Transistor Compact Model Based on Multigradient Neural Network and Its Application in SPICE Circuit Simulations for Gate-All-Around Si Cold Source FETs

Qihang Yang, Guodong Qi, Weizhuo Gan, Zhenhua Wu, *Member, IEEE*, Huaxiang Yin, *Senior Member, IEEE*, Tao Chen, *Member, IEEE*, Guangxi Hu, *Member, IEEE*, Jing Wan, Shaofeng Yu, and Ye Lu

***Abstract*— Transistor compact model (TCM) is the key bridge between process technology and circuit design. Typically, TCM is desired to capture the nonlinear device electronic characteristics and their high-order derivatives. However, for the novel devices in advanced and future technologies, establishing TCM based on analytical equations and extracting model parameters becomes tedious. The model fitting capability for device outputs' high-order derivatives is also limited. These drawbacks hinder fast and accurate device to circuit evaluation cycles. We develop a TCM based on multigradient neural network (MNN) using computational graph in the PyTorch framework. This MNN model is able to simultaneously capture the transistor dc/ac characteristics, such as *I–V*/*Q–V*, their derivatives (*G–V*/*C–V*), and higher order derivatives accurately. Moreover, the model architecture can be widely adapted to various device types. Based on this model scheme, software is developed to enable the automated model generation for standard SPICE simulation. Finally, the model and software are validated for novel gate-all-around (GAA) Si cold source field-effect transistors (CSFET), and 19-stage ring oscillator and two-stage operational amplifier circuit simulations have also been demonstrated. This work reduces the cycle of novel device compact model creation and circuit benchmark simulation from months or weeks to hours. In addition, it enables more precise circuit simulation for analog and RF circuits, and it provides a rapid solution for early stage design technology cooptimization (DTCO).**

***Index Terms*— Circuit simulation, device compact model, field-effect transistor (FET), multigradient neural network (MNN).**

I. Introduction

TRANSISTOR is the key component in modern integrated circuits (ICs), while transistor compact model (TCM) is the bridge between advanced device process technology and electronic chip circuit design. TCM is usually employed to carry out circuit simulation for the circuit benchmark and design sign-off purposes. Therefore, it is desired for the TCM to accurately capture all nonlinear device electronic characteristics and their high-order derivatives, particularly for the analog and RF designs. Conventional TCMs, such as BSIM [1], [2] HiSim [3], [4], and UTSOI [5], [6], are based on analytically physical and/or empirical formulations. These models are widely adopted in industry. However, these compact models still suffer from some limitations:

1) Future device scale approaches to the atomic level, and it requires significant effort and time to build quantum physics-based compact models and prevent rapid circuit evaluations [7], [8].
2) Physical and empirical formulations show relatively weak technology adaptability, and a complicated subcircuit model is usually required for the real application.
3) The optimization and extraction of model parameters are time-consuming and rely on model engineers' experience.
4) The formulas established may not be differentiable [9], and thus, the derivatives of the model output may lose continuity.

This negatively impacts the key analog and RF circuit design simulations related to future 5G and 6G chips. The lookup table (LUT)-based method [10], [11] was proposed as an alternative to physics-based compact model. However, it could cause converge issue in large-scale circuit simulation [12],

Manuscript received May 17, 2021; revised June 9, 2021 and June 23, 2021; accepted June 24, 2021. Date of publication July 7, 2021; date of current version August 23, 2021. This work was supported in part by the Innovation Program of Shanghai Municipal Education Commission under Grant 2021-01-07-00-07-E00077 and in part by the Shanghai Pujiang Program under Grant 20PJ1400900. The review of this article was arranged by Editor A. J. Scholten. *(Qihang Yang, Guodong Qi, and Weizhuo Gan contributed equally to this work.) (Corresponding authors: Ye Lu; Zhenhua Wu.)*

Qihang Yang, Guodong Qi, Tao Chen, Guangxi Hu, Jing Wan, and Ye Lu are with the State Key Laboratory of ASIC and System, School of Information Science and Technology, Fudan University, Shanghai 200433, China (e-mail: lu_ye@fudan.edu.cn).

Weizhuo Gan, Zhenhua Wu, and Huaxiang Yin are with KLMEDIT, Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China (e-mail: wuzhenhua@ime.ac.cn).

Shaofeng Yu is with the School of Microelectronics, Fudan University, Shanghai 200433, China (e-mail: shaofeng_yu@fudan.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TED.2021.3093376.

Digital Object Identifier 10.1109/TED.2021.3093376

and the table size becomes too large to manage when a significant amount of parameters is involved for practical applications [13], [14]. Finally, the LUT approach typically lack of extrapolating capability for extending the model prediction over the known data range.

Artificial neural network (ANN) was proposed to model transistor I–V by Litovski *et al.* [9] in the early 1990s and has been investigated extensively for device modeling since then [14]–[16]. In general, the ANN-based device model adopts iterative mathematical architecture [17], [18] and uses supervised learning with gradient descent to map device characteristics. This model framework is inherently advantageous to capture nonlinear characteristic curves [19]. In addition, the model parameter extraction can be done in an automated fashion [20], [21]. Moreover, the continuity of the high-order derivatives of the model output can be guaranteed [9]. Therefore, the ANN-based device compact model could be a good supplement to the conventional ones and play a substantial role in specific device modeling and circuit simulations.

Although there are numerous works on ANN-based device models [9], [14]–[24], some issues still need to be addressed for future design technology cooptimization (DTCO) needs.

1) Although some early ANN-based TCM works have shown basic fitting for transistor IV and its first derivative gm, they have not specifically focus on the simultaneous fitting capability of I–V/C–V characteristics and their multiple higher order derivatives, e.g., gm, gm', and dC/dV_g. In addition, the circuit simulation impact due to the lack of these high-order derivatives' fitting also has not been carefully studied. For example, a 1% current fitting error could lead to a 100% differential conductance error, and the error rate could be even higher for its second derivatives and so on. This limits the simulation accuracy, particularly for analog and RF circuits.
2) Some other works utilize an adjoint neural network to calculate the first order of the characteristic curve [25]. This approach is effective for the first-order derivative but less efficient due to the additional neural network required, and this brings additional burden for SPICE circuit simulation.
3) While we are preparing this article, we noticed that a very recent work from Samsung exhibited the ANN modeling methodology for advanced silicon transistors [12], our work differs from theirs as we focus on multigradient neuron network fitting capabilities and our model is demonstrated on novel cold source field-effect transistors (CSFETs).

In this article, we propose a TCM scheme based on multigradient neural network (MNN), utilizing the computational graph within the PyTorch framework [26]. This model approach is able to simultaneously capture the transistor dc/ac characteristics such as I–V/Q–V, their high-order derivative terms, e.g., G–V, C–V, dG/dV–V, and dC/dV–V, with one set of neural network. The model creation does not require complete knowledge of detailed underline device physics, and we also show that this model approach could facilitate the prediction of certain electrical characteristics outside the known data range due to the accurate capturing of inherent data curvature. Importantly, the model parameter extraction is done automatically with a customized multigradient-aware backpropagation (MABP) algorithm without manual work.

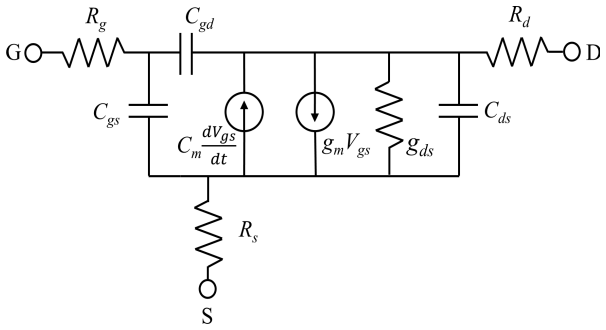

Fig. 1. Equivalent structure of the field-effect transistor.

Based on this algorithm and modeling methodology, we further develop automated model software, named MOSFitApply. MOSFitApply generates the MNN transistor model and extracts model parameters, converts it to Verilog-A code, and then starts the SPICE circuit simulation altogether in an automated fashion, without manual intervention required. Moreover, we tested our model methodology as well as software with Multiphysics TCAD data of a newly designed silicon-based CSFET [27]. The I–V/C–V model was generated within hours, and the fitting of the first and second derivatives of these characteristics is tested and shows a minimum error (<5%) from data. The 19-stage ring oscillator (RO) and two-stage operational amplifier circuit simulations have also been successfully demonstrated. We believe that our model methodology and software will be a good supplement to existing transistor models and particularly contribute to the accurate analog and RF circuit simulations as well as early stage novel device to circuit cooptimization (DTCO). The details of this article is organized as follows. Section II introduces model architecture and algorithm. Section III details the model software. Section IV shows the experiment for CSFETs and circuit simulation results. Section V shows the discussion of model methodology, and finally, key conclusions are summarized in Section VI.

II. Model Architecture and Algorithm

A general equivalent circuit of field-effect transistor is shown in Fig. 1, where G, D, and S are gate, drain, and source of the transistor, respectively. R_g, R_d, and R_s are the corresponding parasitic resistances, $C_{\rm gs}$, $C_{\rm gd}$, and $C_{\rm ds}$ are the intrinsic capacitances, and $C_m = C_{\rm dg}$–$C_{\rm gd}$ is a trans-capacitance taking care of the different effects of the gate and the drain on each other in terms of charging currents [28]. C–V and Q–V characteristic are fit at the same time since capacitance can be obtained by differentiating the charge with corresponding node voltage, and since $C_{\rm ds}$ is very small, we will ignore it in order to simplify the model. The nonlinear current and charge constitutive relations, $I_{\rm ds}$, $Q_{\rm gs}$, and $Q_{\rm gd}$ are presented as follows:

$$I_{\rm ds} = f_{\rm MNN}\left(V_{\rm gs}, V_{\rm ds}, W_i\right) \tag{1-1}$$

$$M = f_{\rm MNN}\left(V_g, V_d, V_s, W_i\right) M = Q_{\rm gs}, Q_{\rm gd} \tag{1-2}$$

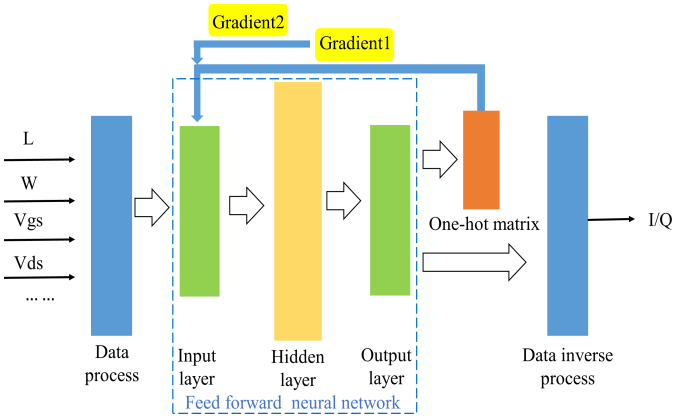

Fig. 2. Structure of MNN modeling method, the blue arrows represent the gradient of the neural network, and the black arrows represent the flow of the calculation.

where $I_{\rm ds}$ represents the current between D and S, M represents charge Q between G and S/D, $V_{\rm gs}$ and $V_{\rm ds}$ are the voltage between G and S/D, and W_i are the device model fitting parameters, i.e., weights in MNN model architecture. The derivatives of $I_{\rm ds}/M$ are presented as

$$\frac{dI_{\rm ds}}{dV_p} = \frac{df_{\rm MNN}(V_{\rm gs}, V_{\rm ds}, W_i)]}{dV_p} = {\rm grad}_{I_{\rm ds}-V_p} p = {\rm gs, ds} \quad (2\text{-}1)$$

$$\frac{dM}{dV_q} = \frac{df_{\rm MNN}(V_g, V_q, W_i)]}{dV_q} = {\rm grad}_{M-V_q} q = s, d \quad (2\text{-}2)$$

where ${\rm grad}_{I_{\rm ds}-V_p}$ is the gradient of output current $I_{\rm ds}$ with respect to input voltage $V_{\rm gs}$ or $V_{\rm ds}$ and while ${\rm grad}_{M-V_q}$ is the gradient of output charge Q with respect to input voltage V_s or V_d.

We propose the MNN model architecture to capture the device input and output characteristic relations in (1) and (2). As shown in Fig. 2, the model architecture is composed of five major parts including data process, feedforward neural network (FFNN), one-hot matrix, gradients, and data inverse process where the structure of FFNN includes a number of neuron nodes, hidden layers, activation functions [29], and so on. Generally, the FFNN structure is adjusted and the specific form of the activation function is chosen for different data and data distribution, in order to achieve high model accuracy. The MNN model inputs are device input variables, such as bias voltage, device length, and width. The outputs are the device characteristics, such as current and charge.

Data process and inverse process are required for successful training and parameter extraction of MNN model. Usually, device characteristics could be linear or nonlinear and across a range of multiple orders of magnitudes. The large data distribution span leads to weights' [i.e., W_i in (1) and (2)] gradient diminishing or exploring in the process of MNN training and this would cause the training failure. Therefore, data are compressed into appropriate distribution by the logarithmic function. Note that different bases of the logarithmic function lead to different degree of compression, and this actually plays an important role in the fitting results. Also, the impact of this effect will be discussed in Section V. MNN learns from the processed data and the FFNN outputs will be compressed data characteristics. Finally, the outputs are converted back by inverse logarithmic function in a data inverse process.

The model is trained and the parameters are extracted by the MABP algorithm. The basic ANN-based model scheme is capable of mapping complex nonlinear device characteristics, which uses a stochastic gradient descent method in common backpropagation (BP) algorithm to learn characteristics from the device data. Here, derived from ANN and BP, we develop the MABP algorithm by utilizing the gradients in BP in PyTorch computational graph and set different gradient constraints in the neural network to capture the high-order derivatives of data. This avoids manual derivation of gradient formulas used in the loss function, and it is particularly effective for modeling high-order derivatives. Note that arbitrary order of derivative of the device electronic characteristics can be captured with a simple loss function shown in the following equation:

$$\text{loss} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - Y_i'\right)^2 + \sum_{1}^{m}\frac{a_m}{n}\sum_{i=1}^{n}\left(\text{grad}_i^m[j] - \text{grad}_i^{m'}[j]\right)^2 \quad (3)$$

where a_m represents the constant weight for different orders derivative, n is the group number of measure data, Y_i and grad_i^m present the output [$I_{\rm ds}$ or M in (1)] and its specific mth gradient we concerned in MNN, respectively, Y_i' and $\text{grad}_i^{m'}$ represent the device data and the corresponding mth partial derivative of the device data, respectively, and j represents the index of grad element we concerned. By considering gradients in the network, our MNN method is generally capable of capturing any order derivatives of device outputs.

The detailed flow of MABP algorithm is shown in Algorithm 1. After data process and FFNN forward propagation, the specific elements, such as I or Q, are extracted in the output matrix, and the corresponding gradients are added into the loss function in (3) as further training constraints. In particular, we use a one-hot matrix [30] technique to conveniently realize this procedure. The output of FFNN multiplied by the one-hot matrix is the desired element. Thus, the entire process includes forward propagation, multiplication of one-hot matrix, constraint gradients, and BP. Moreover, using the same methodology, we can obtain the second-order gradient of the first-order gradient with respect to MNN input, and it is the second-order partial derivative matrix and so forth. We can acquire any number order of gradients and train the MNN model by supervising these gradients with the corresponding high-order data derivatives. In this way, the MNN model could guarantee an accurate fitting of device characteristics and their high-order derivatives.

In order to elaborate the algorithm more clearly, we take a simplest three inputs, two outputs, one hidden layer, and one of 2*1 hot-matrix MNN model as an example to address the training process details. Equation (4) shows the relationship between the input and the output

$$Y = \begin{bmatrix} y1 & y2 \end{bmatrix} = f(W_2 f(W_1 X + b_1) + b_2)$$
$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4)$$

Algorithm 1 MABP

Inputs: Data such as processed I-V/Q-V etc., FFNN, one-hot matrix.
Output: Weights of neural network which has been trained to represent the characteristics of input data
1: Feeding the data into the FFNN and initiate FFNN weights
2: FFNN forward propagation, calculate the output of FFNN
3: Extract the desired element in the output of FFNN by one-hot matrix.
4: Get the (high-order) gradients of the desired element with respect to input data.
5: Calculating the (high-order) derivatives of the input device data
6: Construct the loss function with these gradients constrained by derivatives in **5**
7: FFNN back propagation, minimize the loss function by stochastic gradient descent
8: Iterating 2-7 to update weights in FFNN
9: while over-fitting or under-fitting:
10: Adjust hyper-parameters of FFNN and repeat **8**
11: end while
12: return weights

$$y_1 = \begin{bmatrix} y_1 & y_2 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5}$$

$$\text{grad}_{X-y1} = \left[\frac{\partial y_1}{\partial x_1}, \frac{\partial y_1}{\partial x_2}, \frac{\partial y_1}{\partial x_3} \right] \tag{6}$$

where X represents the neural network input such as bias voltage, Y represents the neural network output such as I–V and Q–V characteristics, W_1 and W_2 are the weights, b_1 and b_2 are the bias, f is the activation function in the MNN, y_1 and y_2 are the elements of neural network output matrix, and x_1, x_2, and x_3 are the elements of neural network input matrix. The output of neural network needs to be multiplied by a one-hot matrix as (5) and calculate the gradient of a particular output node with respect to the input. The partial derivative of y_1 with respect to X is shown in (6) as grad_{X-y1}, and we can also calculate the gradient of grad_{X-y1} with respect to X as second-order partial derivative. Then, we can constrain these gradients in loss function to fit the high-order derivative characteristics.

III. Model Creation

To facilitate the automatic model generation process, we further developed a software based on this MNN model scheme, named MOSFitApply. The general modeling and simulation flow inherent in the MOSFitApply is shown in the blue box of Fig. 3. Also, it includes six major parts: model structure creation, automatic parameter selection, model training, model testing, model transform, and Verilog-A file generation, and we will introduce these six functions in detail in the following.

Model Structure Creation: After the processed device data input MOSFitApply, the software will calculate the number of data and give a suitable model structure that includes the number of hidden layers, the number of nodes, and activation function automatically. Alternatively, users can choose to specify their own network structure manually.

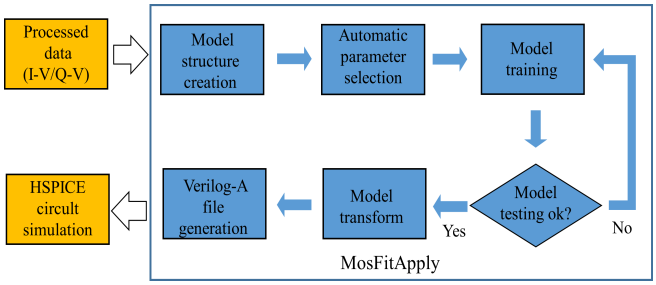

Fig. 3. Model generation flow in the creation software.

Automatic Parameter Selection: Software will select suitable hyperparameters such as learning rate, batch size, and optimizer automatically. Optionally, this process can also be manually adjusted by the user.

Model Training: MOSFitApply uses MABP mentioned above to update the weights in MNN and learn the characteristics and derivative characteristics in transistor data. Model training function automatically trains a neural network model based on the input data, model structure, and hyperparameters already set.

Model Testing: After MNN model creation and weights extraction, it is necessary to test the accuracy of output characteristics of the model. If the accuracy of the model in training is low, then the data are underfit; if the accuracy of the model in training is high, but the output characteristics of untrained part is unphysical in testing, then the data are overfit. If either of these cases occurs, the model should be retrained after adjusting the hyperparameters and MNN structure. MOSFitApply is capable of automatically completing and visualizing the process of model testing.

Model Transform: After testing, the weight matrix of the model is transformed into plain mathematical expressions according to the specific MNN structure and its included activation functions.

Verilog-A Model File Generation: Finally, the mathematical equations with all the trained parameters of the established models are written to a Verilog-A file for further circuit simulation purposes.

As a bridge between process technology and circuit design in modern ICs, the established model should be able to not only accurately fit the data but also converge smoothly in the circuit simulation. After generation the Verilog-A model file by MOSFitApply, device characteristics, such as I–V and C–V, are simulated in circuit netlist using a standard HSPICE simulator, to verify model accuracy and compatibility with SPICE simulation. Circuit simulations using the created models are also tested to ensure the simulation convergence. Based on the circuit simulation results, the circuits' properties, such as power consumption and latency, could be extracted for further device and circuit design and optimization.

IV. Experiment and Results

In this section, we take the novel gate-all-around (GAA) Si CSFET as an example to demonstrate the MNN model

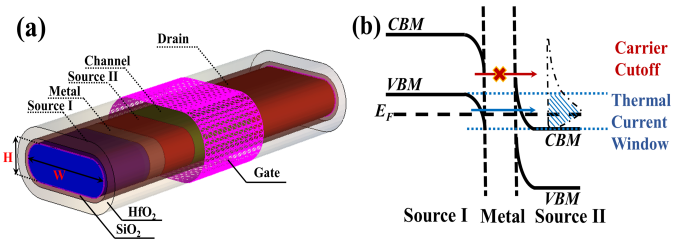

Fig. 4. (a) CSFET structure designed in TCAD and (b) energy band schematic of thermal carrier filtering effect in cold source structure.

capability. Si CSFET is a novel field-effect transistor (FET) design first proposed by Liu *et al.* [31], for which the conventional compact model formulations is not yet available. CSFET combines the benefit of ultralow subthreshold slope (SS), which could overcome the 60-mV/dec thermal limit, as well as high ion (1–2 orders of magnitude higher than typical Si TFET) [32]–[34]. These advantageous properties render the CSFET as a potential candidate for future technology devices. Fig. 4(a) shows a sample CSFET structure designed in TCAD. The lengths of source I/Metal/Source II/Channel/Drain are 5 nm/3 nm/6 nm/12 nm/14 nm, respectively. The thickness and width of NS are 5 nm and 5 nm/10 nm/15 nm with effective oxide thickness of 0.7 nm. Besides, Source I/Source II are n-type/p-type doped in pMOS with the same concentration of 3×10^{20} cm^{-3}. Using the multiphysics framework, we generate the I–V and C–V characteristics of CSFET. Essentially, the thermal carrier filtering effect [Fig. 4(b)] in CSFET is captured by the effective cold carrier distribution model [35]. Quantum correction with modified local-density approximation and bandgap narrowing model is embedded. Surface recombination is implemented at the metal/semiconductor interface. Nonlocal tunneling is used to calculate tunneling current through the channel with tunneling probability computed from the Wentzel–Kramer–Brillouin (WKB) approximation. The novel device mechanism and distinct I–V features are beyond the scope of established conventional compact model such as BSIM. Therefore, CSFET appears as an appropriate candidate to test and validate the proposed MMN-based novel compact model, which provides a rapid solution for early stage CSFET DTCO.

MOSFitApply is used to create an MNN device model based on the I–V and C–V from TCAD simulations. All six steps from model structure creation to Verilog-A file generation are all done automatically without manual work. It takes a total of less than 5 h to process more than 3000 groups of data points and finish the entire procedure, compared to months or weeks to create a pure physics or empirical formula-based compact model and finish model parameter extraction for such novel device.

The fitting results of MNN-based compact model are shown in Fig. 5. In Fig. 5, dots are the TCAD data and lines are the model outputs. Fig. 5(a)–(d) shows the fitting results of I_{ds}–V_{gs}, I_{ds}–V_{ds}, and their derivative characteristics. It should be noted that we fit the I_{ds}–V_{gs} characteristics for every −0.01 V V_{ds} step change between 0 and −0.05 V, and every −0.05-V V_{ds} step change between −0.05 and −0.65 V, we only show limited fitting results in Fig. 5 for graph clarity.

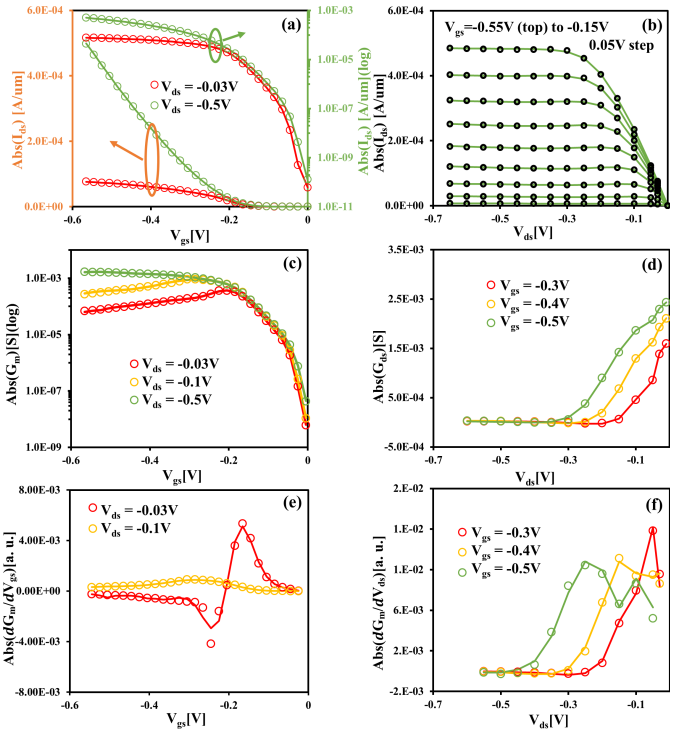

Fig. 5. SPICE simulation results of pMOS CSFET. Symbols are TCAD data and lines are MNN device model output in all figures. (a)–(d) Are I–V/G–V fitting result. (e) and (f) dG_m/dV_{gs} and dG_{ds}/dV_{ds} fitting result, respectively.

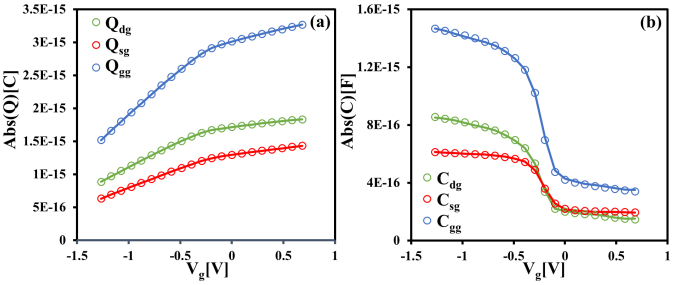

Fig. 6. (a) and (b) Q–V and C–V characteristics fitting result. Symbols are TCAD data and lines are MNN device model output in all figures.

The mean error of I–V/Q–V characteristics of model fitting is within 1% and the mean error of G–V/C–V is within 3%. The MNN device model can even fit the third-order derivative of I_{ds}–V_{gs}/I_{ds}–V_{ds} with high precision, as shown in Fig. 5(e) and (f).

Fig. 6(a) and (b) shows the results of ac characteristics, i.e., Q–V and C–V. Note that Q_{sg}/C_{sg} and Q_{dg}/C_{dg} are different due to the unsymmetrical structure of CSFET. Figs. 5 and 6 only show the fitting results of pMOS CSFET, and nMOS CSFET results are largely the same and thus not shown here. It is noted that we have also tested the MNN model with conventional Si MOSFET TCAD data, and the fitting results are similar to CSFET ones and not shown here.

The created MNN CSFET device model is employed first for a 19-stage RO circuit SPICE simulation. The circuit diagram and simulation results are shown in Fig. 7, and the simulation runs smoothly and completely within ∼1 s, comparable to that of physics-based model such as BSIM

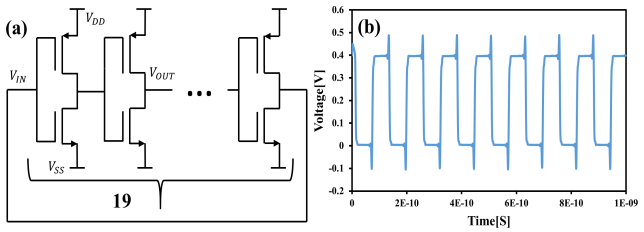

Fig. 7. (a) Schematic of RO used in simulation, it includes 19 pairs of transistors, and each one is modeled by the MNN device model. (b) Output of RO circuit from SPICE simulation.

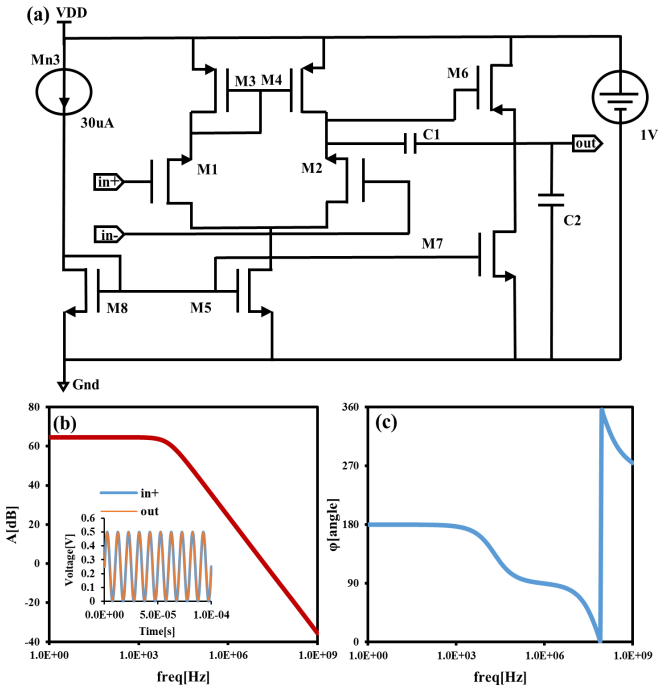

Fig. 8. (a) Two-stage operational amplifier circuit diagram. SPICE simulation of (b) amplitude-frequency characteristics and (c) phase-frequency characteristics.

models. Based on simulation results, the average delay and power per stage are 3.27 ps and 4.41 μW, respectively.

Moreover, the MNN-based device model is particularly advantageous for analog and RF circuit simulations due to its high-order derivative fitting capabilities. The created model is therefore tested with a two-stage operational amplifier circuit shown in Fig. 8(a). The circuit simulation converges and finishes within 1.5 s, and this is also comparable to the widely used BSIM model. In the simulation, the bias circuit at the left of the amplifier generates a current of 30 μA, which is then mirrored to the first-stage differential amplifier through a current mirror. The result is output by the second-stage common source amplifier composed of M6 and M7. The amplitude- and phase-frequency characteristics of the amplifier are shown in Fig. 8(b) and (c), respectively. We find out that the gain can reach 60 dB, the gain–bandwidth product (GBW) is more than 20 MHz, and the phase margin (PM) is greater than 60°.

V. Discussion

The results above show that the MNN device compact model scheme is able to accurately capture all nonlinear electrical characteristics of CSEFT and their high-order characteristics and perform SPICE simulation successfully. Similarly, we argue that this MNN scheme can be used to model linear or nonlinear semiconductor device characteristics of any other semiconductor devices where accurate high-order derivatives' information is important. The same model methodology could also be applied to high-frequency device modeling, such as accurate fitting of S parameters in RF device model.

A. Gradients' Constraints in MNN Model Algorithm

To illustrate the effectiveness of additional gradients' constraints in the MNN model scheme, NN models containing different orders of gradients' constraints are tested for the same CSFET nMOS data. All three models share the same hyperparameters and network structure. We compare the experimental results 1 Grad MNN/common ANN, 2 Grad MNN, and 3 Grad MNN gradients constrained model, where 2 Grad MNN contains both output and first-order output gradient constraints and 3 Grad MNN adds additional second-order gradient constraint to 2 Grad case. The results in Fig. 9(a) clearly show that all three models could fit Q–V. However, the MNN model supervised with 2 and 3 Grad can also capture data's high-order derivative more accurately. As shown in Fig. 9(b), the 1 Grad case fails to fit the dQ/dV_g (C–V) data completely but the 2 and 3 Grad cases succeed. For the d^2Q/dV_g^2 fitting shown in Fig. 9(c), only 3 Grad case successfully captures the data, whereas 1 and 2 Grad are not sufficient for the task. Compared to the 1 Grad case, the 3 Grad MNN decreases the fitting error from 2% to 0.2%, a 10× improvement in the Q–V fitting task. It should be noted that since the device data are compressed by the logarithmic function, the corresponding transformation should be made according to the chain derivative rule when using its derivative to constrain the gradient. The same Q–V data are used for all three training cases for the consistency of our results. As suggested in [12], the different integral path of TCAD C–V data could cause errors in the absolute value of Q–V. Therefore, directly fitting C–V and its higher order derivatives could be good practice for accurate C–V modeling.

Model training time (model turnaround time, TAT) versus model accuracy for different gradients' constraints has also been studied, and the results are shown in Fig. 9(d). MNN (2 and 3 Grad) shows better fitting accuracy (less error) than simple ANN (1 Grad) for all meaning training time between 3 and ~300 s. Simple ANN model training error saturates at ~2.9% after 5 s, whereas MNN (3 Grad) model training error could reach ~0.05% and still improving after 200 s. This is roughly ~60× accuracy improvement.

To explore the influence of MNN on circuit simulation, 19-stage ROs are simulated both with ANN (1 Grad) model and the MNN (3 Grad) model shown in Fig. 9(a)–(c). Fig. 9(e) shows the delay versus V_{dd} results, and the error is defined by the difference between simulation results of 1 and 3 Grad models. The V_{dd}-dependent delay error could be as large as 78% and 9% due to inaccurate C–V fitting of 1 Grad model and 2 Grad model. This clearly shows that the application of MNN could improve the circuit-level simulation accuracy.

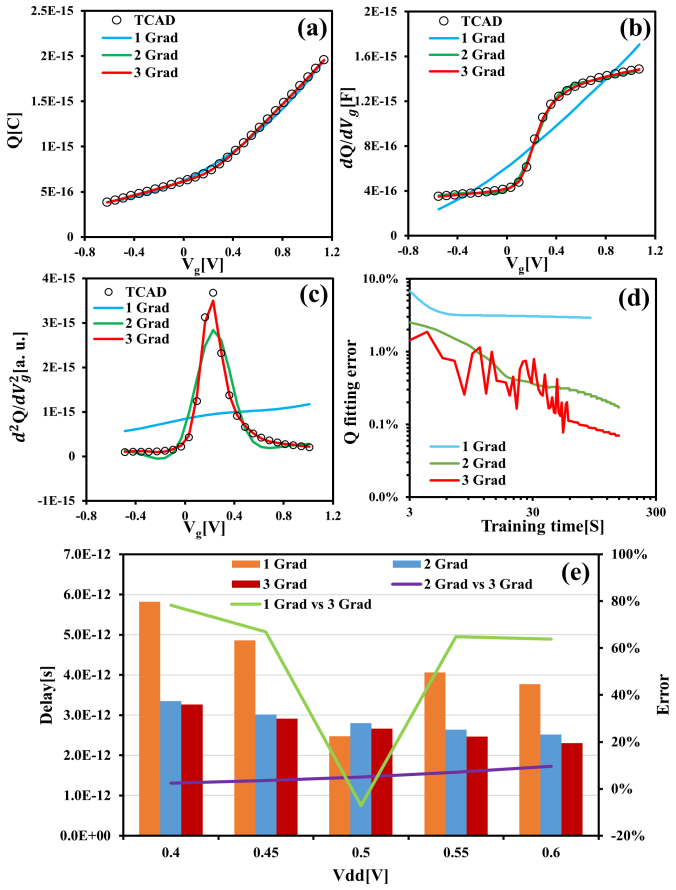

Fig. 9. (a)–(c) Q–V_g, dQ/dV_g–V_g, and d^2Q/dV_g^2–V_g curve fitting result comparison between different numbers of gradients supervised MNN models, respectively. (d) Comparison of training time and fitting accuracy between different NN model constraints. (e) 19 RO simulation delay comparison results between different models.

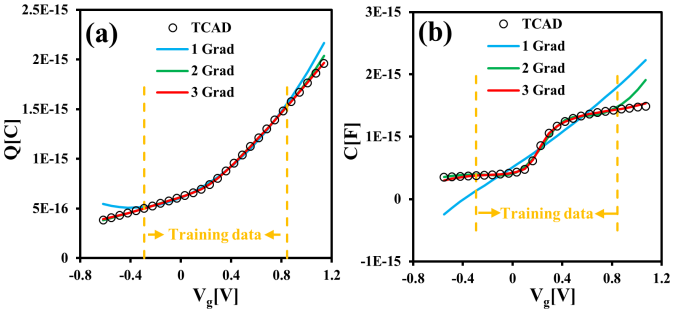

Fig. 10. Model extrapolation capability test for different derivative constraints. (a) Q–V fitting and model prediction over training data range. (b) C–V fitting and model prediction over training data range.

B. Model Extrapolation Capability Experiment of MNN

Both ANN- and LUT-based TCM typically suffer from model extrapolation capability, i.e., model cannot predict the electrical characteristics outside the training data range and causes unphysical model behavior over the known data range. The MNN approach is demonstrated to alleviate this issue. Fig. 10 shows the model extrapolation test for normal ANN model (1 Grad) versus MNN (2 and 3 Grad). The TCAD Q–V ($V_g = [-0.62, 1.14$ V]) data is divided into three parts as shown in Fig. 10, and only the middle part ($V_g = [-0.3, 0.85$ V]) is used for training, while the trained model playback prediction outside this range is compared to the original data. The results show that the MNN model can better extrapolate the characteristics outside the training data range, due to precise capture of original data curvature. It is found that MNN (3 Grad) can predict the characteristics with an error of less than 1% even for the test data 20% outside the training data range and predict its derivative characteristics with an error of less than 5%, as shown in Fig. 10(a) and (b), respectively. This is significant improvement from a typical ANN-based model.

C. Data Process and Superparameters in MNN Model Training

Data process before model creation is critical as MNN method is a data-driven approach, as other ANN-based device models. The data noise will lead to the difficulty of fitting its high-order derivatives as those derivatives generally magnify data noise by tens to hundreds. The excessively large or small degree of data compression will lead to the concentrated data distribution or the large span of data distribution, which are detrimental to accurate MNN modeling and data recovery. We found empirically that it is effective to compress the data into the $[-20, 0]$ by selecting a suitable base of the logarithmic function. Each gradient constraint term has a fitting superparameter a_m in the loss function, which represents the importance of this particular gradient constraint term. Typically, if the derivative of data is smooth, a_m is set to large values; otherwise, a_m may be reduced to avoid the effect of gradient oscillation.

D. Number of Parameters in MNN Model

The ANN-based model may sometimes contain a large number of weights, i.e., model parameters. This could lead to intensive iteration in SPICE simulation, resulting in slow or even convergence difficulty in large-scale circuit simulation. Therefore, in the process of model creation, the model size is compressed to facilitate smooth and fast SPICE circuit simulation. However, the insufficient number of parameters could lead to a decline in fitting accuracy. Therefore, a tradeoff iteration process is usually gone through for MNN model creation. It is found that the total number of weights or parameters in the MNN model should be around 500, to accurately fit the nonlinear characteristics of a transistor device, as well as ensure that it is smooth SPICE circuit simulations. In our experiment, a four-layer neural network is used with two hidden layers both having 20 neurons each, and there are two tanh activation functions between the layers.

VI. Conclusion

We propose a general transistor model scheme based on MNN. This MNN model is able to simultaneously capture the I–V/Q–V characteristics and their high-order derivative characteristics of transistor devices. MNN can be applied to model other new semiconductor devices. Based on the MNN model

scheme, we further develop an automated modeling software MOSFitApply. This software utilizes the MNN-based method and MABP algorithm to automate the entire data-driven transistor modeling process, and it links the device data to SPICE circuit-level simulation effectively. The model methodology and software are tested with novel GAA Si CSFET, and both logic and analog circuits' simulations are successfully accomplished with the created MNN models. The MNN model scheme shortens the compact model creation cycle by more than tenfold for novel semiconductor devices. This enables a rapid solution to verify the device-to-circuit-level characteristics of such devices, and it could be useful for the cooptimization of new devices and circuits both in academic research and in industrial applications.

REFERENCES

[1] H. Agarwal *et al.*, "Modeling of GeOI and validation with Ge-CMOS inverter circuit using BSIM-IMG industry standard model," in *Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC)*, Aug. 2016, pp. 444–447.

[2] Y. S. Chauhan *et al.*, "BSIM—Industry standard compact MOSFET models," in *Proc. ESSCIRC (ESSCIRC)*, Sep. 2012, pp. 30–33.

[3] M. Miyake *et al.*, "The flexible compact SOI-MOSFET model HiSIM-SOI valid for any structural types," in *Proc. Int. Conf. Simulation Semiconductor Processes Devices*, Sep. 2011, pp. 167–170.

[4] H. J. Mattausch, T. Umeda, H. Kikuchihara, and M. Miura-Mattausch, "The HiSIM compact models of high-voltage/power semiconductor devices for circuit simulation," in *Proc. 12th IEEE Int. Conf. Solid-State Integr. Circuit Technol. (ICSICT)*, Oct. 2014, pp. 1–4.

[5] J. Pan, A. Topol, I. Shao, C.-Y. Sung, J. Iacoponi, and M.-R. Lin, "Novel approach to reduce source/drain series and contact resistance in high-performance UTSOI CMOS devices using selective electrodeless CoWP or CoB process," *IEEE Electron Device Lett.*, vol. 28, no. 8, pp. 691–693, Aug. 2007.

[6] S.-H. Lo, K. Das, C.-T. Chuang, and J. Sleight, "Power-gating schemes for ultra-thin SOI (UTSOI) circuits in hybrid SOI-epitaxial CMOS structures," in *Proc. Int. Symp. VLSI Design, Autom. Test*, Apr. 2006, pp. 1–2.

[7] S. Ramey *et al.*, "Aging model challenges in deeply scaled tri-gate technologies," in *Proc. IEEE Int. Integr. Rel. Workshop (IIRW)*, Oct. 2015, pp. 56–62.

[8] P. Kushwaha *et al.*, "Characterization and modeling of flicker noise in FinFETs at advanced technology node," *IEEE Electron Device Lett.*, vol. 40, no. 6, pp. 985–988, Jun. 2019.

[9] V. B. Litovski, J. I. Radjenovic, Z. M. Mrcarica, and S. L. Milenkovic, "MOS-transistor modeling using neural network," (in English), *Electron. Lett.*, vol. 28, no. 18, pp. 1766–1768, Aug. 1992.

[10] R. A. Thakker, C. Sathe, A. B. Sachid, M. Shojaei Baghini, V. Ramgopal Rao, and M. B. Patil, "A novel table-based approach for design of FinFET circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 7, pp. 1061–1070, Jul. 2009.

[11] J. Wang, N. Xu, W. Choi, K.-H. Lee, and Y. Park, "A generic approach for capturing process variations in lookup-table-based FET models," in *Proc. Int. Conf. Simulation Semiconductor Processes Devices (SISPAD)*, Sep. 2015, pp. 309–312.

[12] J. Wang, Y. H. Kim, J. Ryu, C. Jeong, W. Choi, and D. Kim, "Artificial neural network-based compact modeling methodology for advanced transistors," *IEEE Trans. Electron Devices*, vol. 68, no. 3, pp. 1318–1325, Mar. 2021.

[13] M. Lazaro, I. Santamaria, and C. Pantaleon, "A smooth and derivable large-signal model for microwave HEMT transistors," in *Proc. IEEE Int. Symp. Circuits Syst. Emerg. Technol. 21st Century*, May 2000, pp. 713–716.

[14] T. Yildirim, H. Torpi, and L. Ozyilmaz, "Modelling of active microwave transistors using artificial neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 1999, pp. 3988–3991.

[15] V. Markovic, Z. Marinkovic, and N. Males-Ilic, "Application of neural networks in microwave FET transistor noise modeling," in *Proc. 5th Seminar Neural Netw. Appl. Electr. Eng. (NEUREL)*, Sep. 2000, pp. 146–151.

[16] M. Lazaro, I. Santamaria, C. Pantaleon, C. Navarro, A. Tazon, and T. Fernandez, "A modular neural network for global modeling of microwave transistors," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. IJCNN. Neural Comput., New Challenges Perspect. New Millennium*, Jul. 2000, pp. 389–394.

[17] S. Lleo, "Machine learning: An applied mathematics introduction," (in English) *Quant. Finance*, vol. 20, no. 3, pp. 359–360, Mar. 2020.

[18] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[19] H. Huang and C. Wu, "Approximation capabilities of multilayer fuzzy neural networks on the set of fuzzy-valued functions," *Inf. Sci.*, vol. 179, no. 16, pp. 2762–2773, Jul. 2009.

[20] H. Yuan, C. F. Van Der Wiele, and S. Khorram, "An automated artificial neural network system for land use/land cover classification from landsat TM imagery," (in English), *Remote Sens.*, vol. 1, no. 3, pp. 243–265, Sep. 2009.

[21] S. Abreu, "Automated architecture design for deep neural networks," 2019, *arXiv:1908.10714*. [Online]. Available: http://arxiv.org/abs/1908.10714

[22] Q. Chen and G. Chen, "Artificial neural network compact model for TFTs," in *Proc. 7th Int. Conf. Comput. Aided Design Thin-Film Transistor Technol. (CAD-TFT)*, Oct. 2016, p. 11.

[23] J. Xu and D. E. Root, "Advances in artificial neural network models of active devices," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Model. Optim. (NEMO)*, Aug. 2015, pp. 1–3.

[24] A. Jarndal and S. Hamdan, "Global optimization of neural network-based electrothermal model for GaN transistors," in *Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, Nov. 2017, pp. 523–526.

[25] J. Xu, M. C. E. Yagoub, R. Ding, and Q. Jun Zhang, "Exact adjoint sensitivity analysis for neural-based microwave modeling and design," *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 1, pp. 226–237, Jan. 2003.

[26] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," (in English), in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 32, 2019, pp. 8026–8037.

[27] W. Gan *et al.*, "A multi-physics TCAD framework for fast and accurate simulation of SteepSlope Si-based cold source FET," in *Proc. Int. Symp. VLSI Technol., Syst. Appl. (VLSI-TSA)*, Aug. 2020, pp. 66–67.

[28] I. Kwon, M. Je, K. Lee, and H. Shin, "A simple and analytical parameter-extraction method of a microwave MOSFET," *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 6, pp. 1503–1509, Jun. 2002.

[29] T.-Y. Kwok and D.-Y. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 630–645, May 1997.

[30] A. V. Uriarte-Arcia, I. López-Yáñez, and C. Yáñez-Márquez, "One-hot vector hybrid associative classifier for medical data classification," *PLoS ONE*, vol. 9, no. 4, Apr. 2014, Art. no. e95715.

[31] F. Liu *et al.*, "First principles simulation of energy efficient switching by source density of states engineering," in *IEDM Tech. Dig.*, Dec. 2018, p. 33.

[32] K. P. Cheung, "On the 60 mV/dec @300 k limit for MOSFET subthreshold swing," in *Proc. Int. Symp. VLSI Technol., Syst. Appl.*, 2010, pp. 72–73.

[33] F. Liu, C. Qiu, Z. Zhang, L.-M. Peng, J. Wang, and H. Guo, "Dirac electrons at the source: Breaking the 60-mV/decade switching limit," *IEEE Trans. Electron Devices*, vol. 65, no. 7, pp. 2736–2743, Jul. 2018.

[34] E. Gnani, P. Maiorano, S. Reggiani, A. Gnudi, and G. Baccarani, "Performance limits of superlattice-based steep-slope nanowire FETs," in *IEDM Tech. Dig.*, Dec. 2011, pp. 5.1.1–5.1.4.

[35] W. Gan *et al.*, "Design and simulation of steep-slope silicon cold source FETs with effective carrier distribution model," *IEEE Trans. Electron Devices*, vol. 67, no. 6, pp. 2243–2248, Jun. 2020.