



A single neural network global I-V and C-V parameter extractor for BSIM-CMG compact model



Jen-Hao Chen^{a,*}, Fredo Chavez^b, Chien-Ting Tung^a, Sourabh Khandelwal^b, Chenming Hu^a

^a Berkeley Device Modeling Center, University of California, Berkeley, Berkeley 94720, CA, USA

^b School of Engineering, Macquarie University, Macquarie Park, Sydney 2109, NSW, Australia

ARTICLE INFO

Handling Editor: S. Cristoloveanu

Keywords:

Berkeley Short-channel IGFET Model – Common Multi-Gate (BSIM-CMG)
Deep learning
Fin field effect transistor (FinFET)
Compact model
Parameter extraction

ABSTRACT

A global I-V and C-V BSIM-CMG parameter extraction methodology based on deep learning is proposed. 100 k training datasets were generated through Monte Carlo simulation varying 28 IV and CV model parameters in the industry-standard BSIM-CMG FinFET model. For each of the 100 k Monte Carlo-selected BSIM-CMG parameter dataset, the I_D - V_G and C_{GG} - V_G characteristics of seven Monte Carlo-selected gate lengths ranging from 14 nm to 110 nm were generated as the input to train the parameter extraction neural network. The neural network outputs for training are the 28 model parameters' values. The neural network's capability to extract BSIM-CMG model parameters that accurately fit TCAD-generated I_D - V_G and C_{GG} - V_G data over a range of gate lengths was demonstrated. This marks the first time a deep learning compact model parameter extraction flow, employing a single neural network for both I-V and C-V parameters and for a range of gate length, is presented.

1. Introduction

Transistor compact model is an essential computer model for integrated circuit design. With the help of a compact model, circuit performance can be simulated by a SPICE circuit simulator. However, the matching accuracy of a compact model to the transistor's IV and CV characteristics depends on the quality of the model and accurate model parameters. Before a compact model can be used to simulate circuit performance, a crucial procedure called "parameter extraction" needs to be performed to ensure the accuracy of circuit simulation.

Parameter extraction is a tedious process because the number of model parameters for advanced device compact models is large and complex. These parameters make it possible to model many non-ideal effects that are inevitable in state-of-the-art devices [1–3]. Parameter extraction relies on expertise in semiconductor device physics and understanding of the compact model. In order to circumvent this time-consuming process that depends on human skills, some works [4–6] have proposed machine learning-based compact models, aiming to replace the conventional compact model with neural networks. However, all of these works still can't preserve all the functionality in traditional compact model, and it is believed that the entire replacement of conventional compact model still has a long way to go.

As one another direction, Yang et al. [7] proposed a graph-based

compact model that combines the physical equations in conventional compact models with multi-layer perceptron replacing the empirical equations in conventional compact models. This hybrid compact model eases the human's burden in compact model development.

Unlike machine learning-based compact model [4–7], other methods don't aim to replace conventional compact models. They are fully compatible with existing models and preserve all the physics-based formulations in the models. [8–13] One previous work [8] used Levenberg–Marquardt algorithm to reduce human's burden in parameter extraction. However, this method requires iterations to gradually optimize a good enough parameters' values and is too computationally expensive. Moreover, when addressing a global parameter extraction for a large range of gate length, this traditional method might face a huge computational burden due to more data points and more parameters. Another two previous works used decision tree classifier [9] or genetic algorithm [10] to accelerate parameter extraction. Yet, [9] and [10] still need to iterate and optimize the parameters' value step by step. Due to the aforementioned reason, other previous works [11–13] presented fully deep learning methods that do not require iteration. However, Kao et al. [11] only addressed a local parameter extraction case with a fixed gate length and failed to extract parameters that can accurately fit CV curves with high drain bias. Ashai et al. [12] developed a deep learning method capable of accepting various formats of input data, but they still

* Corresponding author.

E-mail address: jenhaochen@berkeley.edu (J.-H. Chen).

encounter the same problem as [11]. Chavez et al. [13] only addressed parameter extraction from IV curves. In general, all of these methods are not able to simultaneously extract I-V and C-V parameters for a large range of gate-length without iteration. We address this need in this work. Table 1 is provided to show the differences between this work and other previous deep learning-based parameter extraction works [11–13].

In this paper, we develop a deep learning-based methodology to extract model parameters from multiple devices with various gate length and demonstrate our work on an industry standard BSIM-CMG FinFET model [14]. To the best of our knowledge, this work is the first demonstration in which a IV and a CV parameter extraction for a large range of gate length, i.e. global IV and CV parameters extraction, is achieved simultaneously by a single neural network. The paper is organized as follows: In section 2, we proposed our methodology, including training dataset generation and neural network training. In section 3, a comparison between different neural network structures and an evaluation of the deep learning parameter extractor accuracy are discussed. Lastly, a conclusion of this work is given in section 4.

2. Proposed method

The process flow of this proposed deep learning-based parameter extraction methodology is depicted in Fig. 1. Firstly, the training datasets will be generated by Monte Carlo technique in which the selected 28 parameters listed in Table 2 are varied within chosen ranges. These 28 parameters were carefully chosen to consider the required length scalability of the IV and CV model. Geometry-related parameters, such as EOT (effective oxide thickness), HFIN (fin height), TFIN (fin thickness), are assigned values according to the TCAD device structure calibrated to Intel 10 nm technology node [15]. The I_D - V_G , I_D - V_D and C_{GG} - V_G data of these TCAD devices are our target data from which the neural network will extract parameters.

In the generation of training data from BSIM-CMG by Monte Carlo simulation, parameter values are randomly chosen with uniform probability distribution within a predefined range. The gate lengths of seven devices are randomly chosen from 7 pre-selected ranges from 14 nm to 110 nm, as depicted in Fig. 2, and within each range, randomly chosen with uniform probability. A group of C_{GG} - V_G and I_D - V_G data for these seven devices is generated in each time of Monte Carlo simulations. For training, the I_D - V_G and C_{GG} - V_G data, and the selected gate lengths serve as the input of a network, and BSIM-CMG parameter values are the corresponding outputs. A total of 100 k training datasets are generated by the Monte Carlo simulations.

The artificial neural network structure is depicted in Fig. 2. There are two different input arrays. The lower one is composed of CV and IV data, which includes C_{GG} - V_G and I_D - V_G in linear ($V_D = 0.05$ V) and saturation ($V_D = 0.7$ V) region. In addition, I_D - V_G in log scale is also used as input data to improve the overall accuracy in IV fitting [16]. The gate voltage in these datasets varies from 0 to 0.8 V with a step size of 0.05 V. $V_G = 0.8$ V is beyond the rated V_{DD} for this technology but is used here intentionally. It has been shown that extending V_G beyond the supply voltage can improve the accuracy at high voltage region. [16] Thus, the

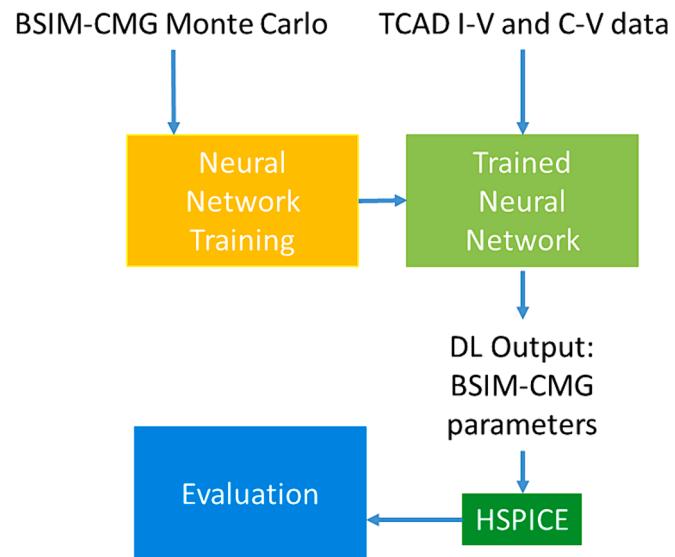


Fig. 1. The workflow of deep learning-based parameter extraction. The neural network is trained with training data generated by BSIM-CMG Monte Carlo simulation. The trained neural network is validated by TCAD IV and CV data.

Table 2

The selected BSIM-CMG parameters and their corresponding physical meaning.

IV Parameter	CV Parameter
1.PHIG (work function)	1.TOX (physical oxide thickness)
2.U0 (low field mobility)	2.QMTCENCV (quantum mechanics)
3.RDSW (series resistance)	3.PQM (quantum mechanics)
4.CIT (interface trap)	4.CFS (fringing capacitance)
5.UA (surface roughness scattering)	5.CGSL (overlap capacitance)
6.EU (surface roughness scattering)	6.CKAPPAS (overlap capacitance)
7.ETAMOB (effective field)	7.VFBSDCV (flatband voltage at S/D)
8.ETA0 (DIBL effect)	8.UOCV (CV low field mobility)
9.CDSCD (drain bias sensitivity)	9.VSATCV (CV saturation velocity)
10.VSAT1 (saturation velocity)	10.LQMTCENCV (quantum mechanics)
11.KSAT1V (saturation Vds)	11.LCFS (fringing capacitance)
12.DVT1SS (subthreshold swing)	12.LVSATCV (CV saturation velocity)
13.DVTO (short channel effect)	
14.UP (mobility)	
15.DSUB (DIBL effect)	
16.AVSAT (saturation velocity)	

lower input array is a $7 \times 6 \times 17$ dimensional input. “7” represents seven different gate lengths. “6” represents I_D - V_G , $\log(I_D)$ - V_G and C_{GG} - V_G in both linear and saturation regions. “17” is the number of data points ranging from 0 V to 0.8 V with a step size of 0.05 V. This lower input array will be passed into a neural network with M hidden layers and N neurons in each hidden layer. A comparison between different M and N based on the performance on validation dataset will be discussed in section 3. On the other hand, the upper input array is composed of seven different gate lengths which is a 7 dimensional input. This gate length

Table 1

The difference between previous works and this work. ✓ means the work is able to address the task listed in the first row. ✗ means the work is not able to address the task listed in the first row.

	IV parameter extraction for both low and high V_D	CV parameter extraction for both low and high V_D	Global IV parameter extraction (for a range of L_G)	Both global IV and CV parameter extraction (for a range of L_G)
[11]	✓	✗	✗	✗
[12]	✓	✗	✗	✗
[13]	✓	✗	✓	✗
This work	✓	✓	✓	✓

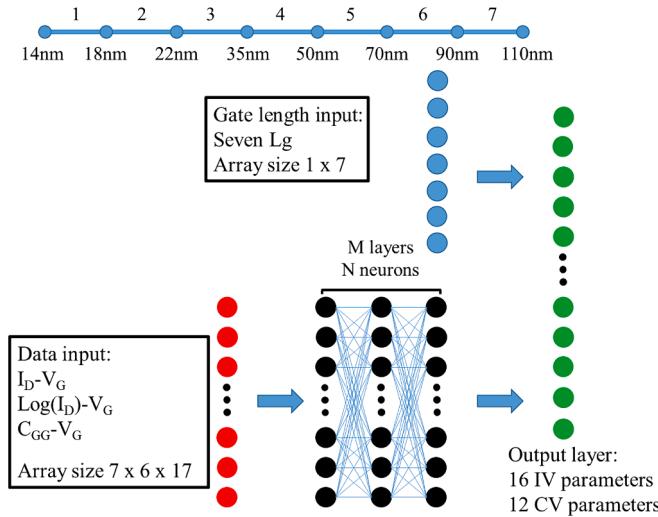


Fig. 2. The Structure of the deep learning neural network. Upper input array is gate length input with seven gate lengths ranging from 14 nm to 110 nm. Lower input array is composed of I_D - V_G and C_{GG} - V_G data points. Output array is composed of 28 IV and CV parameters.

input will be concatenated with the last hidden layer, and this concatenated layer will eventually be fully connected to the output layer.

The inputs and outputs are normalized between 0 and 1 before being passed into the neural network. A non-linear activation function, such as ReLU, Sigmoid, Tanh, Swish and ELU [17–19], is assigned to each neurons in each hidden layer. The output layer with the activation function “Sigmoid” is a 28- dimensional array representing 28 different selected BSIM-CMG parameters.

During the training process, this artificial neural network is constructed with machine learning toolkit Keras [20] and Tensorflow in Python. “Adam” was chosen as an optimizer that adjusts the model weight to minimize the loss function. The loss function used during training process is the mean square error of predicted BSIM-CMG parameters. 10 % of training dataset is used as the validation dataset. The initial learning rate is $1e-5$, and it will reduce to a lower learning rate if the error in the validation dataset does not improve during the training process. This is a widely used approach to overcome the plateau problem [21] in which a scheduled learning rate can help reach a better model accuracy. In addition, Early stopping is implemented to prevent overfitting if the loss function is found to no longer decrease.

After the training process, the TCAD IV and CV data is fed to the trained neural network to generate BSIM-CMG parameters. Finally, this deep-learning extracted BSIM-CMG parameter set is used to recreate IV and CV characteristics by running the BSIM-CMG model in HSPICE, a commercial SPICE simulator. The effectiveness of this neural network is evaluated by comparing the TCAD simulated IV and CV data with BSIM-CMG simulations when using deep learning-extracted parameters.

3. Result and discussion

3.1. Comparison between different neural network structure

In a deep learning model, the number of neurons in hidden layer, the number of hidden layers, and the activation functions are three key parameters determining the performance of a neural network. In order to find a suitable neural network structure, several networks with different layer numbers, neuron numbers, and activation functions were tested. The average percentage error in I_D - V_G and C_{GG} - V_G data points in the validation data was recorded for different neural network structures. Fig. 3 (a) and (b) show the IV and CV percentage error in validation datasets with 3 hidden layers ($M = 3$), different numbers of neurons

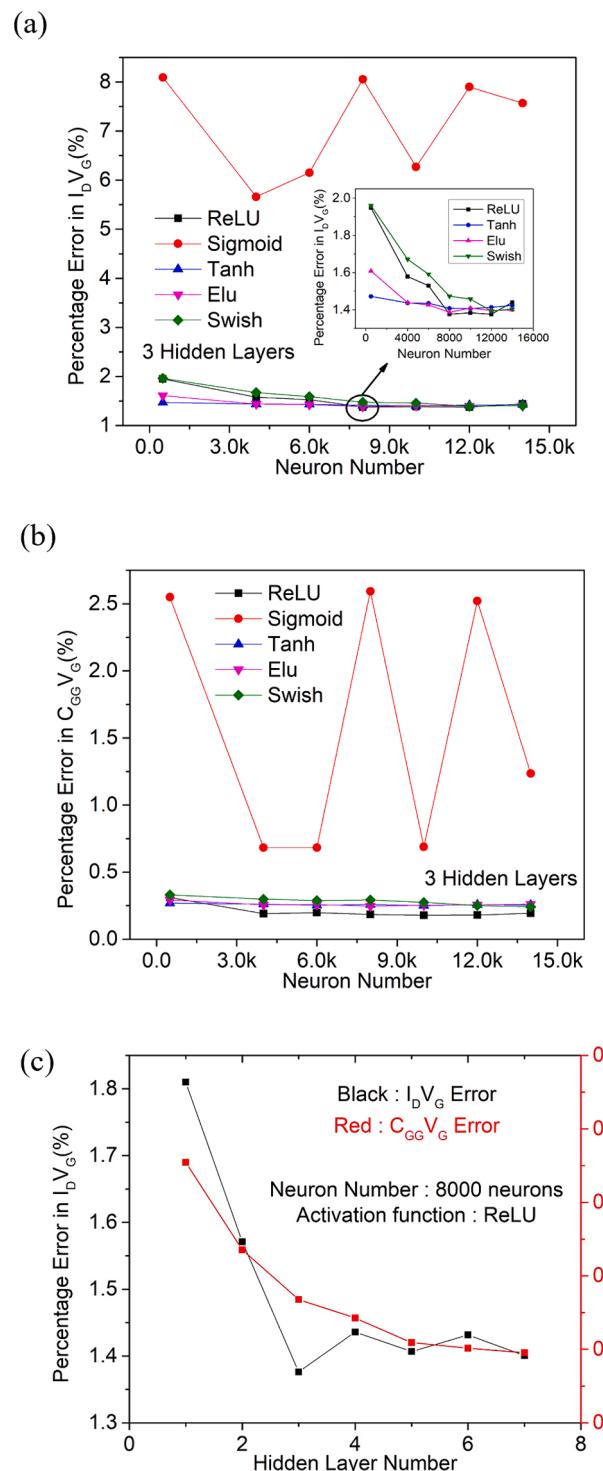


Fig. 3. The average percentage error in validation dataset with different neural network structures and activation functions. Error in validation (a) I_D - V_G (b) C_{GG} - V_G with different neuron number and activation functions. (c) Error in validation I_D - V_G and C_{GG} - V_G with different layer number.

ranging from 500 to 14000, and five different activation functions: ReLU, sigmoid, tanh, ELU, and Swish. The neural networks trained with ReLU have a lower error in validation datasets overall. With ReLU, it is observed that both IV and CV percentage errors in validation datasets are gradually reduced with increasing neuron numbers. If the neuron number is more than 8000 neurons, the average error in the IV can drop below 1.4 % and that in CV can be below 0.2 %. Also, the accuracy in

validation datasets doesn't improve much with neurons number more than 8000.

Fig. 3(c) shows the IV and CV percentage error in validation datasets with 8000 neurons ($N = 8000$), ReLU activation function in each neuron, and different number of hidden layers ranging from 1 to 7. It can be observed that both of the error in IV and CV decreases with increasing number of hidden layers. IV error can drop below 1.4 %, and CV error can be below 0.2 % after three hidden layers. Based on the above analysis, it can be concluded that a neural network with ReLU activation function, more than 3 hidden layer and more than 8000 neurons in each hidden layer can learn the relationship between CV and IV data and their corresponding parameters value well from BSIM-CMG generated dataset.

In the rest of this paper, an artificial neural networks with the ReLU activation function, 3 hidden layers and 8000 neurons in each hidden layer will be adopted to show the performance of this deep learning-based parameter extraction methodology.

3.2. The testing result of deep learning-based parameter extractor

After the training process, TCAD I_D - V_G and C_{GG} - V_G data with seven different gate lengths (16, 20, 30, 40, 60, 80, 100 nm) selected from each of the seven different ranges shown in **Fig. 2**, calibrated to Intel 10 nm technology [15], were used to test the performance of this deep learning-based parameter extractor. In **Fig. 4**, the distribution of the training data generated by BSIM-CMG and this TCAD-generated testing data is plotted. As shown in **Fig. 4**, the seven TCAD-generated testing data curves (corresponding to seven different gate lengths: 16, 20, 30, 40, 60, 80, 100 nm) are fall within the range covered by the training dataset. The goal of this neural network is to learn the relationship between data points, gate lengths and parameters' values and reliably extract a parameter set from I_D - V_G and C_{GG} - V_G data points within this range where the training dataset is generated. Note that this TCAD data is a group of I_D - V_G and C_{GG} - V_G that the neural network has not seen before. This TCAD data and their corresponding gate lengths were put into the input layer of this neural network, and the predicted BSIM-CMG

parameters value and seven gate lengths were then put into BSIM-CMG so that “deep learning-predicted IV and CV curves” can be generated by a circuit simulator. **Fig. 5** shows a good I_D - V_G fitting for devices with gate lengths of 16 nm, 20 nm, 30 nm, 40 nm, 60 nm, 80 nm, and 100 nm. From the fitting result in linear and log scale, it can be observed that this parameter extractor achieves good fitting both in the subthreshold and the strong inversion regions. The root mean square percentage error (%RMSE) in linear ($V_D = 0.05$ V) and saturation ($V_D = 0.7$ V) region are 9.4 % and 8.8 %, respectively. The %RMSE above threshold are 5.6 % and 4.4 % in linear and saturation region, respectively.

In addition to I_D - V_G , the parameter set extracted by neural network was also directly put into BSIM-CMG to generate I_D - V_D . **Fig. 5** shows a good fitting result in I_D - V_D without providing this neural network with any I_D - V_D information. Moreover, **Fig. 6** shows a good C_{GG} - V_G fitting for the same seven devices. The root mean square error (%RMSE) in linear ($V_D = 0.05$ V) and saturation ($V_D = 0.7$ V) region are 2.4 % and 2.5 %, respectively. Therefore, this deep learning-based parameter extraction methodology is demonstrated to have a comparable level of accuracy to conventional manual fitting.

In addition to this single choice of these seven different gate lengths shown in **Figs. 5–7**, a few more different combinations of gate lengths are also used to test the model's accuracy in extracting parameters. **Table 3** shows the fitting accuracy when this trained neural network extractor is tested with different combinations of seven gate lengths selected from each of the seven different ranges in **Fig. 2** one by one. The average %RMSE in I_D - V_G in the linear region ($V_D = 50$ mV) is 7.41 % (Above V_{th} error is 4.36 %), and that in I_D - V_G in the saturation region ($V_D = 0.7$ mV) is 7.38 % (Above V_{th} error is 4.21 %). In terms of CV, the %RMSE in C_{GG} - V_G in the linear region ($V_D = 50$ mV) is 2.79 %, and that in C_{GG} - V_G in the saturation region ($V_D = 0.7$ V) is 2.68 %. All of them show a good accuracy comparable to that in conventional manual parameter extraction.

In addition to extracting parameters from these target TCAD data, this deep learning-based parameter extractor can also be used to extract parameters value from similar I_D - V_G and C_{GG} - V_G data with variation because this neural network has already been trained with a huge

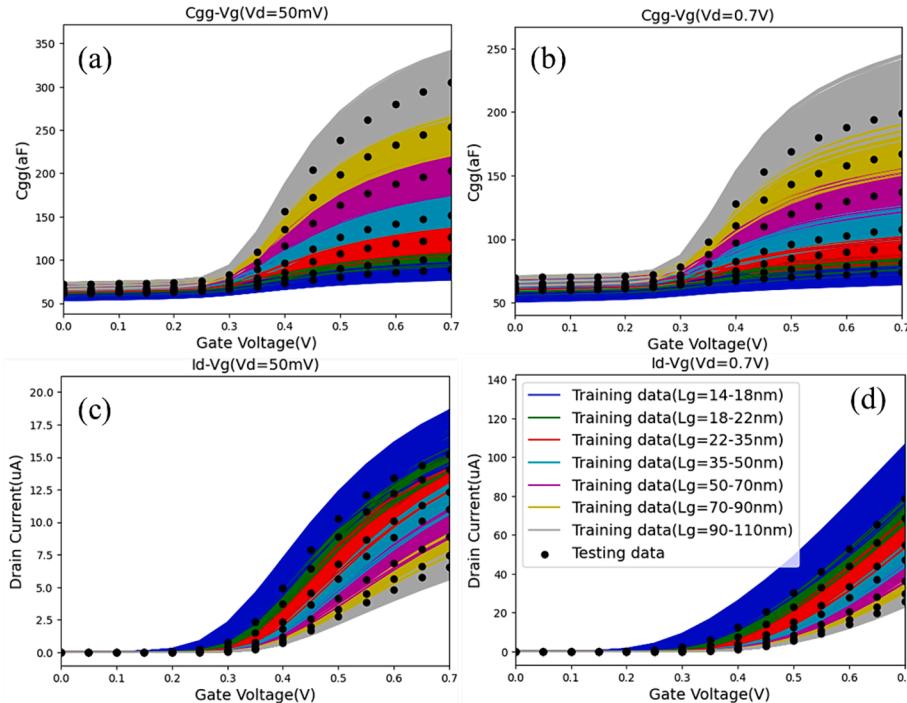


Fig. 4. The distribution of the training dataset and the testing data ($L_G = 16, 20, 30, 40, 60, 80, 100$ nm) of (a) C_{GG} - V_G ($V_D = 50$ mV) (b) C_{GG} - V_G ($V_D = 0.7$ V) (c) I_D - V_G ($V_D = 50$ mV) (d) I_D - V_G ($V_D = 0.7$ V). Training dataset is generated by BSIM-CMG. 10 % of training dataset is used as validation dataset. Testing data are generated by TCAD simulation.

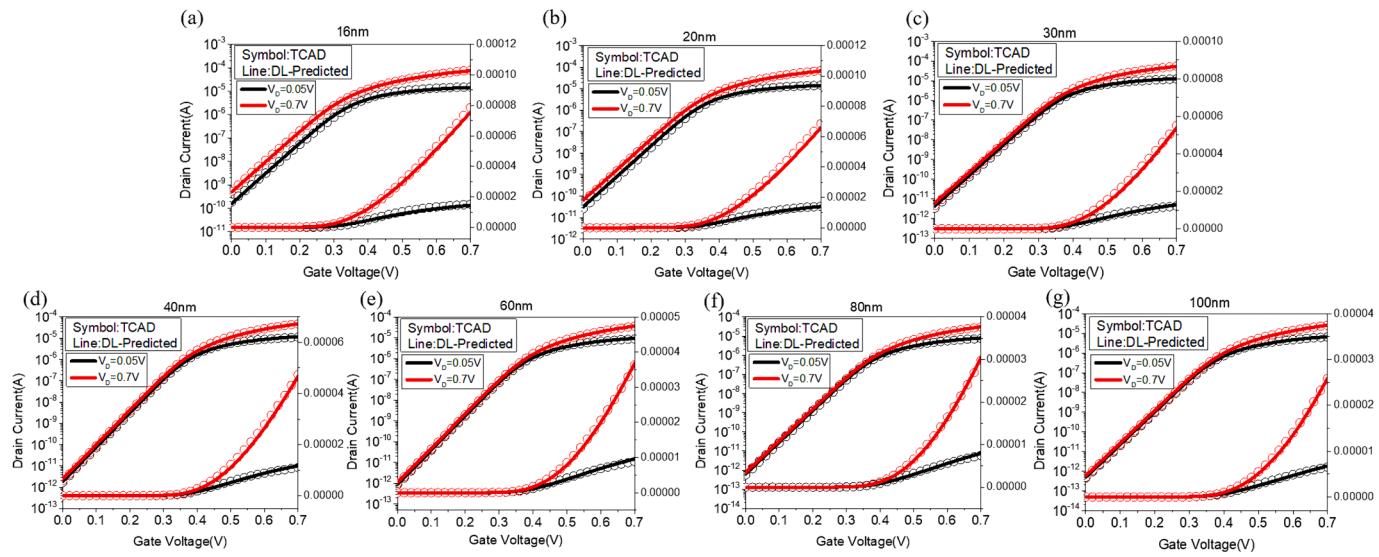


Fig. 5. Fitting result in TCAD I_D - V_G with seven various gate length. (a)16 nm (b)20 nm (c)30 nm (d)40 nm (e)60 nm (f)80 nm and (g)100 nm.

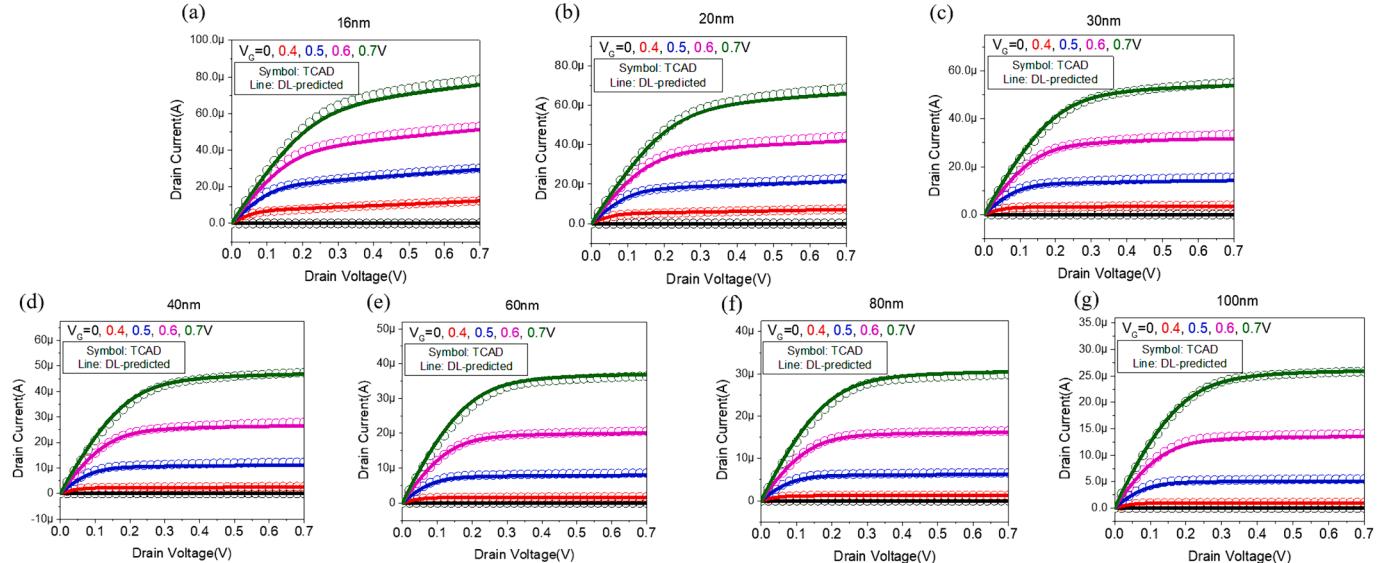


Fig. 6. Fitting result in TCAD I_D - V_D with seven various gate length. (a)16 nm (b)20 nm (c)30 nm (d)40 nm (e)60 nm (f)80 nm and (g)100 nm.

number of training data within a preassigned range. This is of great importance in real application because it can't be expected that IV and CV data with the same technology node are always the same. Variations are expected due to non-idealities in the fabrication process. As long as the I_D - V_G and C_{GG} - V_G with variation are within this preassigned range, this deep learning-based parameter extractor can achieve a good enough fitting comparable to manual fitting due to the fact that the neural network learns the I_D - V_G and C_{GG} - V_G and their corresponding parameters very well. Therefore, this deep learning based parameter extraction methodology will be a useful tool which can facilitate compact model development for a fabrication process.

4. Conclusion

In this work, a deep learning-based compact model parameter extraction framework, which include I_D - V_G and C_{GG} - V_G , is proposed. It is a global parameter extraction methodology because it extracts a set of model parameters that can be applied to transistors within a range of gate lengths (14 nm to 110 nm). A parameter-extraction neural network

is created using this methodology and the accuracy of its extracted model is tested by several sets of TCAD generated data covering a range of gate lengths. The average root-mean-square errors are 7.395 % and 2.735 % for IV and CV, respectively. (The error of IV above threshold is 4.285 %.) This level of accuracy is comparable to parameter extraction by manual fitting, and this neural network can generate parameters within a second. This work lays a foundation for global IV and CV compact model parameters extraction utilizing deep learning.

CRediT authorship contribution statement

Jen-Hao Chen: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Fredo Chavez:** Writing – review & editing, Software, Methodology, Formal analysis. **Chien-Ting Tung:** Writing – review & editing, Formal analysis. **Sourabh Khan-delwal:** Writing – review & editing, Supervision, Project administration, Formal analysis. **Chenming Hu:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Formal

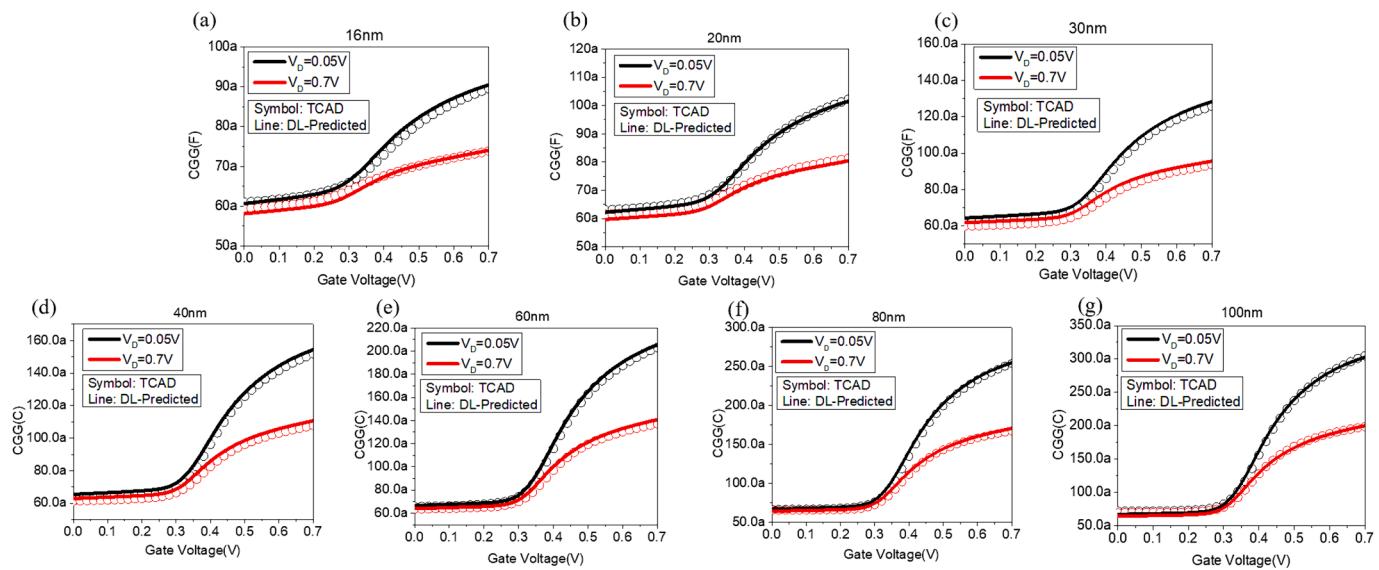


Fig. 7. Fitting result in TCAD C_{GG} - V_G with seven various gate length. (a)16 nm (b)20 nm (c)30 nm (d)40 nm (e)60 nm (f)80 nm and (g)100 nm.

Table 3

The %RMSE in fitting with five different combinations of seven gate lengths with deep learning-extracted parameters.

Lg set (nm)	C_{GG-V_G} ($V_D =$ 50 mV) Error	C_{GG-V_G} ($V_D =$ 0.7 V) Error	I_D-V_G ($V_D =$ 50 mV) Error	I_D-V_G ($V_D =$ 0.7 V) Error	I_D-V_G ($V_D =$ 50 mV) Error above V_{TH}	I_D-V_G ($V_D =$ 0.7 V) Error above V_{TH}
16, 20, 30, 40, 60, 80, 100	2.40 %	2.52 %	8.82 %	9.40 %	5.66 %	4.48 %
17, 20, 28, 42, 63, 85, 102	2.50 %	2.49 %	7.69 %	5.96 %	5.46 %	4.94 %
17, 21, 33, 46, 66, 86, 106	2.76 %	2.51 %	7.21 %	7.11 %	2.76 %	2.35 %
18, 22, 34, 47, 69, 89, 109	3.46 %	3.25 %	8.62 %	8.40 %	4.24 %	3.79 %
17, 21, 31, 45, 63, 83, 103	2.83 %	2.65 %	5.57 %	5.19 %	3.94 %	3.44 %
Average Error	2.79 %	2.68 %	7.41 %	7.38 %	4.36 %	4.21 %

analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgement

This work was supported by the Berkeley Device Modeling Center, University of California at Berkeley, Berkeley, CA, USA.

References

- [1] Gill A, Madhu C, Kaur P. Investigation of short channel effects in Bulk MOSFET and SOI FinFET at 20nm node technology. In: IEEE; 2015. p. 1–4. <https://doi.org/10.1109/INDICON.2015.7443263.P>.
- [2] Hutchby JA, Bourianoff GI, Zhirnov VV, Brewer JE. Extending the road beyond CMOS. IEEE Circuits Device Magazine 2002;18(2):28–43. <https://doi.org/10.1109/101.994856>.
- [3] Chin A, McAlister SP. The power of functional scaling: beyond the power consumption challenge and the scaling roadmap. IEEE Circuits Device Magazine 2005;21(1):27–35. <https://doi.org/10.1109/MCD.2005.1388766>.
- [4] Cha Y-S, Park J, Park C, Chong S, Kim C-H, Lee C-S, et al. A novel methodology for neural compact modeling based on knowledge transfer. Solid State Electron 2022; 198(108450). <https://doi.org/10.1016/j.sse.2022.108450>.
- [5] Tung C-T, Kao M-Y, Hu C. Neural network-based modeling with high accuracy and potential model speed. IEEE Trans Electron Devices 2022;69(11):6476–649. <https://doi.org/10.1109/TED.2022.3208514>.
- [6] Wang J, Kim Y-H, Ryu J, Jeong C, Choi W, Kim D. Artificial neural network-based compact modeling methodology for advanced transistors. IEEE Trans Electron Devices 2021;68(3):1318–25. <https://doi.org/10.1109/TED.2020.3048918>.
- [7] Yang Z, Gaidhane AD, Anderson K, Workman G, Cao Y. Graph-Based Compact Model (GCM) for Efficient Transistor Parameter Extraction: A Machine Learning Approach on 12 nm FinFETs. In: IEEE Transactions on Electron Devices; 2023. <https://doi.org/10.1109/TED.2023.3327973>.
- [8] Le Duc-Hung, Pham Cong-Kha, Nguyen Thi Thien Trang and Bui Trong Tu, “Parameter extraction and optimization using Levenberg-Marquardt algorithm,” 2012 Fourth International Conference on Communications and Electronics (ICCE), Hue, Vietnam, 2012, pp. 434–437, doi: 10.1109/CCE.2012.6315945.
- [9] Alia G, Buzo A, Maier-Flaig H, Pieper K-W, Maurer L, Pelz G. Machine learning-based acceleration of genetic algorithms for parameter extraction of highly dimensional mosfet compact models. In: 2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS). IEEE; 2021. p. 1–4. <https://doi.org/10.1109/ICECS53924.2021.9665517>.
- [10] Li Y, Cho Y-Y. Intelligent bsim4 model parameter extraction for sub-100 nm mosfet era. Jpn J Appl Phys 2004;43(4S):1717. <https://doi.org/10.1143/JJAP.43.1717>.
- [11] Kao M-Y, Chavez F, Khandelwal S, Hu C. Deep learning-based bsim-cmg parameter extraction for 10-nm finfet. IEEE Trans Electron Devices 2022;69(8):4765–8. <https://doi.org/10.1109/TED.2022.3181536>.
- [12] Ashai A, Jadhav A, Behera AK, Roy S, Dasgupta A, Sarkar B. Deep Learning-based fast BSIM-CMG Parameter Extraction for general input dataset. IEEE Trans Electron Devices July 2023;70(7):3437–41. <https://doi.org/10.1109/TED.2023.3278615>.
- [13] Chavez F, Tung C-T, Kao M-Y, Hu C, Chen J-H, Khandelwal S. Deep learning-based I-V global Parameter Extraction for BSIM-CMG. Solid State Electron 2023;209 (108766). <https://doi.org/10.1016/j.sse.2023.108766>.
- [14] BSIM-CMG 111.2.1, 2022, [online] Available: <https://bsim.berkeley.edu/models/bsimcmg/>.
- [15] Auth C, et al. A 10 nm high performance and low-power CMOS technology featuring 3rd generation FinFET transistors self-aligned quad patterning contact

- over active gate and cobalt local interconnects. p. 29.1.1–4. In: 2017 IEEE international electron devices meeting. IEEE; 2017. <https://doi.org/10.1109/IEDM.2017.8268472>.
- [16] Chavez F, Kao M-Y, Hu C, Khandelwal S. Optimization of deep learning-based bsim-cmg iv parameter extraction in seconds. In: IEEE; 2022. p. 124–6. <https://doi.org/10.1109/RFIC54256.2022.9882451>.
- [17] Dubey SR, Singh SK, Chaudhuri BB. Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark. arXiv:2109.14545.
- [18] Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). arXiv:1511.07289.
- [19] Prajit Ramachandran, Barret Zoph, Quoc V. Le. Searching for Activation Functions. arXiv:1710.05941.
- [20] Ketkar N. Introduction to keras. In: Deep learning with python. Springer; 2017. p. 97–111. https://doi.org/10.1007/978-1-4842-2766-4_7.
- [21] Ainsworth M, Shin Y. Plateau Phenomenon in Gradient Descent Training of ReLU networks: Explanation, Quantification and Avoidance arXiv:2007.07213.