Real-Time Dynamic IR-drop Prediction for IR ECO

Yu-Che Lee[1], Yu-Hsuan Chen[1], Yu-Chen Cheng[1], Yong-Fong Chang[2], Jia-Wei Lin[2], Hsun-Wei Pao[2],
Yung-Chih Chen[3], Yi-Ting Li[1], Wuqian Tang[1], Shih-Chieh Chang[1], and Chun-Yao Wang[1]

[1]National Tsing Hua University, Taiwan
[2]Mediatek Inc., Taiwan
[3]National Taiwan University of Science and Technology, Taiwan

2025 62nd ACM/IEEE Design Automation Conference (DAC) | 979-8-3315-0304-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/DAC63849.2025.11133331

Abstract*—During the IR Engineering Change Order (ECO) stage, cell moving leads to uncertain IR-drop results, requiring designers to explore multiple ECO candidates in each iteration to find a solution that effectively mitigates IR-drop, resulting in a long evaluation time. Although machine learning (ML)-based predictors have been proposed to expedite IR-drop evaluation, partial simulations are still needed to update features after ECO, taking over an hour and delaying IR-drop results. In this work, we propose a real-time dynamic IR-drop estimation method based on an XGBoost model with a global view of a cell's surroundings. After ECO, our method provides dynamic IR-drop results in minutes without running any simulations and thus achieves real-time estimation. This allows designers to evaluate multiple ECO candidates concurrently in a single iteration. We conducted the experiments on five ECO candidates of an industrial design with $3nm$ technology. The results show that the proposed model can effectively predict the IR-drop variations of moved cells after ECO with over 93% of fixed cells detected and an average MAE of $8.75mV$ achieved. Furthermore, our method achieves an $88X$ speedup over Voltus (commercial tool) and a $64X$ speedup over traditional ML predictors when evaluating a single ECO candidate. The speedup is expected to increase as the number of ECO candidates increases.

I. Introduction

According to Ohm's law, when current flows through the power delivery network (PDN) and reaches the standard cells, the supply voltage drops due to the resistance of power rails. The voltage consumed in this process is known as IR-drop. IR-drop can increase the cell delay and potentially lead to circuit performance degradation and functional errors. As a result, IR-drop analysis for identifying violations and IR Engineering Change Order (ECO) for fixing the violations are extremely crucial in the sign-off flow.

The IR sign-off flow mentioned in previous works [7], [12], [10] indicates that an ECO and an IR-drop analysis are performed iteratively to identify and fix violations. However, the flow is oversimplified for the industry. In real-world scenarios, designers conduct cell moving-based ECO by adjusting the physical positions of violation cells in the layout to mitigate IR-drop. Due to the uncertain IR-drop variations caused by cell moving, designers typically experience several trials and errors within an iteration to reach a desired result. Consequently, multiple ECO candidates may exist in a single iteration. When evaluating each ECO candidate, even minor cell movements require running computationally expensive simulations for the entire circuit to obtain the IR-drop report. It is therefore time-consuming for designers to evaluate all candidates and select the best one for the next iteration. Additionally, due to design cycle time constraints and computational resource limitations during sign-off, the number of ECO candidates that can be attempted per iteration is limited.

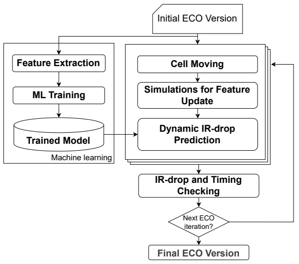


Fig. 1: Industrial flow + IR-drop prediction

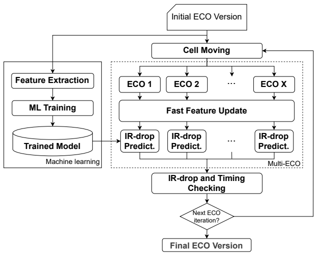


Fig. 2: Industrial flow + Real-time IR-drop prediction

Several machine learning (ML)-based IR-drop predictors have been proposed to accelerate IR-drop estimation [7], [12], [9], [3], [10], [8], [13], [15]. The ML-aided IR ECO flow is depicted in Fig. 1, where the IR-drop analysis process is replaced by IR-drop prediction. However, this flow still requires power analysis and partial rail analysis for resistance extraction to update the features of the modified circuit for prediction. This prevents designers from quickly acquiring IR-drop result caused by cell moving and becomes a primary bottleneck in parallelizing the evaluations of multiple ECO candidates due to the time- and resource-intensive nature of simulations.

In this paper, we propose a new scheme for real-time prediction of dynamic IR-drop after cell moving-based ECO, aiming to enhance the efficiency of the industrial IR sign-off flow. Firstly, we develop an XGBoost model [4] to predict the dynamic IR-drops of cells after ECO. Unlike previous works [7], [3], [9], [12] that let the model learn neighboring features in a local manner, we globally consider neighboring features in advance and provide representative ones for the model. Then, to achieve real-time IR-drop prediction, instead of rerunning simulations to update the features of the revised circuit, we use the physical information and raw features of cells before ECO to quickly estimate the changes in neighboring features. With real-time dynamic IR-drop prediction, we further propose an innovative industrial IR sign-off flow as shown in Fig. 2. In this flow, we can rapidly estimate the dynamic IR drop result after ECO. Since no simulations are required for updating features, multiple ECO candidates can be evaluated in parallel,

This work is supported in part by the National Science and Technology Council (Taiwan) under NSTC 112-2218-E-007-014, NSTC 112-2221-E-007-106-MY2, NSTC 112-2221-E-007-108, NSTC 113-2218-E-007-020, NSTC 113-2221-E-007-082-MY3, NSTC 113-2425-H-007-004, NSTC 113-2640-E-011-003, NSTC 114-2425-H-007-002, and National Tsing Hua University under NTHU 112A0316 and NTHU 113A0314.

significantly reducing the time for designers to explore ECO candidates. Moreover, with the realization of parallelization, designers can explore more ECO candidates within an iteration and increase the probability of selecting a better ECO version for the next iteration.

The contributions of this paper are summarized as follows:

- An XGBoost-based dynamic IR-drop predictor. It utilizes newly proposed features designed to represent a cell's surrounding information globally and successfully improves the accuracy by 20.2% compared to the local approach.
- A feature update technique. It rapidly updates features after ECO without the need for resimulations, enabling real-time IR-drop estimation. It makes the predictor effectively capture the IR-drop variations of moved cells with more than 96% of fixed cells detected and an average MAE of $8.75mV$ achieved.
- An innovative ML-aided IR sign-off flow. It allows designers to evaluate multiple ECO candidates concurrently, enabling the exploration of more candidates within an iteration. It achieves $88X$ and $64X$ speedups over Voltus and traditional ML predictors for an ECO candidate.

The rest of this paper is organized as follows: Section II reviews the related works. Section III presents the proposed sign-off flow and details each step of it. Section IV shows the experimental results. Finally, Section V concludes our work.

II. Prior Works

Being aware of the time-consuming nature of IR-drop analysis, several ML-based methods for IR-drop prediction have been proposed [7], [12], [9], [3], [8], [13], [15], [6], [5], [11]. These methods can be classified based on the prediction target. [8], [13] focus on static IR-drop prediction of power nodes in the PDN; [15], [6], [5], [11] focuses on tile-based dynamic IR-drop prediction; [7], [12], [9], [3] focus on cell-based dynamic IR-drop prediction. We review the methods in [7], [12], [9], [3] in more detail, as we also focus on cell-based dynamic IR-drop prediction.

The method in [7], [12], [9] aims to accelerate the estimation of IR-drop in a modified circuit after cell moving-based ECO. In [7], numerous small regional XGBoost models are constructed to predict dynamic IR-drop results post-ECO. [12] introduces an Artificial Neural Network model for predicting dynamic IR-drop post-ECO. [9] proposes a dynamic IR-drop ECO optimization through cell movement, and uses a Random Forest model to predict dynamic IR-drop post-ECO to guide the cell movement process. Meanwhile, the work in [3] aims to speed up the vector-based IR-drop analysis by using an XGBoost model trained by certain vectors and predicting the other vectors.

Despite their innovations, [7], [9], [3] rely on specific neighboring features (e.g., $21x11$ partitions, rhombus-shaped 13 partitions, rhombus-shaped 13 vias) to capture the environmental effect on IR-drop. While effective, this approach may limit tree-based models' ability to learn relationships between neighboring features, potentially leading to confusion. Additionally, the neighboring features in [12] do not account for the simultaneous switching of the target cell and surrounding cells.

A shared drawback of [7], [12], [9] is their reliance on partial simulations to update the features of modified circuits before prediction. The time required for partial simulations is comparable to that for IR-drop simulations, making it time-intensive for the industry to evaluate multiple ECO candidates within an IR sign-off iteration.

Furthermore, [7], [12], [9] treat the revised circuit post-ECO as an independent entity, neglecting its relationship with the pre-ECO circuit. This oversight could result in the model missing crucial information.

III. Proposed IR Sign-off Flow

In this section, we first provide an overview of the proposed IR sign-off flow and present how the proposed methods facilitate IR sign-off. Next, we present the proposed features and the extraction method. Then, we present the proposed ML model for dynamic IR-drop prediction. Finally, we present an estimation method to update the features of the revised circuit after ECO for obtaining real-time IR-drop prediction.

A. Overview

The proposed IR sign-off flow is shown in Fig. 2. Given an initial circuit with its IR-drop report, we first extract features and IR-drop labels to train an XGBoost model as a dynamic IR-drop predictor. Meanwhile, designers start the process of cell moving-based ECO manually or automatically with an ECO tool for fixing the IR-drop violations. When using an ECO tool for automatic cell location adjustment, three key parameters influence its behavior: (1) the $design\ rule\ violation$ (DRV), which establishes the trade-off between timing- and IR-drop-aware ECO, (2) the $maximum\ distance$, which determines the furthest allowable displacement of violation cells, and (3) the $padding\ size$, which specifies the blockage size surrounding violation cells. Designers cannot expect to achieve the best cell movement result in a single attempt. Thus, they configure various parameter combinations to generate a set of ECO candidates for exploration, where the number of candidates, denoted as n, can vary widely. Then, we utilize the fast feature update technique to estimate the features of these candidates, enabling the trained model to predict the dynamic IR-drop results. This process, from feature update to IR-drop prediction, can be completed rapidly and eliminates the need for simulation, so that the parallel evaluation of all ECO candidates is allowed. Finally, designers select the most suitable ECO candidate based on the predicted IR-drop and timing results reported by *Tweaker* [14] to proceed to the next iteration.

B. Feature Extraction

Both the target cell and its surrounding environment can impact its dynamic IR-drop. We categorize the extracted features into cell-based features and tile-based features. Cell-based features represent the electrical characteristics of the cell, while tile-based features capture effects from the neighboring cells.

1) Cell-based features: The cell-based features listed below capture the raw features relevant to the dynamic IR-drop of the target cell itself.

- *Cell loc* (x, y): The lower-left coordination of the cell
- *Cell size* (w, h): Cell width and cell height
- C_{load}: Load capacitance which is the sum of the net capacitance and the output pin capacitance

- I_{peak}: Peak current which is the maximum current during the simulation
- P_{switch}: Switching power which is caused by the charging and discharging of C_{load}
- TR_{out}: Output toggle rate which is the average toggle count per cycle of the output pin
- $R_{eff}(2)$: Effective resistance which is the equivalent path resistance from the cell's VDD (VSS) pin to the voltage source within the die
- $R_{bump}(2)$: The resistance of the bump that the cell's VDD (VSS) pin connects with
- $L_{bump}(2)$: The inductor of the bump that the cell's VDD (VSS) pin connects with

Among the features, C_{load}, I_{peak}, P_{switch}, and TR_{out} can be extracted after power analysis, and R_{eff} can be extracted after rail analysis. R_{bump} and L_{bump} reflect the package effect on IR-drop. They can be extracted using a commercial package extraction tool and remain constant before and after ECO.

2) Tile-based features: Nearby cells have a significant impact on the dynamic IR-drop of the target cell, particularly those that switch simultaneously with the target cell. To effectively capture the neighboring information, we generate the tile-based features in three steps. We first construct the local tile-based features [7], [3], [9]. Then, we transform them to the global tile-based features. Finally, among them, we select the features that are highly correlated with IR-drop as our final tile-based features.

In the first step, we create a $tile\ map$ by dividing the design into tiles and assign each cell to the corresponding tile based on its $cell\ loc$. The $tile\ map$ records the physical information of cells in the layout. For a target cell, its neighboring tiles are the $5x5$ tiles surrounding the tile where the cell is located. We then calculate the tile value by summing up the features of cells in each neighboring tile. To consider the simultaneous switching effect on dynamic IR-drop, we sum only the features related to timing, including peak current (I_{peak}), switching power (P_{switch}), internal power ($P_{internal}$), output toggle rate (TR_{out}), and IR-drop (IR). Moreover, we include only cells whose timing window (tw) overlaps with the target cell's tw in the summation. The inclusion of IR-drop is based on our observation that the ECO tool often mitigates IR-drop by moving cells from regions of higher IR-drop to those of lower IR-drop, indicating a strong correlation between the cell's IR-drop and the regional IR-drop. Now, there are 125 local tile-based features (5 timing-related features × 25 neighboring tiles).

In the second step, for each timing-related feature, we extract the following three global tile features among the neighboring tile features.

- *Min. tile feature*: the minimum tile among the neighboring tiles
- *Max. tile feature*: the maximum tile among the neighboring tiles
- *Overall tile feature*: sum of the neighboring tiles

The previous works [7], [3], [9] directly use all the neighboring tiles as features, which could confuse the XGBoost model. Fig.3 demonstrates one example. Assume that there are three cells, each surrounded by 3x3 neighboring tiles. Two of them (left and middle) are IR-drop violations and the other (right) is a non-violation. In the neighboring tiles, the red one indicates a high-power tile and the green one indicates a low-power tile. When the XGBoost model grows the tree by splitting nodes for training, one node considers only one feature at a time. Suppose the model is using $tile_3$ to split a node. The model learns that: (1) a high-power tile leads to a violation; (2) a low-power tile leads to a violation; (3) a low-power tile leads to a non-violation. Obviously, there are conflicts between (1) and (2), and between (2) and (3), which confuses the model. This is mainly because the model learns the neighboring tiles separately, lacking a global view of them. Therefore, rather than learn all neighboring tiles in a local manner, we use *min.*, *max.* and *overall* tile features which have taken the relationship between neighboring tiles into account, providing the XGBoost model with the global view. In the above example, *max tile feature* and *overall tile feature* cause zero conflict. We currently have 15 (5 timing-related features x 3 tiles) tile-based features.

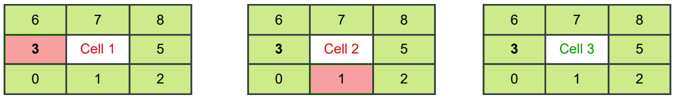


Fig. 3: Neighboring tiles of three cells

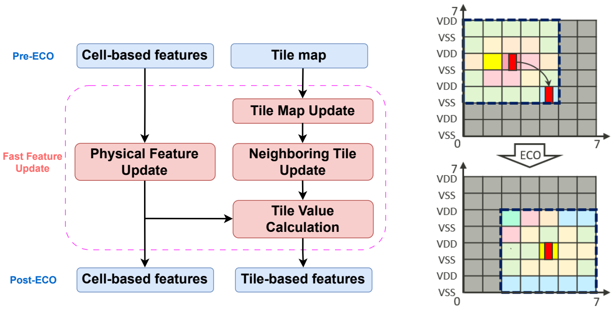
TABLE I: Selected tile-based features

CC rank	Tile-based features	Description
1	*Max. IR tile*	The maximum tile among all neighboring tiles of IR-drop
2	*Overall IR tile*	Sum of all neighboring tiles of IR-drop
3	*Min. IR tile*	The minimum tile among all neighboring tiles of IR-drop
4	*Overall P_{switch} tile*	Sum of all neighboring tiles of switching power
5	*Overall $P_{internal}$ tile*	Sum of all neighboring tiles of internal power
6	*Overall TR_{out} tile*	Sum of all neighboring tiles of output toggle rate
7	*Max. TR_{out} tile*	The maximum tile among all neighboring tiles of output toggle rate

In the final step, to mitigate the curse of dimensionality and overfitting during learning, we compute the correlation coefficients (CC) between each tile-based feature and the golden IR-drop, and select the top half of these features as our final set of tile-based features. The selected tile-based features are listed in TABLE I, and the results reveal that the regional IR-drop is highly correlated with the cell's IR-drop, which is overlooked by previous works. In the end, our final feature set has a total of 21 features (14 cell-based features + 7 tile-based features). Compared with previous works, we use fewer features to represent the surrounding information more globally.

C. XGBoost-Based IR-Drop Predictor

XGBoost [4] is a decision tree ensemble based on gradient boosting and it is a state-of-the-art machine learning model utilized for supervised learning. In this work, we use XGBoost as our regression model since (1) it is faster than other supervised learning such as Random Forest and Neural Network (NN) models; (2) it can achieve IR-drop prediction for each cell; (3) it is explainable, making it easier to optimize; (4) XGBoost-based IR-drop predictor is used in existing works, such as [7], [8], [13], and shows the competitive result of prediction.



(a) Flowchart

(b) Example of tile-based feature update

Fig. 4: Fast feature update

We have a training dataset $D = \{(x_i, y_i) : |D| = N, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$, where N is the number of cells and m is the number of features in our final feature set ($m = 21$). $x_i = (x_i^{(1)}, x_i^{(2)}, ..., x_i^{(m)})^T$ denotes the i_{th} cell's feature set. y_i denotes the golden IR-drop of the i_{th} cell. Let $\hat{y_i}$ be the predicted IR-drop of the ensemble tree model, which is generated as follows:

$$\hat{y_i} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{1}$$

f_k denotes the function of the k_{th} tree model. K is the number of tree models. F represents the set of functions of all trees. We need to find the best set of functions above by minimizing the objective function as follows:

$$L(\phi) = \sum_{i=1}^{N} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

l is a loss function that indicates the difference of the predicted IR-drop $\hat{y_i}$ and the golden IR-drop y_i, and Ω is a measure of model complexity to avoid overfitting. We employ asymmetric Mean Square Error (MSE) as our loss function to deal with the imbalanced data set. When violation cells are underpredicted, we apply a penalty to MSE as shown in Eq. 3.

$$\alpha\left(1 + \frac{y_i - \hat{y_i}}{y_i - thrs}\right) \cdot \|\hat{y_i} - y_i\|^2 \tag{3}$$

α is a penalty factor and $thrs$ is the violation threshold. Within the fraction part, the numerator indicates the magnitude of underprediction, and the larger value results in a greater penalty. The denominator indicates how close the golden IR-drop is to the threshold. The closer value results in a greater penalty as it is more likely to be misclassified as non-violation.

D. Fast Feature Update

After designers perform cell moving-based ECO, the features corresponding to the moved cells change. It is time-consuming to re-conduct power analysis and partial rail analysis to update the new feature values for prediction. Thus, an alternative to obtain the new feature values without the need to run simulations is desirable.

TABLE II: Design information

Technology node	3nm
Cell count	1,033,849
Chip size	$6984 \times 4135 \ (\mu m \times \mu m)$
Supply voltage	$0.8V$
IR violation threshold	$120mV$

We propose a simple yet powerful feature update method that utilizes the original information before cell moving to estimate the new feature values after cell moving. The flowchart is shown in Fig. 4(a), where the left and right parts are responsible for the cell-based and tile-based feature updates, respectively. In the cell-based feature update, since moved cells do not change their cell types, the electrical characteristics of the cell itself are affected slightly. Therefore, given the cell-based features of the initial circuit as input, we only update the physically related features, including cell location and cell size, which can be immediately obtained without simulation. The other cell-based features remain the same as those before ECO.

As for the tile-based feature update, moving cells mainly alters the relationships between cells and their surrounding cells, so the tile-based features, which represent the neighboring information, are our focus for update. There are three steps. Firstly, given the *tile map* constructed in the feature extraction stage, we update it to account for layout changes resulting from moved cells. This is accomplished by removing each moved cell from its original tile before ECO and adding it to the new tile it moves to after ECO. Secondly, due to the update of *tile map* or the movement of cells, the neighboring tiles for both moved and non-moved cells may change and need to be updated. Thus, we iterate over each cell, determine the corresponding tile where the cell is located, and redefine the 5x5 tiles surrounding the tile in the updated tile map as the new neighboring tiles. Finally, after updating the neighboring tiles, we calculate the tile value and extract the tile-based features listed in TABLE I, as described in Section III-B2. Note that the timing-related features, I_{peak}, P_{switch}, $P_{internal}$, TR_{out}, and IR of cells used to calculate the tile value, are taken from the input cell-based features, as these features are considered unchanged after ECO in our estimation method. In Fig. 4(b), we use a small tile-based design to demonstrate the tile-based feature update. Before ECO, the red cell was originally in $tile_{2,4}$, and the surrounding 5x5 tiles were its neighboring tiles. After ECO, the red cell moves to $tile_{4,2}$, and the colors of $tile_{2,4}$ and $tile_{4,2}$ change due to the tile map update. Then, the new neighboring tiles are updated to the 5x5 tiles surrounding $tile_{4,2}$.

The time complexity of the fast feature update is dominated by the tile value calculation, where each cell needs to interact with its neighboring cells to determine tw overlaps and summation. The number of neighboring cells is bounded by a constant because it is independent with the cell count n in the design. Therefore, the estimation method can update the features in linear time.

IV. Experimental Results

A. Experiment Setup

In the experiments, we used Cadence Voltus [1] to conduct power analysis and rail analysis to obtain the features and the

TABLE III: ECO candidates

ECO version	# Moved cells	# Moved cells in $\triangle$ IR intervals (mV)			# Fixed cells	Max IR-drop (mV)
		0-20	20-40	40-80		
ECO0	-	-	-	-	-	162.62
ECO1-1	184	52	76	56	168	142.64
ECO1-2	195	74	75	46	167	139.01
ECO1-3	192	56	86	50	166	142.40
ECO1-4	159	55	60	44	133	144.64
ECO1-5	179	75	72	32	152	155.67

TABLE IV: Training data preprocessing

Preprocessing step	# Cells	# Vio.	# Non-vio.	#Vio. : #Non-vio.
Initial	1,033,849	233	1,033,616	1:4436
Create regions	12,219	233	11,986	1:51
Oversampling	13,184	1,198	11,986	1:10

IR-drop report. We utilized Synopsys Tweaker [14] as the ECO tool for automated cell moving. Our method was implemented on a machine with an NVIDIA GeForce RTX 3090 GPU.

We evaluated the proposed methods on an industrial design with $3nm$ technology. The information about the design is shown in TABLE II. A cell is considered a violation cell if its IR-drop exceeds $120mV$, which corresponds to 15% of the supply voltage $0.8V$. To illustrate the scenario where our proposed methods can concurrently explore multiple ECO versions within a single iteration, we took the industrial design as the initial design, denoted as ECO0, and performed cell moving separately on ECO0 to generate five ECO candidates, ECO1-1 to ECO1-5. ECO1-1 to ECO1-3 were generated through manual cell moving by designers, and ECO1-4 and ECO1-5 were generated with Tweaker.

TABLE III presents the statistics of these ECO versions. Column 2 shows the number of moved cells in each ECO version. All the moved cells in ECO1-1 to ECO1-5 are originally violation cells in ECO0. Columns $3 \sim 5$ show the numbers of moved cells within different IR-drop variation intervals caused by ECO. Column 6 shows the number of fixed cells in each ECO version. The final column shows the maximum IR-drop value of a cell in each ECO version. For example, in ECO1-1, there are 184 moved cells after ECO. Among them, there are 52, 76, and 56 cells whose IR variations are within $0 \sim 20mV$, $20 \sim 40mV$, and $40 \sim 80mV$, respectively. Additionally, 168 out of the 184 violation cells are fixed. In ECO1-1, the maximum IR-drop value is $142.64mV$.

B. Real-Time IR-Drop Prediction for ECOs

We trained the dynamic IR-drop predictor with ECO0 and tested it on moved cells in the other ECO versions. Since the industrial design is nearing tape-out with only a few violation cells, the training data is highly imbalanced, which may lead to a biased model. Therefore, we preprocess the training data to address the issue. The changes in the cell numbers are outlined in TABLE IV. First, we frame a $6\mu m \times 6\mu m$ region around each violation cell, and only the cells covered in the regions are collected to be the training data. This significantly reduces the number of non-violation cells, making a more balanced dataset. Second, due to the limited number of violation cells, we utilize the Synthesized Minority Oversampling Technique (SMOTE) [2] to generate pseudo-violation cells.

TABLE V shows the prediction results. In the five ECO versions, the majority of moved cells are fixed. To check if the trained model can capture the trend within them, we define the confusion matrix as follows:

- TP: Fixed cells are predicted as fixed cells.
- FP: Non-fixed cells are predicted as fixed cells.
- TN: Non-fixed cells are predicted as non-fixed cells.
- FN: Fixed cells are predicted as non-fixed cells.

$Recall = TP/(TP + FN)$ indicates the proportion of fixed cells that were correctly detected; $Precision = TP/(TP + FP)$ indicates the proportion of cells correctly identified as fixed among all cells detected as fixed. We achieve an average recall of approximately 96% and an average precision of approximately 93%, demonstrating that our predictor can effectively capture the IR-drop variations of the moved cells through fast feature update. Then, we evaluate the prediction accuracy with mean absolute error (MAE), maximum absolute error (MaxE), and correlation coefficient (CC). The MAE is further classified into three classes to analyze the errors of all, fixed, and non-fixed cells within moved cells, respectively. The average MAE is $8.75mV$ for all moved cells, $7.41mV$ for non-fixed cells, and $8.96mV$ for fixed cells. The MAE for fixed cells is higher than that for non-fixed cells. This is mainly because fixed cells have larger IR-drop variations, compared with non-fixed cells after ECO, posing a greater challenge for prediction. Even so, the average MAE for fixed cells is less than $10mV$, which is within the acceptable error range for the industry.

For the evaluation time comparison, Voltus spends about 2 hours on power analysis and rail analysis, while our real-time method only spends 1.5 minutes on fast feature update and IR-drop inference.

Fig. 5 shows the scatter plots of the prediction results of five ECO candidates. x-axis is the golden IR-drop and y-axis is the predicted IR-drop. The red line is the perfect prediction line, and the two black lines next to it form the $\pm$ $10mV$ error range. The two green lines, which are the IR violation thresholds, divide the scatter plot into four quadrants. Before ECO, we can imagine that all moved cells (green dots) are in the upper right quadrant, and most of them should move to the lower left quadrant after ECO. As shown in the scatter plots, we successfully capture the IR-drop changes, predicting the majority of cells to the lower left quadrant.

Fig. 6 visualizes the relations among IR-drop before ECO (red), golden IR-drop after ECO (green), and predicted IR-drop after ECO of moved cells in ECO1-2. The x-axis represents the index of the moved cell, with a total of 195 cells, and the y-axis represents the IR-drop value.

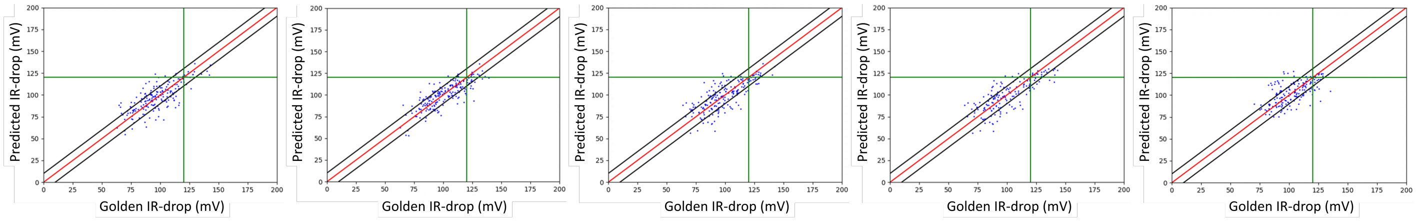
C. Local vs. Global Tile-based Features

In Section III-B2, we introduced the features to capture neighboring information. The local tile-based features are first constructed. Then we propose the global tile-based features, and then further extract the global tile-based features having higher CC with IR-drop.

TABLE VI takes the local approach adopted from the work [7] as a baseline and compares our two global approaches with the baseline using the metrics MAE and CC. For the local approach, the average MAE is $10.96mV$ exceeding the acceptable error range of $10mV$. The average CC is 64% and the CC in ECO1-5 is even lower than 50%. In contrast, for our global approach, the average MAE is $9.75mV$ and the average CC is 70.5%, where the improvements are 11% and

TABLE V: Prediction results of moved cells after ECO

ECO cand.	# Moved cell	# Fixed cell	TP	FP	TN	FN	**Recall**	**Precision**	All MAE (mV)	Non-fixed MAE (mV)	Fixed MAE (mV)	MaxE (mV)	CC	Voltus eval. time	Our eval. time
ECO1-1	184	168	162	3	13	6	0.964	0.982	8.84	7.20	9.00	31.08	0.760	127.10m	1.45m
ECO1-2	195	167	162	15	13	5	0.970	0.915	7.60	8.33	7.48	33.23	0.810	100.97m	1.45m
ECO1-3	192	166	161	11	15	5	0.970	0.936	9.08	5.62	9.62	32.87	0.757	129.72m	1.43m
ECO1-4	159	133	129	8	18	4	0.970	0.942	9.68	8.22	9.96	30.01	0.773	99.77m	1.47m
ECO1-5	179	152	142	14	13	10	0.934	0.910	8.56	7.69	8.72	33.39	0.641	127.23m	1.52m
Avg.	182	157	151	10	14	6	**0.962**	**0.937**	8.75	7.41	8.96	32.12	0.748	116.95m	1.47m



(a) ECO1-1 (b) ECO1-2 (c) ECO1-3 (d) ECO1-4 (e) ECO1-5

Fig. 5: Scatter plots of moved cells after ECO

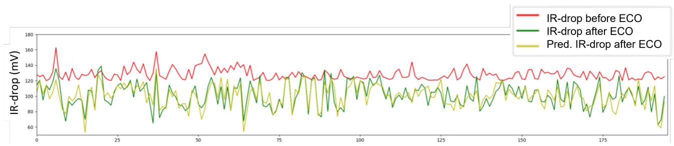


Fig. 6: Line chart of moved cells after ECO

TABLE VI: Tile-based features comparison

ECO cand.	Local [7]		Global		Global with CC ranking	
	MAE (mV)	CC	MAE (mV)	CC	MAE (mV)	CC
ECO1-1	11.01	0.630	9.85	0.720	8.84	0.760
ECO1-2	10.65	0.679	8.73	0.764	7.60	0.810
ECO1-3	10.75	0.674	9.83	0.723	9.08	0.757
ECO1-4	10.54	0.731	10.63	0.737	9.68	0.773
ECO1-5	11.86	0.484	9.72	0.580	8.56	0.641
Avg.	10.96	0.640	9.75	0.705	8.75	0.748
Impr.(%)	-	-	11.0	10.2	20.2	16.9
# Tile features	125		15		7	

10.2%, respectively, compared to the baseline. Furthermore, when we further select the top half of the features based on the CC ranking, the average MAE is $8.75mV$ and the average CC is 74.8%, showing the improvements of 20.2% and 16.9%, respectively, compared to the baseline. Additionally, the number of features in these three approaches decreases from 125 to 15, and finally to 7. In summary, by employing global tile-based features and removing features that are less relevant to IR-drop, we can not only reduce the feature dimension but also make significant improvements in accuracy for the XBGoost predictor.

D. Evaluation Time Comparison

TABLE VII compares the evaluation time of Voltus, traditional ML, and our real-time ML in ECO1-1 with 1.03 million cells. In Voltus, the evaluation time includes power analysis and rail analysis; In traditional ML, it includes (1) partial simulations: run power analysis and partial rail analysis, (2) feature extraction: transform raw data to the features, and (3) ML inference: IR-drop prediction; In real-time ML, it includes fast feature update and ML inference. Our real-time ML takes only 1.45 minutes for each ECO candidate, achieving 88X speedup over Voltus and 65X speedup over traditional ML.

TABLE VII: Evaluation time comparisons

Step	Voltus	Traditional ML	Real-time ML
Partial simulations	127.1 mins	87 mins	1.45 mins
Feature extraction		7.1 mins	
ML inference		0.014 secs	0.014 secs
Eval. time	127.1 mins	94.1 mins	1.45 mins

Additionally, considering the real-world scenario in industrial IR sign-off flow, designers must experience multiple ECO candidates within an iteration. Thus, the evaluation time for multiple ECO candidates is crucial. Due to the time and resource-intensive nature of simulations, both Voltus and traditional ML (Fig. 1) are unable to evaluate multiple ECO candidates in parallel. Therefore, if there are n ECO candidates, the evaluation time of Voltus would be $127.1 \times n$ minutes, and that of traditional ML would be $87 \times n + 7.1$ minutes. In contrast, since real-time ML (Fig. 2) does not require any simulations, ECO candidates can be evaluated in parallel, so that the evaluation time would ideally remain 1.45 minutes.

The result shows that using Voltus or traditional ML to the industrial IR sign-off flow is very time-consuming and designers have to sacrifice considering multiple ECO candidates to speed up the process. In contrast, our real-time ML can not only obtain the IR-drop results in a very short time but also allow designers to explore more ECO candidates within an iteration, increasing the probability of selecting a better ECO version for the next iteration.

V. Conclusion

We propose a real-time dynamic IR-drop estimation method based on an XGBoost model equipped with the global view of cells' neighboring tiles. After ECO, our method produces the dynamic IR-drop result without the need for any simulations. This enables designers to concurrently evaluate multiple ECO candidates within a single iteration to facilitate IR sign-off. The results show that we can capture the dynamic IR-drop variations in five ECO candidates in a very short time compared to Voltus and the traditional ML predictors, with $< 10mV$ MAE overhead.

References

[1] Cadence. Voltus.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

[3] Jia-Xian Chen, Shi-Tang Liu, Yu-Tsung Wu, Mu-Ting Wu, Chieo-Mo Li, Norman Chang, Ying-Shiun Li, and Wen-Tze Chuang. Vector-based Dynamic IR-drop Prediction Using Machine Learning. In *2022 27th Asia and South Pacific Design Automation Conference*, pages 202–207, 2022.

[4] 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*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[5] 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 *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 690–696, 2021.

[6] 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, 2021.

[7] 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*, page 1–7. IEEE Press, 2018.

[8] 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, 2019.

[9] Xuan-Xue Huang, Hsien-Chia Chen, Sheng-Wei Wang, Iris Hui-Ru Jiang, Yih-Chih Chou, and Cheng-Hong Tsai. Dynamic IR-Drop ECO Optimization by Cell Movement with Current Waveform Staggering and Machine Learning Guidance. In *2020 IEEE/ACM International Conference On Computer Aided Design*, pages 1–9, 2020.

[10] Santanu Kundu, Manoranjan Prasad, Sashank Nishad, Sandeep Nachireddy, and Harikrishnan K. 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*, 2022.

[11] 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, 2021.

[12] Shih-Yao Lin, Yen-Chun Fang, Yu-Ching 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, 2018.

[13] Chi-Hsien Pao, An-Yu Su, and Yu-Min Lee. XGBIR: An XGBoost-based IR Drop Predictor for Power Delivery Network. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1307–1310, 2020.

[14] Synopsys. Tweaker.

[15] 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, 2020.