, and is the Director of the Multimedia Communication Laboratory (MCL) at the University of Southern California, Los Angeles, California. He is a fellow of the National Academy of Inventors (NAI), the Association for Computing Machinery (ACM), the American Association for the Advancement of Science (AAAS), and the International Society for Optical Engineers (SPIE). He is also an Academician of Academia Sinica in Taiwan. His research activities lie in multimedia computing and green AI. He co-authored 375 journal papers, 1000 conference papers, 30 patents, and 15 books. Dr. Kuo has guided 178 students to their Ph.D. degrees and supervised 31 postdoctoral research fellows.



Yu-Hao Cheng (Student Member, IEEE) received his B.S. degree in electronics and electrical engineering from National Yang Ming Chiao Tung University, Hsinchu, Taiwan, in 2024. He is currently pursuing an M.S. degree in electronics engineering at National Yang Ming Chiao Tung University, Hsinchu, Taiwan. His research interests include machine learning and electronic design automation.



Shao-Yu Wu (Student Member, IEEE) received his B.S. degree in electrical and computer engineering from National Yang Ming Chiao Tung University, Hsinchu, Taiwan, in 2024. He is currently pursuing an M.S. degree in electronics engineering at National Taiwan University, Taipei, Taiwan. His research interests include machine learning and electronic design automation.