Data-Driven Fast Electrostatics and TDDB Aging Analysis*

Shaoyi Peng, Wentian Jin, Liang Chen and Sheldon X.-D. Tan
Department of Electrical and Computer Engineering, University of California at Riverside
Riverside, CA

ABSTRACT

Computing the electric potential and electric field is a critical step for modeling and analysis of VLSI chips such as TDDB (Time dependent dielectric breakdown) aging analysis. Data-driven deep learning approach provides new perspectives for learning the physics-law and representations of the physics dynamics from the data. In this work, we propose a new data-driven learning based approach for fast 2D analysis of electric potential and electric fields based on DNNs (deep neural networks). Our work is based on the observation that the synthesized VLSI layout with multi interconnect layers can be viewed as layered images. Image transformation techniques via CNN (convolutional neural network) are adopted for the analysis. Once trained, the model is applicable to any synthesized layout of the same technology. Training and testing are done on a dataset built from a synthesized CPU chip. Results show that the proposed method is around 138x faster than the conventional numerical methods based software COMSOL, while keeping 99% of the accuracy on potential analysis, and 97% for TDDB aging analysis.

CCS CONCEPTS

• **Hardware** → **Aging of circuits and systems**; **Metallic interconnect**; *Modeling and parameter extraction.*

KEYWORDS

Reliability, TDDB, Lifetime, Machine Learning

ACM Reference Format:
Shaoyi Peng, Wentian Jin, Liang Chen and Sheldon X.-D. Tan. 2020. Data-Driven Fast Electrostatics and TDDB Aging Analysis. In *2020 ACM/IEEE Workshop on Machine Learning for CAD (MLCAD '20), November 16–20, 2020, Virtual Event, Iceland.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3380446.3430623

1 INTRODUCTION

Electrostatics is an important subject of study as it is pivotal in many VLSI modeling applications. The goal is to compute electric potentials and electric fields with some given voltage and current sources (boundary conditions). In the back-end of VLSI chips, strong electric field can induce failure of the dielectrics, which is known as TDDB [7]. Simulation of this aging effect requires electrostatics. Also, several methods of parasitic extraction involve electrostatics simulations in the chip layouts [1, 14]. Furthermore, global placement is also proposed to be modeled as electrostatic problem recently [6].

Traditionally, electrostatics is primarily solved by numerical methods, with spatial discretization of the governing equation. Such numerical methods typically require a mesh of the complicated layout or geometry, which can be computationally prohibitive for large designs.

Recently, machine learning, especially deep learning has revolutionized fields such as image, text, and speech recognition [5]. These fields require statistical approaches which can model nonlinear dynamic functions very well. Deep learning has shown potential and promise in practice for such tasks. But using DNN to learn the spatial and temporal dynamics in physics laws is a less explored field. Some early works have been proposed recently [12, 13, 15] to solve the partial differential equation based on supervised learning. However, the problems those methods solve are too small to be practically significant.

Inspired by these observations, in this work, we propose a new data-driven learning based approach for fast analysis of electric potential and electric fields, based on which TDDB aging analysis can also be done efficiently. Our new contributions are as follows:

- Our work is based on the observation that the synthesized VLSI layout with multi interconnect layers can be viewed as layered images. We adopt image transformation techniques to solve electrostatics. Specifically, a CNN based neural network is used. A method called layout partition is used to reduce the size of to suitable for neural networks.
- There is no need to retrain the model for new input layouts. Once trained, the model is applicable to any synthesized layout of the same technology.
- Results on a few VLSI layout tile examples show that the proposed method can deliver 138x speedup over the conventional numerical method like FEM (finite element method), while keeping 99% of the accuracy on electric potentials, and 97% for TDDB aging analysis.

This paper is organized as follows: Section 2 reviews some similar work. Section 3 gives a brief introduction of electrostatics and TDDB. Section 4 introduces in detail the way to encode the electrostatics problem into layered images, architecture of the neural network. Section 5 introduces the dataset used to train and test the model and lists some experimental results to show the performance in speed and accuracy. Finally, Section 6 concludes this paper.

2 RELATED WORK

In convention, electrostatics problems are solved using discretization methods such as FEM or finite difference method (FDM) [14]. Results from commercial tools based on FEM such as COMSOL

*This work is supported in part by NSF grants under No. CCF-1816361, in part by NSF grant under No. CCF-2007135 and No. OISE-1854276.

MLCAD '20, November 16–20, 2020, Virtual Event, Iceland

ACM ISBN 978-1-4503-7519-1/20/11.
https://doi.org/10.1145/3380446.3430620

are usually deemed golden. However, the speed of these methods is limited.

Recently, deep learning is gaining a lot of attention and has been explored in some practical simulation problems, mainly because of its speed once the model is well trained. Tang et al. [12] proposed to solve Poisson's equation, specifically in the case of electrostatics, using modern CNN-based model. The setup of the problem is similar to this work, where the PDEs (partial differential equations) remain unchanged (Poisson's equation), and the input variables are locations of excitation sources and distribution of permittivity (a coefficient in the PDE). Error was reported as low as 3%. However, the problem size is relatively small (the grid size is 64×64 with no extension). The similar method was also used by Zhang et al. [15], where charge distribution and boundary condition (voltage) are used as the inputs of the neural network. A regular grid is used in all these works. Tompson et al. [13] try to deal with a more complicated time-dependant fluid flow problem. Yet convolutional network is used to speed up some steps in the overall simulation instead of solving the whole problem on its own.

This work also aims to use CNN-based network to solve electrostatics. The problem size is comparatively larger, and it can be further expanded through the used method of layout partition.

3 PRELIMINARIES OF ELECTROSTATICS AND TDDB

Electrostatics studies the distribution of electric potential and field in cases where the charges are static, i.e., there is no electric current. Electric potential is governed by the first equation of the Maxwell's equations, also known as Gauss's law:

$$\nabla^2 u = \frac{-\rho}{\epsilon} \tag{1}$$

where u is electric potential, ρ is static charge density, and ϵ is the permittivity.

The equation usually comes with the following Dirichlet and Neumann boundary conditions:

$$\begin{aligned} u &= f(x),\ x \in \Gamma_D, \\ \nabla u \cdot \vec{n} &= g(x),\ x \in \Gamma_N, \end{aligned} \tag{2}$$

where Γ_D is the part of the boundary where Dirichlet (voltage) boundary conditions are given, Γ_N is the part of the boundary where Neumann (electric field or current) boundary conditions are given, u is the unknown potential to be solved, $f(x)$, and $g(x)$ are given voltage sources and electric field (or current sources) at the boundaries.

In cases where static charge is absent, which this work focuses on, (1) becomes Laplace equation:

$$\nabla^2 u = 0 \tag{3}$$

With the solution of u, distribution of electric field is usually obtained by calculating the gradient as per its definition:

$$\vec{E} = -\nabla u \tag{4}$$

Solving for $\vec{E}$ under given voltage boundary conditions $f(u)$ is often of more interest for many practical problems. For example, the integral of $\vec{E}$ is used to evaluate reliability effects. TDDB is a reliability effect caused by high electric field over time. The time-to-failure (TTF) of an interconnect wire induced by TDDB can be modeled through $\sqrt{E}$ model, which is calculated as [9]:

$$TTF \propto \oint_L \exp\left(-\gamma \sqrt{E(l)}\right) dl \tag{5}$$

where γ is a coefficient fitted from experiment data, and L is the perimeter of the wire. By finding the wire with the minimum TTF, the hotspot of the designed layout can be picked out for further optimization.

Note that in this application concerning electric fields, it is sufficient that only two voltages are involved: VDD and GND. In this work, VDD is set to 1V for the ease of computation.

Further note that (5) assumes a simplification that the wires are modeled in 2D. Although interconnects in fact have 3D structures, they can still be analyzed in 2D by dropping the dimension of height and analyzing layer by layer. This is possible because the thickness of wires is relatively constant for a certain layer.

4 THE PROPOSED DATA-DRIVEN ELECTROSTATIC ANALYSIS

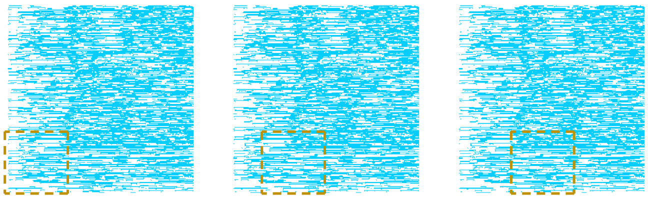
In this section, we introduce the encoding scheme of the electrostatics problem, along with the training and test data generation process, and details of the neural network used.

4.1 Problem formulation

As introduced above, this work aims to solve the distribution of electric potential and electric field in dielectrics in the VLSI back-end-of-line. Typical VLSI designs have dimensions in the order of hundreds of micrometers to millimeters. Interconnect wires that are as wide as only nanometers form complex geometries in such large area. This problem size poses the first challenge to feed this problem into a neural network. A typical input image in modern machine learning research [11] has a size of 224 pixels by 224 pixels, which is far coarser than the required complexity of VLSI back-end-of-line.

This work uses the simplification called layout partition [9] to solve this problem. The key idea is to partition one interconnect layer into smaller tiles and analyze them separately. The logic is that interconnect wires of the same layer are typically routed in the same direction, either horizontal or vertical. Thus, the voltage applied on one wire can hardly affect the potential (or electric field) several channels away, as there can be other wires in between blocking the electric field. Results at the boundaries of tiles would be inevitably inaccurate in many cases because of missing information from neighboring tiles. To fix this problem, regions at boundary can be solved in a new tile that places them in the center. Fig. 1 shows the process of layout partitioning, where the partition is labeled by brown dashed lines. The first three partitioned tiles from the layout are shown in the three subfigures respectively. The rightmost region from the first tile in (a) can be solved correctly inside the second tile in (b), which solves the inaccuracy close to boundaries. Through layout partition, electrostatics of one interconnect layer can be solved by separately solving small tiles and merging the results eventually.

Another challenge is to encode the tiles so that it fits machine learning. The chip layout is originally stored in gdsii format, in which the shapes and positions of elements are stored in a binary format. An easy approach is to convert it into images. In our problem, there are two sets of inputs, namely geometry information of



(a) (b) (c)

Figure 1: The first three partitioned tiles. The layout is cropped for demonstration purposes.

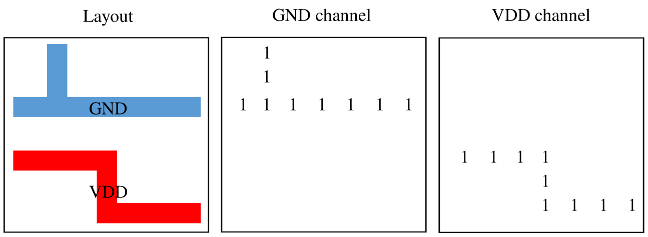
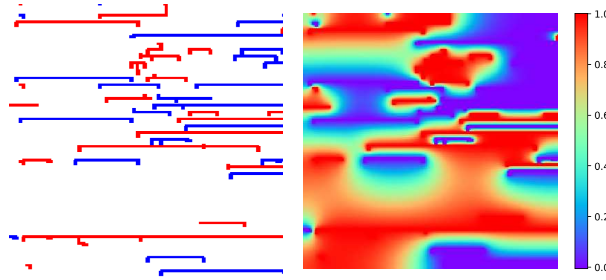


Figure 2: Encoding a layout tile into image



(a) VDD wires: red; GND wires: blue **(b) Solution of electric potential (unit: V)**

Figure 3: An example of encoded tile image of and its corresponding solution of potential

the interconnect wires and the voltage boundary conditions. Note that in the two targeted electrostatics applications introduced previously in Section 3, two voltages, VDD and GND, are sufficient for the analysis. Also, the routine used in reliability analysis [9] that VDD and GND are set alternately on wires is followed here. An important result is that every wire is set a voltage. Consequently, the locations of VDD wires and GND wires are sufficient to describe the problem, as the locations where voltages are set indicate the existence of wires. Also, the lifetime analysis is done on a worst case scenario. A simple value of $1V$ can be assumed for VDD as more complicated cases involving duty cycles and other voltage values can be simply scaled by adding coefficients [4]. The two sets of boundary conditions (VDD and GND) are encoded in different channels in the encoded image. For each pixel, the value of a channel is set to 1 if the corresponding voltage boundary condition is set at this pixel, and 0 if not. The corresponding size of a pixel can be set to the pitch size of the technology (60nm in this work), as it is basically the unit length in the geometry. The results of electric potential distribution can also be displayed as images, using the same method. This encoding is demonstrated with an example shown in Fig. 2 and Fig. 3.

4.2 Structure of the neural network

The overall architecture of the neural network is shown in Fig. 4. The hyper parameters are listed in the figure as well. Also, the output tensor sizes of each layer are listed in square brackets. The number of channels are listed last in the tensor sizes. The structure is very similar to autoencoders [2] as well as U-Net [10]. It is composed of a contracting network, called encoder, and an expansive network, called decoder. Skip connections [3] are added between the encoder and decoder layers that have same output size, shown as dashed lines.

First, the encoded layout tiles are expanded to 256×256 pixels sized square images by padding zeros on four sides. Then they are sent into the first convolutional layer, composed of four convolution kernels with sizes of 3×3, 5×5, 7×7, 9×9 respectively. They are used to capture features nearby, however in different distances. The four feature maps are concatenated and sent to the next layer. The following convolutional layers use single 5×5 convolutional kernels. All convolutional layers are followed by max pooling layer to down-sample the tensor. Through the encoder network, the image is down-sampled to 1×1×512 vector, which can be seen as latent features of the layout tile extracted by the encoder. Then, the remaining part of the network, namely the decoder, reversely up-samples the latent feature vector to the output image, which is the solution of electric potential, or u.

Instead of traditional transposed convolutional layers, the decoder uses resize-convolution blocks [8] to up-sample the latent vector, which helps avoid checkerboard artifacts in the final image and thus increases the accuracy.

Overall, once trained, the network is expected to work like a finite difference method based solver, which is able to output the electric potential distribution from the given interconnect geometry and voltage. Since all layouts synthesized with the same technology have the same pitch size, they can all be analyzed efficiently with the trained model. In other words, the model is universal as it applies to any layout of this technology.

5 NUMERICAL RESULTS AND DISCUSSIONS

The aforementioned neural network has been implemented with TensorFlow 2.0 and Python 3.7. All the following experiments are run on a Linux server equipped with 2 Intel Xeon E5-2698v3 2.3GHz processors and a Nvidia Titan X GPU.

5.1 Data preparation and training

This work is done with an example layout synthesized with a 32nm educational technology. The size of the layout is 200μm×200μm, and the interconnect layers are partitioned into tiles of 12μm×12μm.

Example tiles are extracted from two interconnect layers, specifically M3 (Metal 3) and M4, as they have horizontal and vertical routing directions respectively. In total about 12000 tiles are collected as data samples for training and testing the model. COMSOL (a commercial FEM tool) is used to simulate electrostatics on them for distributions of electric potential and electric field.

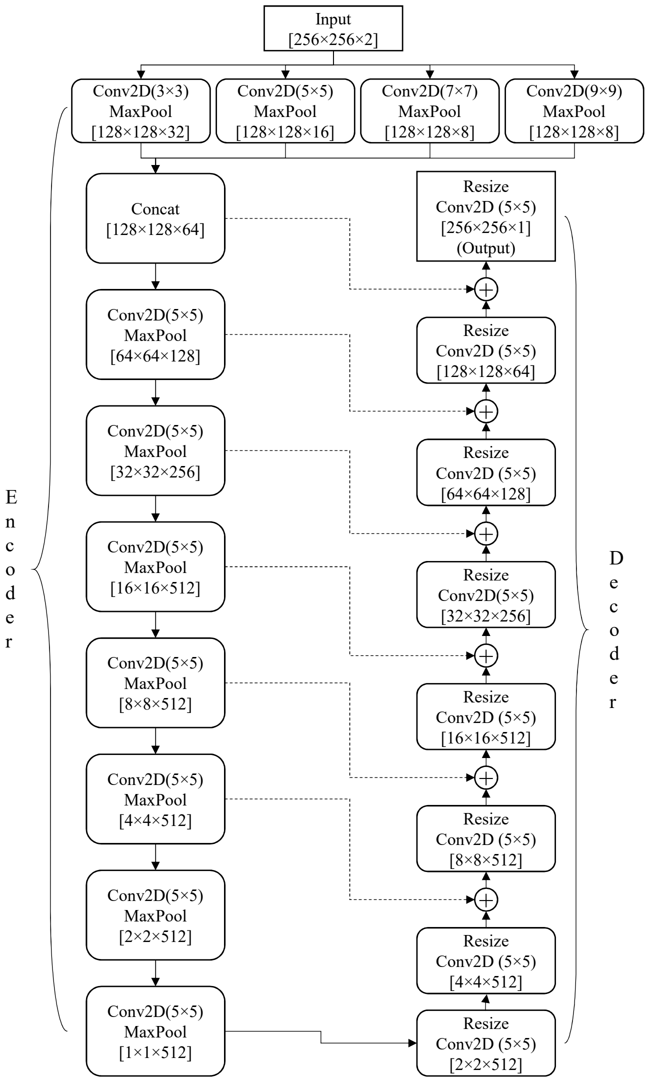


Figure 4: Structure of the used neural network

As tiles are highly overlapped with each other, we cannot just randomly divide the dataset into training, validation and test set. Otherwise, for any tile from the test set, it is highly possible that there is another tile that is highly overlapped with it in another, say training set, thus defying the test. To overcome this problem, the dataset is divided by region. As shown in Fig. 5, the test set is composed of the layout tiles that come from $0 \leq y \leq 36\mu m$, the validation set is composed of those from $36\mu m \leq y \leq 72\mu m$, and the training set includes all the remaining tiles. Furthermore, tiles that cross the dividing horizontal lines $y=36\mu m$ and $y=72\mu m$ are dropped from the dataset to avoid the overlap problem between data sets. Overall, 18% of the tiles are used as validation set, another 18% are used as test set, and the rest are training set.

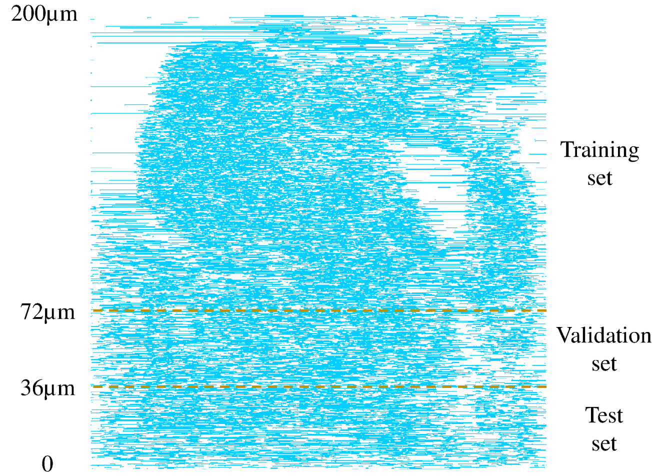


Figure 5: The split of the dataset from layout (M3 shown here)

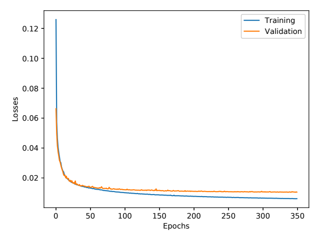


Figure 6: Training curves

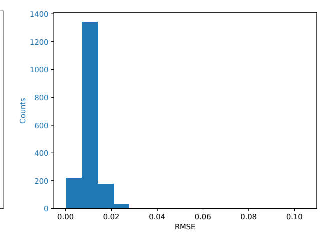


Figure 7: Distribution of RMSE over the test set

In training, the root-mean-square error (RMSE) is used as the loss function:

$$RMSE = \sqrt{\frac{\sum_{x=1}^{W}\sum_{y=1}^{H}[u(x,y) - u_0(x,y)]^2}{W \times H}} \tag{6}$$

where u and u_0 are output electric potential and golden result simulated with FEM respectively, W and H dimension are sizes of the tile. Adam optimizer with the learning rate of 0.00001 is used. The training runs for 350 epochs. Progress of training and validation losses are shown in Fig. 6. There is no sign of over-fitting till the end of training. This training process takes in total 12 hours.

The final model after 350 epochs of training are tested with the test set. The RMSE of all test samples is illustrated as histogram in Fig. 7. The average RMSE across the whole test set is 0.01, As the range of potential is 0V to 1V in this problem, so the RMSE value is equivalent to error in percentage, which means the average error of the model is 1%. This is very close to the accuracy achieved in training and validation. The maximum RMSE is 10.4%. Only 5 out of all 1760 test samples have errors larger than 5%. It is worth noting that these 5 samples turn out all to be extreme cases. The sample of largest RMSE is shown in Fig. 8. It can be seen that this is an extreme case where wires only exist in corners, and it is the other empty areas that contribute to the error. However, these empty areas are actually not studied, as all electrostatic problems focus

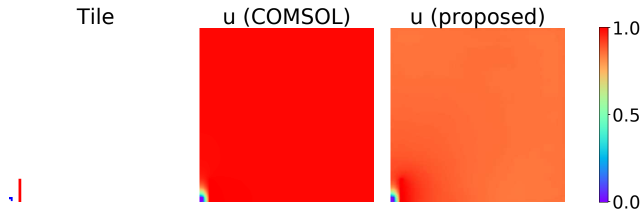


Figure 8: The test sample with the largest RMSE. The tile is shown using the same color convention as in Fig. 3a

on wires and their nearby dielectrics. Furthermore, these test samples of extreme cases would not appear in practical applications if layout partition is used, because the related wires would already have been solved in neighboring tiles that have these wires located away from the tile boundaries and thus can achieve more accurate results.

5.2 Results of electric potential analysis

To further demonstrate the potentials calculated using the trained model, 4 samples, 2 from M3 and 2 from M4, have been randomly picked from the test set. The calculated distributions of potential are plotted and compared against COMSOL simulation results, in Fig. 9. The tiles are also shown in the figure. The RMSE of these samples are 0.011, 0.013, 0.008, and 0.007 respectively. It can be seen that the electric potential calculated by the proposed method matches the golden results very well for the given four example samples.

5.3 Results of electric field analysis

With the solution of potential, electric field can be easily derived by numerically calculating gradients depicted by equation (4). In practice, it is calculated through the gradient function from the numpy library. The calculated results are plotted and compared with COMSOL results, shown in Fig. 10. The same 4 samples from test set are used for comparison. The largest RMSE of the eight plots are 0.01276MV/cm, with the range being -0.0818 to 0.0825MV/cm, so the error is equivalent to 7.8%. Although the overall distribution is matched well as shown in the figure, it is obvious the results derived using the proposed method appear noisier. Furthermore, it can be seen that the values at corners of wires show some discrepancies. The reason of this is mainly that the gradient at such positions is in fact singular in problems of Poisson's equation. Thus, its value is highly related to the mesh used. In spite of the mentioned problems, the overall error, as mentioned, is within 8%.

5.4 Results of TDDB aging analysis

The derived solutions of electric field are used to analyze TDDB aging using equation (5).

Two sample wires are chosen from each sample tile used in the previous comparison, one set to VDD and one set to GND, and the integral results on them are again compared to those from COMSOL. Table 1 lists these results. Note that they are direct integral results so they are expected only to be proportional to the final capacitance values or lifetimes. The differences are calculated with

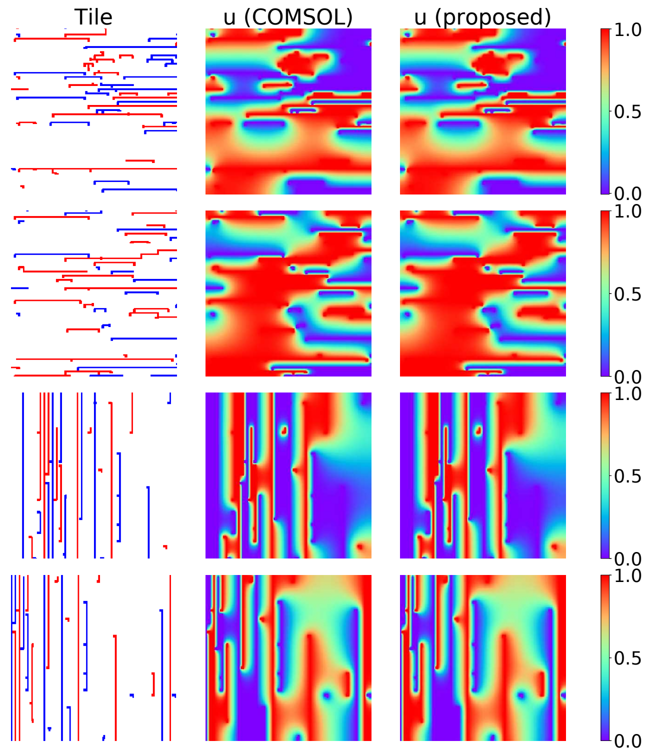


Figure 9: Results of electric potential of sample tiles (left) from COMSOL (middle) and the proposed method (right)

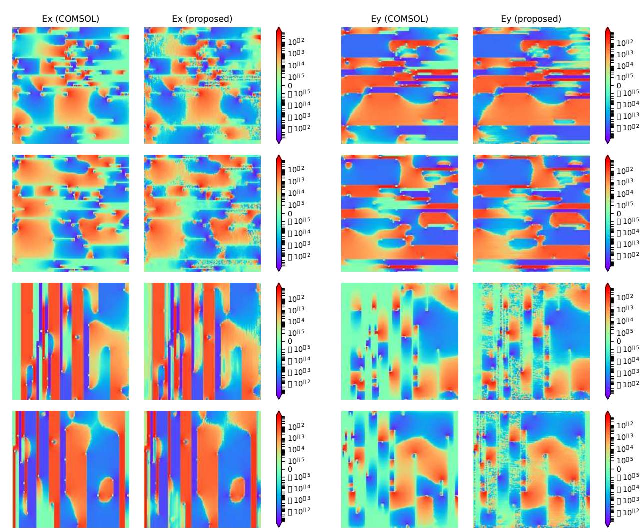


Figure 10: Results of electric field of sample tiles from COMSOL and the proposed method. The unit of electric field is MV/cm**.**

Table 1: Integral results in capacitance and aging analysis of sample wires, calculated using the proposed method and COMSOL

	COMSOL	Proposed	Diff
Line1	3.486e-5	3.492e-5	0.19%
Line2	1.093e-4	1.162e-4	6.28%
Line3	5.864e-6	5.804e-6	-1.03%
Line4	1.581e-5	1.543e-5	-2.39%
Line5	1.058e-5	1.080e-5	2.11%
Line6	1.019e-5	1.008e-5	-1.08%
Line7	1.221e-4	1.140e-4	-6.71%
Line8	3.750e-5	3.725e-5	-0.66%

COMSOL results treated as golden reference. The differences between results from the proposed method and from COMSOL are are all within 6.71%, and the average of their absolute values are 2.57%, which is equivalent to 97.43% accuracy.

It is worth noting that COMSOL uses a finer mesh in the simulation, which can help improve the accuracy of numerical integrals, which is similar to the cause of discrepancy in electric fields themselves. Despite all the simplifications and comparatively larger errors in electric field analysis results, the proposed method is still able to achieve around 97% accuracy for the application of TDDB aging analysis.

5.5 Simulation efficiency study

As the model is trained on synthesized layout tiles, the trained model can be used to find the electric potential distribution for any layout with the discussed method, as long as the layout is synthesized with the same technology.

The runtime of the proposed method is tested with the batch size set to 100, on average it takes 34ms to complete one inference, which means 34ms per simulation. In comparison, it takes COMSOL 2 seconds to finish one simulation, which means the proposed method achieves 58x speedup. Both times include data pre-processing and moving to or from GPU, but do not include the library set-up time.

Another major drawback of COMSOL is that the architecture of this software is so well designed that the potential of parallelism in multiple simulations is limited. Normally, one machine can only run one COMSOL simulation at a time as it is already multi-threaded. This parallelism is crucial for the applications discussed as the layout is partitioned into a lot of tiles so running simulations in parallel would be of great help. The proposed method, on the other hand, can easily handle this level of parallelism by encoding simulation problems into batches. This can further increase the advantage of the proposed method over COMSOL as all previous tests are done with batch size set to 1. Although it would take about 2 seconds to load the Tensorflow libraries and the saved neural network, this time can be neglected or amortized if there is a large number of problems to be solved, again because of the large number of tiles generated from layout partition.

Following this idea, the batch size is set to 100 for our model and the time to calculate electrostatics and TDDB lifetime for the complete M3 shown in Fig. 5 is compared. It takes 175 seconds with the proposed method. However, it takes COMSOL 6.7 hours. This means 138x speed up. Although COMSOL supports larger tiles, it takes longer to compute each tile and the total time would not reduce significantly.

6 CONCLUSIONS

This work proposes a method to use machine learning, specifically a CNN based neural network to solve electrostatics problems. The proposed method first encodes the electrostatic problem to be solved into an image. Taking it as input, the neural network then solves the problem through inference and outputs an image of electric potential distribution. Training and testing are done on a dataset from a synthesized CPU chip. Once trained, the model is applicable to any synthesized layout of the same technology. Compared to the conventional FEM based solver, the proposed method achieves 138x speedup, while keeping 99% of the accuracy on potential analysis, and 97% for TDDB aging analysis.

REFERENCES

[1] Tai-Yu Chou and Zoltan J Cendes. 1994. Capacitance calculation of IC packages using the finite element method and planes of symmetry. *IEEE transactions on computer-aided design of integrated circuits and systems* 13, 9 (1994), 1159–1166.
[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.
[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[4] X. Huang, V. Sukharev, Z. Qi, T. Kim, and S. X.-D. Tan. 2016. Physics-based full-chip TDDB assessment for BEOL Interconnects. In *Proc. Design Automation Conf. (DAC)*. IEEE, 1–6.
[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (May 2015), 436–444.
[6] Jingwei Lu, Pengwen Chen, Chin-Chih Chang, Lu Sha, Dennis Jen-Hsin Huang, Chin-Chi Teng, and Chung-Kuan Cheng. 2015. EPlace: Electrostatics-Based Placement Using Fast Fourier Transform and NesterovâĂŹs Method. *ACM Trans. Des. Autom. Electron. Syst.* 20, 2, Article 17 (March 2015), 34 pages. DOI: http://dx.doi.org/10.1145/2699873
[7] J.W. McPherson and H.C. Mogul. 1998. Underlying Physics of the Thermochemical E Model in Describing Low-Field Time-Dependent Dielectric Breakdown in SiO_2 Thin Films. *Journal of Applied Physics* 84, 3 (1998), 1513–1523.
[8] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Checkerboard Artifacts. *Distill* (2016). DOI: http://dx.doi.org/10.23915/distill.00003
[9] Shaoyi Peng, Ertugrul Demircan, Mehul D Shroff, and Sheldon X.-D. Tan. 2019. Full-chip wire-oriented back-end-of-line TDDB hotspot detection and lifetime analysis. *Integration* (2019).
[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.
[11] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
[12] Wei Tang, Tao Shan, Xunwang Dang, Maokun Li, Fan Yang, Shenheng Xu, and Ji Wu. 2017. Study on a Poisson's equation solver based on deep learning technique. In *2017 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*. IEEE, 1–3.
[13] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. 2017. Accelerating eulerian fluid simulation with convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 3424–3433.
[14] Wenjian Yu and Xiren Wang. 2014. *Advanced field-solver techniques for RC extraction of integrated circuits*. Springer.
[15] Zhongyang Zhang, Ling Zhang, Ze Sun, Nicholas Erickson, Ryan From, and Jun Fan. 2019. Solving Poisson's Equation using Deep Learning in Particle Simulation of PN Junction. In *2019 Joint International Symposium on Electromagnetic Compatibility, Sapporo and Asia-Pacific International Symposium on Electromagnetic Compatibility (EMC Sapporo/APEMC)*. IEEE, 305–308.