

# PCT-Cap: Point Cloud Transformer for Accurate 3D Capacitance Extraction\*

Ye Cai<sup>1,\*</sup>, Yuyao Liang<sup>1,\*</sup>, Zhipeng Luo<sup>3</sup>, Biwei Xie<sup>4,2</sup>, Xingquan Li<sup>2,3,✉</sup>

<sup>1</sup>College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China,

<sup>2</sup>Peng Cheng Laboratory, Shenzhen, China,

<sup>3</sup>Center of Machine Learning and Optimization, Minnan Normal University, Zhangzhou, China,

<sup>4</sup>State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China,

Email: ycai@szu.edu.cn, 2100271085@email.szu.edu, lzp2680@mnnu.edu.cn, xiebiwei@ict.ac.cn, ✉lixq01@pcl.ac.cn

**Abstract**—Accurate parasitic capacitance extraction becomes increasingly essential in advanced technology nodes. 2.5D extraction requires more effort to maintain accuracy comparable to the 3D solver. In this work, two experiments reveal that error deviation in the 2.5D method enlarges as wire width shrinks and length increases. A novel Gauss law-based point cloud is proposed to model the 3D capacitance problem. With the proposed data representation, a transformer-based neural network architecture, called PCT-Cap, is designed for 3D pattern matching. PCT-Cap exhibits much better performance than the ResNet-based capacitance models. Extensive experiments on 28nm technology demonstrate that PCT-Cap can accurately predict over 95.41% of total capacitance with an error margin of less than 5%. Additionally, PCT-Cap achieves 95.90% accuracy in predicting coupling capacitance within a margin of error of less than 10%. PCT-Cap achieves at least a 55× speeding up compared with the commercial extraction tool while consuming negligible memory.

**Index Terms**—3D capacitance extraction, point cloud, transformer network, pattern match

## I. INTRODUCTION

Accurate parasitic extraction is increasingly significant in delay estimate, power calculation, and crosstalk analysis during the nanometer process. As the feature size shrinks, the back-end interconnect wires are getting narrower and closer, which enables a more flexible routing strategy and high integration. Multiple dielectric environments under advanced process technology nodes bring additional challenges to capacitance extraction [1].

In chip manufacturing, the process has inherently been 3D. 3D numerical methods, such as field solvers [2], represent the most precise approach, which inherently captures the intricacies of realistic geometry. However, solvers face challenges for full-chip capacitance extraction under the demands of large-scale chip analysis.

Nowadays, the prevailing methodology for full-chip capacitance extraction is pattern matching, characterized by two primary steps [3]: 1) pattern library preparation and 2) layout parasitic extraction. The current pattern matching primarily refers to 2.5D extraction technology, which projects the 3D interconnect wires onto the XOZ and YOZ planes.

These projected 2D structures are then matched against patterns represented in various forms such as look-up tables (LUTs) [4], analytical and empirical formulas [5], or rule-based patterns [6], [7]. The creation, classification, and refinement of adequate pattern structures are meticulously executed, followed by precise calculation using field solvers like [2]. Subsequently, a pre-characterized patterns library is constructed, facilitating large-scale capacitance extraction with commendable efficiency. Nevertheless, the potential pattern mismatch is an obstacle to meeting the precision requirements for capacitance extraction in advanced process technologies, due to the escalating complexity of interconnects and the limited coverage of available patterns.

The prevalence of machine learning and neural networks has been boosting capacitance extraction, fostering some significant advancements in research [7]–[10]. These studies focus on encoding conductor information using various feature representations and leveraging sophisticated models. Additionally, certain related investigations [11] employ data augmentation to alleviate the considerable burden of dataset construction. Such endeavors partially reduce pattern mismatches, enhance pattern coverage, and optimize memory utilization.

As technology progresses, the complexity of interconnect structures and dielectric environments increases. The accuracy of 2.5D extraction methods may eventually reach its limit. 3D field solver methods can achieve the best accuracy when given sufficient runtime and memory. Nevertheless, artificial intelligence (AI) models can meet accuracy requirements through elaborated feature representation, exquisite network framework, and appropriate training tricks. Because the values in LUT are discrete, interpolation is often required when using LUT. AI models serve as comprehensive pattern libraries, offering continuous output even for 3D structures. Moreover, AI models can significantly reduce runtime via GPU acceleration [9]. Given time, AI models may emerge as desirable alternatives to pattern-matching extraction methodologies.

However, previous AI-based capacitance extraction suffers from two main problems. Firstly, the feature representation inadequately incorporates dielectric properties. Consequently, models closely associated with the manufacturing process tend

\*Equivalent contribution author.

to favor layer-wise training, limiting their adaptability to the trend of increasing layers. Secondly, existing 2D feature-based networks lack robustness in representing 3D data effectively. Consequently, models heavily reliant on geometric features struggle to discern the intricate relationship between solid geometry and realistic capacitance values. In this paper, we first analyze the deviation between 2.5D and 3D extraction technologies, then propose a point cloud-based feature to represent 3D layout, and further input point cloud into the transformer network. The main contributions of this paper are listed as follows:

- To explore the 3D geometry influence on capacitance extraction under advanced technologies, two experiments are carried out based on the essential factors of interconnect. Compared with the 3D method, the 2.5D extraction brings accuracy loss, and this deviation will enlarge with width shrinking and length increasing.
- To efficiently model the 3D task and reduce storage consumption, the point cloud is utilized to represent the 3D layout. This data representation can flexibly characterize the process variation of metal conductors and dielectrics under advanced process nodes.
- To clearly describe the capacitance problem of a multi-conductor system in a multi-dielectric environment, the point cloud feature based on Gauss's law is proposed. In addition, a framework of self-attention mechanism is designed, called PCT-Cap. This methodology implements 3D pattern matching, which directly increases the upper limit of accuracy for extraction.
- Experiment comparisons show the effectiveness and efficiency of the proposed PCT-Cap. PCT-Cap can predict over 95.41% of total capacitance with an error of less than 5%. PCT-Cap can predict over 95.90% of coupling capacitance with an error of less than 10%. PCT-Cap run on the CPU server is at least  $55\times$  speeding up compared with a commercial extraction tool.

## II. 3D CAPACITANCE EXTRACTION IS NEED

### A. Capacitance Calculation Theory

For an electrostatic balance system with  $n + 1$  conductors, whose standard capacitance equations is

$$\begin{cases} Q_1 = C_{10}\varphi_{10} + \dots + C_{1i}\varphi_{1i} + \dots + C_{1n}\varphi_{1n} \\ \dots \\ Q_i = C_{i1}\varphi_{i1} + \dots + C_{i0}\varphi_{i0} + \dots + C_{in}\varphi_{in} \\ \dots \\ Q_n = C_{n1}\varphi_{n1} + \dots + C_{ni}\varphi_{ni} + \dots + C_{n0}\varphi_{n0} \end{cases} \quad (1)$$

where  $Q_i$  is the charge on conductor  $i$ ,  $C_{ij}$  is the capacitance value between conductors  $i$  and  $j$ ,  $\varphi_{ij}$  is the potential difference between conductors  $i$  and  $j$ . Conductor 0 is always regarded as the ground, i.e.,  $\varphi_0 = 0$ .

Supposed that conductor  $i$  is the main conductor we are concerned, we set  $\varphi_i = 1$  and  $\varphi_j = 0$ , where  $j \neq i$ . Then, we have

$$\begin{cases} Q_i = \sum_{j=0, j \neq i}^n C_{ij} = C_{ii} \\ Q_j = -C_{ij}, \quad j \neq i \end{cases} \quad (2)$$

Finally, We can obtain the capacitance value by Gauss's law

$$Q = \oint \vec{D} \cdot d\vec{S} = \oint \varepsilon \vec{E} \cdot d\vec{S} = - \oint \varepsilon \nabla \varphi \cdot d\vec{S} \quad (3)$$

where  $Q$  is the charge in the calculation area,  $\vec{D}$  is electric displacement,  $\varepsilon$  is the permittivity, and  $\vec{E}$  is electric field intensity. To obtain the potential distribution within the computational domain, the field solver figures out the Laplace equation (4).

$$\nabla^2 \varphi = 0 \quad (4)$$

### B. 3D Downstream to 2.5D Model

Modeling and application scenarios lead to different capacitance calculation formulas. The 1D capacitance model only considers a single dimension for the Laplace equation, which has no practical application scenarios. The 2D capacitance models are often used to construct pattern libraries. The 2.5D method utilizes multiple 2D capacitance models to approximate the 3D capacitance effects, due to the high computational cost of the 3D field solver. Many simplified formulas can be used to fasten 3D capacitance calculation. In [4], we have the area capacitance formula:

$$C_a = P_{l1,l2}(s) \times s \quad (5)$$

The lateral capacitance formula:

$$C_l = P_l(d) \times l \quad (6)$$

And the fringe capacitance formula:

$$C_f = \begin{cases} P_{l1,l2}(d) \times l + P_{l2,l1}(d) \times l, & d \geq 0 \\ 0, & d < 0 \end{cases} \quad (7)$$

Fig. 1 shows an ordinary cross structure in adjacent metal layers. Ref. [3] provides an approximate calculation formula for capacitance for this structure.

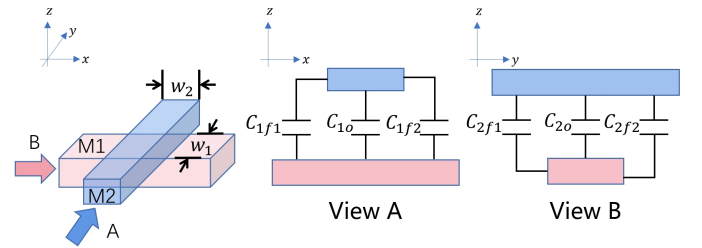


Fig. 1. 2.5D capacitance calculation for a crossover structure.

In Fig. 1, suppose the 2D capacitance between M1 and M2 in the view along direction A is  $C_A$ , then we have

$$C_A = C_{1f1} + C_{1o} + C_{1f2} \quad (8)$$

where  $C_{1f1}$  and  $C_{1f2}$  are two fringe capacitance values, and  $C_{1o}$  is the overlapping capacitance. So does the view along direction  $B$ :

$$C_B = C_{2f1} + C_{2o} + C_{2f2} \quad (9)$$

Then, the 2.5D capacitance value,  $C_{2.5}$ , is

$$C_{2.5D} = C_A w_1 + (C_B - C_{2o}) w_2 \quad (10)$$

where  $w_1$  and  $w_2$  are widths of M1 and M2, respectively.

### C. Deviation between 2.5D and 3D Capacitance Extraction Technologies

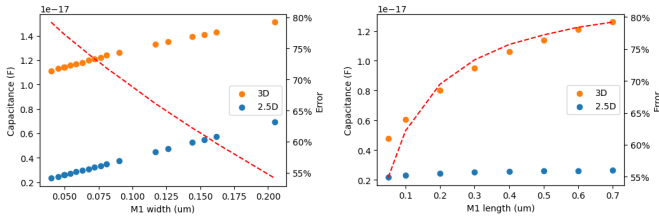
In this section, we try to analyze the accuracy deviation between the 3D numerical method and the 2.5D extraction. It turns out that this deviation will enlarge with the width shrinking and length increasing.

Two simple experiments are based on relevant details of the 28nm interconnect technology format (ITF) specifications file. Two crossover conductors lie in the center of a  $50 \times 50 \times 50$  region, with relative permittivity 1. The center coordinate of the bottom surface of M1 is (25, 25, 25), while M2 is (25, 25, 25.175). The width of M1 and M2 are both 0.05, length 0.5, and height 0.09 microns. The electric field intensity is supposed to be 0 on the Neumann boundary. The capacitance between M1 and M2 is calculated by a commercial tool by setting the number of grids to  $10^6$ .

In Fig. 2, the blue points show the 2.5D method by Eqs.(8)-(10), while the orange is the 3D numerical calculation result. The red line depicts *Error* by

$$Error = \left| \frac{C_{2.5D} - C_{3D}}{C_{3D}} \right| \times 100\% \quad (11)$$

where  $C_{3D}$  is the golden capacitance value. The width and length sampling is based on details of ITF files.



(a) Deviation as width. (b) Deviation as length.

Fig. 2. Capacitance deviation between 2.5D and 3D.

First, all parameters are fixed except for the width of M1. Fig. 2(a) shows that the error of the 2.5D method gradually increases as the width of M1 shrinks. Second, the length of M1 is an independent variable. Fig. 2(b) shows that the error of the 2.5D method gradually increases as the length rises.

In the previous technology nodes, the 2.5D method and layer-wise capacitance models are widely applied with the balanced trade-off between accuracy and time by simplifying the physical model. However, wires are getting closer and narrower. Dielectric environments are more complicated. 2D or 2.5D patterns are making more efforts to minimize such an accuracy gap.

### D. 3D Capacitance Extraction Based on Neural Network

This subsection describes the flow of 3D pattern matching based on neural networks. This methodology can model conductors with a variable quantity, 3D complicated interconnect structures, and process-related dielectric environments. Furthermore, it improves accuracy from the perspective of model capability.

The patterns established through 3D geometric structures can directly increase the upper limit of precision because they comprehensively factor in the actual physical model. Capacitance extraction is such a difficult problem that sophisticated geometry, complex dielectric environments, and specific boundary conditions concurrently constitute the Laplace equation. It is a challenging task to define universal 3D patterns with particular forms and parameters.

Even so, the neural network helps to avoid designing handcrafted patterns because it can learn through backward propagation. A neural network model serves a comparable functionality as a more versatile and powerful pattern library. Arising from regression tasks, NN-based capacitance models have achieved noticeable success in parasitic extraction with the help of elaborated feature representation and exquisite network framework.

Fig. 3 shows the flow of 3D pattern matching based on the neural network. According to [9], The sample structures for 3D extraction are randomly chopped from tape-out IC design under the same process technology. The field solver calculates the capacitance label for the regression task. Finally, Well-trained capacitance models are capable of performing full-chip extraction.

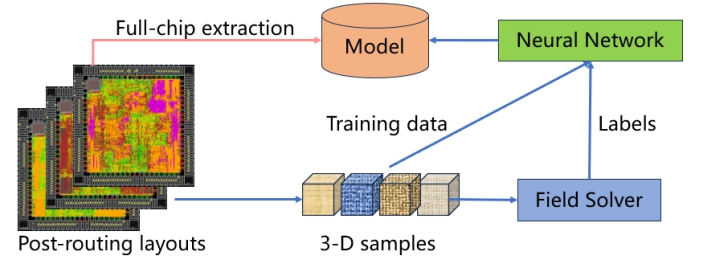


Fig. 3. The flow of 3D capacitance extraction based on neural network.

## III. PCT-CAP: POINT CLOUD TRANSFORMER FOR 3D CAPACITANCE EXTRACTION

A novel data representation by point cloud is present, inspired by the traditional flow of capacitance calculation. With the proposed feature representation, a framework based on the self-attention mechanism is designed for capacitance prediction. An effective data augment method is also introduced to improve accuracy.

### A. Gauss Law-Based Point Cloud Feature

3D data can be represented in different formats, including depth images, point clouds, meshes, and volumetric grids. As a commonly used format, point cloud representation preserves

the original geometric information in 3D space without any discretization [12].

In the scene of parasitic capacitance extraction, manufacturing effects, such as etch, thickness variation, and conformal dielectric, grow the geometric complexity of the metal structure and the dielectric distribution. To uniquely describe a specific 3D computational task, the point cloud maintains flexibility and versatility since others format, like depth image data, fail to be continuous in the third dimension. Furthermore, because of the electrostatic shielding effect, point clouds can generate points where we are concerned, with controlled density, even in the planar situation. Point cloud saves storage costs compared with image, which records all information in a plane.

From Eqs. (1) - (3), the basic capacitance-solving idea shows that the capacitance value can directly relate to variables in Gauss's law. The point cloud features based on Gauss's law are inspired and exhibited in Table I.

TABLE I  
SELECTED FEATURES.

Feature symbols	Description
$x, y, z$	Coordinates in the space
$n_x, n_y, n_z$	The normal vector of a dielectric boundary
$\varepsilon_r$	Relative permittivity
$\Phi$	Sign of the electric flux

Unlike resistance calculation, the Laplace equation for capacitance problems calculates the region of the dielectric space excluding the conductors. The capacitance values may be distinct with different environments, even if the electrostatic system has the same conductor distribution within the calculation region, as the Maxwell Equation boundary condition for electric flux density shows:

$$\vec{D}_{2n} - \vec{D}_{1n} = \sigma \quad (12)$$

where  $\vec{D}_{2n}$  and  $\vec{D}_{1n}$  are the normal components of  $\vec{D}$  on the two sides of the boundary, and  $\sigma$  is the surface density of free charges. Consequently, the generation region of the point cloud should be based on the boundary of the dielectric.

Channel  $\Phi$  is inspired by the sign of the electric flux:

$$\Phi = \int_S \vec{E} \cdot d\vec{S} \quad (13)$$

The electric field lines  $\vec{E}$  originate from the master conductor and end at the environment conductor. So we can get different signs at different conductor surfaces. Prior works [7], [9] mix density-based and voltage-based channels together, destroying the channels' original physical meaning.

Fig. 4 gives some examples of channel setting. Conductor 1 is a master conductor while 2 and 3 are environment conductors. The Point lists some cases.  $\varepsilon_r$  means relative permittivity channel.  $\Phi_{C_{11}}$  means channel  $\Phi$  setting when predicting total capacitance  $C_{11}$ .  $\Phi_{C_{12}}$  means channel  $\Phi$  setting when predicting coupling capacitance  $C_{12}$ .

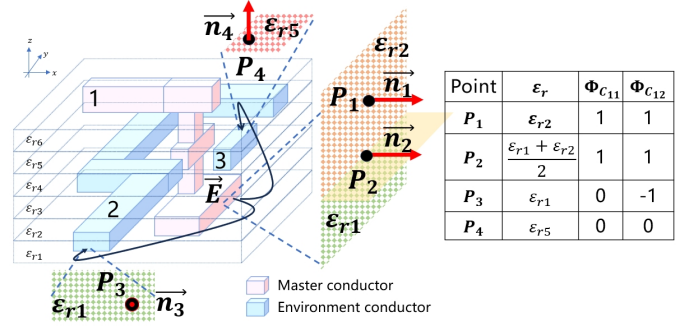


Fig. 4. Feature illustration.

$P_1$  is a point in the outer boundary of conductor 1, the master conductor. So we set channel  $\Phi$  to 1 no matter predicting  $C_{11}$  or  $C_{12}$ , i.e.  $\Phi_{C_{11}} = \Phi_{C_{12}} = 1$ .  $P_1$  is in the dielectric area in  $\varepsilon_{r2}$ . So we set its channel  $\varepsilon_r$  to  $\varepsilon_{r2}$ .

$P_2$  is also a point in the outer boundary of conductor 1. So we set  $\Phi_{C_{11}} = \Phi_{C_{12}} = 1$ .  $P_2$  is in the interface of the dielectric area in  $\varepsilon_{r1}$  and the dielectric area in  $\varepsilon_{r2}$ . This work sets channel  $\varepsilon_r$  to the mean relative permittivity value at the multi-dielectric interface, i.e.  $\varepsilon_r = \frac{\varepsilon_{r1} + \varepsilon_{r2}}{2}$ .

$P_3$  is a point in the outer boundary of conductor 2, environment conductor. So we set  $\Phi_t = 0$  when predicting  $C_{11}$ . we set  $\Phi_{C_{12}} = -1$  when predicting  $C_{12}$ , as conductor 2 is target environment conductor.

$P_4$  is a point in the outer boundary of conductor 3, environment conductor. So we set  $\Phi_t = 0$  when predicting  $C_{11}$ . we set  $\Phi_{C_{12}} = 0$  when predicting  $C_{12}$ , as conductor 3 is non-target environment conductor.

## B. Framework

This subsection introduces a framework for capacitance prediction based on the self-attention mechanism. The novel architecture named *Point Cloud Transformer for Capacitance (PCT-Cap)*, in Fig. 5.

There is a point cloud with the feature map  $F \in \mathbb{R}^{N \times d_{in}}$ , where  $N$  is the number of points, and  $d_{in}$  is the dimension of input channels. In this work,  $N$  is 1024, while  $d_{in}$  is 8 as Table I shows. Second, the data augment module performs plane symmetry, such as symmetry about  $x = x_0$ , on the input point cloud without changing the size of  $F$ . Third, with the transposed  $F^T$ , appropriate convolutional layers expand the feature channel to  $d_e$  so the network can better perform regression tasks for capacitance prediction. Then, the feature map  $F_{in} \in \mathbb{R}^{d_e \times N}$ , where  $d_e$  is 256.

The self-attention module is based on the inner product below:

$$\begin{cases} A = [a_{ij}] = \text{softmax}(Q^T K) \\ F_{out} = \frac{VA}{\sum_{j=1}^N a_{ij}} + F_{in} \end{cases} \quad (14)$$

where linear transformations generate  $Q$ ,  $K$ , and  $V$ . Repeat this module  $N$  times as illustrated in Fig. 5. With the transposed  $F_{out}^T$ , an average pooling and MLP layers shrink the feature map into a single value for practical capacitance prediction.

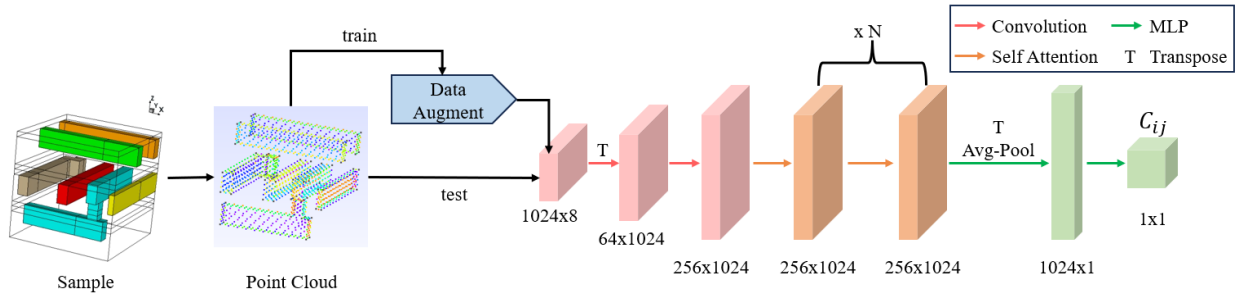


Fig. 5. Our PCT-Cap framework.

The training uses the Adam optimizer with a learning rate of 0.0001, the scheduler with Cosine Annealing Warm Restarts, and a batch size of 64. The loss function is RMSE.

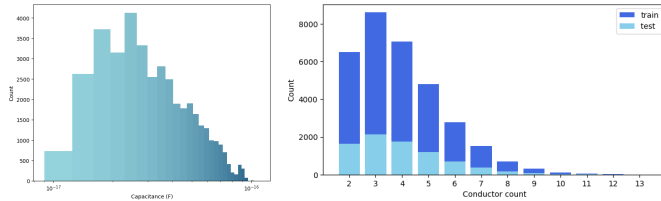
#### IV. EXPERIMENTAL RESULT

All experiments are performed on a Linux server with 4 Intel Xeon Platinum 8380 CPUs at 2.3 GHz, 2 NVIDIA A100 GPUs, and 1TB RAM.

##### A. Dataset

The 3D capacitance dataset is randomly sampled from a 28nm tape-out design whose core size is  $1160.04\mu m \times 1155.6\mu m$ . The 3D sampling window is height-adaptive with XOY-plane area sized  $0.5\mu m \times 0.5\mu m$ . Every sample can accommodate three routing layers with the capacity for five parallel wires in the same layer. Each 3D window contains only one master conductor, which is almost located at the center of the sample. The capacitance value, the label, is calculated by a golden standard commercial parasitic extraction tool.

The 3D dataset consists of two parts: total capacitance and coupling capacitance. Both sub-datasets are randomly split into two parts: 80% point clouds in the training dataset and 20% in testing. Meanwhile, the distribution of sample quantities remains consistent between the training dataset and the testing based on the number of conductors in a 3D window, which is determined by routing tools. Fig. 6(a) shows the total capacitance distribution or label distribution, and Fig. 6(b) shows the dataset splitting in terms of conductor counts in a 3D sampling window. For the total capacitance dataset, 32438 point clouds are for training and 8106 for testing. Fig. 7 shows the coupling capacitance dataset overview with 36747 training point clouds and 9180 testing.



(a) Capacitance distribution. (b) Dataset splitting.

Fig. 6. Total ground capacitance dataset.

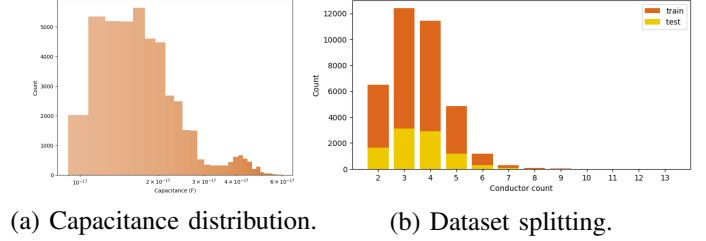


Fig. 7. Coupling capacitance dataset.

##### B. Comparisons on NN Framework

Tabel II and III outline the accuracy performance of different capacitance models in the total and coupling capacitance datasets, respectively. Error evaluation of each testing sample is denoted as  $Err$ :

$$Err = \left| \frac{C_{pred} - C_{label}}{C_{label}} \right| \times 100\% \quad (15)$$

where  $C_{pred}$  is the prediction of the NN-based model, and  $C_{label}$  is the reference capacitance.  $Err_{avg}$  is the mean value of all predicted  $Err$ .  $Err_{max}$  is the maximum of  $Err$ .  $Std$  is the standard deviation of  $Err$ .  $P(< 5\%)$  in Tabel II means that a model predicts capacitance value with a probability of  $P$  to achieve the accuracy requirement ( $Err < 5\%$ ). So does  $P(< 10\%)$  in Tabel III. There are some minus signs ( $-$ ) in Tabel II and Tabel III, which stand negligible experiment results because of the poor accuracy performance.

RFR is a random forest regression model. PointNet2 is based on [13], a classical and versatile framework for point cloud tasks. PCT develops from [14], which integrates the attention mechanism. PointNet2 and PCT are not designed for regression tasks initially. Hence, a simple regression network CNN3-MLP3, organized by 3-layer convolution and 3-layer MLP, is conceived. Four ResNet models are developed from [15], with mere modification of the head of the network to facilitate regression tasks. ResNet also serves as the basic architecture of [9]. PCT-Cap is the architecture proposed in this work. PCT-Cap- $N$  repeats the self-attention module  $N$  times. All models are implemented with PyTorch.

$P(< 5\%)$  of PCT-Cap architecture is at least 4.96% higher than the ResNet series in Tabel II. The PCT-Cap architecture outperforms the ResNet family by at least 4.20% in terms of  $P(< 10\%)$  in Tabel III.



TABLE II  
MODEL ACCURACY FOR TOTAL CAPACITANCE DATASET

Model Name	$P(< 5\%)$	$Err_{avg}$	$Err_{max}$	$Std$
RFR	8.06%	-	-	-
PointNet2	11.98%	-	-	-
PCT	38.34%	-	-	-
CNN3-MLP3	78.27%	-	-	-
ResNet-18	86.79%	-	-	-
ResNet-34	87.43%	-	-	-
ResNet-50	90.45%	2.27%	164.41%	0.0313
ResNet-101	90.38%	2.29%	59.34%	0.0275
PCT-Cap-6	95.41%	1.59%	40.64%	0.0196
PCT-Cap-8	96.03%	1.52%	37.92%	0.0193
PCT-Cap-10	96.53%	1.40%	30.97%	0.0189

TABLE III  
MODEL ACCURACY FOR COUPLING CAPACITANCE DATASET

Model Name	$P(< 10\%)$	$Err_{avg}$	$Err_{max}$	$Std$
ResNet-18	89.89%	-	-	-
ResNet-34	90.65%	-	-	-
ResNet-50	91.61%	-	-	-
ResNet-101	91.70%	-	-	-
PCT-Cap-6	95.90%	2.64%	48.62%	0.0351
PCT-Cap-7	96.13%	2.62%	46.09%	0.0336
PCT-Cap-8	96.38%	2.41%	42.82%	0.0322

In Fig. 8, (a) shows the relative error distribution of an instance of PCT-Cap-10 for total capacitance prediction, and (b) shows PCT-Cap-8 for coupling capacitance prediction.

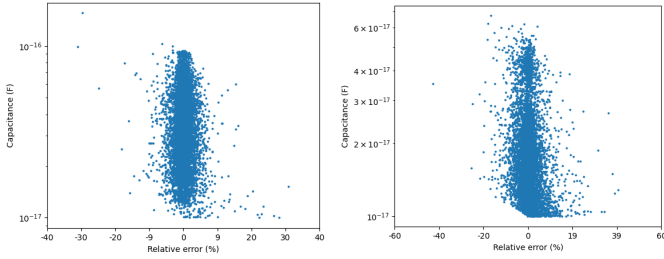


Fig. 8. Relative error distribution of PCT-Cap.

### C. Comparisons on Runtime and Model Size

The Comparison of runtime and storage cost is shown in Table IV. The testing batch size is 64. Runtime measurement is based on the total capacitance dataset with size of 8106. All models are deployed on the CPU for total capacitance inference. PCT-Cap family runs at least  $55\times$  faster than the commercial tool with desirable accuracy performance.

As for model size, a PCT-Cap-6 model has only 2248129 parameters occupying 8.6 MB storage, which is enough to achieve an accuracy requirement of over 95% probability within 5% relative error.

## V. CONCLUSION

In this work, two simple experiments show the 2.5-D method enlarges error with the wire width shrinking and length

TABLE IV  
RUNTIME AND STORAGE COST FOR TOTAL CAPACITANCE DATASET

Method	Time (s)	Speedup	Parameters	Storage (MB)
Commercial	18488.6	1	-	-
ResNet-50	87.6	211	30568769	117.0
ResNet-101	175.6	105	60780353	232.7
PCT-Cap-6	177.9	104	2248129	8.6
PCT-Cap-8	253.7	73	2641345	10.1
PCT-Cap-10	333.1	55	3034561	11.6

increasing, compared with the 3-D solver. The Gauss law-based point cloud feature is proposed to represent the 3-D capacitance problem. PCT-Cap, an architecture based on self-attention, is designed for capacitance prediction. PCT-Cap takes advantage of accuracy and storage over ResNet-based models by extensive experiments.

## ACKNOWLEDGMENT

This work is supported in part by the Major Key Project of PCL (No. PCL2023A03), the National Natural Science Foundation of China (No. 42301503).

## REFERENCES

- [1] W. Yu, M. Song, and M. Yang, "Advancements and challenges on parasitic extraction for advanced process technologies," in *Proc. of ASP-DAC*, Jan 2021.
- [2] S. Wu, B. Xie, and X. Li, "An adaptive partition strategy of galerkin boundary element method for capacitance extraction," in *Proc. of ASP-DAC*, pp. 541–546, 2023.
- [3] W. Yu and Z. Wang, "Capacitance extraction," *Encyclopedia of RF and Microwave Engineering*, Apr 2005.
- [4] S. S. Bo Yang, "ICCAD 2018 CAD contest on timing-aware fill insertion." [Online]. Available: [https://iccad-contest.org/2018/Problem\\_C/2018ICCADContest\\_ProblemC.pdf](https://iccad-contest.org/2018/Problem_C/2018ICCADContest_ProblemC.pdf).
- [5] A. Zhang, W. Zhao, Y. Hu, J. He, Q. He, L. Song, H. Zhou, and Y. Wu, "A field-based parasitic capacitance model with 3-D terminal and terminal fringe components," in *Proc. of ASQED*, pp. 166–170, IEEE, 2015.
- [6] N. K. Karsilayan, J. Falbo, and D. Petranovic, "Efficient and accurate RIE modeling methodology for BEOL 2.5D parasitic extraction," in *Proc. of MWSCAS*, pp. 519–522, 2014.
- [7] M. S. Abouelyazid, S. Hammouda, and Y. Ismail, "Accuracy-based hybrid parasitic capacitance extraction using rule-based, neural-networks, and field-solver methods," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 12, pp. 5681–5694, 2022.
- [8] Z. Li and W. Shi, "Layout capacitance extraction using automatic pre-characterization and machine learning," in *Proc. of ISQED*, pp. 457–464, 2020.
- [9] D. Yang, W. Yu, Y. Guo, and W. Liang, "CNN-Cap: Effective convolutional neural network based capacitance models for full-chip parasitic extraction," in *Proc. of ICCAD*, pp. 1–9, 2021.
- [10] M. S. Abouelyazid, S. Hammouda, and Y. Ismail, "Connectivity-based machine learning compact models for interconnect parasitic capacitances," in *Proc. of MLCAD*, pp. 1–6, 2021.
- [11] Y. Ma, X. Xu, S. Yan, Y. Zhou, T. Zheng, Z. Ren, and L. Chen, "Extraction of interconnect parasitic capacitance matrix based on deep neural network," *Electronics*, vol. 12, no. 6, 2023.
- [12] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021.
- [13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Proc. of NIPS*, Jun 2017.
- [14] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Computational Visual Media*, p. 187–199, Jun 2021.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of CVPR*, Jun 2016.