ML-Based Wire RC Prediction in Monolithic 3D ICs with an Application to Full-Chip Optimization

Sai Surya Kiran Pentapati
sai.pentapati@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Bon Woong Ku
bon.ku@synopsys.com
Synopsys Inc.
Mountain View, California, USA

Sung Kyu Lim
limsk@ece.gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

ABSTRACT

The state-of-the-art Monolithic 3D (M3D) IC design methodologies [1, 2] use commercial electronic design automation tools built for 2D ICs to implement a pseudo-3D design and split it into two dies that are routed independently to create an M3D design. Therefore, an accurate estimation of 3D wire parasitics at the pseudo-3D stage is important to achieve a well optimized M3D design. In this paper, we present a regression model based on boosted decision tree learning to better predict the 3D wire parasitics (RCs) at the pseudo-3D stage. Our model is trained using individual net features as well as the full-chip design metrics using multiple instantiations of 8 different netlists and is tested on 3 unseen netlists. Compared to the Compact-2D [1] flow on its own as the reference pseudo-3D, the addition of our predictive model achieves up to 2.9× and 1.7× smaller root mean square error in the resistance and capacitance predictions respectively. On an unseen netlist design, we observe that our model provides 98.6% and 94.6% RC prediction accuracy in 3D and up to 6.4× smaller total negative slack of the design compared to the result of Compact-2D flow resulting in a more timing-robust M3D IC. This model is not limited to Compact-2D, and can be extended to other pseudo-3D flows.

CCS CONCEPTS

• **Hardware → 3D integrated circuits**; **Modeling and parameter extraction**; **Static timing analysis**.

ACM Reference Format:
Sai Surya Kiran Pentapati, Bon Woong Ku, and Sung Kyu Lim. 2021. ML-Based Wire RC Prediction in Monolithic 3D ICs with an Application to Full-Chip Optimization. In *Proceedings of the 2021 International Symposium on Physical Design (ISPD '21), March 22–24, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3439706.3447266

1 INTRODUCTION

In monolithic 3D (M3D) ICs, dies are fabricated sequentially on top of each other with the transistor stacking process [3]. This allows the logic gates to be placed in 3D space, adding a new degree of freedom for cell placement. With two dies (tiers) fabricated on top of each other, the entire chip can fit into half the 2D footprint, leading to $(1/\sqrt{2})\times$ smaller half perimeter wire length (HPWL) in an M3D IC compared to corresponding 2D IC on average. This HPWL saving has been known to be the main driver to reduce the power and improve the timing in M3D ICs.

The state-of-the-art M3D IC physical design methodology named Compact-2D [1] starts from performing placement, clock design, routing, and post-route optimization in the pseudo-3D design. Here the design is still in a 2D layout but tries to mirror the parasitics of a 3D IC design. The footprint of the pseudo-3D design is twice that of the final 2-tier 3D design. This ensures all the cells of the final 3D design fit in the 2D footprint with the same overall density as the 3D design. The design is later compacted into the 3D footprint by scaling the linear dimensions by $1/\sqrt{2}$. To account for this compaction step, the wire resistance and capacitance parasitics are also scaled by $1/\sqrt{2}$ in the 2D layout. Next, tier partitioning splits the design into two dies (tiers) and monolithic inter-tier via (MIV) planning finds the optimal 3D connection locations between the die, and finally, a die-by-die routing completes the 3D design.

The Back-End-Of-Line (BEOL) and Front-End-Of-Line (FEOL) of a pseudo-3D stage are significantly different from the final 3D stage. The cell pins which were on a single plane in the pseudo-3D design are now split between the tiers, and the BEOL stack in the final 3D design is a concatenation of the BEOL from two different tiers. Therefore, there is a noticeable difference in routing between the two stages. This is especially true for the nets that connect cells on different dies (referred to as 3D nets). Even in the nets that only connect to cells within a single die (2D nets), the additional legalization stage in 3D after tier partitioning causes a parasitic mismatch. If the parasitic estimation in the pseudo-3D stage is smaller than the original value, the cells connected to these nets show timing degradation when routed with 3D BEOL stack. It is also possible that some nets will have a smaller capacitance in 3D design due to the unpredictability of legalization after the tier partitioning. These cells have immoderate drive-strength and consume excessive power. This is why an accurate parasitic prediction at the pseudo-3D stage mirroring the final 3D parasitics is essential to achieve a good timing and power optimization in 3D.

In this work, we minimize such deviations in the RC values using XGBoost [4], a decision tree learning model. The model predicts the final 3D parasitics using 23 total features based on metrics related to the net (wirelength), full-chip (average fanout), and scaled net features (ratio of total and specific-net wirelength). With better RC prediction, the optimization in pseudo-3D stage is closer to the 3D routing optimization leading to improved timing. We aren't focused on inductance estimation, as on-chip nets have negligible inductance and are not considered by the EDA tools during on-chip optimization. Our predictive model achieves an overall total

ISPD '21, March 22–24, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8300-4/21/03…$15.00
https://doi.org/10.1145/3439706.3447266

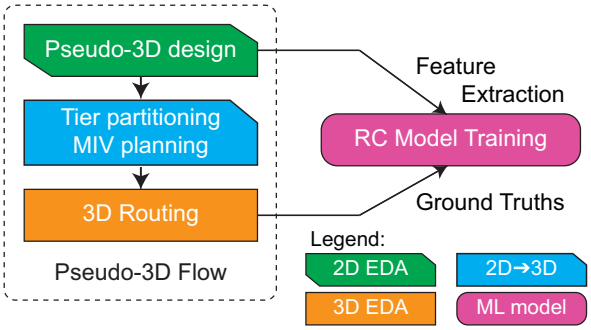

Figure 1: A high-level overview of a pseudo-3D flow along with the feature extraction, training stage used in this work.

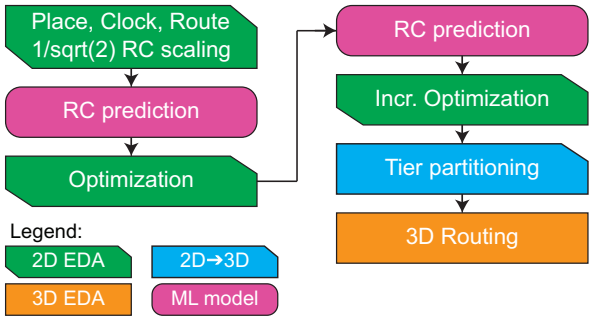

Figure 2: Our RC prediction model applied to Compact-2D (C2D) flow.

negative slack reduction up to 6.4× smaller than the traditional parasitic estimation model on an unseen netlist.

2 EXPERIMENTAL SETUP

2.1 Model Training

Figure 1 shows the overall flow used for data collection in each design along with the pseudo-3D flow. To train the model, we use a total of 8 different verilog netlists from Opencores [5] and ISPD contests [6]. These are first implemented as a pseudo-3D design using Compact-2D flow. Features extracted at the end of this stage serve as the input to the training model. The cells are partitioned into two tiers based on physical (placement, cell size) and logical (net connectivity) information with bin-based placement driven FM min-cut algorithm [7]. Legalization and routing are then done in the 3D design to extract the ground truth 3D parasitics used to train the ML model. Each netlist is designed at four target frequencies, and with three different tier partitioning options at each frequency to generalize the model and reduce model bias to specific implementations.

2.2 Model Application

The trained model is integrated with the pseudo-3D flow so that the optimization in this stage uses predicted RCs. As the input net features for training are extracted after detail routing, we integrate the model after the same stage. Once the design is routed, the input features are extracted and passed to the ML-model to get the predicted 3D RCs of the nets. These RCs are assigned to the routed nets with a net-by-net annotation, and the design optimization is done as usual. During optimization, connectivity of the netlist changes due to the addition and removal of buffers, changes in the cell types, etc. Due to this, the annotated nets are discarded and replaced with a non-annotated net.

To annotate these new nets, the net features are once again collected (stage 5 in Figure 2) and re-annotated using model predictions. Now, an incremental in-place optimization using 'do not touch' attribute for the nets (stage 6 in Figure 2) is performed so that the cells are sized in-place using model predicted RCs, and no (or a very few) new nets are added. The first optimization stage doesn't use 'do not touch' attribute as it severely constrains the EDA tool. This is the final pseudo-3D design with ML-based RCs assigned to the nets during final optimization. The design is then split into two tiers as per the normal pseudo-3D flow and perform 3D routing. The overall flow is given in Figure 2.

3 RC PARASITICS DISCREPANCY ANALYSIS

We implement the 2D and 3D designs based on a commercial 28 nm technology node. For the 2D design, we use six metal layers for routing. For the 3D design, technology files are created to reflect the full 3D metal stack that has 12 routing layers (6 layers per die). An inter-layer dielectric separates the two BEOL stacks to represent the top-die FEOL's insulating oxide in 3D. Two sets of nearly identical cell files are generated for 3D FEOL from the corresponding 2D technology files. The $1/\sqrt{2}$ RC scaling assumption during the pseudo-3D stage in Compact-2D flow is based on the fact that HPWLs are scaled down by $1/\sqrt{2}$ when the design is converted from pseudo-3D to 3D stage. While this is true on a global scale, RC parasitics are not scaled down uniformly for all the nets. In this section, we analyze this RC parasitics discrepancy between pseudo-3D and 3D designs in detail.

3.1 Via Resistance

In modern technology nodes, the parasitics of a via start playing a considerable role in the total resistance of a net. For instance, in the 28 nm technology node, the default via from each metal layer has a resistance equivalent to 1.4 µm of signal wire, and a capacitance equal to a wire of 0.1 µm in length. As the number of vias on a net increase, the parasitics are no longer dependent solely on the length of the wires. This is much more apparent in the resistance of the net than the capacitance due to the large resistance of the vias.

From Table 1, the 128-bit AES design has an average of 0.76 vias per 1 µm of net in the pseudo-3D design and the number of vias increase by ~10% to 0.83 vias per 1 µm of net in the final 3D design. This is due to the usage of more metal layers in the smaller footprint of 3D design. The routing in 3D needs uses more metal layers requiring more vias to complete connectivity than the pseudo-3D stage. In pseudo-3D flow, as the global BEOL RCs are scaled by 0.707×, it doesn't account for the via RCs correctly.

The via impact is observable in the 3D RC vs. wirelength plots in Figure 3. In these plots, each point on the graph is a net in the 3D design and they are colored based on the number of vias on the net. The RC per unit length values for the 28 nm technology are

Table 1: 28nm 128-bit AES implementation statistics of the Pseudo-3D and the final 3D routed designs

128-bit AES	Pseudo-3D	Final-3D
Metal Stack	6 Layers	12 Layers
Cell Pin Count	381,907	381,907
Wire length (µm)	1,141,178	1,234,332
Via Count	872,931	1,028,040
Via Count per unit wire	0.76	0.83
Ground Capacitance (pF)	119.34	140.83
Coupling Capacitance (pF)	35.20	31.71
Wire Resistance (MΩ)	15.26	21.23

within 5% of each other for all the 6 metal layers used for signal routing. So the total RC of a net is closely approximated using the total wirelength instead of having to break it into different wire segments on various metal layers. The wire capacitance is almost linearly dependent on the total wirelength with only a small deviation from the mean caused by effects such as number of vias, coupling capacitance between the nets, and small differences in the RC per unit length of different metal layers. On the other hand, the resistance of nets, while linearly increasing with wirelength, is much wider (high variance) due to the larger contribution from the vias. Considering the nets with similar via count, the resistance values are close to linear with wirelength. So, properly considering the via resistance contribution is important to get a good resistance estimate of the net in 3D.

3.2 Tier Partitioning

Another important difference in the pseudo-3D and final 3D comes from the cell movement due to the legalization after tier partitioning. Since it is not possible to determine the cell partitioning during the pseudo-3D stage, we make use of features like wirelength, half-perimeter bounding box (HPBB), and the number of connections made by a net, to predict if the net is likely to be partitioned. This gives the ML model an insight into the additional routing required in 3D due to cell movement and 3D metal stack. Depending on the tier-partitioning constraints used, the type of nets that are likely to be split into two-tiers as a '3D net' will vary, this is discussed using various features in the following sub-section 3.3.

3.3 Scaling Error

First, we introduce a new metric called Scaling Error for resistance (SE_R) and capacitance (SE_C) as follows:

$$SE_{R(C)} = \frac{\text{R(C) of nets in pseudo-3D}}{\text{R(C) of the nets in final-3D}}$$

By plotting $SE_{R(C)}$[1] for various features of the net, we can understand their effects on RC scaling. The valid range of $SE_{R(C)}$ is $(0, \infty)$.

$SE_{R(C)} \approx 1$ implies that the scaling done in the pseudo-3D stage is adequate, and the parasitics in the pseudo-3D design are a good estimate for the final 3D parasitics.

$SE_{R(C)} \ll 1$ implies that the estimation in pseudo-3D is much lower than the final 3D parasitics, this results in worse timing after the design is 3D routed.

$SE_{R(C)} \gg 1$ implies that the parasitic value is over-estimated resulting in aggressive cell sizing to meet timing in pseudo-3D. This creates higher power consumption in the 3D routed design.

Figure 4 shows these plots for some of the net features. We group the nets based on various features and find the $SE_{R(C)}$ for the group. The numerator and denominator are now summed over the nets in the group, to get R(C) of the group of nets in pseudo-3D and final-3D respectively.

Figure 4(a) shows the $SE_{R(C)}$ vs. wirelength. A point 'x' on the X-axis corresponds to the group containing all the nets with wirelength (wl) such that $x < wl \leq x + 1$. To clearly understand this graph, let us divide it into three regions: Region 1($0 \leq x < 10$), Region 2 ($10 \leq x < 32$), Region 3 ($32 \leq x$).

We see that for small wirelengths in Region 1 $SE_{R(C)} \ll 1$, the scaling error (ratio between the estimated pseudo-3D parasitics and the actual 3D parasitics) is very small, and pseudo-3D parasitics are much smaller than the real 3D parasitics. This is because when cells are legalized, short nets are heavily affected by minor displacements of the cell. Therefore, these are much likely to have an increased wirelength and higher parasitics in the 3D design. The sensitivity to the cell movement decreases as the wirelength increases, which can be seen by the fact that the $SE_{R(C)}$ starts increasing with wirelength

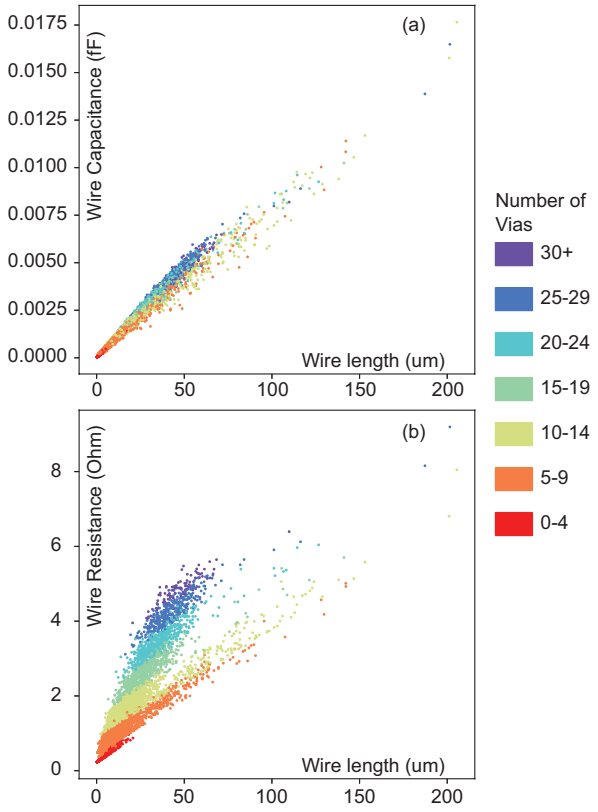

Figure 3: (a) Capacitance, (b) Resistance of nets in the AES-128 design w.r.t. routed wirelength of the net. The points are color coded based on the number of vias on each net.

[1]It is important to note that Scaling Error=1 implies exact scaling, any deviation away from 1 is undesirable. $SE_{R(C)}$ close to either 0 or ∞ are considered bad

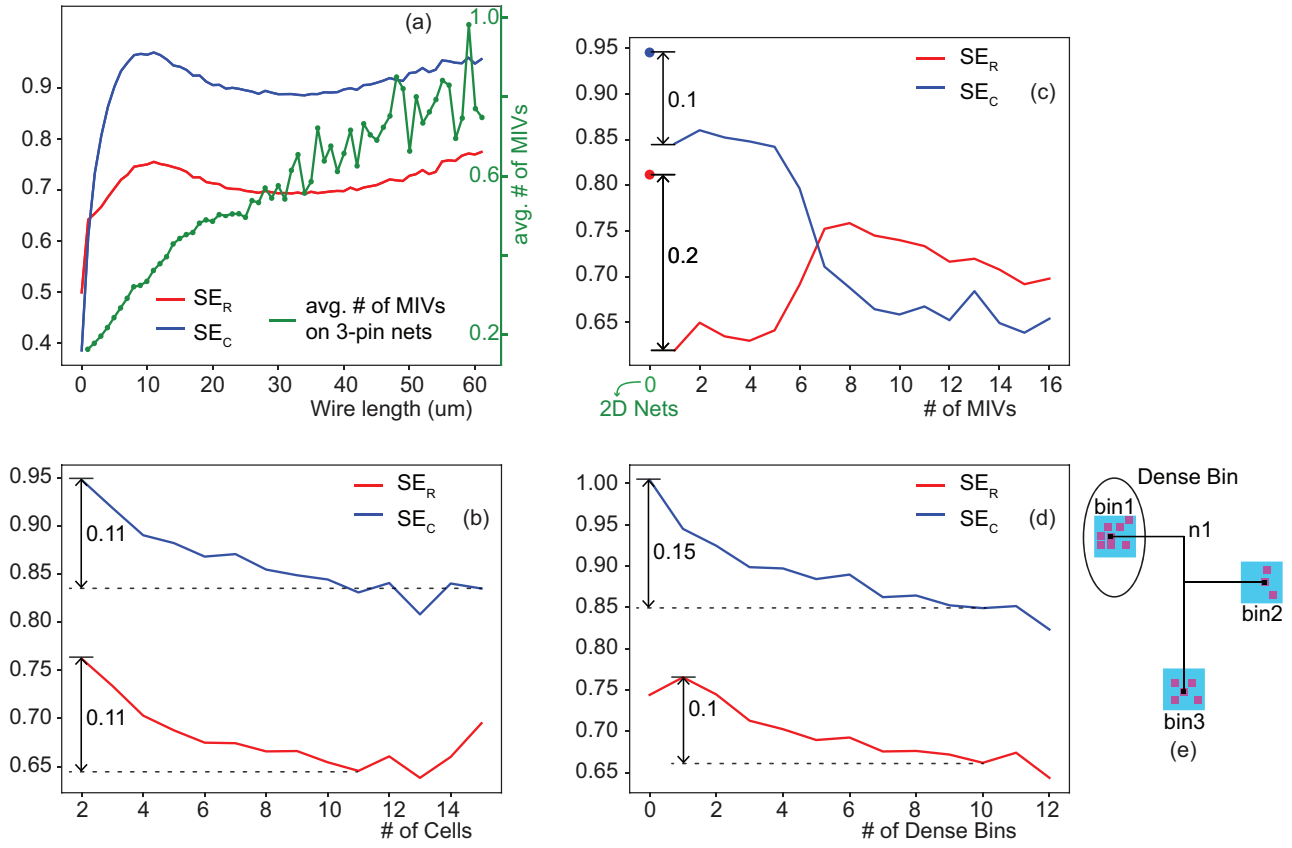

Figure 4: Average RC scaling errors w.r.t. various net features such as (a) Wire length (Also shows average number of MIVs vs. WL), (b) Number of cells connected to the net, (c) Number of MIVs on the net, (d) Number of dense bins of the net. (e) Example of net along with the bins at endpoints of the net and example of a dense bin

in this Region signifying that the estimated parasitics in pseudo-3D are a better match for final 3D parasitics.

After this, we see a slight dip in $SE_{R(C)}$ for wirelengths in Region 2. This relates to the fact that as nets become longer they become more likely to be partitioned into two tiers and connect to pins on different tiers. There is a two-fold reasoning for this wirelength impact on partitioning likelihood: 1. Longer wirelengths mean that they are likely connected to more cells and are favorable to be partitioned (more analysis on this part is done with the help of Figure 4(b) next). 2. The bin-based area-balancing of tier partitioning is more likely to convert long nets in pseudo-3D to '3D nets' by partitioning them. This trend is observed in the average MIV count vs. wirelength graph given by the green line in Figure 4(a). We only include 3-pin nets (connecting 3 pins) in average MIV count calculation to remove the effect of net connectivity on tier partitioning and MIV count. For shorter nets in the wirelength Region 2, the additional metal layer routing becomes a measurable portion of the total routing of the net. As the net length keeps increasing, the probability of partitioning first increases and more nets in the wirelength group are partitioned, leading to a dip in $SE_{R(C)}$.

As the wire length keeps increasing beyond $\approx 32\,\mu m$ in Region 3, we again see an increase in $SE_{R(C)}$. This is because the extra metal layer cost due to partitioning keeps decreasing in comparison to the wirelength, and the estimated RC values once again keep improving.

Figure 4(b) shows the avg. $SE_{R(C)}$ vs. the net connectivity (= # of Cells on the net). Each point on the x_i X-axis corresponds to the group of nets with x_i number of cells connected to it. As the number of cells increases, the net becomes most likely to be partitioned and $SE_{R(C)}$ decreases showing that the pseudo-3D estimation becomes worse. In hypergraph partitioning, the cut-size of a net is either '0' or '1'. So, the cut-size cannot increase beyond '1' irrespective of the number of times a net crosses a partition. Because of this, it is more efficient to partition the nets with a large number of cells than with fewer cells. A large cell-count net can be arranged in various configurations without increasing the cut-size beyond 1. So, when such nets are not partitioned, all the cells connected to it are placed in a single partition which restricts the solution space greatly and the expected cut-size increases. So, it is highly likely for nets with large connectivity to be partitioned compared to nets connecting fewer (say, 2–3) cells. This is still just a likelihood estimate of partitioning and not a hard rule.

Figure 4(c) shows the scaling error due to the increased routing cost of going back and forth the 3D metal stack. While the cell-count

on the net is only a probabilistic measure of the tier partitioning, the MIV count used here in Figure 4(c) is a derived net feature from 3D routing and can give us the exact impact of MIVs. This feature is only shown for analysis and is not used in model training as it is not available at the pseudo-3D stage. Here, $x = 0$ MIVs corresponds to the 2D nets, and the points with $x \geq 1$ are the 3D nets. There is a clear difference in the scaling factor between these two groups, and the $SE_{R(C)}$ deviates further away from 1(ideal estimation) as the number of MIVs increases. For low MIV count $x \leq 5$, the nets correspond to relatively shorter nets on average. So, when these nets are partitioned, the number of vias increases dramatically as the nets travel the complete routing stack from the bottom to top metal layers. So, $SE_R < 1$ in this region. The capacitance value is weakly dependent on the number of MIVs and this error is not as drastic. As the number of MIVs increases and the average wirelength becomes greater, the additional via cost decreases and SE_R rises towards the correct scaling factor of 1. This rise is not present in SE_C as the number of vias has very little impact, and the scaling simply decreases due to the increasing metal layer cost of going back and forth the 3D stack.

In Figure 4(d), we introduce a new feature called "Dense Bins". Consider net n1 as shown in Figure 4(e) that is connected to three cells c1, c2, c3. At each endpoint of the net, we consider a square of side $3 \times CellHeight$, where cell height is given by the technology. The bin is defined as a "dense bin" if the density within the bin is more than a certain threshold (75% in our case). The number of dense bins of each net is used as an input feature. Bins with high density are likely to have overlapped cells after tier-partitioning which would be displaced during legalization. As the number of dense bins increases, more ends of nets would be displaced when 3D routing is performed. We see that the decrease in the SE_C with the number of dense bins that is stronger than the net-connectivity feature, showing its importance. We also see that the SE_C is very close to 1, for the net group with #Dense Bins = 0, as these cells in the grouping are more likely to stay unchanged during legalization and have the best capacitance estimation among all the groups discussed so for. Another input feature used in our model based on these bins is the average density of the bins. During training, we see that this average bin density is an important feature for both resistance and capacitance modelling.

4 LEARNING MODEL

To train the machine learning model that can predict the RC parasitics, we use XGBoost [4] open-source python library. XGBoost is based on gradient boosting using decision trees and is considered state-of-the-art for regression and classification tasks. The decision trees in XGBoost are chosen to maximally reduce the loss function, which in our case is given by the following equations:

$$L_C = \sum_{nets} (C_{pred} - C_{true})^2 \qquad L_R = \sum_{nets} (R_{pred} - R_{true})^2$$

4.1 Input Features

As discussed in Section 3.3, we use several input features that learn if a net will be split in 3D, or if the cells connected to the net are being displaced. Apart from these, we use features of the full chip design for the model to identify between different designs if needed. Another set of features we use are derived (scaled) features that are a combination of multiple input features. All the input features used (derived and independent), and their meaning and/or significance in training the model are presented in Table 2.

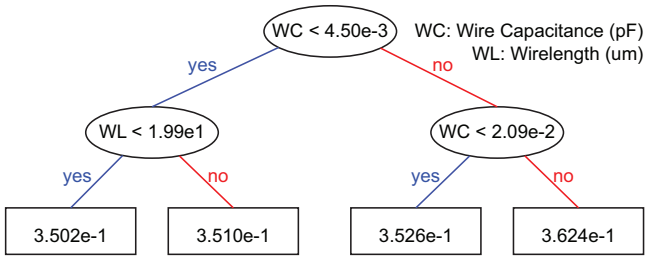

Figure 5: A simple decision tree from XGBoost when trained on a small netlist for capacitance modelling. The leaf nodes are the contribution of this tree to the final capacitance

4.2 Data Processing

The data to be used in training the model is collected as described in Section 2.1. The netlists are split into two sets for training, and testing. Here we use 8 netlists for training, and 3 for testing. The testing data is never used by the model to train or to modify hyper-parameters and remains unseen throughout the training process. Further, not all nets in the training data set are used. Since the designs that we use vary widely in the total number of cells and nets, it becomes unfair to the small netlists to use the entire data from each design. This introduces bias towards large netlists and can result in loss of accuracy when used on a generalized design. For each design, we randomly select N nets, where $N \sim$ *min netlist size*. The data is then further shuffled to remove any possible bias in the order used nets during data collection.

5 RESULTS

5.1 Training and Inference

The training data set has $\sim 3 \times 10^6$ data-points (nets) in total from different netlist implementations, and each data-point has a total of 23 input features. Using 80:20 split for training and cross-validation sets gives us $\sim 0.75 \times 10^6$ nets for cross-validation set. Training is a one-time task and takes only up to $\approx$ 10 min in total. Using cross-validation set, we choose the training model with 30 decision trees as the cross-validation error saturates at this point and starts to increase as more decision trees are included due to overfitting.

Figure 6 shows the feature importance of the top-18 features used for the RC models. In gradient boosting algorithms, feature importance quantifies the average improvement in the loss function due to the feature. Unsurprisingly, resistance and capacitance in pseudo-3D are the most important features for the respective models. Note that, this still does not mean the scaling factor is uniform as in Compact-2D. In the resistance model, the number of pins (net connectivity) and vias of the net in pseudo-3D also become crucial as they dictate the number of vias after partitioning. In the R(C) models, the other parasitic variable C(R) has high importance which shows that the model uses both resistance and capacitance of

Table 2: Input Features used to train the XGBoost model, and their importance and/or explanation

Feature Name	Significance
	Individual Net Features
Wirelength	Long, medium, short wires have varying average $SE_{R(C)}$
HPBB	Wirelength estimation from placement; not affected by congestion
Net connectivity	Number of cells connected by the net; estimates partitioning probability
Via Count	Number of vias on the net; useful for resistance calculation
Wire Cap	capacitance in pseudo-3D design, has strong effect on final capacitance
Wire Res	resistance in pseudo-3D design, has strong effect on final resistance
Average Local density	Average density of all the bins of a given net as shown in Figure 4(e); useful for deciding legalization errors
Dense Bins	Number of dense bins of each net; useful for deciding legalization errors
	Full Chip Features (Design Identifiers)
Total WL	Total routed signal wirelength, design identifier
Number of nets	Total number of nets in the design
Number of cells	Total number of cells in the design
Average Fanout	Number of cell pins/ number of nets; design connectivity information
Chip Area	Footprint of the design
Cell Area	Total standard cell area in the design
Utilization	Standard cell density
Bin Count	Number of partitioning bins to be used
	Derived Features
net WL/ total WL	Identifying global nets among various designs; detouring, cross coupling capacitance information
HPBB / $\sqrt{\text{Chip area}}$	Identifying global nets among various designs, less affected by pseudo-3D congestion
Wire Cap / Total Cap	Fractional capacitance of net w.r.t. total, importance of net within a design
Wire Res / Total Res	Fractional resistance of net w.r.t. total, importance of net within a design
net WL / Bin size	For a given wirelength, a increase in bin-size would decrease the probability of net being partitioned
HPBB / Bin size	Similar to net WL/ Bin size, but only considers placement information
HPBB + 0.1*VC	A combined HPBB, via count feature

the nets during modeling of each variable. An interesting outcome is that the average bin density is the 5^{th} most important feature in both the models, and sometimes better than the usefulness of wirelength or HPBB of the net. The full-chip features always have medium-to-low importance as they do not have any net information, and the derived features which use both the full chip and the net features on the other hand have medium importance. The full-chip and derived features are mainly useful in distinguishing nets from different designs.

Table 3 shows the final timing, power, and the RC accuracy for C2D and our flow using predicted 3D RCs. Using the modeled RCs, the root mean square error of the resistance estimation (M_R) decreases by up-to 61% ($\sim$ 2.5× decrease) in testing set and 65% ($\sim$ 2.8× decrease) in the training set from the results shown in Table 3. Capacitance RMSE (M_C) goes down by up-to $\sim$ 40% in both training and testing datasets. The improvements come from the better estimation of 3D displacement, partitioning, double metal stack overhead by the RC models. Overall, the decrease of M_R is much larger than the decrease in M_C because of the poor resistance prediction in the uniform scaling model in pseudo-3D design, and difficulty of predicting coupling capacitance in our model.

From the histograms of the RC values shown in Figure 7, the pseudo-3D design has more predictions of low capacitance values, leading to an optimistic parasitic estimation. This is also seen is table 1, where the pseudo-3D ground capacitance estimation is

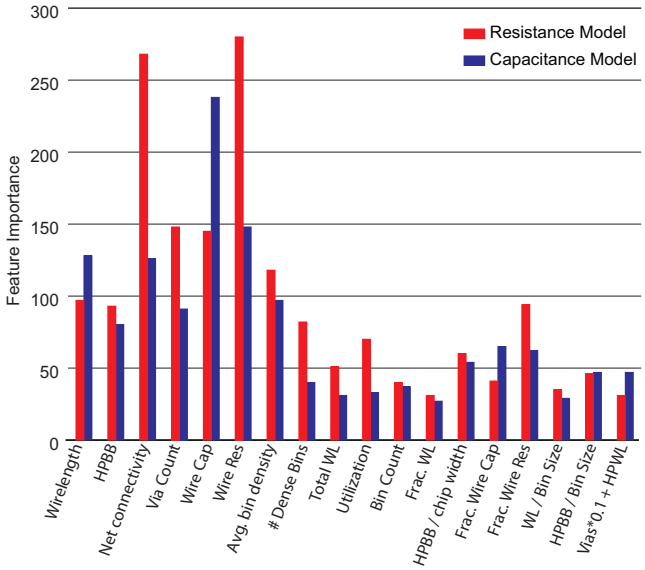

Figure 6: The 18 most important features for RC modelling.

$\sim$ 14% smaller than the 3D parasitics. When comparing the overall design of the unseen netlist 'tate' in Table 4, the total capacitance

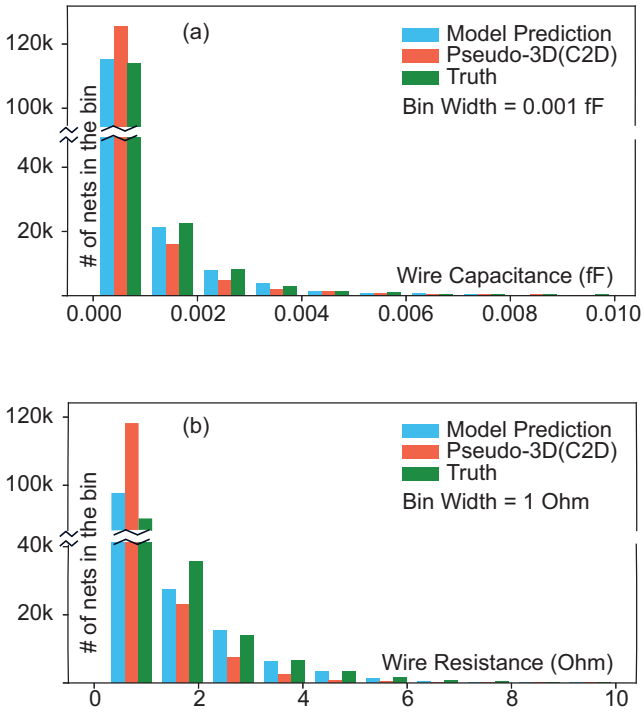

Figure 7: RC Histograms in the unseen 'tate' netlist design

error from pseudo-3D to 3D stage is reduced to just 5.1% from -21.1%, and the total resistance error is 1.4% as compared to -32.3% in pseudo-3D. The capacitance error is higher than the resistance modeling error as coupling capacitance cannot be easily predicted from per net features, and requires physical layout information and neighboring routing structures of each wire segment which are not provided in our feature list.

5.2 Full-chip PPA

The complete 3D flow with RC predictions, annotations, and in-place optimization still takes the similar amount of time as the Compact-2D flow for M3D ICs with only a < 1% increase in total run-time on average. With better overall RC predictions, the Total Negative Slack of the design goes down significantly by as much as 89.0%(≈ 9× smaller) in the training set and 84.6%(≈ 6.5× smaller) in the testing set.

TNS improves in all the designs except LDPC as this is an extremely wire-dominant design with many long wires. The M_R, M_C for LDPC are significantly higher than every other design for both pseudo-3D and ML-based predictions. The small improvement in M_R, M_C with predictive scaling is not enough to improve the WNS, TNS of the design. Out of the 11 total netlists, 4 of them (ecg, ldpc, matrix-multiplier, tate) have a worse WNS with ML-modelled RCs. The WNS of design is heavily dependent on a very few critical nets, but we do not have definitive features to learn the timing criticality of the nets. Moreover, the slack of a path depends on all the cells and nets on the path, making it more complex to predict with ML models. Features that reflect global netlist connectivity, cell strengths, etc. are required to target the WNS reduction. When

Table 3: Timing (in ns), Power (in mW), and RC accuracy. Root Mean Square Error of Capacitance(M_C) is in aF, and resistance(M_R) in dΩ. For context, the mean capacitance value in the training set is ~ 1.5 aF, mean resistance is ~ 1.8 dΩ. $\Delta > 0$ if the metric improves with our ML based 3D RC model.

Netlist		PPA Metrics			Accuracy Metrics			
Scaling Type→		C2D	ML	Δ%		C2D	ML	Δ%
Trained Netlists								
aes-128	Pwr	141.35	142.47	-0.8	M_C	0.76	0.54	28.9
110k Cells	WNS	-0.100	-0.081	19.0	M_R	1.15	0.52	54.8
2.25 GHz	TNS	-55.37	-22.87	58.7				
cordic	Pwr	20.32	21.69	-6.7	M_C	0.54	0.36	33.3
30k Cells	WNS	-0.132	-0.124	6.1	M_R	0.89	0.43	51.7
0.75 GHz	TNS	-62.9	-32.0	49.1				
des	Pwr	129.6	133.6	-3.1	M_C	0.78	0.52	33.3
50k Cells	WNS	-0.102	-0.075	26.5	M_R	1.03	0.50	51.5
2.25 GHz	TNS	-52.1	-20.4	60.8				
ecg	Pwr	117.0	105.3	10.0	M_C	0.81	0.48	40.7
95k Cells	WNS	-0.134	-0.464	-246	M_R	1.29	0.45	65.1
1.50 GHz	TNS	-222.4	-94.7	57.4				
edit	Pwr	122.0	122.0	0.0	M_C	1.05	0.82	21.9
73k Cells	WNS	-0.261	-0.158	39.5	M_R	1.39	0.58	58.3
1.50 GHz	TNS	-208.6	-161.8	22.4				
fpu	Pwr	22.2	22.6	-1.8	M_C	1.18	0.74	37.3
40k Cells	WNS	-0.173	-0.115	33.5	M_R	1.58	0.56	62.9
0.75 GHz	TNS	-129.8	-14.3	89.0				
ldpc	Pwr	104.6	109.0	-4.2	M_C	1.89	1.50	20.6
45k Cells	WNS	-0.209	-0.223	-6.7	M_R	1.57	0.82	47.8
1.25 GHz	TNS	-167.1	-178.6	-6.9				
matrix	Pwr	82.1	82.8	-0.9	M_C	1.09	0.96	11.9
60k Cells	WNS	-0.125	-0.196	-56.8	M_R	1.10	0.56	49.1
1.00 GHz	TNS	-31.5	-21.7	31.1				
Testing (Unseen) Netlists								
b19	Pwr	40.1	40.1	0.0	M_C	0.73	0.44	39.7
35k Cells	WNS	-0.161	-0.102	36.6	M_R	1.27	0.49	61.4
1.25 GHz	TNS	-155.8	-37.3	76.1				
tate	Pwr	132.4	131.8	0.5	M_C	1.26	0.85	32.5
150k Cells	WNS	-0.401	-0.450	-12.2	M_R	1.42	0.62	56.3
1.00 GHz	TNS	-497.3	-76.8	84.6				
vga	Pwr	28.40	28.47	-0.2	M_C	0.20	0.20	0.1
35k Cells	WNS	-0.193	-0.191	1.0	M_R	1.80	1.11	38.3
1.25 GHz	TNS	-48.4	-12.7	73.8				

arranged according to reduction in RMSE, the top 3 least performing netlists are: vga, matrix-multiplier, ldpc, and they also have a worse or no improvement in WNS. A lower reduction in RMSE means that the model is more likely to make errors in these designs compared to others. When these errors occur on the timing critical nets, the WNS worsens.

Total power increases by a small margin when pseudo-3D optimization is done with ML modeled RCs. This is mainly because the RC prediction with 0.707× uniform scaling is optimistic leading to smaller than expected RCs in pseudo-3D. The netlist ECG is an exception to this rule, as it has a ~ 10% reduction in power for this case. While this is accompanied by a TNS that is ~ 2× smaller,

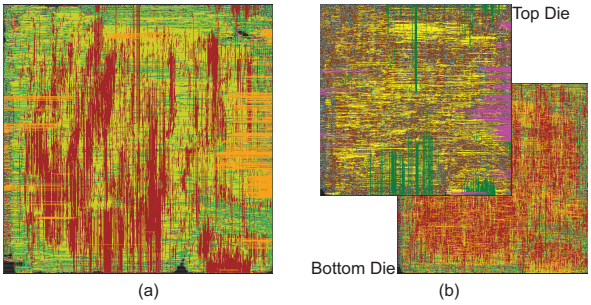

Figure 9: Full-Chip layouts of the tate netlist design with ML modelled RCs in (a) Pseudo-3D stage, (b) final 3D stage

Table 4: Pseudo-3D statistics of the unseen netlist 'tate' designed at 1.0 GHz using Compact-2D flow with and without the 3D RC modelling.

tate 1 GHz	w/o ML modelling	w/ ML modelling
Chip Area (μm^2)	116,690	116,690
Cell Area (μm^2)	187,492	187,609
Buffer Area (μm^2)	1,443.7	1,495.6
Capacitance Error (%)	-21.1	5.1
Resistance Error (%)	-32.3	1.4
WL (m)	1.610	1.612
Total Power (mW)	126.6	132.8
TNS (ns)	-0.0	-0.0
Final 3D Power (mW)	132.4	131.8
Final 3D TNS (ns)	-497.3	-76.8

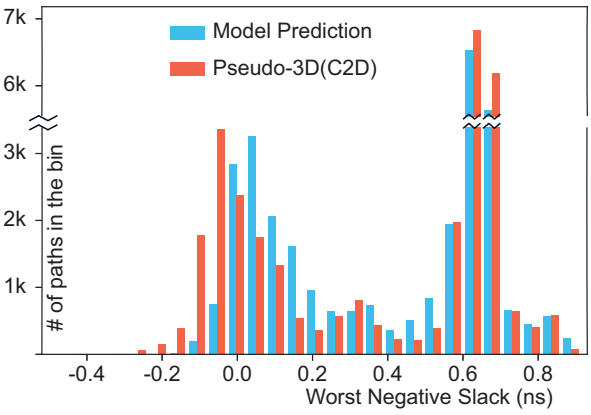

Figure 8: Timing Histograms of all the paths in the final 3D stage of unseen netlist 'tate' with and without RC modelling in the pseudo-3D design

the WNS increases dramatically once again showing its sensitive nature even when M_C, M_R are significantly lower with the model RC prediction.

An in-depth study of the timing paths of 'tate' design is useful in understanding the slack evolution. From Table 3, we observe that the WNS of 'tate' becomes worse with ML predicted RCs, but the timing histograms in Figure 8 show a more complete picture. While the worst path slack degrades, the overall number of paths with negative slack (left end of the X-axis) is much smaller in the 3D design with ML modeled RCs. Another interesting fact is that the number of paths towards the right end decreases with ML modeling. The decrease in the number of over-optimized paths stems from the fact that the ML modeling is not abuse scaling by simply over-scaling all the RCs to get better TNS. This shows that the ML model not only helps improve worst timing paths but also reduces the over-optimization of the cells. The number of over-optimized paths cannot go down to zero as some cells in this group will also be present in other timing-critical paths making it impossible to downsize such cells.

The evolution of the PPA on the unseen netlist 'tate' is further detailed in Table 4 and its full chip layouts in Figure 9. The PPA metrics in this table are from the pseudo-3D stage unless specified as 3D. Even with the better parasitic estimation, the TNS cannot be matched from pseudo-3D to 3D stages. The total power, on the other hand, is much closely predicted in the pseudo-3D with the RC modeling. Due to the optimistic RC prediction (-ve error % value) in pseudo-3D without ML modeling, the power estimation is also very low with pseudo-3D and there's an increase in the 3D stage with true 3D RC values being higher. With the ML modeled RCs, we also provide a better estimation of the total power for faster feedback during chip design.

6 CONCLUSION

In this work, we presented an ML-based RC modeling framework for monolithic 3D designs. Using this, we have shown a significant reduction in the TNS of the 3D designs without over-optimizing the design or increasing the design run-time. As a result of our model, we also have a better total power prediction at pseudo-3D stage. The WNS is more sensitive and correctly identifying the timing criticality of the nets and assigning corresponding weights is required to minimize the WNS further. The training model can also be incrementally improved after each new design to improve its accuracy.

ACKNOWLEDGMENTS

This research is partially funded by the DARPA ERI 3DSOC Program under Award HR001118C0096, the Semiconductor Research Corporation under Task 2929, and the National Research Foundation of Korea under NRF-2020M3F3A2A02082445.

REFERENCES

[1] B. W. Ku, K. Chang, and S. K. Lim. Compact-2d: A physical design methodology to build two-tier gate-level 3d ics. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2019.
[2] S. Panth, K. Samadi, Y. Du, and S. K. Lim. Shrunk-2-D: A Physical Design Methodology to Build Commercial-Quality Monolithic 3-D ICs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017.
[3] L. Brunet and C. Fenouillet-Beranger et al. Breakthroughs in 3D Sequential technology. In IEEE International Electron Devices Meeting (IEDM), 2018.
[4] Scalable and flexible xtreme gradient boosting (xgboost). https://xgboost.ai.
[5] Opencores Bemchmarks. https://opencores.org/projects.
[6] M. M. Ozdal, C. Amin, A. Ayupov, S. Burns, G. Wilke, and C. Zhuo. An Improved Benchmark Suite for the ISPD-2013 Discrete Cell Sizing Contest. In Proc. ACM International Symposium on Physical Design, 2013.
[7] S. Panth, K. Samadi, Y. Du, and S. K. Lim. Placement-Driven Partitioning for Congestion Mitigation in Monolithic 3D IC Designs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015.