Fast Electrostatic Analysis For VLSI Aging based on Generative Learning

Subed Lamichhane*, Shayoi Peng*, Wentian Jin*, Sheldon X.-D. Tan*
*Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521
slami002@ucr.edu, speng004@ucr.edu, wjin018@ucr.edu, stan@ece.ucr.edu

2021 ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD) | 978-1-6654-3166-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/MLCAD52597.2021.9531320

***Abstract*—Electrostatic analysis, which computes electrical potential and electrical field, is important for VLSI reliability and high speed circuit design. Deep learning provides new opportunities and challenges to speedup the analysis process by learning physical laws and feature representations. In this work, we propose an image generative learning framework for electrostatic analysis for VLSI dielectric aging estimation. This work leverages the observation that the synthesized multi layer interconnect VLSI layout can be viewed as layered 2D images and the analysis can be viewed as the image generation. The efficient image-to-image translation property of generative learning is therefore used to obtain the potential distribution on the respective interconnect layers. Compared with the recent CNN-based electrostatic analysis method, the new method can lead to 1.54x speedup for inference due to reduced neural network structures and parameters. We demonstrate the proposed method for time-dependent dielectric breakdown analysis and show the significant speedup compared to the traditional numerical method.**

***Index Terms*—TDDB Analysis, Generative Adversarial Networks**

I. Introduction

Electrostatics analysis deals with computation and analysis of electric potentials, electric fields and electric forces subjected to some given voltage and current boundary conditions. Electrostatic analysis is a crucial part of VLSI modeling and reliability analysis. It is well known that with technology scaling the size of interconnects, feature length and VLSI chip as a whole is decreasing in order to save area and power consumption. With scaling, dielectrics in the VLSI back-end are also getting thinner. With long-term use, the performance of digital circuits gets deteriorated with varying levels of dielectric degradation [1]. Strong electric fields can cause failure of these degraded dielectrics, this effect is known as time-dependent dielectric breakdown (TDDB) [2]. TDDB typically is determined by electric field between the dielectrics. In addition to TDDB based reliability analysis, electrostatic analysis has been used effectively for simulations related to parasitic extraction [3], [4], and also for modeling of global placement.

Conventional approaches to solve electrostatic problems involve numerical methods such as finite element or finite difference methods. These approaches use spatial democratization of the governing equations. Complex geometrical mesh and computation complexity of such numerical method limits the size of the problem and speed of the computation.

Recently deep learning-based data-driven analysis method has been proposed for fast electrostatic analysis based on CNN-based structure [5]. This method shows that we can treat the synthesized VLSI layout with multiple interconnect layers as layered images and the electrostatic analysis can be viewed as image processing.

Recently, generative learning framework based on Generative Adversarial Networks (GAN) [6] has gained much popularity for image related applications due to its effectiveness of learning form latent features without extensive labeled training data. GANs have been used in a variety of applications such as image synthesis, semantic image editing, style transfer, image super-resolution and classifications. In VLSI physical designs GAN-based methods have been used for applications including layout lithography analysis [7], sub-resolution assist feature generation [8], analog layout well generation [9], routing congestion estimation [10] and for thermal map estimation for multicore-chips [11].

In this work, we are proposing a new GAN-based method for fast electrostatic analysis which involves learning electrical potential distribution from VLSI layout and using it to calculate electric field and hence to model TDDB based aging using the calculated electric field. The new contributions include:

- Leverage the very effective image transformation approach of GANs to obtain potential distribution of tiles obtained from layout partitioning [5] of synthesized VLSI layout. As a result, the new method uses less complex neural network model for generator compared to convolution neural network (CNN) model used in [5], thus has less training and inference time.
- Results of this method show that this proposed method offers 213x speedup over the conventional FEM (Finite Element Method) based method COMSOL with about 99 % of accuracy on electric potential and 97 % accuracy on TDDB aging analysis. The new method also leads to 1.54x more speed up over the CNN-based method [5] with similar accuracy.

In this paper, Section II presents some related works. Section III provides some underlying physics behind electrostatics and TDDB analysis for VLSI interconnects. Section IV contains elaborated problem formulation and neural network models used in this work. Section V will provide detailed insights on dataset used for training, validation and testing and also provides experimental results for performance analysis. Finally, this paper ends with concluding Section V.

II. Related Work

Traditionally, FEM or FDM (finite difference method) approaches are used for electrostatic analysis [4]. Recently deep learning based approaches have been investigated for electrostatic analysis. A CNN based method was proposed to solve Poisson's equation, specifically in the case of electrostatics [12]. However the problem size is quite small as the grid size is set to 64×64 with no extension. Another work proposed by Zhang et al. [13] used charge distribution and boundary condition as the inputs of the neural network. A regular grid is used in all these works. Tompson et al. [14] studies a more complicated time-dependant fluid flow problem.

Recently a CNN-based method for electrostatic analysis was proposed in [5] in which an CNN-based autoencoders was used. This work achieves about 138x speedup over COMSOL with an average 0.01V RMSE error in electrical potential estimation, 8% average error in electric field and 97.43 % accuracy in TDDB analysis.

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.

III. Physics behind Electrostatics and TDDB analysis

In this section, we first briefly review the electrostatic physic models and related TDDB analysis.

For static charges, the electric potential is given by the Maxwell's first equation. This equation is also known as Gauss's law and is given by relation:

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

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

This Maxwell's equation can be solved using Dirichlet and Neumann boundary conditions which are given by:

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

Here, 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 provided. While u is the unknown potential to be solved, $f(x)$ is given voltage sources and $g(x)$ is electric field or current sources at the boundaries.

In this work since we are dealing with interconnects in the VLSI back-end-of-line, there are no static charges involved. Therefore $\rho = 0$ for this condition and (1) becomes Laplace equation given as:

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

After solving for u based on (3) and given boundary conditions, Electric field $\overrightarrow{\mathrm{E}}$ can be calculated as negative gradient of u, i.e.

$$\overrightarrow{\mathrm{E}} = -\nabla u \tag{4}$$

Once we get $\overrightarrow{\mathrm{E}}$, it can be used to model TDDB since TDDB is a reliability effect caused by high electric field over time. The TDDB induced time-to-failure(TTF) of an interconnect is proportional to $\sqrt{\overrightarrow{\mathrm{E}}}$, the relation is given in [15] which is :

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

Here, γ is a coefficient that is fitted from experimental data. L is the parameter of the wire.

Here, equation 5 assumes that the wires are modeled in 2D for simplicity. In actual implementation, interconnects are 3D, but they can be analyzed in 2D by ignoring the height and analyzing layer by layer [5]. For this work only VDD and GND are used as boundary conditions and VDD equal to 1V is used.

IV. Proposed Method

A. Problem formulation and data preparation

As we discussed above, FDM or FEM based methods like COMSOL have speed limitations. Hence, the main goal of this work is to speed up the process to solve the electric potential distribution and electric fields in dielectrics in VLSI back-end-of line using neural network models. We also discussed previously that to solve electric potential and hence electric field we need boundary conditions. This boundary condition information in VLSI circuits can be obtained from VLSI design layouts. However, there are some challenges for us to solve before we can use these layouts for neural network-based models.

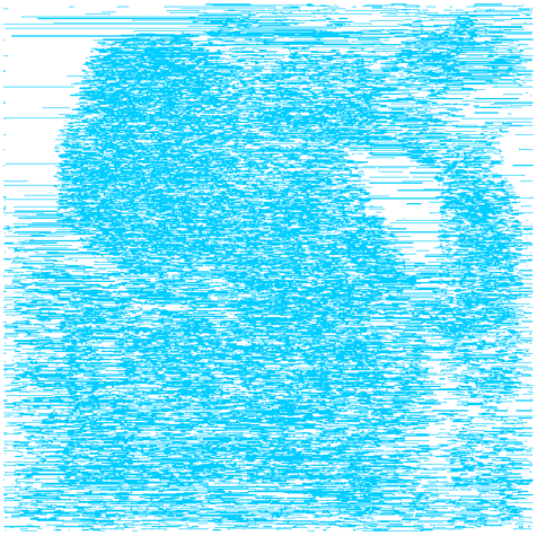
One of the first challenges in front of us is the size of the VLSI design layouts. Typically, the size of such layouts is in the order of micrometers to millimeters. Since the interconnect wires are of nanometer sizes, the resolution of image of such layouts is very large in pixel size. On the other hand, typical input image size in modern machine learning problems has the size of around 224 pixels by 224 pixels [16]. This problem can be solved using the layout partition [15] method. In layout partitioning, one interconnect layer is partitioned into smaller tiles and analyzed separately. The reason why this method works is that the interconnect wires of the same layer are generally routed in the same direction, i.e. either vertical or horizontal routing. So, the electric field on one wire has a very low effect on another wire that is several channels away [5]. One drawback of using the layout partition method is that the solution of electric potential or field at the boundaries of tiles would be inaccurate since we are not considering the effect of adjacent wires. This problem can be fixed by introducing a tile that places the boundaries of two adjacent tiles in the center. Fig. 1b shows the process of layout partitioning along with the introduction of a new tile between two adjacent tiles.

With the layout partition method we can get the tiles suitable for neural network models. However, the tiles are still not in the compatible format. The VLSI chip layout is stored in a gdsii format where the geometric and electric properties of the elements are stored in a binary format. Since we are using an image-based neural network model, these tiles in gdsii format are converted to images. To solve the electric potential of the interconnect layer we need two information: geometric information of the interconnects and the voltage boundary conditions required to solve the potential distribution i.e. VDD or GND [5], [15]. To convert the tiles in gdsii format to images, the two boundary conditions (VDD and GND) are encoded in two different channels of the image. For the geometry of interconnects, the value of channel is set to 1 if that pixel belongs to the corresponding boundary condition(VDD or GND), and set to 0 if not. Fig. 2 shows the encoding process where VDD is encoded in red channel and GND is encoded in blue channel.

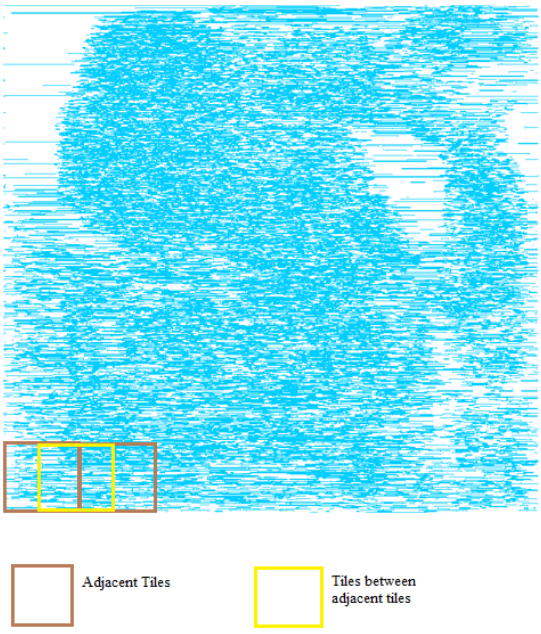
In this work, an example layout synthesized with 32nm technology is used. The physical size of the layout is $200\mu m$x$200\mu m$. This layout is partitioned into tiles of size $12\mu m$x$12\mu m$. Interconnect layers with horizontal routing direction(M3:Metal 3) and vertical routing direction(M4) are used to extract the tiles. About 12000 tiles are extracted in total. For ground truth, COMSOL is used to solve electric potential and electric field distribution of the tiles. As the tiles are highly overlapped to each other, training, testing and validation sets can not be selected randomly. It is because a tile in training set can have some overlapped portion of tile in test set. Therefore three regions are divided for training, testing and validation. For testing, tiles from the region $0 \leq y \geq 36\mu m$ are used. Similarly, region $36 \leq y \geq 72\mu m$ is used for validation and region $72 \leq y \geq 200\mu m$ is used for training. Here, y is the vertical length of the tile. To avoid overlap, tiles that cross the dividing horizontal lines $y = 36\mu m$ and $y = 72\mu m$ are removed. Overall, 64% of data is used for training, 18% are used for validation and 18% for testing.

B. Generative Adversarial Networks(GANs)

GANs [17] are composed of two separate deep neural networks, generator G and discriminator D. The job of generator is to generate real-like outputs and the job of discriminator is to distinguish between output generated by generator and the real ones. In a conventional GAN, generator takes a random noise vector(z) as input and generates G(z) as output. A typical discriminator is a binary classifier. Real data and generated data are alternatively given to the discriminator. The discriminator then gives the score of the given input being real or fake. Gradients from the discriminator are passed



(a) Original VLSI layout



(b) Partitioning layout into small tiles

Fig. 1: VLSI layout and layout partitioning

to both discriminator and generator network through back propagation. These gradients are then used to train both generator and discriminator model.

While conventional generator generates output randomly based on training data. There is no control on what type(label) of output to be generated. The conditional GAN (CGAN) [18] learns to generate the controlled output. In this case, a condition label is also given to generator as input along with the vector z. If x is the condition vector and z is the random noise, the output of the generator is now G(x,z). This condition is passed to generator as well. Therefore in CGANs both generator and discriminator are conditioned on vector x. Recently, CGANs have shown effective results in image translation problems. In image translation problems, input images are used as condition to control the generator output [19]. For our problem, the encoded tile images are used as conditional input to the CGAN as shown in Fig. 3.

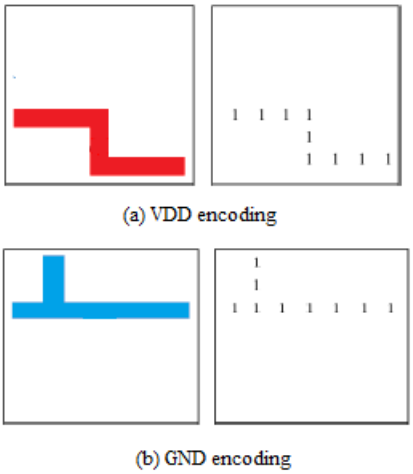


Fig. 2: Encoding layout tile into image.

Conventional GANs including CGAN are not so easy to train. Two major problems with these networks are that the training process is not stable and the training might not converge due to vanishing gradient problem. To solve these issues, Wasserstein GAN (WGAN) [20] was introduced. In WGAN, the conventional JS-Divergence is replaced by the Wasserstein distance as the score of realness or fakeness of the input data. In this work, we have implemented WGAN for better convergence and stability during the training.

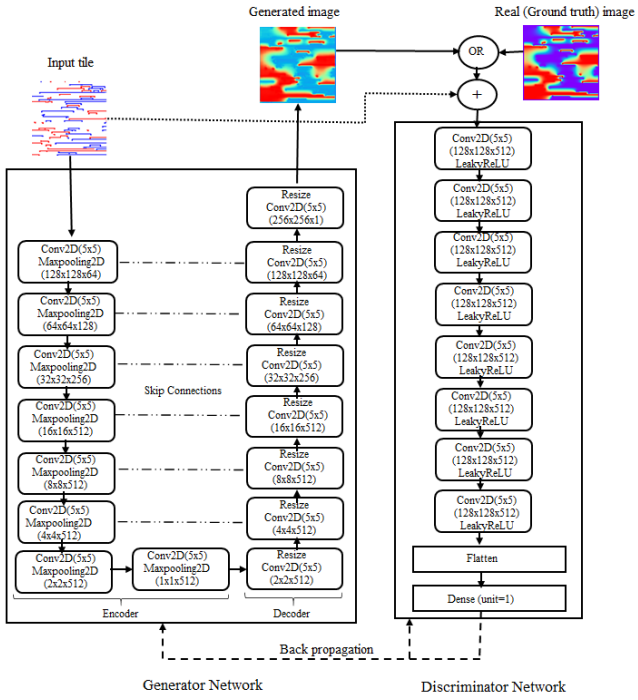


Fig. 3: GAN model for electric potential distribution estimation

As shown in Fig. 3, our GAN model is based on encoder-decoder based generator which is a widely used model for

image translation. The 256x256x2 size tile image is given as input to the encoder input. In encoder network, the image is passed through various convolutional layers. The latent features of the tile image are extracted in the encoder network as it gets downsampled through the convolutional layers. The extracted features are then upsampled through convolutional layers to generate potential distribution image of size 256x256x1. To pass the gradient information between layers of encoder and decoder, skip connections are used. For discriminator network, series of convolutional layers followed by flatten layer and fully connected dense layers are used. The generated potential distribution image and ground truth images from COMSOL are fed alternatively to the discriminator. The size of these images is 256x256x1. Along with the real and generated images, input tiles are also given to discriminator as condition as mentioned above.

As mentioned above we used WGAN for this work. For WGAN, the classification output of discriminator is not probabilistic. Therefore, no activation function is used after the last fully connected dense layer. To further improve the performance of WGAN, gradient penalty is applied in the cost function. This approach is called WGAN-GP [20]. The WGAN-GP based objective function for the discriminator is:

$$\max_{D}\{\mathbb{E}_{x,y}[D(y,x)] - \mathbb{E}_{x}[D(G(x),x)] - \lambda_{gp}\mathbb{E}_{\hat{x}}[(||\nabla_{\hat{x}}D(\hat{x},x)||_2 - 1)^2]\} \quad (6)$$

Here, $D(y,x)$ is the discriminator output for real ground truth image and $D((G(x),x)$ is discriminator output for generated images. The goal of the discriminator is to give higher score to real potential distribution images than the generated images. That is to minimize $D((G(x),x)$ and maximize $D(y,x)$. $\hat{x}$ is the interpolation between the real image and generated image. λ_{gp} is the weight of gradient penalty to maintain the 1-Lipschitz continuity of the discriminator.

For generator, the training goal is to deceive the discriminator and get higher scores for generated images. Since the generator has nothing to do with the real images, $D(y,x)$ is not included in the objective function. L2-norm to the loss of generator is added to improve the objective function [21]. The strength of the L2-norm distance penalty on the loss of generator is controlled by λ_{L2}. The objective function for generator is:

$$\min_{D}\{\mathbb{E}_{x}[-D(G(x),x)] + \lambda_{L2}\mathbb{E}_{x,y}[||y - G(x)||_2]\} \quad (7)$$

V. Experimental Results and Discussion

The CGAN model explained in section IV with WGAN-GP is implemented using python 3.8.8 and tensorflow 2.4.1. For optimization, RMSProp optimizer with the learning rate of 0.0001 and batch size of 10 is used. Further, λ_{gp}=10 and λ_{L2}=100 is used for gradient penalty implementation. The model with these parameters is then trained on a Linux server 2 Intel Xeon E5-2698v3 2.3GHz processors and a Nvidia Titan X GPU. The training converges after 50 epochs as shown in Fig. 4 with training time of 10.25 hours.

A. Electric potential estimation results

As the metric of accuracy, RMSE (Root mean square error) between real image (from COMSOL) and generated image is used. Fig. 4 shows the average RMSE error curve for training and validation set. For both training and validation data, the average RMSE converges around 0.013V after 50 epochs. This trained model is saved and used to estimate electric potential of test set consisting of 1781 input tiles. The average RMSE of 0.015V is observed for the test set. The histogram showing RMSE distribution of test set is shown by Fig. 5. For testing batch size of 100 images is used to leverage the parallelism offered by neural network based approach [5]. Here, since we have 1781 test data and batch size of 100 is used for testing, around 18 batches are shown. Only 1 batch i.e. 5% of the test data have RMSE greater than 0.02V. These 5% of extreme cases are caused by empty areas in the tile where wires exist only in corners [5]. Theses empty areas are not much problematic when we use layout partitioning since these corner wires would already have been solved in neighbouring tiles. As the range of potential is 0V to 1V, average 0.015V RMSE in test set is equivalent to 1.5% of error. This result is almost same as in CNN based model in [5]. Fig. 6 shows example of potential distribution estimation using this approach. For this GAN approach, the average time for one inference is 22ms. This is equivalent to 22ms per simulation. For COMSOL, average time per simulation is around 2 seconds. The CNN-based model has average inference time of 34 ms. All these times include data pre-processing and moving to or from GPU and do not include library setup time. Our GAN based approach offers 90x speedup in simulation time compared to COMSOL. For CNN based model the speedup is only 58x.

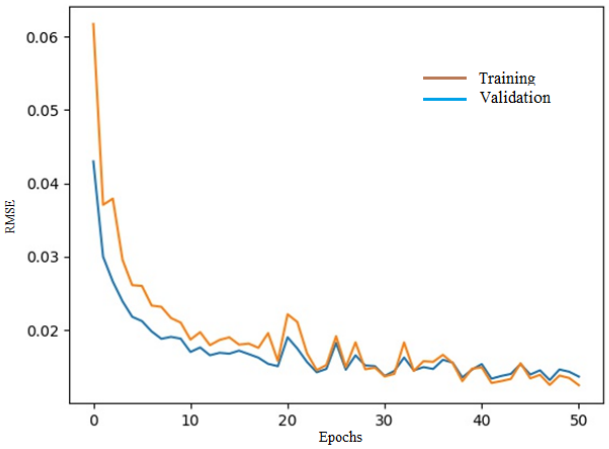


Fig. 4: The training errors vs epochs

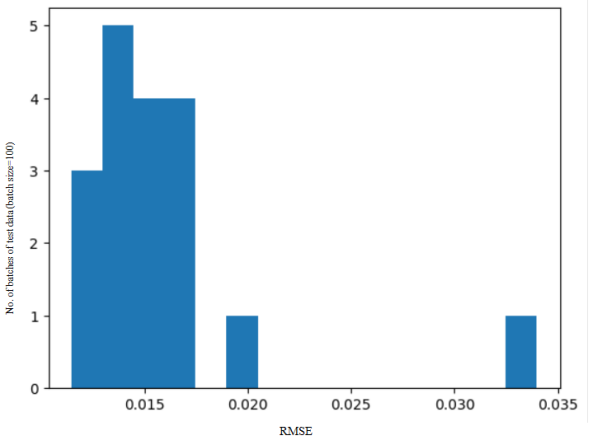
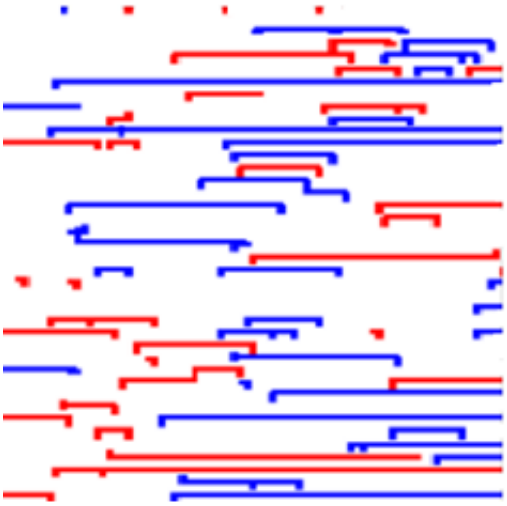
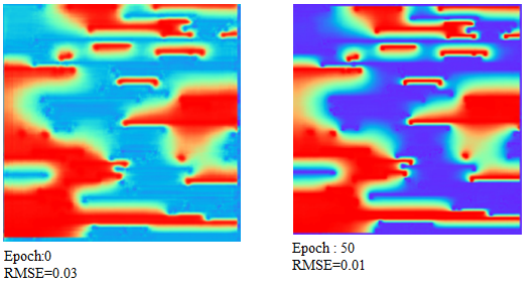


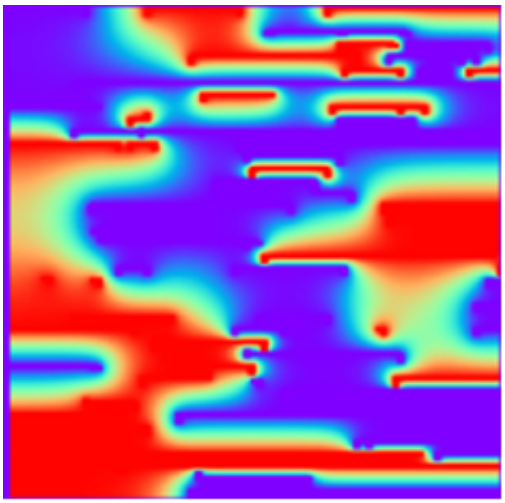
Fig. 5: RMSE distribution over test data



(a) The input tile with a layered layout image



(b) Generated potential results using GAN based method at different training epochs



(c) Ground truth image from COMSOL

Fig. 6: Results for electric potential estimation using GAN

B. Electric field analysis results

Once the solution of potential is estimated, we can use equation (4) to calculate the electric field numerically. The gradients are calculated using gradient function of Numpy library. For 4 randomly selected samples, the largest RMSE is observed to be 0.00085 MV/cm. The range of electric field for these samples is calculated to be between -0.08333 to 0.08333 MV/cm. Hence, the average error can be approximated to be around 0.55%. Some discrepancies can be seen in the electric field values at the corner of the wires due to singular gradients [5]. In spite of the discrepancies, average error of around 1% is observed. This average error is 8% for CNN based model in [5]. Fig. 7 shows the example of electric field distribution estimation using GAN-based approach.

C. TDDB aging analysis results

As discussed in section III, equation (5) can be used to analyze the TDDB aging with the electric field solution obtained above. For TDDB analysis, two sample wires(one set of VDD and one set of GND) are selected from each sample tiles used in electric field estimation. The line integral value is calculated for these lines using equation (5). This integral value is used to estimate the TTF due to TDDB aging [15]. Table I presents the integral values calculated using COMSOL and the same values calculated using the proposed method. The difference between two methods is also presented in the table.

TABLE I: Integral values for TDDB analysis using COMSOL and GAN

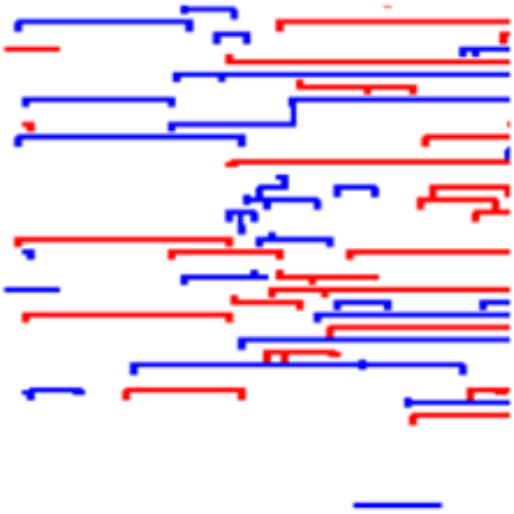
	COMSOL	***GAN***	***Error (%)***
Line1	3.486e-05	3.478e-05	0.2
Line2	1.093e-04	1.157e-04	5.8
Line3	5.864e-06	6.008e-06	2.45
Line4	1.581e-05	1.697e-05	7.3
Line5	1.058e-05	1.103e-05	4.2
Line6	1.019e-05	1.059e-05	3.9
Line7	1.221e-04	1.140e-04	-6.63
Line8	3.750e-05	3.768e-05	0.4

We can see from Table I that the differences between the results from the proposed GAN-based method closely matches the results from COMSOL. The differences are all within 7.3%. The average of the absolute values of the differences are around 3.6%, which results in an accuracy of around 96.3% compared to golden results from COMSOL. For CNN-based method in [5] this accuracy is around 97.4%.

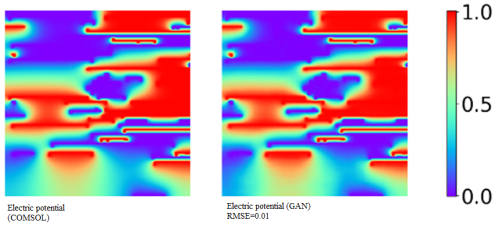
Using the proposed GAN-based method, the time to calculate electrostatics and TDDB aging for the complete layer i.e M3 as shown in Fig. 1a is around 113 seconds using a batch size of 100. COMSOL takes around 6.7 hours for the same task. Therefore the proposed method offers around 213x speedup compared to COMSOL. The speedup of CNN based method in [5] is 138x. This infers that using GAN we can get around 1.54x more speedup compared to CNN method around the same accuracy. This is because the complexity of the generator network used in GAN is less complex than CNN method with less number of layers and parameters. In [5], the convolutional network uses 19 Conv2D layers and one concatenation layer whereas, the generator network in this method consists of 15 Conv2D layers only.

VI. Conclusions

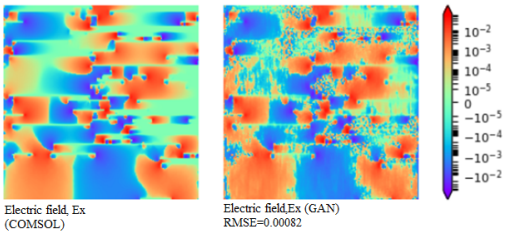
The main limitation of electrostatic analysis using conventional FEM and FDM based method like COMSOL is speed. This work proposes a GAN-based machine learning model to overcome that limitation. The proposed GAN-based model takes input tiles partitioned from the interconnect layer of VLSI circuit and gives the potential distribution as output. This model is trained on synthesized layout tiles. So, once



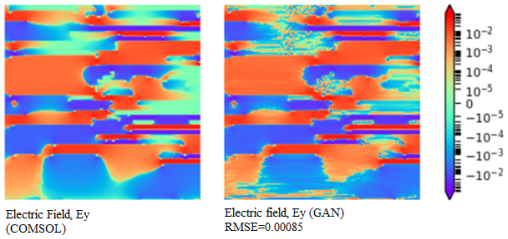
(a) The input tile from the layered layout image



(b) Generated electric potential



(c) Electric field comparison, X component



(d) Electric field comparison, Y component

Fig. 7: Results for electric field estimation using GAN.

trained this model can be used to solve electric potential for any layout as long as the layout is synthesized with the same technology. The proposed method offers 213x speedup compared to COMSOL with an expense of only around 3% in accuracy. Compared to the similar CNN-based method, the proposed GAN-based approach gives 1.54x more speedup with around similar accuracy.

REFERENCES

[1] J. Fang and S. S. Sapatneka, "Scalable methods for analyzing the circuit failure probability due to gate oxide breakdown," in *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, 2011, pp. 1960–1973.

[2] F. Lu, J. Shao, X. Liu, and X. Wang, "Validation test method of tddb physics-of-failure models," in *Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing*, 2012.

[3] C. Tai-Yu and J. C. Zoltan, "Capacitance calculation of ic packages using the finite element method and planes of symmetry," in *IEEE transactions on computer-aided design of integrated circuits and systems 13, 9 (1994)*, 2015, pp. 1159–1166.

[4] W. Yu and X. Wang, *Advanced field-solver techniques for RC extraction of integrated circuits*. Springer, 2014.

[5] S. Peng, W. Jin, L. Chen, and S. X.-D. Tan, "Data-driven fast electrostatics and tddb aging analysis," in *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*, ser. MLCAD '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 71–76. [Online]. Available: https://doi.org/10.1145/3380446.3430620

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016, http://www.deeplearningbook.org.

[7] W. Ye, M. B. Alawieh, Y. Lin, and D. Z. Pan, "Lithogan:end-to-end lithography modeling with generative adversarial networks," in *56th Annual Design Automation Conference 2019, DAC '19*. ACM, 2019, pp. 107:1–107:6.

[8] M. B. Alawieh, Y. Lin, Z. Zhang, M. Li, Q. Huang, and D. Z. Pan, "Gan-sraf: Sub-resolution assist feature generation using conditional generative adversarial networks," in *56th Annual Design Automation Conference 2019, DAC '19*. ACM, 2019, pp. 149:1–149:6.

[9] B. Xu, Y. Lin, X. Tang, S. Li, L. Shen, N. Sun, and D. Z. Pan, "Wellgan: Generative-adversarial-network-guided well generation for analog/mixed-signal circuit layout," in *56th Annual Design Automation Conference 2019, DAC '19*. ACM, 2019, pp. 66:1–66:6.

[10] C. Yu and Z. Zhang, "Painting on placement: Forecasting routing congestion using conditional generative adversarial nets," in *56th Annual Design Automation Conference 2019, DAC '19*. ACM, 2019, pp. 219:1–219:6.

[11] W. Jin, S. Sadiqbatcha, J. Zhang, and S. X.-D. Tan, "Full-chip thermal map estimation for commercial multi-core cpus with generative adversarial learning," in *iccad*. New York, NY, USA: ACM, Nov. 2020, pp. 1–9.

[12] W. Tang, T. Shan, X. Dang, M. Li, F. Yang, S. Xu, and J. Wu, "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, 2020, pp. 1–3.

[13] Z. Zhang, L. Zhang, Z. Sun, N. Erickson, R. From, and F. Jun, "Solving poisson's equation using deep learning in particle simula-tion of pn junction," in *2019 Joint International Symposium on Electromagnetic Compatibility, Sapporo and Asia-Pacific International Symposium on Electromag- netic Compatibility (EMC Sapporo/APEMC)*. IEEE, 2019, pp. 305–308.

[14] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin, "Accelerating eulerian fluid simulation with convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3424–3433.

[15] S. Peng, E. Demircan, M. D. Shroff, and S. X.-D. Tan, "Full-chip wire-oriented back-end-of-line tddb hotspot detection and lifetime analysis," *Integration*, 2019.

[16] K. Simonyan and A. Zisserman, "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*, 2015.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[18] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv e-prints*, p. arXiv:1411.1784, Nov. 2014.

[19] W. Jin, S. Sadiqbatcha, Z. Sun, H. Zhou, and S. X.-D. Tan, "Em-gan: Data-driven fast stress analysis for multi-segment interconnects," in *iccd*, Oct. 2020, pp. 296–303.

[20] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv e-prints*, p. arXiv:1701.07875, Dec. 2017.

[21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.