Machine Learning Prediction for Design and System Technology Co-Optimization Sensitivity Analysis

Chung-Kuan Cheng, *Life Fellow, IEEE*, Chia-Tung Ho, *Graduate Student Member, IEEE*, Chester Holtz, *Student Member, IEEE*, Daeyeal Lee, *Student Member, IEEE*, and Bill Lin, *Member, IEEE*

***Abstract*—As technology nodes continue to advance relentlessly, geometric pitch scaling starts to slow down. In order to retain the trend of Moore's law, design technology co-optimization (DTCO) and system technology co-optimization (STCO) are introduced together to continue scaling beyond 5 nm using pitch scaling, patterning, and novel 3-D cell structures [i.e., complementary-FET (CFET)]. However, numerous DTCO and STCO iterations are needed to continue block-level area scaling with considerations of physical layout factors: 1) various standard cell (SDC) library sets (i.e., different cell heights and conventional FET); 2) design rules (DRs); 3) back end of line (BEOL) settings; and 4) power delivery network (PDN) configurations. The growing turnaround time (TAT) among SDC design, DR optimization, and block-level area evaluation becomes one of the major bottlenecks in DTCO and STCO explorations. In this work, we develop a machine learning model that combines bootstrap aggregation and gradient boosting techniques to predict the sensitivity of minimum valid block-level area of various physical layout factors. We first demonstrate that the proposed model achieves 16.3% less mean absolute error (MAE) than the previous work for testing sets. Then, we show that the proposed model successfully captures the block-level area sensitivity of new SDC library sets, new BEOL settings, and new PDN settings with 0.013, 0.004, and 0.027 MAE, respectively. Finally, compared to the previous work, the proposed approach improves the robustness of predicting new circuit designs by up to 6.76%. The proposed framework provides more than 100× speedup compared to conventional DTCO and STCO exploration flows.**

***Index Terms*—Cell synthesis, complementary-FET (CFET), design technology co-optimization (DTCO), DTCO and STCO sensitivity prediction, machine learning (ML), standard cell (SDC), system technology co-optimization (STCO).**

I. Introduction

As VLSI technology continues to advance relentlessly beyond 5 nm, geometric pitch scaling starts to slow down. Moreover, design technology co-optimization (DTCO) [1] based on pitch scaling and patterning is unable to continue the cost scaling in 2-D IC technology. In order to keep the trend of Moore's law, system technology co-optimization (STCO) has been introduced to assist DTCO scaling with 3-D integrated logic and novel 3-D cell structure (CS) [i.e., complementaryFET (CFET)] [2], [3] beyond 5 nm. However, technology development beyond sub-5 nm demands enormous engineering effort for identifying the optimal technology options (i.e., evaluation of cost and determination of standard cell (SDC) heights, 2-D/3-D SDC architectures, design rules (DRs), power delivery networks (PDNs), and back end of line (BEOL) settings). Furthermore, process architects must be aware of the impact of the technology transition on the power, performance, area, and cost (PPAC) for further optimization. Therefore, finding the optimal technology option necessitates numerous DTCO and STCO iterations among SDC optimization, DR optimization, and block-level area evaluation. This results in exploding turnaround time (TAT) in DTCO and STCO explorations. There is a high demand for a holistic, fast, and robust prediction methodology that provides information on the potentially optimal technology options and the impact on PPAC from the technology transition.

Manuscript received 20 December 2021; revised 18 March 2022 and 18 April 2022; accepted 2 May 2022. Date of publication 13 May 2022; date of current version 26 July 2022. The work of Chung-Kuan Cheng and Bill Lin was supported by NSF under Grant CCF-2110419 and Grant IIS1956339. *(Corresponding author: Chia-Tung Ho.)*

Chung-Kuan Cheng and Chester Holtz are with the Department of Computer Science, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: ckcheng@ucsd.edu; chholtz@ucsd.edu).

Chia-Tung Ho, Daeyeal Lee, and Bill Lin are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: c2ho@ucsd.edu; ldaeyeal@ucsd.edu; billlin@ucsd.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2022.3172938.

Digital Object Identifier 10.1109/TVLSI.2022.3172938

A. DTCO and STCO Frameworks

Song *et al.* [4] proposed a unified technology platform using integration analysis for DTCO and STCO at sub7-nm node. Kahng *et al.* [5] proposed a routability metric k_{th} to evaluate the routing capacity of BEOL stacks, but this work lacks of explorations on various CSs and does not provide the change of block-level metrics (i.e., area) from the technology transition. Recently, in [6], a novel DR evaluation technique using automatic cell layout generation for DTCO exploration is proposed, but the focus is limited to conventional FET (Conv. FET) structures (i.e., FinFET). Cheng *et al.* [7] also proposed a novel automatic CFET cell layout synthesis framework for DTCO and STCO explorations. However, these works use conventional block-level placementand-route (P&R) to evaluate block area and thus result in longer TAT for DTCO and STCO technology development.

B. Machine Learning (ML)-Based DTCO and STCO Approaches

Recently, many ML-based DTCO and STCO approaches have been proposed to shorten the DTCO and STCO exploration time. In [8], an ML-based modeling framework is developed to generate compact models of novel devices, but this work does not consider block-level evaluations. Ceyhan *et al.* [9] used ML techniques to search and find

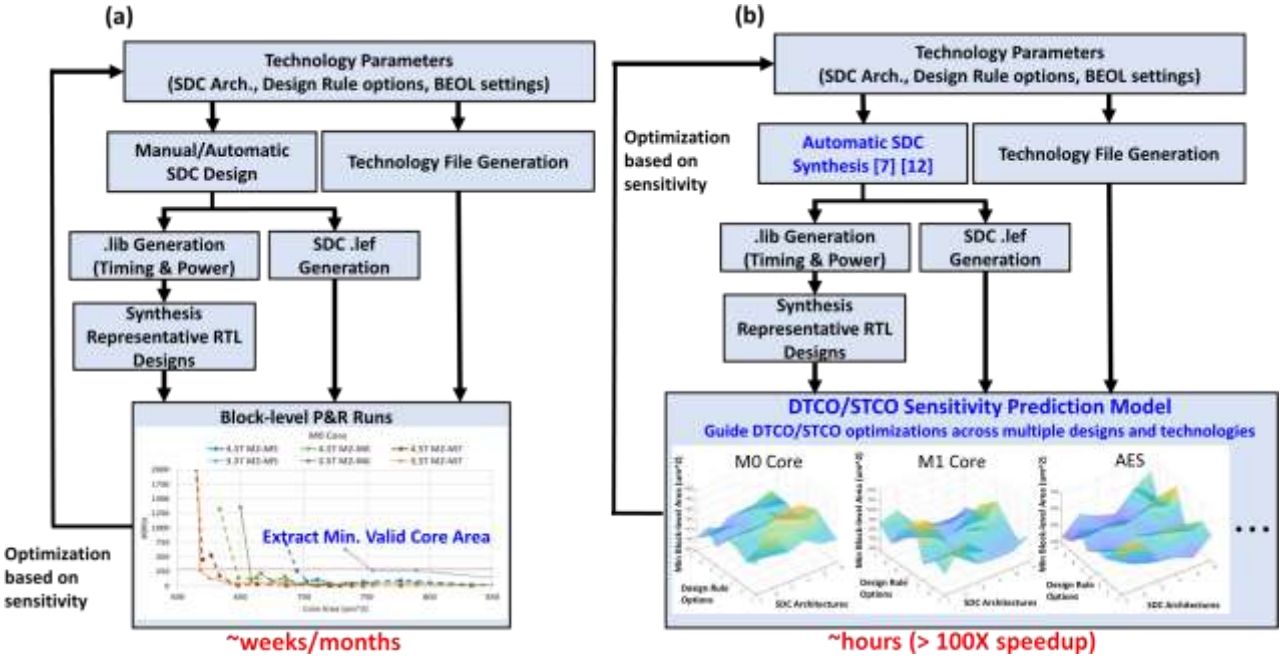


Fig. 1. Illustrations of (a) traditional DTCO and STCO exploration flow and (b) proposed DTCO and STCO sensitivity prediction framework.

optimal combinations of design, technology, and flow recipes for high-performance CPU designs in the enormous solution space, but its performance on 3-D SDC architectures (i.e., CFET) has not been explored, and the methodology requires 4–6 weeks of TAT. Recently, Cheng *et al.* [10] extended routability metric, k_{th}, to cell level and block level and applied ML-assisted prediction on the k_{th} for various technology options. However, we focus on exploring technology options with Conv. FET SDC structure and does not predict/provide the changes in block-level metrics (i.e., area) induced by the technology transition from one option to another in this work. In [11], a modeling approach for DTCO and STCO sensitivity prediction has been proposed, but they performed limited exploration on ML models.

In this article, we propose a novel DTCO and STCO sensitivity prediction framework, which provides information on the change/gradient of block-level metrics from the technology transition. Also, we develop an ML model that combines bootstrap aggregation and gradient boosting techniques to improve the prediction accuracy. Fig. 1(a) and (b) shows the difference between the traditional DTCO and STCO exploration flow and the proposed DTCO and STCO sensitivity prediction framework. The proposed prediction model and automatic cell synthesis [7], [12] significantly reduce the TAT of DTCO and STCO explorations on various physical layout factors: 1) SDC library sets (i.e., different cell heights (CHs), Conv. FET, and CFET SDC architectures); 2) DRs; 3) BEOL parameters; and 4) PDN configurations. In this work, we focus on the sensitivity of block-level area variations according to different technology features and demonstrate the feasibility of ML techniques in DTCO and STCO exploration flows.[1] Our main contributions are given as follows.

1) We propose a novel DTCO and STCO sensitivity prediction framework that improves the efficiency of explorations by orchestrating the proposed ML model and automatic cell synthesis [7], [12].
2) We develop an ML model using bootstrap aggregation and gradient boosting techniques to predict the change/gradient of block-level metrics from the technology transition.
3) We perform extensive studies on various ML algorithms for block-level area sensitivity prediction and demonstrate that the developed ML model outperforms other ML algorithms on DTCO and STCO sensitivity prediction.
4) We identify key features of each SDC and extract cell and block-level features for prediction. We validate the extracted features via feature importance analysis in Exp. III-B.
5) We perform extensive studies on model accuracy for new technologies and model robustness for new designs across Conv. FET and CFET SDC architectures, and various CHs, DRs, PDNs, and BEOL settings.

The remaining sections are organized as follows. Section II describes our DTCO and STCO sensitivity prediction approach. Section III presents our main experiments. Section IV concludes this article. Section V discusses the important directions of future research.

II. Design and System Technology Co-Optimization Sensitivity Prediction Framework

We apply ML techniques to predict the sensitivity of DTCO and STCO explorations on block-level areas considering

[1] DTCO and STCO sensitivity prediction for incorporating block-level power and performance are one of the future works as discussed in Section V.

various physical layout factors: 1) SDC architectures (e.g., CH, multirow/single-row, CFET, and Conv. FET.); 2) DRs; 3) BEOL parameters; and 4) PDN configurations. In this section, we describe the specifics of our prediction methodology: 1) DTCO and STCO sensitivity; 2) overall modeling flow; 3) methodology for feature extraction; 4) input features; and 5) ML techniques.

A. DTCO and STCO Sensitivity

The DTCO and STCO sensitivity for block-level area of two technologies of a block-level circuit is the percentage of the block-level area difference of these two technologies, $A_{i,j}$, as shown in the following equation:

$$A_{i,j} = (A_i - A_j)/A_i \quad (1)$$

where A_i and A_j are the minimum valid block-level areas of the ith technology and the jth technology, respectively. Here, a technology is the combination of SDC library set, DRs, BEOL parameters, and PDN configuration. Fig. 2(a) and (b) shows an example of DTCO and STCO block-level area sensitivity of #BEOLs and DRs using 4.5T and 3.5T CFET SDC library sets of AES and M0 Core block-level circuits, respectively. Different SDC library sets, #BEOLs, and DRs can potentially impact the block-level area up to 17.9%. The importance of knowing the information of change/gradient of block-level area from the technology transition is needed for holistic technology development.

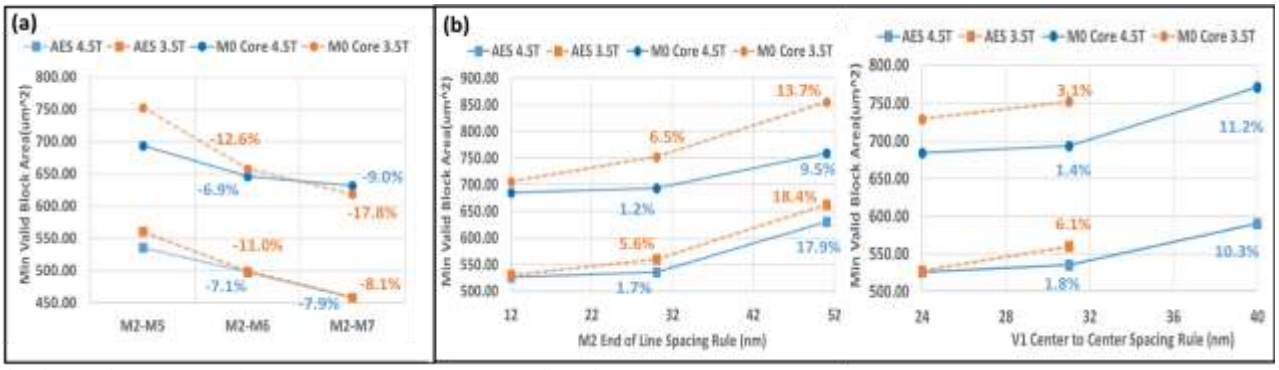
In this work, we focus on studying the minimum block-level area of various CHs, cell architectures (i.e., CFET and Conv. FET), cell pin accessibility, DRs, BEOL parameters, and PDN structures and develop a model to predict $A_{i,j}$ for reducing the TAT of DTCO and STCO explorations.

B. Overall Modeling Flow

Fig. 3 shows the proposed training flow and prediction model for DTCO and STCO exploration flow. In the training phase as shown in Fig. 3(a), we generate multiple SDC library sets, BEOL parameters, and power delivery configurations to perform multiple block-level P&R runs with the synthesized block-level circuits through a commercial P&R suite [13]. The minimum valid block-level area of a technology combination (i.e., SDC library set, DRs, and BEOL combination) is extracted with 300 DR violations (#DRVs)[2] with multiple P&R runs, as shown in Fig. 1(a). Then, the percentage of the block-level area difference of two technologies (i.e., $A_{i,j}$) are extracted in the feature extraction stage for training.

We show the prediction model for DTCO and STCO exploration flow in Fig. 3(b). Prediction flow utilizes the same input types with new technology parameters to explore. The proposed DTCO and STCO sensitivity prediction model outputs the predicted $A_{i,j}$.

Fig. 2. Example of DTCO and STCO block-level area sensitivity of (a) #BEOLs and (b) DRs using 4.5T and 3.5T CFET SDC library sets of AES and M0 Core circuits. The number represents the block-level area difference as changing DTCO and STCO parameters from left to right. Many 3.5T CFET SDCs (i.e., NAND2 × 2 and NAND3 × 1) do not have feasible solutions when V1 center-to-center spacing is 40 nm [7]. As a result, there are no data points of the block-level area of 3.5T CFET using 40-nm V1 center-to-center spacing rule.



In our envisioned usage scenario, technology developers define and generate multiple circuit designs, SDC library sets, tech lef files (.tf), and PDN configurations for DTCO and STCO explorations. The proposed framework assists and guides the technology tuning process to find one of the optimal technology candidates by predicting the gradient of the blocklevel area, $A_{i,j}$, for block-level area cost evaluation. With the predicted $A_{i,j}$ of all the technology pairs, technology developers can find the technology, which provides the largest improvement on block-level area metric compared to the baseline technology. If the selected technology combination is a new technology, which involves systematic physical layout change (i.e., backside PDN technology), to the prediction model, the block-level P&R is launched to extract the minimum valid block-level area and the data are used to update the prediction model; otherwise, technology developers adopt the selected technology for the next phase in technology development.

C. Methodology for Feature Extraction

We describe the feature extraction component of our framework. Table I summarizes four categories of input features: 1) synthesized block-level circuit statistics; 2) SDC architectures; 3) BEOL parameters; and 4) PDN configurations.

1) Synthesized Block-Level Circuit Statistics: We extract the statistics of the block-level circuit, which is derived after logic synthesis and before physical layout. The data include

[2] As a common industrial practice, once the number of DRVs increases beyond 300, the block layout is deemed too troublesome to fix with laborious engineering change orders (ECOs).

circuit structures, instance numbers, and SDC area from the synthesized block-level circuit. For circuit structures, we consider the distribution of fan-out counts (#fanouts), Rent's multiplier k, and exponent p component of Rent's rule [14]. These terms define an empirical power-law relationship between the number of gates "N" and the number of terminals "T" as shown in the following equation:

$$T = kN^{p}. \tag{2}$$

For each circuit, we extract the (T, N) pairs and perform linear regression to obtain k and p for each design. In addition, we extract the number of fanouts per net (#fanouts), the number of sequential cells (#Seq), the number of combinational cells (#Comb), the number of buffers (#Buf), and SDC area from the report of synthesis tool.

2) SDC Architectures: We extract key metrics that impact routability at the block level and lead to larger minimum valid block-level area, such as average remaining pin access (RPA) value [15], [16] of I/O pins, the number of M2 Track usage (#M2Track) [7], and M2 metal length ($M2ML$) [7] in the cell level. Then, we calculate the block weighted RPA_d, M2Track_d, and M2ML_d with the corresponding SDC metrics and cell percentage of the synthesized block-level circuit d using the following equation [7]:

$$\text{Metric}_d = \sum_c \text{Metric}_c * CP_{d,c} \tag{3}$$

where Metric_d denotes the block weighted metric of design d. Metric_c is the cell level metric of cell c, such as average RPA value, #M2Track, and M2ML. $\text{CP}_{d,c}$ is the percentage of cell c in the synthesized block-level circuit d. In addition, we use CH as one of the features since the CH limits the horizontal routing tracks/resources, which shows greater impacts on SDC less than 5T [15], for accessing M1 pin in SDC.

3) BEOL Parameters: We introduce BEOL parameters related to the DR and BEOL settings. We use representative DRs such as min spacing rule, end-of-line (EOL) spacing rule, via rule (VR), same net VR, and fat metal spacing rule for metal and via layers as the input features. For BEOL settings, the pitch of each routing metal layer and the total number of routing metal layers (#BEOLs) are selected as the input features of our model.

4) Power Delivery Network Configurations: We categorized the PDN into front-side PDN and backside PDN categories [17]. For front-side PDN, we mainly study the M3 power strap period, which is critical to the power integrity and signal routing. With a denser M3 power strap, the IR drop will be improved, but it will result in poor routability and a larger core area because it takes more metal resources for signal routing. On the other hand, a sparser M3 power strap may lead to power integrity issue and causes functional failure. For backside PDN, we set the power strap period feature to a large

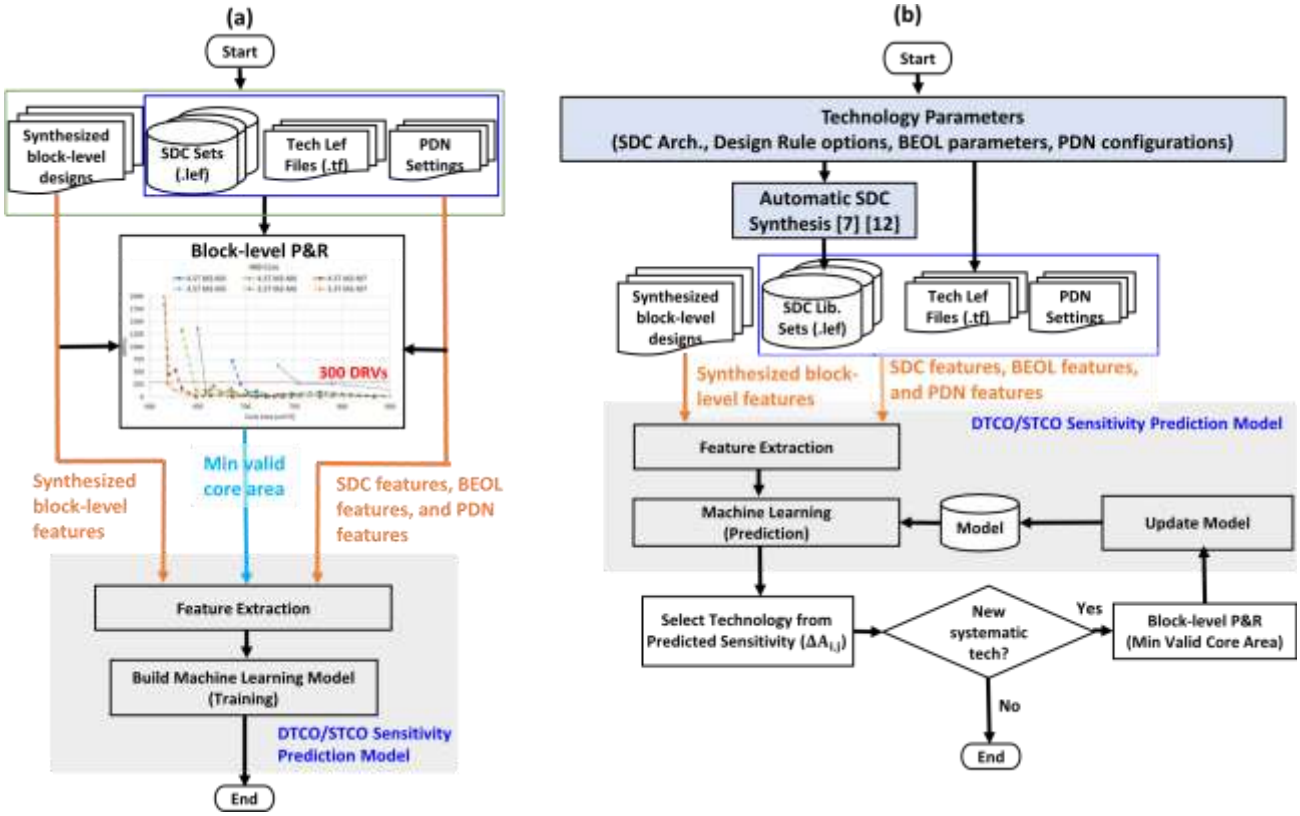


Fig. 3. Overall flow of DTCO and STCO sensitivity prediction: (a) Training flow and (b) prediction model for DTCO and STCO exploration flow. Technology developers can select the optimal technology candidate, which provides the largest improvement of block-level area metric compared to the baseline technology from the predicted $A_{i,j}$ in (b). If the selected technology is new technology, which brings systematic physical layout change at block level, the model can be updated with the block-level P&R data of the new technology.

number (i.e., $1e^6$) since there are no power straps on the front side at the block level.

D. Input Features

We describe the input features used to predict $A_{i,j}$ in (1). Fig. 4 shows an illustration of the input features of the proposed DTCO and STCO sensitivity prediction model. The input features consist of the extracted features, which are shown in Table I, of the *i*th and *j*th technologies.

E. ML Techniques

We develop our ML model with bootstrap aggregation and gradient boosting regression tree (GBRT) techniques to achieve state-of-the-art results on DTCO and STCO sensitivity prediction. We introduce the overview of the proposed model, the feature selection technique, and the modeling approach in the following.

1) Model Overview: Fig. 5 shows the developed ML model, which combines bootstrap aggregation and GBRT techniques. In the bootstrap aggregation technique, the bootstrap sampling is used to estimate statistics on a population by sampling a dataset with replacement and can be used to create meaningful simulated datasets to control the variance of a model. Then, the simulated datasets are used to train a set of GBRT models. Finally, the outputs of GBRT models are aggregated for predicting DTCO and STCO sensitivity. We use XGBoost [18] for implementing the GBRT models in the proposed model. XGBoost implements ML algorithms using a GBRT, which achieves state-of-the-art results on tabular data prediction. To avoid the structural similarity of GBRT trees and have a high correlation of their predictions, we set *colsample_bytree*[3] to 0.7 for each GBRT model. Finally, the final predicted $\hat{A}_{i,j}$ values are calculated by averaging the prediction of all GBRT models.

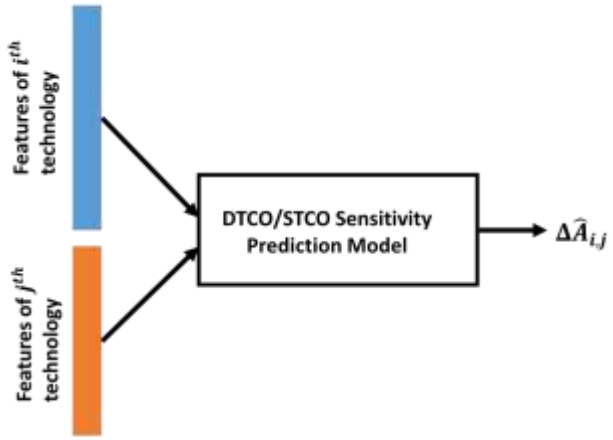


Fig. 4. Illustration of input features of the proposed DTCO and STCO sensitivity prediction model. Features of the *i*th and *j*th technologies are described in Table I.
Fig. 4.

TABLE I

EXTRACTED FEATURES TABLE

Feature Scope	Feature Types	Feature Name
Synthesized block-level circuit design statistics	Net complexity	#Fanouts
		Rent's multiplier
		Rent's exponent
	Instance	#Seq
		#Comb
		#Buf
	Synthesized Design Area	SDC area
SDC features	Block weighted SDC metric	RPA_d
		M2Track_d
		M2ML_d
	Horizontal Routing Resource	Cell Height
Design Rule & BEOL settings	Design Rules for Metal and Via layers	Min spacing
		EOL
		VR
		Same net VR
		Fat metal spacing
	BEOL settings	BEOL Pitches
		#BEOL
Power Delivery Network (PDN) features	PDN settings	M3 power strap period

Algorithm 1 Feature Selection

*/*Input: Data set D, and Feature set F; Output: Feature subset $\hat{F}$.*/*
1: Split data set D into 80% training set, T, and 20% validation set, V;
2: Train a model with T and V with F using GBRT with early stopping;
3: Get the validation error, E_{val};
4: Set $E_{val}^{min} = E_{val}$;
5: Set $\hat{F} = F$;
6: Set $F_{sub} = \{\}$;
7: Set F = Sort F based on the gain of features in descending order;
8: Set $m = |F|$;
9: **for** $i = 1, 2, \ldots, m$ **do**
10: Set $F_{sub} = F_{sub} + f_i$;
11: Extract F_{sub} from T and V to T_{sub} and V_{sub}, respectively;
12: Train a model with T_{sub} and V_{sub} with F_{sub} using GBRT with early stopping;
13: Get validation error, E_{val};
14: Calculate the VIF of each feature in data set T_{sub} and get $f_{i,vif}$ value;
15: **if** $E_{val} > E_{val}^{min}$ && $f_{i,vif} \geq VIF_{th}$ **then**
16: Remove f_i from F_{sub};
17: **end if**
18: **if** $E_{val} \leq E_{val}^{min}$ **then**
19: Set $\hat{F} = F_{sub}$;
20: Set $E_{val}^{min} = E_{val}$;
21: **end if**
22: **end for**
23: Return $\hat{F}$

2) Feature Selection Technique: We describe the feature selection technique here. First, we extract the feature importance of a trained GBRT model. Then, we use the variance inflation factor (VIF) [19] to detect instances of multicollinearity, which result in the high sensitivity to small changes in correlated features. Finally, we perform feature selection as described in Algorithm 1. Here, we use the "gain" for feature importance. The gain of a leaf node is the difference of metric before and after splitting at the leaf node [18]. The "gain" of the feature is the total gain of using the feature to split

[3] *colsample_bytree* is the fraction of features (randomly selected) that will be used to train each tree in the XGBoost library [18].

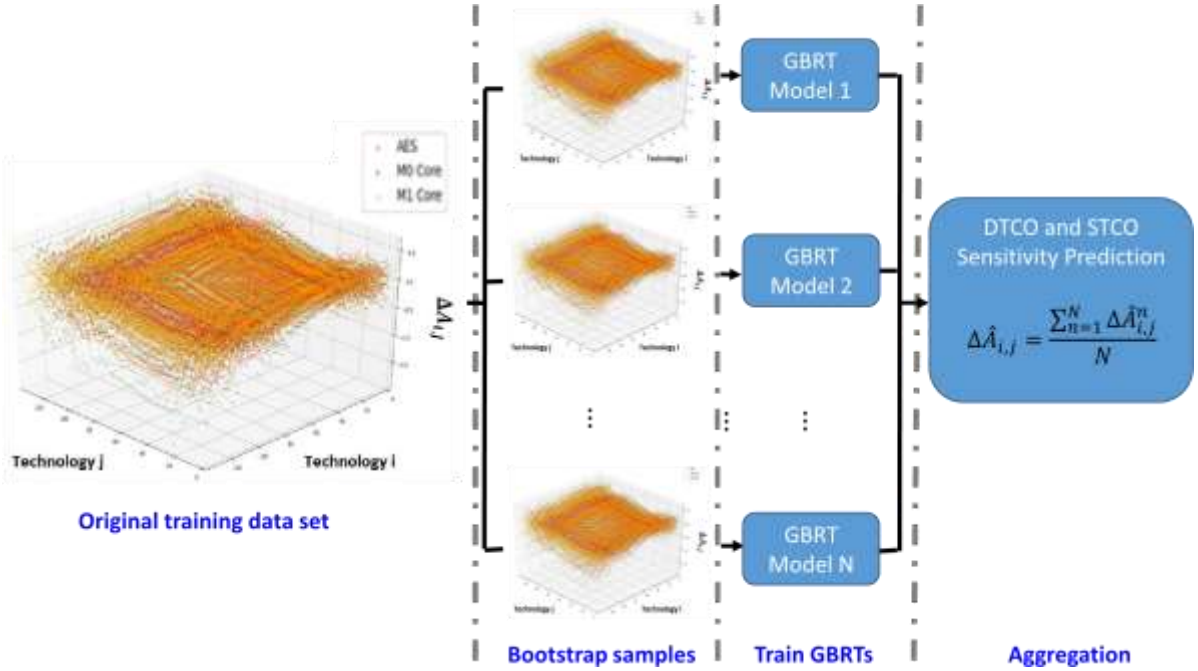


Fig. 5. Overview of the developed ML model. The model combines bootstrap aggregation and GBRT techniques.

nodes divided by the number of times the feature used to split a node. The feature selection technique reduces the average MAE by 0.02 (i.e., 25%) for new design prediction in Exp. III-E.

In Algorithm 1, first, we split the dataset D into the training set, T, and validation set, V (Line 1). We train a GBRT model with training and validation sets (Line 2). Then, we sort the features based on the gain of the GBRT model in the descending order (Line 7). After that, we sequentially add features to F_{sub} and train a GBRT model with selected features F_{sub} (Line 9–12). Then, we calculate VIF of each feature in dataset T_{sub} and extract the VIF, $f_{i,vif}$, of f_i (Line 13). If the validation error, E_{val}, is larger than the minimum validation error, E_{val}^{min}, and the VIF of f_i is larger than a VIF threshold, we remove f_i from F_{sub} (Line 14–16). If the validation error is smaller than the minimum validation error, we record F_{sub} as $F\hat{}$ and update the minimum validation error (Lines 17–20). Finally, we return the feature subset, $F\hat{}$, which has minimum validation error (Line 22).

3) Modeling Approach: We extract the input features from all the technologies as shown in Section II-D, compose all the technologies into pairs, and perform feature selection to compose a dataset $D = (x_{i,j}, A_{i,j})$, where $x_{i,j} \in \mathbb{R}^m$ corresponds to the m input features after feature selection and $A_{i,j} \in \mathbb{R}$ is the percentage of the ith and jth block-level area difference of technologies. We aim to predict $A_{i,j}$ using the developed ML model. D is resampled to generate N datasets, $D\hat{}^n$. We increase the number of samples until each bootstrap sample (i.e., $D\hat{}^n$) contains approximately 63.2% of the data points in the training set [20].

For each GBRT model, XGBoost sequentially builds an ensemble of K regressors. Predictions, $A\hat{}^n_{i,j}$, are made by taking the weighted sum of predictions made by the individual members of the ensemble as shown in the following equation:

$$\hat{A}^n_{i,j} = \sum_{k=1}^{K} g_k(x_{in,j}), \quad g_k \in G \tag{4}$$

where G is the space of regression trees and n represents the nth GBRT model. The goal is to minimize $L(A^n_{i,j}, \hat{A}^n_{i,j})$ in the following equation:

$$L(A^n_{i,j}, \hat{A}^n_{i,j}) = \sum_i l(A^n_{i,j}, \hat{A}^n_{i,j}) + \sum_k (f_k)$$

$$\text{where } (f) = \gamma T + \frac{1}{2}\lambda ||w|| \tag{5}$$

where each $l(A^n_{i,j}, \hat{A}^n_{i,j})$ is a differentiable convex function that measures the difference of $A^n_{i,j}$ and $\hat{A}^n_{i,j}$. We use the mean absolute error (MAE) as the evaluation metric. is a function that penalizes the complexity of the model. T is the number of leaves in the tree and w is the leaf weight. We use tenfold cross validation [21] to perform hyperparameter tuning (i.e., *min_child_weight*, and *etc*) to train our model. Then, to predict $A_{i,j}$, $\hat{A}_{i,j}$ is obtained using the average of the prediction results of N GBRT model, $\hat{A}^n_{i,j}$, as shown in the following equation:

$$\hat{A}_{i,j} = \frac{\sum_{n=1}^{N} \hat{A}^n_{i,j}}{N}. \tag{6}$$

III. Experimental Results

Our framework is implemented in Python and is executed on a workstation with 2.4-GHz Intel Xeon E5-2620 CPU and 256-GB memory. For the proposed model in Fig. 5, we implement the bootstrap sampling technique with sklearn library [22] and GBRT tree models with XGBoost library [18].

A. Experiment Setup

We use the synthesized block-level circuits, SDC library sets generated from [7] and [12], DRs, BEOL settings, and PDN configurations to generate the data for our experiments. We run multiple block-level P&R runs through a commercial test suite [13] and use a 300 #DRV threshold to measure the minimum valid block-level area of each synthesized blocklevel circuit for each technology combination.

1) Synthesized Block-Level Circuits: For synthesized blocklevel circuits, six open-source RTL designs [23], M0 Core, M1 Core, AES, MPEG, JPEG, and DarkRiscV that, respectively, have 17k, 20k, 14k, 18k, 45k, and 7k instances using 30 representative SDCs [7]. The worst negative slack (WNS) of each synthesized block-level circuit is carefully adjusted between ±50 ps for a fair comparison to study the change of

TABLE II SYNTHESIZED BLOCK-LEVEL CIRCUIT TABLE

Design Name	#Instance	Rent's multipliers (k)	Rent's exponent (p)
M0 Core	17k	2.69	0.73
M1 Core	20k	2.72	0.71
AES	14k	2.62	0.70
MPEG	18k	3.58	0.61
JPEG	45k	2.71	0.78
Darkriscv	7k	5.78	0.25

minimum block-level area of various CHs, cell architectures (i.e., CFET and Conv. FET), cell pin accessibility, DRs, BEOL parameters, and PDN structures.

Rent's multipliers, k, and Rent's exponent, p, of each design are listed in Table II. For the number of fanouts per net (#fanouts), we categorize the number of fanouts per net into eight bins, which are 1–3 #fan-out nets, 4–6 #fan-out nets, 7–9 #fan-out nets, 10–50 #fan-out nets, 50–100 #fan-out nets, 100–500 #fan-out nets, 500–1000 #fan-out nets, and more than 1000 #fan-out nets. Fig. 6(a) and (b) shows the #Fanouts distribution and cell statistics of these six block-level circuits, respectively.

2) SDC Library Sets Generation: To evaluate the blocklevel PPA during early DTCO exploration, we select 30 representative SDCs [7]. We generate 19 SDC library sets with 4.5T, 3.5T, and 2.5T CHs, different EOL and VR DR parameters, and two cell architectures (i.e., CFET and Conv. FET) using [7] and [12]. The top layer is M2 for SDC generation. We generate SDC library sets with variations on three dimensions as follows.

1) *CSs:* We generate Conv. FET and CFET SDC layouts for explorations on 2-D and 3-D CS in the experiments.
2) *CH:* The CFET SDC CH is scaling from 4.5T to the extreme 2.5T CH [7], [25]. For Conv. FET SDC, we generate 4.5T and 3.5T CH because using two horizontal routing tracks for Conv. CS cannot be implemented due to the limitation of p-n separation [26].
3) *DRs:* We use grid-based DR parameters to generate SDC layouts for layers up to M2, and they are applied to block-level using the corresponding metal pitch values [7]. Here, the baseline DR parameters are EOL = 1 and VR = 1.

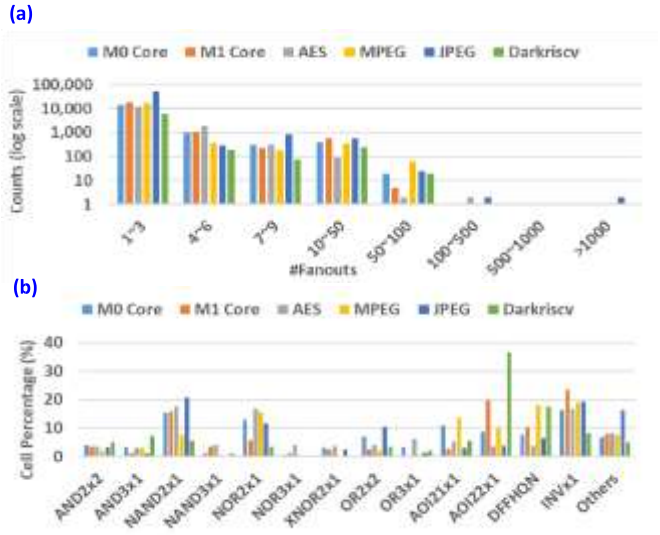


Fig. 6. (a) #Fanouts distribution. (b) Cell statistics of M0 Core, M1 Core, AES, MPEG, JPEF, and Darkriscv.

Table III shows the average cell area, average RPA [16], average M2Track, and average *M2ML*, which are extracted for predicting $A_{i,j}$ as described in Section II-D, of each SDC library set. Note that the M0/M2 pitches are 24 nm, and contacted poly pitch (CPP) is 42 nm for all the SDC library sets in Table III. Fig. 7 shows an example of generated DFFHQN SDC layouts with variations on these three dimensions: 1) CSs; 2) DR; and 3) CH. Notice that the *AvgRPA* metric of a cell library might be larger than the CH because the limited horizontal M0 routing resource and the connection of SDC external pins need to be promoted to M2 for connecting FET terminals and satisfy the minimum pin opening constraint [7] for medium or large cell (i.e., XOR2 × 1 and FAx1). Fig. 8 shows the RPA value of each pin of XOR2 × 1 in 3.5T CFET EOL = 1 VR = 0 SDC set. Pins A and B are promoted to M2 for connecting internal FET terminals and satisfy the minimum pin opening constraints.

Considering the coverage of CFET and Conv. FET CSs, 4.5T, 3.5T, and 2.5T CHs, and various DRs, we select 15 SDC library sets as listed in the train column of Table III to build our prediction model for Exp. III-B, Exp. III-C, and Exp. III-D. Then, to test the accuracy of the proposed prediction model on new SDC library sets, we use the remaining four SDC library sets in Exp. III-C.

3) BEOL Parameters: We adjust DRs in the block level based on the DR parameters used in the SDC library set generation [7] for M1, VIA12, and M2 layers. Then, the metals' pitch and width of layers above M2 are set based on LEF/DEF guide [27]. For via layers above M2, the via spacing is set to allow diagonal via, and the same net via spacing is set to allow adjacent via.

For the BEOL settings, we generate various M4–M7 metal pitches by varying the baseline metal pitches from 0.5× to 1.5×. If the metal pitch is smaller/larger than the smallest pitch/largest pitch after scaling, its metal pitch is set to the smallest pitch/largest pitch. Here, the smallest vertical/horizontal metal pitch is M1/M2 metal pitch; the largest horizontal/vertical metal pitch is M8/M9 metal pitch. For the BEOL routing layers, we use M2–M5, M2–M6, and M2–M7 options for block-level routing.

TABLE III

SDC FEATURE VALUES OF 19 SDC LIBRARY SETS. CH = CELL HEIGHT. CS = CELL STRUCTURE. CONV. = CONV. FET. THE BASELINE DR PARAMETERS ARE EOL = 1 AND VR = 1. TRAIN = USED FOR TRAINING THE PROPOSED PREDICTION MODEL IN EXP. III-B AND EXP. III-C

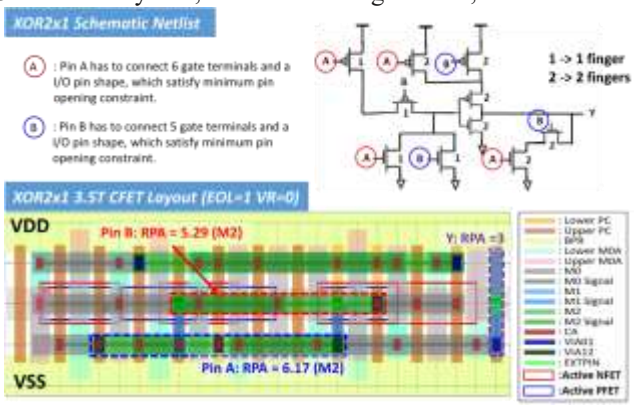
CH	CS	DR parameters	Avg Cell Area (um^2)	Avg RPA (access point)	Avg $M2Track$ (track)	Avg $M2ML$ (segment)	Train
4.5T	Conv.	Baseline	0.04415	3.290	0.433	4.900	V
		EOL 2 VR 1	0.04551	2.830	0.600	9.533	
		EOL 0 VR 1	0.04309	2.805	0.267	2.867	V
		EOL 1 VR 1.5	0.05232	4.119	1.067	13.900	
		EOL 1 VR 0	0.04355	2.831	0.500	5.667	V
	CFET	Baseline	0.04249	3.204	0.200	2.200	V
		EOL 2 VR 1	0.04324	3.100	0.500	5.867	V
		EOL 0 VR 1	0.04234	3.266	0.133	1.800	V
		EOL 1 VR 1.5	0.04581	3.813	0.933	12.533	V
		EOL 1 VR 0	0.04234	3.198	0.267	2.067	V
		EOL 0 VR 0	0.04229	3.058	0.167	1.333	V
3.5T	Conv.	Baseline	0.04151	2.839	1.233	19.233	V
	CFET	Baseline	0.03657	2.784	1.033	14.400	V
		EOL 2 VR 1	0.04057	3.692	1.367	23.500	
		EOL 0 VR 1	0.03422	3.734	1.033	12.300	V
		EOL 1 VR 0	0.03410	3.516	0.933	11.333	V
		EOL 0 VR 0	0.03375	3.496	0.833	9.267	V
2.5T	CFET	EOL 0 VR 0	0.02915	3.010	2.000	19.167	-
		EOL 0 VR 0 FC/M0A R	0.02764	2.874	1.700	18.100	V

In total, there are 45 various DRs and BEOL pitches technologies. For each BEOL technology, there are three BEOL routing options. As a result, there are 135 BEOL settings in the experiment.

4) Power Delivery Network Configurations: We study frontside PDN and backside PDN in the following experiments. For front-side PDN structure, the PDN is constructed with top power mesh on M8 and M9, and they are designed as spaces are allowed. Then, the power is delivered through M3 power straps to SDCs. Here, we vary the M3 power strap period with 24 CPPs, 32 CPPs, 48 CPPs, and 64 CPPs based on the PDN studies in [17] and [28] for early DTCO exploration. For backside PDN architecture, there is no PDN in the front side at block level.

5) Minimum Valid Block-Level Area Extraction: Multiple block-level P&R runs are launched for minimum valid blocklevel area extraction, as shown in Fig. 1(a). In each blocklevel P&R run, the floorplan (i.e., including PDN generation), placement (i.e., including placement optimization), clock tree synthesis (CTS), and routing (i.e., including global routing and detail routing) stages are performed. Table IV shows the breakdown of the runtime in each stage of an automated M0 core block-level P&R implementation using 2.5T CFET EOL = 0 VR = 0 library and M2–M7 routing layers. The routing stage takes 94% of the total runtime because fixing DRC violations in the detail routing stage is time-consuming and usually needs many iterations (i.e., 69 iterations in this example). As a result, it takes more than 8 h to extract minimum valid block-level area of a technology combination.

We generate the data using the synthesized block-level circuit, SDC library sets, BEOL parameters, and PDN configurations for our experiments. The total runtime to extract input features and to train the proposed model is around 15 h. However, it takes two months to generate all the block-level P&R data from 19 SDC library sets, five PDN configurations,



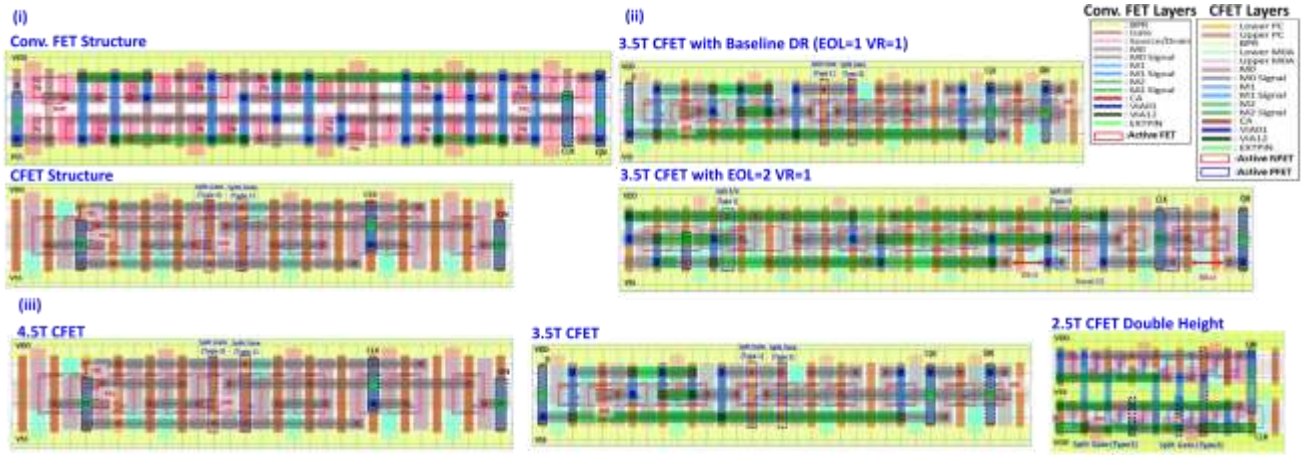


Fig. 7. Example of generated DFFHQN SDC layouts with variations on three dimensions: (i) 2-D versus 3-D CS (Conv. FET versus CFET), (ii) DR, (iii) CH.

Fig. 8. Example of RPA counts of XOR2 × 1 in 3.5T CFET EOL = 1 VR = 0 SDC library. Pins A and B are promoted to M2 for connecting internal FET terminals and satisfy the minimum pin opening constraints.

TABLE IV

BREAKDOWN OF RUNTIME IN EACH DESIGN STAGE OF AN AUTOMATED M0 CORE BLOCK-LEVEL P&R IMPLEMENTATION USING 2.5T CFET EOL = 0 VR = 0 LIBRARY, AND M2–M7 ROUTING LAYERS.

Runtime	Design Stages					
	Floorplan	Placement	CTS	Routing	others	Total
wall time (s)	3	294	92	8412	118	8919

135 DRs and BEOL settings, and six block-level circuits for the experiments.[4] The experiments are organized as follows.

1) *Exp. III-B:* We explore various ML algorithms and demonstrate our prediction model accuracy on training, validation, and testing data.
2) *Exp. III-C:* We show the accuracy of our prediction model on prediction of new SDC library sets and BEOL parameters.
3) *Exp. III-D:* We show the accuracy of the proposed model on prediction of various PDN configurations.
4) *Exp. III-E:* We study the robustness of our prediction model on new block-level circuit prediction.

For Exp. III-C, Exp. III-D, and Exp. III-E, we introduce gradient accuracy (gradient ACC) metric to measure the accuracy of the direction of $A_{i,j}$. If the signs of actual $A_{i,j}$ and predicted $A_{i,j}$ are the same, we consider that the prediction is accurate for gradient ACC metric.

B. Prediction Model Accuracy

We study various ML algorithms and demonstrate prediction model accuracy with training, validation, and testing datasets. We first split the generated data of 15 SDC library sets (i.e., Table III), six block-level synthesized designs, 102 DRs and BEOL settings, and three PDN settings, using 80% as training data and 20% as testing data based on the empirical study in [29]. Then, we split the 80% training data after bootstrap sampling such that 80% is used for model training and 20% is used for model validation. The validation dataset is used to avoid overfitting with the early stopping technique in the training phase of each GBRT model. In the following experiments, XGBoost_DTCO is a GBRT tree model used in [11].

1) Hyperparameter Tuning: We explore multilayer perceptron (MLP) neural network, radial basis function (RBF) neural network, random forest (i.e., implemented with sklearn library), XGBoost_DTCO [11], and the proposed ML algorithm, which integrates bootstrap aggregation and GBRT techniques. We tune the hyperparameters of each ML modeling algorithm for our DTCO and STCO sensitivity prediction. For optimizing neural network structure for our regression problem, we adopt Hyperband [24] to set the number of layers, the number of neurons of each layer, dropout rate, batch size, and learning rate of MLP and RBF neural networks. Fig. 9 shows the selected MLP and RBF neural network structures with Hyperband [24] algorithm. For the XGBoost_DTCO [11], random forest, and the proposed ML model, we use tenfold cross validation [21] to set the hyperparameters of our prediction model. Table V shows the range of each hyperparameter in the explored ML algorithms. In the proposed model, the max_depth, sub_sample, min_child_weight, and learning rate of each GBRT model are 9, 1.0, 7, and 0.05, respectively. We select 100 #GBRTs for the proposed model after hyperparameter tuning.

2) Prediction Accuracy: Table VI shows the prediction accuracy results of MLP, RBF neural network, XGBoost_DTCO [11], random forest, and the proposed method. The MAE of the proposed model is 4.1×10^{-3} on the testing set. Compared to MLP and RBF neural networks,

[4] We use eight CPU cores for each block-level P&R job and run multiple block-level P&R jobs simultaneously.

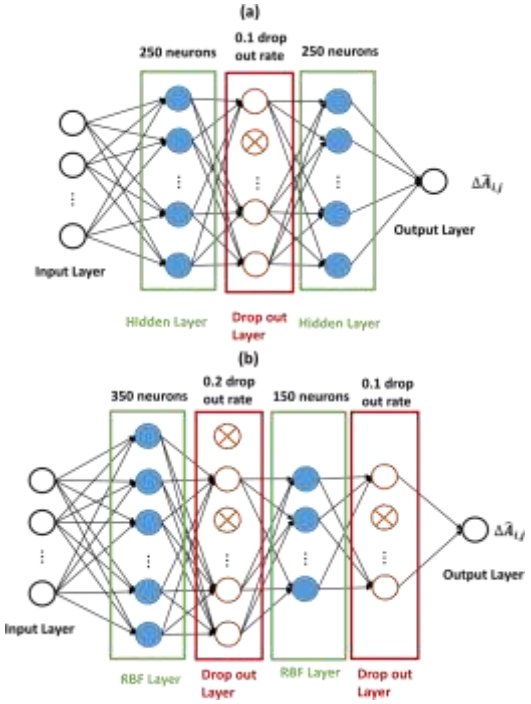


Fig. 9. (a) MLP and (b) RBF neural network structures after Hyperband [24] search on #layers (i.e., 2–10), #neurons per layer (i.e., 25–500), and dropout rate (i.e., 0.0–0.5).

the proposed model achieves 92.9% and 74.2% less MAE on the testing set, respectively. Moreover, the proposed model provides 65.8% and 16.3% less MAE on the testing set than random forest and XGBoost_DTCO [11], respectively.

Fig. 10(a) and (b) shows the predicted $A_{i,j}$ values versus golden $A_{i,j}$ values for training and testing sets of the proposed model. The blue solid line in the middle indicates a perfect correlation between golden $A_{i,j}$ and predicted $A_{i,j}$. The upper and lower black solid lines are 5% away from the blue solid line. We can observe that most of the errors of predicted $A_{i,j}$ are within 5% in the training and testing sets. The MAEs are 3.2×10^{-3} for the training set and 4.1×10^{-3} for the testing set. Fig. 10(c) shows the error distribution of testing set. The mean is 3.47×10^{-5}, with a standard deviation of 0.0075 (hence, 99.7% of predicted $A_{i,j}$ values are within the three-σ range of ±0.023). Furthermore, compared to XGBoost_DTCO [11], the proposed model reduces the standard deviation of error distribution by 0.0011 (i.e., 12.8%) for the testing set. This shows that the proposed model is more robust than XGBoost_DTCO [11] on the model accuracy.

3) Key Features Study: To further study the key features, we combined the gain of the same features of the first and second technologies of each technology pair in XGBoost_DTCO [11] model and the proposed model. Fig. 11(a) and (b) shows the average combined important features of 100 GBRTs in the proposed model and top 15 combined important features in XGBoost_DTCO [11] after feature selection (see Section II-E), respectively.

The most important feature in the proposed model and XGBoost_DTCO [11] is CH. CH is highly related to the block-level area because it determines the size of each cell row in the block level. For the pin accessibility and routing congestion metrics in SDCs, the proposed weighted RPA_d,

TABLE V

HYPERPARAMETER EXPLORATION OF ML ALGORITHMS TABLE

Machine Learning Alg.	Hyperparameter	Value Range
MLP/ RBF neural network	#layers	2 - 10
	#neurons per layer	25 - 500
	drop out rate	0.0 - 0.5
	learning rate	{1e-2, 5e-3, 1e-3, 5e-4, 1e-4}
	batch size	{128, 256, 512}
Random Forest (sklearn) [22]	#estimators	50 - 500
	max_depth	{10 - 100, None}
	min_samples_leaf	1 - 4
	min_samples_split	2 - 10
XGBoost_DTCO [11]	max_depth	6 - 15
	min_child_weight	5 - 11
	sub_sample	0.7 - 1.0
	learning rate	{5e-2, 1e-2, 5e-3, 1e-3, 5e-4, 1e-4}
Proposed Method	#GBRTs	50 -500
	GBRT parameters	Same as XGBoost_DTCO [11]

TABLE VI

PREDICTION ACCURACY TABLE. IMPR. MAE = (MAE_{MLALG} – $MAE_{PROPOSED}$)/MAE_{MLALG} × 100. HERE, MAE_{MLALG} REPRESENTS THE MAE ERROR OF MLP/RBF NEURAL NETWORK/XGBOOST_DTCO [11]/ RANDOM FOREST [22]

Machine Learning Alg.	MAE		Impr. MAE (%)	
	Training set	Testing set	Training set	Testing set
MLP	0.0570	0.0578	94.3	92.9
RBF Neural Network	0.0155	0.0159	79.4	74.2
Random Forest (sklearn) [22]	0.0066	0.0120	51.5	65.8
XGBoost_DTCO [11]	0.0034	0.0049	5.9	16.3
Proposed	0.0032	0.0041	-	

weighted $M2Track_d$, and weighted $M2ML_d$ are also very important for $A_{i,j}$ prediction in both XGBoost_DTCO [11] model and the proposed model in the block level. For the synthesized design feature, the 1–3 fanouts and the number of sequential cells (#Seq) features are recognized as top 15 average important features in the proposed model.

For DR feature, we can observe that the V1 spacing, M2 minimum spacing, V3 spacing, and V3 same net spacing all have large gains in both XGBoost_DTCO [11] model and the proposed model because these layers are mainly used for accessing the SDC pins on M1/M2. For the DR features of layers above M4, their gains are smaller since these layers are mainly used to connect above and below metal layers instead of accessing SDC pins. Note that the M2 and M4 fat metal spacing rules (FatMSpace), which are usually related to the wider metal used for power straps, are recognized as important features in the proposed model and XGBoost_DTCO [11] model. In addition to the DRs related to power strap, the power strap period feature is also in the top 15 important features in both models because its impact on the block level is nontrivial, as shown in Fig. 13. Here, although via spacing and same net via spacing has high correlation, they could be remaining in the input features after feature selection stage since we remove the

feature only if the validation error, E_{val}, is larger and the VIF of the feature is larger than a VIF threshold in Algorithm 1. The "V3 via same net space" and "V3 via space" are both used as input features in Fig. 11(a).

With more simulated datasets generated by bootstrap aggregation, we observe various important features of each GBRT model in the proposed model, as shown in Fig. 11(c). Fig. 11(c) shows the #Counts of the important features, which are extracted from the top 3 gain of each GBRT model, in the proposed model. The top 9 important feature with larger average gain in Fig. 11(b) is also in the top 3 important features of 100 GBRTs frequently in the proposed model. Here, the 7–9 fanouts net feature also frequently appears in the top 3 important features of 100 GBRTs in the proposed model though its average gain across 100 GBRT models is not in Fig. 11(b).

C. Prediction of New Technologies

We apply the trained model from Exp. III-B to predict $A_{i,j}$ of new SDC library sets and new BEOL parameters. Here, we implement a benchmark utilization prediction model (Util. model) with XGBoost algorithm, which takes the features of a technology (i.e., Table I) and predicts the utilization after block-level P&R. Then, we calculate the block-level area after P&R from the output of Util. model and obtain $A_{i,j}$ of every technology pairs for comparison. In this experiment, we compare the accuracy of the proposed model, random forest, XGBoost_DTCO [11], and Util. model on DTCO and STCO sensitivity prediction of new SDC library sets and new BEOL parameters.[5]

For new SDC library sets, we study the accuracy of the proposed prediction model to predict $A_{i,j}$ of 20% of 19 SDC library sets in Table III. The four new SDC library sets are

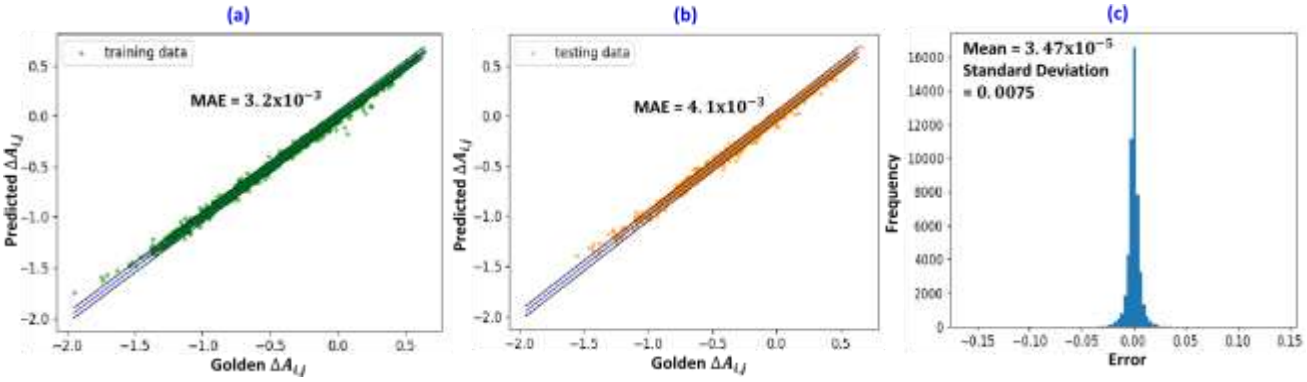


Fig. 10. Predicted $A_{i,j}$ versus golden $A_{i,j}$ of (a) training set and (b) testing set, and (c) error distribution of testing set of the proposed model. The mean of MAE is 3.47×10^{-5}, with standard deviation of 0.0075 for testing set. Hence, 99.7% of predicted $A_{i,j}$ are within the three-σ range of ±0.023.

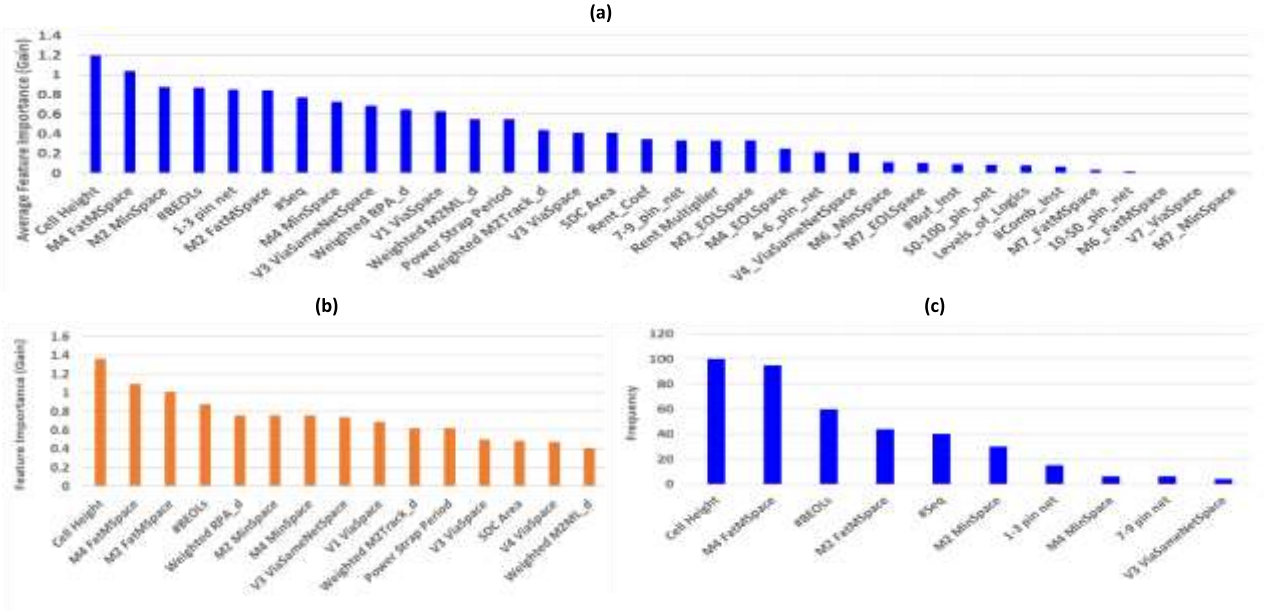


Fig. 11. Feature importance (gain) of the proposed model and XGBoost_DTCO [11] for key feature study. (a) Average combined feature importance (gain) of GBRTs in the proposed model. (b) Top 15 combined feature importance (gain) in the trained XGBoost_DTCO model [11]. (c) #Counts of important features, which are extracted from top 3 gain of 100 GBRT models, in the proposed model.

[5] We mainly study the tree-based ML models (i.e., the proposed model, random forest, and XGBoost_DTCO [11]) since the MAEs of tree-based ML models on testing set are better than neural network models in Exp. III-B.

carefully selected to include different CHs (i.e., 4.5T, 3.5T, and 2.5T), different CSs (i.e., Conv. and CFET), and DRs including strict and loose DR parameters
(i.e., EOL = 2 VR = 1 and EOL = 0 VR = 0) to demonstrate the prediction of new SDC library sets. The BEOL routing layer options for these four testing SDC library sets are M2–M5, M2–M6, and M2–M7. Table VII shows the prediction results of $A_{i,j}$ using Util. model, and the proposed DTCO and STCO sensitivity prediction approach with random forest, XGBoost_DTCO [11], and the proposed model.

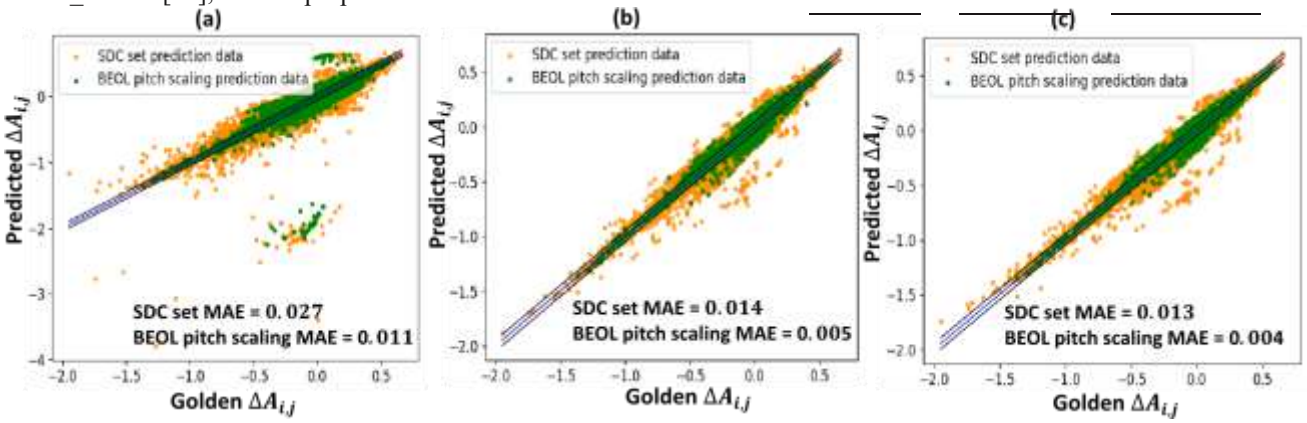


Fig. 12. Predicted $A_{i,j}$ versus golden $A_{i,j}$ of new SDC library set technology prediction (i.e., orange points) and new BEOL pitch scaling technology prediction (i.e., green points) with (a) random forest (i.e., implemented with sklearn), (b) XGBoost_DTCO [11], and (c) proposed model.

The proposed model provides 0.013 MAE and 97.3% gradient ACC on new SDC library set prediction. Compared to the Util. model, our proposed model achieves 91.3% less MAE error and 20.0% better gradient ACC. Compared to random forest and XGBoost_DTCO [11], the proposed model still maintains 51.9% and 7.1% less MAE error, respectively. Fig. 12(a)–(c) shows the predicted $A_{i,j}$ versus golden $A_{i,j}$ for SDC library set prediction (i.e., orange point) with random forest, XGBoost_DTCO [11], and the proposed model, respectively. There are clearly more data points of random forest prediction outside of the black solid line, which represents 5% away from the perfect correlation line in the middle. This matches the larger standard deviation and MAE in Table VII.

For new BEOL pitch scaling settings, we study the accuracy of the proposed model on prediction of 11 BEOL pitch scaling technologies. Combining these 11 BEOL pitch scaling technologies with 15 SDC library sets, three #BEOL layer options (i.e., M2–M5, M2–M6, and M2–M7), and three PDN settings, there are 1485 technology combinations for prediction in this experiment. In addition, the BEOL pitch scaling also affects DRs, such as minimum spacing, EOL spacing, via spacing, and same net via spacing. In Table VII, the MAE and gradient ACC of the proposed model are 0.004 and 97.1%, respectively. Compared to the Util. model, the proposed model achieves 97.2% less MAE error and 8.3% better gradient ACC. Moreover, the MAEs of the proposed model are 63.6% and 20.0% smaller than random forest and XGBoost_DTCO [11], respectively. Fig. 12(a)–(c) shows the predicted $A_{i,j}$ versus golden $A_{i,j}$ for BEOL pitch scaling prediction (i.e., green points) with random forest, XGBoost_DTCO [11], and the proposed model, respectively. Here, we can observe that there are obviously more data points of random forest prediction outside of the black solid line.

To summarize, the proposed DTCO and STCO sensitivity prediction modeling approach achieves better accuracy than the Util. model because it directly minimizes the MAE of $A_{i,j}$ and $\hat{A}_{i,j}$ during the training phase. On the other hand, there are utilization prediction errors from the Util. model and inherent differences between synthesized block-level circuit area and block-level area after P&R in Util. model

$$= (\hat{A}_i - \hat{A}_j) - (A_i - A_j) = E_i \bar{A}_j + E_j A_i \quad (7)$$

$$\hat{A}_i \quad A_i \quad A_i(A_i \quad Ei)$$

Equation (7) shows the DTCO and STCO sensitivity error E_{sen}. when we use the predicted minimum block-level area from

TABLE VII
$A_{i,j}$ PREDICTION RESULTS OF NEW TECHNOLOGIES USING UTILIZATION MODEL (UTIL.), RANDOM FOREST, XGBOOST_DTCO [11], AND THE PROPOSED MODEL. MAE: MEAN ABSOLUTE ERROR. GRADIENT ACC: GRADIENT ACCURACY OF $A_{i,j}$. ERROR DIST.: ERROR DISTRIBUTION. STD. DEV.: STANDARD DEVIATION

Prediction Type	Model	MAE	Gradient ACC (%)	Error Dist.	
				Mean	Std. Dev.
New SDC lib. set	Util.	0.150	77.3%	-0.012	0.233
	Random Forest [22]	0.027	94.8%	0.002	0.069
	XGBoost_DTCO [11]	0.014	97.2%	0.001	0.031
	Proposed	0.013	97.3%	0.001	0.031
New BEOL pitch scaling tech.	Util.	0.147	88.8%	-0.025	0.210
	Random Forest [22]	0.011	96.9%	0.001	0.049
	XGBoost_DTCO [11]	0.005	96.9%	0.000	0.013
	Proposed	0.004	97.1%	0.000	0.012

Util. model. Here, E_{sen} and E_i are the error of DTCO/STCO sensitivity and predicted minimum block-level area, respectively. $\hat{A}_i = A_i + E_i$. When E_i is very small and $E_j > A_i$, the predicted block-level error (i.e., E_j) leads to large E_{sen} on DTCO and STCO sensitivity prediction. For example, from one of the data points in new SDC library set technologies prediction, A_i, A_j, $\hat{A}_i$, and $\hat{A}_j$ are 276.652, 1138.511, 280.911, and 814.554, respectively. E_{sen} is 1.19, which is larger than 99.7% (i.e., three-σ range 0.093 ± 0.001) of the error of the proposed model. As a

result, we observe that the Util. model has larger standard deviation of error than the proposed model in Table VII. Moreover, compared to random forest and XGBoost_DTCO [11], the proposed model provides smaller MAE and better gradient ACC for DTCO and STCO sensitivity prediction with bootstrap aggregation and GBRT techniques.

D. Prediction of New Power Delivery Network Setting

We study the model accuracy on predicting $A_{i,j}$ of new PDN grid scales and architectures (i.e., backside PDN). Here, we select front-side PDN with 24 CPPs, 48 CPPs, and 64 CPPs power strap period to train our model and use the trained model to predict $A_{i,j}$ of new PDN setting with 32 CPPs power strap period and backside PDN architecture.

Fig. 13(a) shows the snapshots of M0 Core design with various M3 power strap periods. For backside PDN architecture, there is no power strap in the front side at block level, as shown in Fig. 13(b). We can observe that the power strap period and PDN architecture (i.e., backside power delivery) can potentially impact the block-level area from 6% to 25% in Fig. 13. Since the Util. model performs poorly in Exp. III-C, we mainly study the accuracy of random forest, XGBoost_DTCO [11], and the proposed model in this experiment.

1) Prediction of New PDN Setting: Table VIII shows the prediction results of new front-side PDN setting prediction (i.e., 32 CPPs). For new front-side PDN setting prediction, the proposed model achieves 0.027 MAE and 94.4% gradient ACC, which are 27.0% less MAE and 1.9% better gradient ACC than random forest. Compared to XGBoost_DTCO [11], the proposed model achieves 3.6% less MAE and 0.3% better gradient ACC. Fig. 14 shows the predicted $A_{i,j}$ versus golden $A_{i,j}$ of new PDN setting prediction (i.e., green points). Although there are few green points located far away from the perfect center line in the proposed model, the gradient ACC is 94.4%. Therefore, the accuracy of the proposed model can be calibrated along the gradient of $A_{i,j}$ from one technology to another technology, as shown in Fig. 3(b).

2) Prediction of Backside PDN: Here, to further study the robustness of the proposed model on new PDN architecture, we first use the trained model, which is trained using frontside PDN with various power strap periods, to predict $A_{i,j}$ of backside PDN architecture. Then, we further study the improvement of prediction accuracy of XGBoost_DTCO [11] and the proposed model using various ratios (i.e., 10%–80%) of backside PDN data points to update the models.

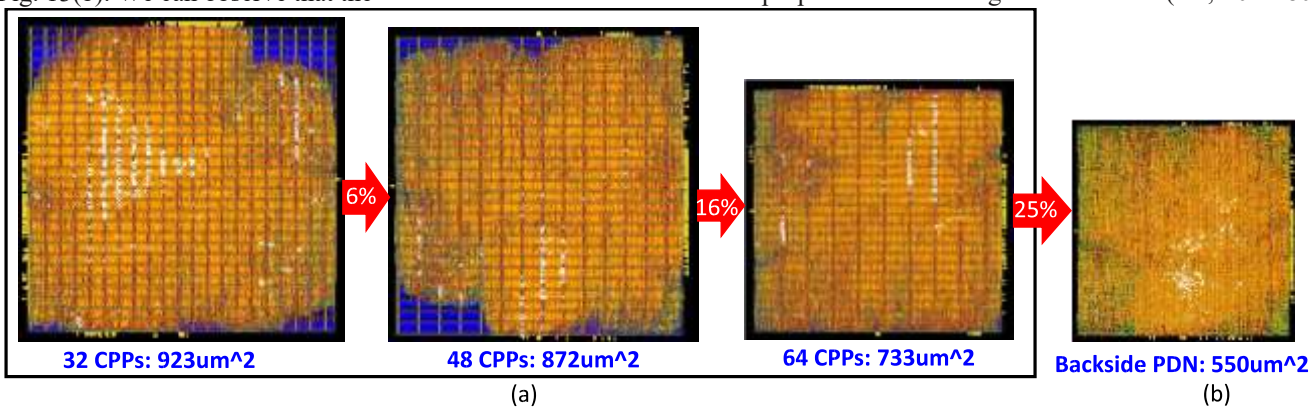


Fig. 13. Minimum block-level area of M0 Core with various (a) front-side PDN grid scales (i.e., 32 CPPs, 48 CPPs, and 64 CPPs) and (b) backside PDN architecture using M2–M6 for signal routing. Compared to 32 CPPs front-side PDN setting, the core area of backside PDN is 40% smaller. The SDC library is 3.5T CFET with baseline DR parameters in Table III.

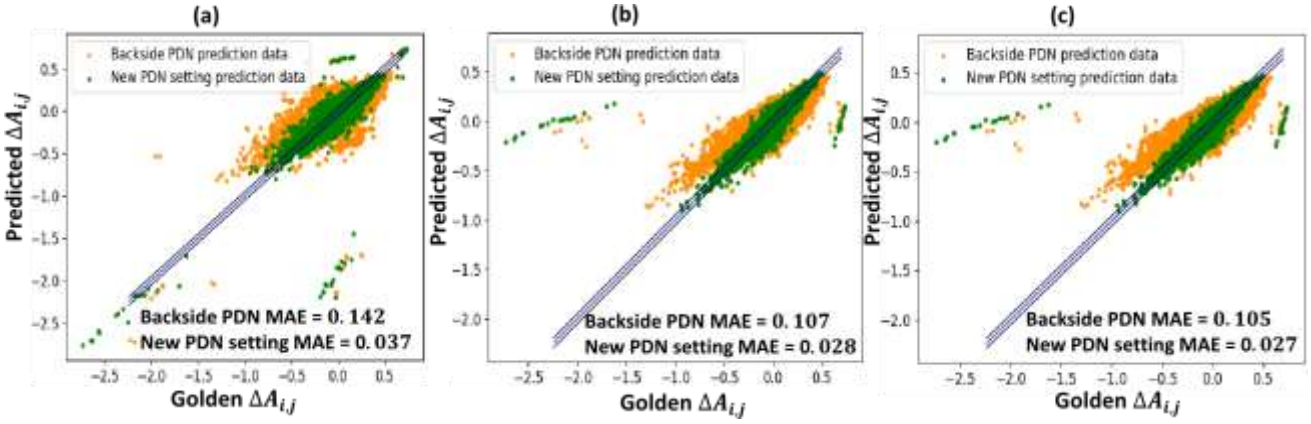


Fig. 14. Predicted $A_{i,j}$ versus golden $A_{i,j}$ of new PDN setting (32 CPPs) prediction (green points) and backside PDN prediction (orange points) with (a) random forest (implemented with sklearn), (b) XGBoost_DTCO [11], and (c) proposed model.

TABLE VIII

$A_{i,j}$ PREDICTION RESULTS OF NEW FRONT-SIDE PDN SETTING AND BACKSIDE PDN ARCHITECTURE USING RANDOM FOREST, XGBOOST_DTCO [11], AND THE PROPOSED MODEL.

Prediction Type	Model	MAE	Gradient ACC (%)	Error Dist.	
				Mean	Std. Dev.
	Random Forest [22]	0.037	92.5%	0.004	0.102

New front side PDN setting (32 CPPs)	XGBoost_DTCO [11]	0.028	94.1%	0.000	0.107
	Proposed	0.027	94.4%	0.000	0.106
Backside PDN Architecture	Random Forest [22]	0.142	77.8%	-0.014	0.198
	XGBoost_DTCO [11]	0.107	84.8%	-0.017	0.147
	Proposed	0.105	86.9%	-0.015	0.155

First, for predicting backside PDN without any backside PDN data for training, the MAE and gradient ACC of the proposed model are 0.105% and 86.9%, respectively. The proposed model achieves 35.2% and 1.8% less MAE than random forest and XGBoost_DTCO [11], respectively. In addition, compared to random forest and XGBoost_DTCO [11], the proposed model provides 9.1% and 2.1% better gradient ACC, respectively. Fig. 14 shows the predicted $A_{i,j}$ versus golden $A_{i,j}$ of backside PDN prediction (i.e., orange points). The block-level area difference of backside PDN technology, which brings the systematic physical layout change at the block level, cannot be fully captured (i.e., MAE is larger than 0.1) with only front-side PDN training data using random forest, XGBoost_DTCO [11], and the proposed model. As a result, we further study the accuracy improvement of prediction models using various ratios (i.e., 10% 80%) of backside PDN data points to update the models, which is the outer loop in Fig. 3(b).

Fig. 15(a) shows the MAE of backside PDN prediction of XGBoost_DTCO [11] and the proposed model with various ratios (i.e., 10%–80%) of backside PDN data points for updating the models. The proposed model provides larger accuracy improvement than XGBoost_DTCO [11] when giving a ratio of backside PDN data for model update. Moreover, the proposed model achieves up to 60.8% MAE reduction when updating the model with 20% backside PDN data. Fig. 15(b) shows that predicted $A_{i,j}$ versus golden $A_{i,j}$ of 0%, 10%, and 20% backside PDN data for model update. From Fig. 15(b), the proposed model can efficiently capture the block-level area difference ($A_{i,j}$) of backside PDN with 10%–20% backside PDN data for model update. This shows that the proposed model can be updated efficiently and robustly with small amount of data of new technologies, which leads to the systematic physical layout change at the block level.

To summarize, the bootstrap aggregation technique creates meaningful simulated datasets from the given training dataset, which can reduce model variance while avoiding overfitting. For each GBRT model in the proposed model in Fig. 5, the gradient boosting tree technique improves the accuracy by sequentially building an ensemble of K regressors to minimize the prediction error. Therefore, the proposed model can provide better accuracy and robustness than random forest and XGBoost_DTCO [11] model on predicting new PDN setting and backside PDN architecture.

E. Robustness of New Circuit Prediction

We study the robustness of the proposed modeling approach for predictions on new block-level circuits in this experiment. Here, we iteratively select one synthesized block-level circuit out of the six synthesized block-level circuits (i.e., Table II) for testing the robustness of model prediction on new designs. Then, train the model with the rest of the five synthesized block-level circuits with all SDC library sets, DRs, and BEOL settings and apply the trained prediction model to predict $A_{i,j}$ of the selected synthesized block-level circuit with all the SDC library sets, DRs, and BEOL settings.

Table IX shows the robustness of random forest, XGBoost_DTCO [11], and the proposed model to make predictions on designs unseen in the training set. The average MAE and average gradient ACC are 0.0555 and 87.91% for DTCO and STCO sensitivity prediction on new designs using the proposed model, respectively. Moreover, the proposed modeling approach achieves 24.22% and 3.40% smaller average MAEs than random forest and XGBoost_DTCO [11], respectively. Also, the proposed model provides 9.84% and 0.73% better gradient ACC on average than random forest and XGBoost_DTCO [11], respectively. This shows that the proposed model is able to robustly guide DTCO optimization on designs unseen during training.

Regarding runtime performance, it takes less than 1 min to predict 10k block-level area sensitivities of one technology to another technology. On the other hand, it takes more than 8 h to extract the minimum valid block-level area of a new technology combination for block-level metric comparison (i.e., $A_{i,j}$) as described in Section III-A. The proposed prediction model achieves more than 100× speedup on finding the optimal technology candidate in the potential technology list compared to running the block-level P&R runs for multiple potential technology candidates, extracting the minimum valid

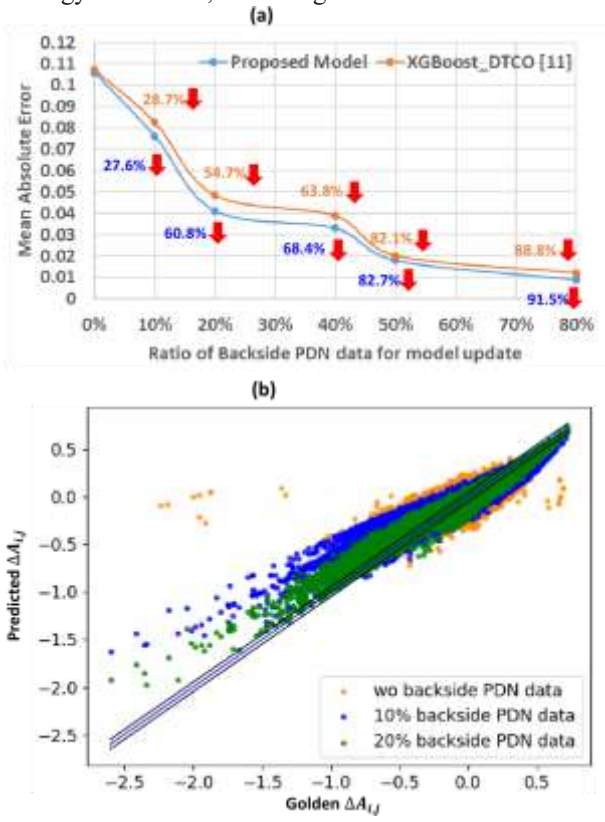


Fig. 15. Accuracy improvement with various ratios of backside PDN data for model update. (a) MAE versus ratio of backside PDN data for model update. Orange/blue number is the reduced MAE percentage of XGBoost_DTCO [11]/proposed model compared to 0% backside PDN data for model update. (b) Predicted $A_{i,j}$ versus golden $A_{i,j}$ of 0%, 10%, and 20% backside PDN data for model update. The 10%–20% backside PDN data for model update greatly reduce up to 60.8% MAE for the proposed model.

block-level area, and finding the optimal technology candidate. In summary, we show that our modeling approach not only captures the block-level area difference on new SDC library sets, BEOL parameters, and various PDN configurations but also capable of robustly predicting $A_{i,j}$ of various technology options for new circuit designs.

IV. Conclusion

We propose an overall framework along with the proposed DTCO and STCO sensitivity prediction model, and automatic SDC synthesis [7], [12] to significantly reduce the TAT of DTCO and STCO explorations. In addition, we develop an ML model using bootstrap aggregation and gradient boosting techniques to predict the difference of block-level area between two different technology options for reducing the runtime of block-level P&R in DTCO and STCO explorations.

We first demonstrate that the MAEs of the proposed DTCO and STCO sensitivity prediction model are 3.2×10^{-3} for the training set and 4.1×10^{-3} for the testing set. In addition, 99.7% of prediction errors are within ±0.023. Then, we validate the importance of the proposed block-level SDC metrics (i.e., weighted RPA, M2Track, and M2ML) through the feature importance in the proposed model. For prediction on new technologies, we showed that our ML model not only achieves 7.1% less MAE on predicting new SDC library sets across different designs but also provides 20.0% less MAE on predicting new BEOL settings than XGBoost_DTCO [11].

TABLE IX

Prediction Result of Selected Synthesized Block-Level Circuit Table.

Improvement = ($\text{Metric}_{\text{MLA\_LG}} - \text{Metric}_{\text{proposed}}$)/$\text{Metric}_{\text{MLA\_LG}}$. Metric = MAE/Gradient ACC

Selected Design for Prediction	Random Forest [22]				XGBoost DTCO [11]				Proposed Model				Improvement			
			Error Dist.				Error Dist.				Error Dist.		Random Forest [22]		XGBoost DTCO [11]	
	MAE	Gradient ACC	Mean	Std. Dev.	MAE	Gradient ACC	Mean	Std. Dev.	MAE	Gradient ACC	Mean	Std. Dev.	MAE	Gradient ACC	MAE	Gradient ACC
M0	0.0929	80.68%	-0.0114	0.1195	0.0742	87.19%	-0.0082	0.0846	0.0725	87.86%	-0.0067	0.0846	21.96%	8.17%	2.29%	0.77%
M1	0.0487	87.00%	0.0070	0.0680	0.0445	88.10%	0.0117	0.0622	0.0446	88.13%	0.0122	0.0623	8.42%	1.28%	-0.22%	0.03%
AES	0.0988	77.75%	0.0339	0.1223	0.0637	87.47%	0.0098	0.0840	0.0608	88.17%	0.0085	0.0802	38.46%	11.82%	4.55%	0.80%
MPEG	0.058	73.86%	0.0047	0.0770	0.0516	87.89%	-0.0025	0.0700	0.0492	87.97%	-0.0042	0.0674	15.17%	16.04%	4.65%	0.09%
JPEG	0.0706	78.26%	0.0256	0.0818	0.0562	87.65%	0.0049	0.0730	0.0524	87.64%	0.0130	0.0680	25.78%	10.70%	6.76%	-0.01%
Darkrisc	0.0831	78.01%	0.0075	0.1107	0.0549	85.36%	0.0012	0.0718	0.0536	87.67%	0.0013	0.0692	35.50%	11.02%	2.37%	2.71%
Avg	0.0754	79.26%	0.0112	0.0965	0.0575	87.28%	0.0028	0.0743	0.0555	87.91%	0.0040	0.0720	24.22%	9.84%	3.40%	0.73%

For the studies on predicting $A_{i,j}$ of new PDN setting and backside PDN structure, we not only show that the proposed model achieves 0.027 MAE for new front-side PDN configuration but also demonstrate that the MAE of the proposed model is reduced up to 60.8% with only 10%–20% backside PDN data for model update. Finally, we demonstrate that the proposed modeling approach achieves 5.55×10^{-2} MAE and 87.91% gradient ACC on average in the robustness experiment of new design prediction. For the performance, it takes less than 1 min to predict 10k block-level area sensitivities of one technology to another technology and provide more than 100× speedups compared to running block-level P&R for technologies and extracting minimum valid block-level area.

V. Future Works

Future research directions include: 1) conducting an extensive study on multiple 3-D SDC architectures, such as many-tier VFET SDC [30]; 2) incorporating more circuit designs in the study (i.e., deep learning accelerators [31]); and 3) extending the DTCO and STCO area sensitivity prediction model for power and performance metrics.

References

[1] L. W. Liebmann and R. O. Topaloglu, "Design and technology cooptimization near single-digit nodes," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 2014, pp. 582–585.

[2] L. Liebmann, "Design technology co-optimization for 3 nm and beyond," in *IEDM Tech. Dig.*, Sep. 2019, p. 1.

[3] L. Liebmann *et al.*, "DTCO acceleration to fight scaling stagnation," *Proc. SPIE*, vol. 11328, Sep. 2020, Art. no. 113280C.

[4] S. C. Song *et al.*, "Unified technology optimization platform using integrated analysis (UTOPIA) for holistic technology, design and system co-optimization at <= 7 nm nodes," in *Proc. IEEE Symp. VLSI Circuits (VLSI-Circuits)*, Jun. 2016, pp. 1–2.

[5] A. Kahng *et al.*, "PROBE: A placement, routing, back-end-of-line measurement utility," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 7, pp. 1459–1472, Jul. 2018.

[6] K. Jo *et al.*, "Design rule evaluation framework using automatic cell layout generator for design technology co-optimization," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1933–1946, Aug. 2019.

[7] C.-K. Cheng *et al.*, "Complementary-FET (CFET) standard cell synthesis framework for design and system technology co-optimization using SMT," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 6, pp. 1178–1191, Jun. 2021.

[8] Z. Zhang *et al.*, "New-generation design-technology co-optimization (DTCO): Machine-learning assisted modeling framework," in *Proc. Silicon Nanoelectronics Workshop (SNW)*, Jun. 2019, pp. 1–2.

[9] A. Ceyhan *et al.*, "Machine learning-enhanced multi-dimensional cooptimization of sub-10 nm technology node options," in *IEDM Tech. Dig.*, Dec. 2019, pp. 6–36.

[10] C.-K. Cheng *et al.*, "PROBE2.0: A systematic framework for routability assessment from technology to design in advanced nodes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 5, pp. 1495–1508, May 2021.

[11] C.-K. Cheng *et al.*, "Design and system technology co-optimization sensitivity prediction for VLSI technology development using machine learning," in *Proc. ACM/IEEE Int. Workshop Syst. Level Interconnect Predict. (SLIP)*, Nov. 2021, pp. 8–15.

[12] D. Lee *et al.*, "SP&R: SMT-based simultaneous Place-and-Route for standard cell synthesis of advanced nodes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 10, pp. 2142–2155, Oct. 2021.

[13] (2020). *Cadence Innovus User Guide*. [Online]. Available: http:/www.cadence.com
[14] P. Christie and D. Stroobandt, "The interpretation and application of Rent's rule," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 6, pp. 639–648, Dec. 2000.
[15] C.-K. Cheng *et al.*, "A routability-driven complimentary-FET (CFET) standard cell synthesis framework using SMT," in *Proc. 39th Int. Conf. Comput.-Aided Design*, Nov. 2020, pp. 1–8.
[16] J. Seo *et al.*, "Pin accessibility-driven cell layout redesign and placement optimization," in *Proc. 54th Annu. Design Autom. Conf.*, Jun. 2017, pp. 1–6.
[17] B. Chava *et al.*, "DTCO exploration for efficient standard cell power rails," *Proc. SPIE*, vol. 10588, Mar. 2018, Art. no. 105880B.
[18] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
[19] R. S. Gómez *et al.*, "Collinearity diagnostic applied in ridge estimation through the variance inflation factor," *J. Appl. Statist.*, vol. 43, no. 10, pp. 1831–1849, Jul. 2016.
[20] B. Efron and R. Tibshirani, "Improvements on cross-validation: The 632+ bootstrap method," *J. Amer. Stat. Assoc.*, vol. 92, no. 438, pp. 548–560, Jun. 1997.
[21] S. Arlot and A. Celisse "A survey of cross-validation procedures for model selection," *Statist. Surv.*, vol. 4, pp. 40–79, Jul. 2010.
[22] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
[23] (2020). *OpenCores: Open-Source IP Cores*. [Online]. Available: https://opencores.org/
[24] L. Li *et al.*, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6765–6816, 2017.
[25] C.-K. Cheng *et al.*, "Multirow complementary-FET (CFET) standard cell synthesis framework using satisfiability modulo theories (SMTs)," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 7, pp. 43–51, 2021.
[26] P. Weckx *et al.*, "Novel forksheet device architecture as ultimate logic scaling device towards 2nm," in *IEDM Tech. Dig.*, Dec. 2019, pp. 5–36.
[27] (2020). *LEF/DEF Language Reference*. [Online]. Available: http://www.ispd.cc/contests/18/lefdefref.pdf
[28] A. Gupta *et al.*, "High-aspect-ratio ruthenium lines for buried power rail," in *Proc. IEEE Int. Interconnect Technol. Conf. (IITC)*, Jun. 2018, pp. 4–6.
[29] A. Gholamy *et al.*, "Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation," Univ. Texas El Paso, El Paso, TX, USA, Tech. Rep. UTEP-CS-18-09, 2018. [Online]. Available: https://scholarworks.utep.edu/cs_techrep/1209/
[30] D. Lee *et al.*, "Many-tier vertical gate-all-around nanowire FET standard cell synthesis for advanced technology nodes," *IEEE J. Explor. SolidState Comput. Devices Circuits*, vol. 7, pp. 52–60, 2021.
[31] (2018). *NVIDIA Deep Learning Accelerator (NVIDIA)*. [Online]. Available: https://github.com/nvdla/hw