

# A Novel Neural-Network Device Modeling Based on Physics-Informed Machine Learning

Bokyeom Kim<sup>ID</sup> and Mincheol Shin<sup>ID</sup>, *Member, IEEE*

**Abstract**—In this work, we present a novel physics-informed machine learning (PIML)-based neural-network device modeling that predicts both device performance and spatial physical quantities in real-time. Using cutting-edge technologies such as physics-informed neural network (NN) and physics-informed deep operator networks, our approach suggests interpolation and extrapolation strategies in device physics modeling. Despite being trained with a small number of bias voltages, our model demonstrates remarkable accuracy, with a mean absolute percentage error (MAPE) of 0.12% for predicting potential for interpolation and 0.19% for extrapolation. Our approach can be used for data-efficient NN modeling for TCAD and real-time physics analysis in the spatial domain.

**Index Terms**—Deep operator network, nanowire, neural network (NN), physics-informed machine learning (PIML).

## I. INTRODUCTION

AS ELECTRONIC devices continue to become smaller and more complex, incorporating 3-D structures to improve their performance, traditional device modeling methods such as numerical self-consistent simulation and compact model generation have become increasingly time-consuming. To address this issue, researchers have turned to neural networks (NNs) to accelerate device modeling [1], [2], [3], [4]. Previous studies exploring NN-based methods have focused on two primary research aspects: spatial physical quantities and device performance. Studies on NN-based spatial physical quantities aim to accelerate numerical simulation by modeling spatial physical quantities (e.g., potential, charge density) using NNs [1], [2]. However, these studies have relied on purely data-driven (DD) approaches, requiring dense data collection or training with large amounts of data [1], [2]. On the other hand, researchers on NN-based device performance have modeled the correlation between device parameters (e.g., gate-drain bias  $V_g/V_d$ , length, width) and device performance (e.g., current, conductance) [3], [4], [5], [6]. However, NNs in those studies do not provide information on spatial physical

Manuscript received 3 July 2023; revised 12 September 2023; accepted 14 September 2023. Date of publication 3 October 2023; date of current version 24 October 2023. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science and Information Communication Technology (ICT) under Grant RS-2023-00243425. The review of this article was arranged by Editor F. Bonani. (*Corresponding author: Mincheol Shin.*)

The authors are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea (e-mail: kimbou@kaist.ac.kr; mshin@kaist.ac.kr).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TED.2023.3316635>.

Digital Object Identifier 10.1109/TED.2023.3316635

quantities, making it difficult for engineers to obtain physical insights for optimization [2].

Recently, a technique called physics-informed machine learning (PIML) has been proposed to combine physical knowledge with machine learning to make model predictions physically consistent, increasing model accuracy and reducing the amount of data required for training [7]. In this study, we have developed a PIML-based NN device modeling that predicts both device performance and spatial physical quantities using the state-of-the-art PIML technique: physics-informed NN (PINN) and PI-deep operator network (PI-DeepONET). The techniques use residuals of partial differential equation (PDE) as a regularizer by constructing a PDE loss function with the help of automatic differentiation [7], [8], [9]. We have explored strategies to increase model accuracy and reduce data requirements for modeling spatial physical quantities. Although there has been an attempt to replace TCAD simulation by predicting spatial physical quantities and device performance based on a multitask ML with a DD approach, we have investigated, for the first time, a modeling strategy to accelerate TCAD simulation in a data-efficient manner [10].

Our approach is demonstrated on a 3-D nanowire field-effect transistor (NW FET) as shown in Fig. 1. A simulation of the NW FET requires electrostatics, quantum confinement, and charge/current density calculations, which are evaluated in this work by the Poisson equation (PE), Schrödinger equation (SE), and Boltzmann transport equation (BTE), respectively. In our NN device modeling, the physics equations, the PDEs with the boundary conditions, are directly used in the form of physics loss function ( $L_{\text{physics}}$ ) as displayed in Figs. 4–6, which remarkably improves NN modeling efficiency and accuracy.

## II. METHODS

For the target device of Fig. 1, the training data were generated by solving the PE, SE, and BTE self-consistently based on traditional numerical methods [11], [12]. The device specifications and physical parameters are provided in Fig. 1(d). In contrast to a previous study that densely sampled data at an interval of 0.01 V [2], we sampled our training data at various intervals ranging from 0.1 to 0.3 V. This resulted in a total of 25 training bias points as shown in Fig. 1(e). Note that the primary motivation behind using a nonregular grid is to gain a clearer understanding of the impact of PDE residuals under extreme conditions, without the need for densely sampled bias values.

As shown in Fig. 2, our model consists of three networks: the self-consistent potential network (SCP-NET), SE-NET, and BTE-NET. The SCP-NET predicts SCP value ( $V_{\text{NN}}$ ), which

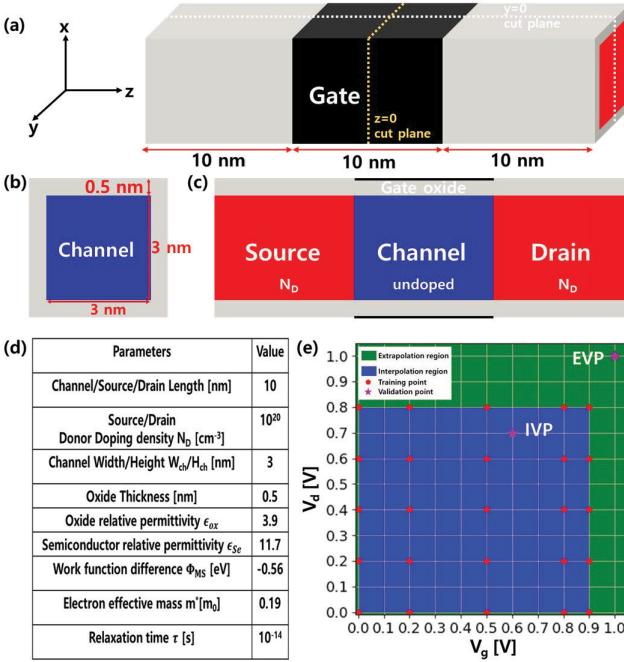


Fig. 1. (a) Simulated device, (b)  $z = 0$  cut plane, and (c)  $y = 0$  cut plane. (d) Device simulation parameters. (e) Domain for training data sampling.

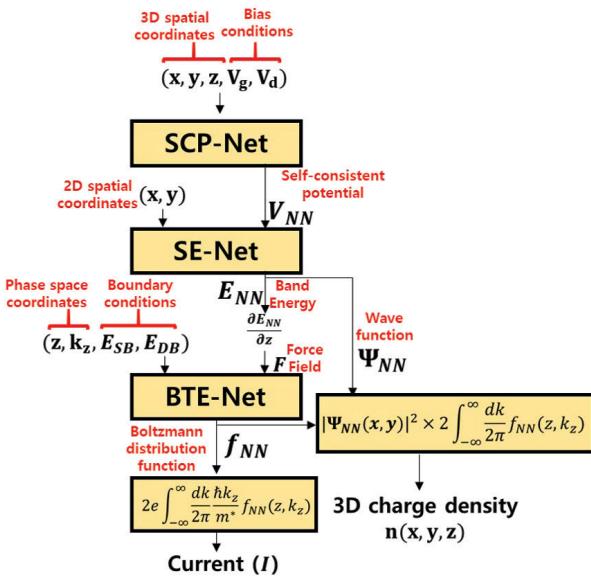


Fig. 2. Schematic flowchart of our PIML-based NN device modeling.

is then fed into the SE-NET to produce the electron band energy ( $E_{NN}$ ) and wave function ( $\Psi_{NN}$ ). The BTE-NET takes the force field ( $F$ ) as an input and outputs the Boltzmann distribution function ( $f_{NN}$ ). Unlike previous works [1], [2], our model predicts not only spatial quantities but also device performance (i.e., current).

The activation functions in the SE-NET and the BTE-NET are adaptive hyperbolic tangents [13], while the swish activation function is used for the SCP-NET [14]. Each network has six hidden layers with 100 nodes. The weight normalization is applied for the SE-NET and the BTE-NET. Adam optimizer is used with 200 000 training iterations [15]. The total loss function of the networks is the summation of data loss ( $L_{data}$ ) and

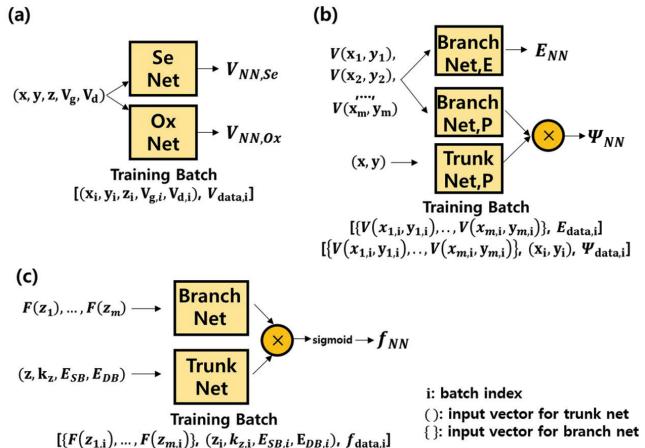


Fig. 3. Major components of the NN device model in this work. Schematics of (a) SCP-NET, (b) SE-NET, and (c) BTE-NET with the training batch configuration for each network.

$$\begin{aligned}
 L_{total}(\theta) &= \lambda_d L_{data}(\theta) + \lambda_p L_{physics}(\theta) \\
 L_{data}(\theta) &= RMS \left( \frac{V_{data,Se} - V_{NN,Se}}{V_{data,Se} + \delta} \right) + RMS \left( \frac{V_{data,Ox} - V_{NN,Ox}}{V_{data,Ox} + \delta} \right) \\
 L_{physics}(\theta) &= L_{pde,Se}(\theta) + L_{pde,Ox}(\theta) + L_{if}(\theta) + \lambda_b L_{BC}(\theta) \\
 L_{pde,Se}(\theta) &= MS \left( \nabla \cdot \epsilon_{se} \nabla V_{NN,Se} - \frac{e}{\epsilon_0} (n - N_D) \right) \\
 L_{pde,Ox}(\theta) &= MS \left( \nabla \cdot \epsilon_{ox} \nabla V_{NN,Ox} \right) \\
 L_{if}(\theta) &= L_{ifx}(\theta) + L_{ify}(\theta) + L_{ifc}(\theta) \\
 L_{ifx}(\theta) &= MS \left( \epsilon_{ox} \partial V_{NN,Ox} / \partial x \Big|_{x=\pm W_{ch}/2} - \epsilon_{se} \partial V_{NN,Se} / \partial x \Big|_{x=\pm W_{ch}/2} \right) \\
 L_{ify}(\theta) &= MS \left( \epsilon_{ox} \partial V_{NN,Ox} / \partial y \Big|_{y=\pm H_{ch}/2} - \epsilon_{se} \partial V_{NN,Se} / \partial y \Big|_{y=\pm H_{ch}/2} \right) \\
 L_{ifc}(\theta) &= MS(V_{NN,Ox} - V_{NN,Se}) \\
 L_{BC}(\theta) &= MS(V_{NN,Ox} - (V_g - \Phi_{MS})) \\
 \end{aligned}$$

Fig. 4. Loss function of the SCP-NET.  $\theta$  indicates the NN parameters.

$L_{physics}$ , where  $L_{physics}$  is added as a regularizer. Each loss term depends on NN parameter  $\theta$ . The weights for data, physics, and boundary conditions are  $\lambda_d$ ,  $\lambda_p$ , and  $\lambda_b$ , respectively, where  $\lambda$  values are manually adjusted to avoid overfitting and underfitting. For implementation, we use Python with JAX [16]. We use NVIDIA GeForce RTX 3090 GPU for NN training and inference.

#### A. SCP-NET

To develop the SCP-NET, we used the PINN approach. The model is trained using bias values ( $V_g, V_d$ ) and spatial coordinates ( $x, y, z$ ) as input parameters to learn the correlation between the self-consistent potential and input parameters. Domain decomposition is applied to enforce flux continuity at the semiconductor/oxide interface [17]. The SCP-NET consists of two separate fully connected NNs: Se Net and Ox Net which predict the potential values in the semiconductor and oxide regions [Fig. 3(a)].

The loss function of the SCP-NET is presented in Fig. 4. The interface loss  $L_{if}$  is estimated by adding the interface flux loss along the  $x$ - and  $y$ -directions ( $L_{ifx}/ify$ ) and the interface value continuity loss ( $L_{ifc}$ ). PE residual is evaluated using an automatic differentiation of NN with respect to the spatial coordinates. The residual value is used as  $L_{pde}$ . We use the root mean squared (rms) relative error (RE) to calculate  $L_{data}$  of the SCP-NET. An infinitesimal number  $\delta$  of  $1^{-10}$  is applied to

$$\begin{aligned}
L_{\text{total}}(\theta) &= \lambda_d L_{\text{data}}(\theta) + \lambda_p L_{\text{physics}}(\theta) \\
L_{\text{data}}(\theta) &= L_{\text{data},E}(\theta) + L_{\text{data},P}(\theta) \\
L_{\text{data},E}(\theta) &= MS(E_{\text{NN}} - E_{\text{data}}) \\
L_{\text{data},P}(\theta) &= MS(\Psi_{\text{NN}} - \Psi_{\text{data}}) \\
L_{\text{physics}}(\theta) &= MS(-\hbar/2m^* \cdot \nabla^2 \Psi_{\text{NN}} + V\Psi_{\text{NN}} - E_{\text{NN}}\Psi_{\text{NN}})
\end{aligned}$$

Fig. 5. Loss function of the SE-NET.  $\theta$  indicates the NN parameters.

$$\begin{aligned}
L_{\text{data}}(\theta) &= MS(f_{\text{NN}} - f_{\text{data}}) \\
L_{\text{physics}}(\theta) &= \lambda_p L_{\text{pde}}(\theta) + \lambda_b L_{\text{BC}}(\theta) \\
L_{\text{pde}}(\theta) &= MS\left(\frac{\hbar k_z}{m^*} \cdot \frac{df_{\text{NN}}}{dz} + \frac{1}{\hbar} \cdot F \cdot \frac{df_{\text{NN}}}{dk_z} + \frac{1}{\tau} (f_{\text{NN}} - f_{\text{eq}})\right) \\
L_{\text{BC}}(\theta) &= MS(f_{\text{NN}} - f_{\text{BC}_1}) + MS(f_{\text{NN}} - f_{\text{BC}_2}) \\
f_{\text{BC}_1}(z_{\text{min}}, k_z(k_z > 0)) &= 1/(1 + \exp(1/kT * (\hbar^2 k_z^2 / 2m^* + E_{\text{SB}}))) \\
f_{\text{BC}_2}(z_{\text{max}}, k_z(k_z < 0)) &= 1/(1 + \exp(1/kT * (\hbar^2 k_z^2 / 2m^* + E_{\text{DB}}))) \\
E_{\text{SB}}: E_S - \mu_s, E_{\text{SB}}: E_D - \mu_d & \\
E_{\text{SD}}: \text{band energy at Source/Drain end} & \\
\mu_{\text{SD}}: \text{Fermi-level at Source/Drain end} & \\
f_{\text{eq}}: \text{equilibrium Boltzmann distribution function} &
\end{aligned}$$

Fig. 6. Loss function of the BTE-NET.  $\theta$  indicates the NN parameters.

avoid numerical overflow when evaluating RE. The boundary condition is imposed by adding the boundary loss ( $L_{\text{BC}}$ ) to  $L_{\text{physics}}$ . For numerical simulations, we discretize the spatial domain into a grid 21, 21, and 120 points along the  $x$ -,  $y$ -, and  $z$ -axes, respectively. This results in 52 920 spatial points and potential values for each bias. For training the SCP-NET, we sample 20 000 batches for each iteration.

## B. SE-NET

The SE-NET is built with PI-DeepONET, which is an infinite function space mapping consisting of a branch net and a trunk net [Fig. 3(b)]. The branch net parameterizes spatially varying input function at predefined fixed positions [9]. The trunk net parameterizes the spatial coordinates. The outputs of branch net and trunk net are inner-producted to approximate the PDE output values. The SE-NET consists of two NNs for, respectively, predicting  $E_{\text{NN}}$  and  $\Psi_{\text{NN}}$ . Potential values along the cross section of NW ( $xy$  plane at each  $z$  coordinate) are fed into the branch net. For training loss,  $L_{\text{data}}$  is evaluated by mean squared error (MSE), while  $L_{\text{physics}}$  is evaluated by the residual of SE (Fig. 5). In all, 1000 training batches are sampled for each iteration.

## C. BTE-NET

The BTE-NET is based on PI-DeepONET [Fig. 3(c)]. We put the force field values  $F$  into the branch net and feed the phase space coordinates ( $z, k_z$ ) and the boundary conditions into the trunk net. A sigmoid function is used to impose nonnegativity for  $f_{\text{NN}}$  values. The BTE residual and boundary condition loss are added to  $L_{\text{physics}}$  (Fig. 6).

For numerical simulations, we discretize the phase space into 120 and 250 points along the  $z$ - and  $k_z$ -axes, respectively. As a result, there are 30 000 coordinate points for each bias condition. During training, 59 000 batches are sampled for each iteration. After 200 000 iterations, we change the criteria for evaluating  $L_{\text{data}}$  from MSE to rms RE and execute additional 1000 iterations as a fine-tuning process because  $f_{\text{data}}$  values vary exponentially across the phase space.

TABLE I ELAPSED TRAINING TIME FOR EACH NETWORK			
Training time [min]	Network		
Training Type	SCP-NET	SE-NET	BTE-NET
PI	98	20	106
DD	27	15	36

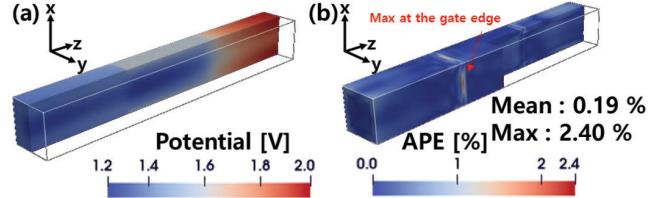


Fig. 7. Prediction capability of the SCP-NET at EVP. (a) Potential profile and (b) APE.

## III. RESULTS AND DISCUSSION

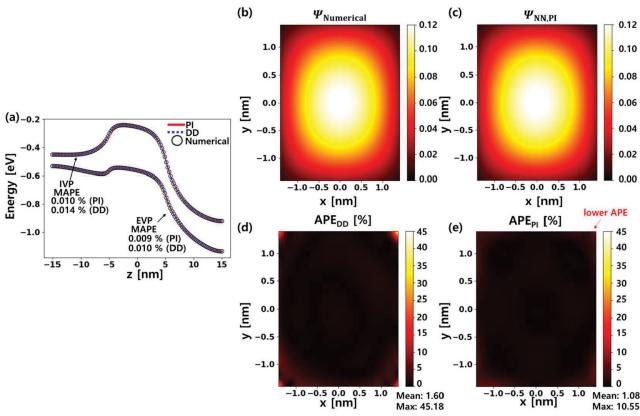
To show the effectiveness of  $L_{\text{physics}}$  which contains the physics information, the DD approach ( $\lambda_p = 0$ ) is compared with the PI approach ( $\lambda_p > 0$ ). We validate our model at two representative points: the interpolation validation point (IVP) of  $V_g = 0.6$  V,  $V_d = 0.7$  V and the extrapolation validation point (EVP) of  $V_g = 1.0$  V,  $V_d = 1.0$  V [Fig. 1(e)]. The training time for each network is tabulated in Table I. The PI method requires a total of 224 min, while DD only requires 78 min. The incorporation of automatic differentiation for evaluating  $L_{\text{pde}}$  and its impact on backpropagation result in longer training time for PIML.

### A. Prediction of Spatial Physical Quantities

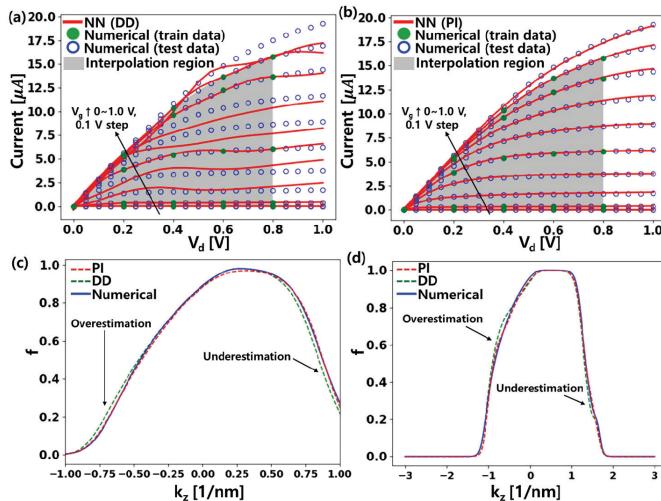
For the SCP-NET at IVP, despite the training data being sampled with a maximum step of 0.3 V in  $V_g$ , the mean absolute percentage error (MAPE) of only 0.12% outperforms a previous study showing MAPE of 0.14% [2]. Note that 61  $V_g$  conditions were sampled with a 0.01-V interval for training to forecast potential values in the 0.0–0.6-V range in that study, while only 5  $V_g$  is sufficient to more precisely predict potentials in the 0.0–1.0-V range in our study. Compared with the DD approach, the PI approach shows a reduction in MAPE from 0.20% to 0.19% at EVP. As shown in Fig. 7(b), the maximum absolute percentage error (APE) occurred at the gate edge.

For the SE-NET, the prediction of  $E_{\text{NN}}$  for interpolation and extrapolation is accurate with  $\text{MAPE} \leq 0.015\%$  [Fig. 8(a)].  $\Psi_{\text{NN}}$  values predicted by the PI model also match well with the numerical data [Fig. 8(b) and (c)]. The prediction accuracy of  $\Psi_{\text{NN}}$  is improved with the PI regularizer with MAPE decreased from 1.60% to 1.08% [Fig. 8(d) and (e)]. Note that the PI model exhibits much lower maximum APE of 10.55% than the DD approach of 45.18% at semiconductor/oxide interface.

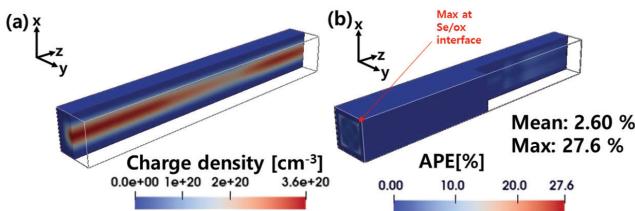
$F$ , exact force field values derived from numerical simulations, is fed to the BTE-NET to output  $f_{\text{NN}}$  and the drain current for the bias range of 0.0–1.0 V. The prediction results from the BTE-NET clearly show the effect of PIML. For the BTE-NET, the PI model predicts  $f_{\text{NN}}$  excellently although the predicted bias conditions are far from the training conditions [Fig. 9(b)]. On the other hand, the DD model shows poor performance when predicting the test data, predicting accurate values only within the training data. This limitation arises from both underestimation and overestimation of the Boltzmann distribution functions [Fig. 9(c) and (d)]. We found that PIML significantly enhances the accuracy of ML models in both



**Fig. 8.** Prediction capability of the SE-NET. (a)  $E_{\text{NN}}$  prediction by the PI model (red lines) at IVP and EVP, which are compared with the DD model (blue dashed lines) and the numerical data (black circles). At EVP,  $\psi$  values at  $z = 0$  cross section from (b) numerical data and (c) PI model are depicted. APE values of (d) DD and (e) PI model are plotted.



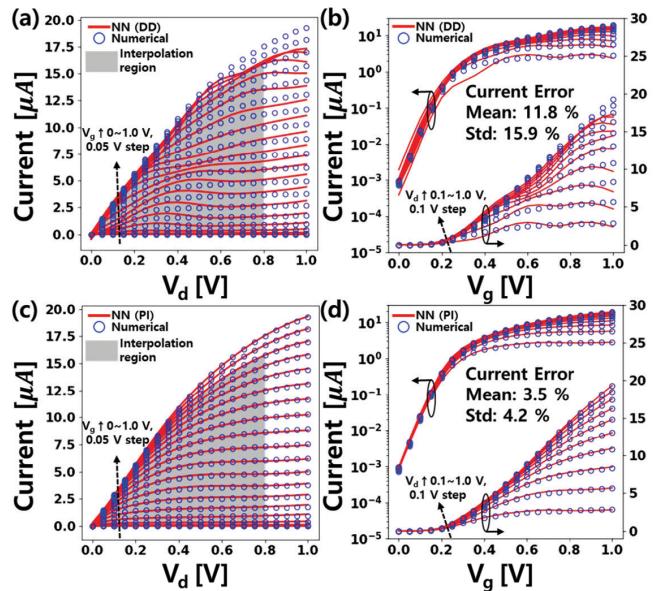
**Fig. 9.** Drain currents predicted by the BTE-NET are represented by red lines for (a) DD model and (b) PI model, which are compared with the numerical data (circles). Interpolation region is filled with a gray color. The data in the training set (green circles) and the test set (blue empty circles) are separately plotted to clearly show the effects of a PDE regularizer. The Boltzmann distribution function at  $z = 0$  for the PI model (red dashed line), the DD model (green dashed line), and numerical data (blue line) at (c) IVP and (d) EVP.



**Fig. 10.** Three-dimensional charge density predicted by our NN device model at EVP. (a) Charge density profile and (b) its APE.

interpolating within the known data range and extrapolating beyond it.

As shown in Fig. 10(a) and (b), our model can also precisely forecast quantum mechanical charge density, with an MAPE of only 2.06% and 2.60% at the IVP and EVP, respectively. The maximum APEs are 33.2% and 27.6% at the IVP and EVP, respectively. This performance is much superior to a previous



**Fig. 11.** Drain currents predicted by our NN device model. (a)  $I_d$ - $V_d$  curves and (b)  $I_d$ - $V_g$  curves predicted by NN device model trained in the DD approach (red lines) are plotted with numerical data (blue empty circles). (c)  $I_d$ - $V_d$  curves and (d)  $I_d$ - $V_g$  curves in the PI approach (red lines) are plotted with numerical data (blue empty circles) (red lines). Interpolation region is filled with a gray color in (a) and (c).

research that showed an MAPE of 5.73% (maximum 71%) for the interpolation of charge densities under different gate bias conditions [2]. At EVP, the largest APE is observed at the semiconductor/oxide interface, where the charge density is negligibly small.

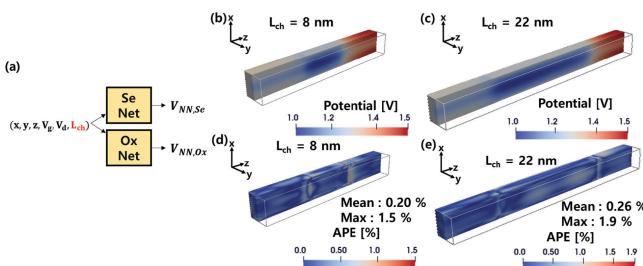
## B. Prediction of Current

Although our PIML-based NN of in Fig. 2 which is a serial connection of SCP, SE, and BTE-NET is not trained with the drain current data, it has a power to predict the current. If we train our NN device model without PI, currents are predicted poorly with an MAPE of 11.8% [Fig. 11(a) and (b)]. However, as shown in Fig. 11(c) and (d), the values predicted by PIML-based NN agree well with numerical data, with the MAPE of 3.5%. Unlike the approach described in [10], which uses 2000 TCAD data points to predict 50 samples, our model leverages only 25 training bias conditions to accurately predict 441 bias conditions. This highlights the remarkable improvement in data efficiency achieved through the use of PIML. Note that in the previous study, the NN was directly trained to learn current using the bias-current dataset, while only the spatial physical quantity data are used in the NN's learning process in our model. While our current work has primarily focused on predicting current, there is room for improvement in accurately predicting other device performance metrics, such as conductance [5], [6].

Our model only requires 15 ms to predict the spatial physical quantities and current for each bias condition. In Table II, we compared the numerical simulation results with those obtained from our NN device model, in the same manner as presented in [2]. Numerical simulation is conducted using Intel Xeon Gold 6126 CPU at 2.6 GHz (24 core). NN training and inference is performed using NVIDIA GeForce RTX 3090 GPU. Predicting the drain currents at 441 bias conditions shown in Fig. 11(c) takes 6.6 s using our model while the numerical simulation requires 96 300 s, indicating

**TABLE II**  
COMPARISON OF OUR PIML-BASED NN DEVICE  
MODEL WITH THE TRADITIONAL METHOD

Numerical simulation of 441 bias conditions	NN training with 25 bias conditions	Prediction of physical quantities under 441 bias conditions
96,300s	13,440s	6.6s



**Fig. 12.** (a) Modification of the SCP-NET by increasing the input dimension of NN. 3-D potential profile predicted at  $L_{ch}$  of (b) 8 nm and (c) 22 nm. APE of the SCP-NET  $L_{ch}$  of (d) 8 nm and (e) 22 nm.  $V_g$  and  $V_d$  were fixed to 0.7 and 0.4 V, respectively.

that our model exhibits  $1.46 \times 10^4$  times faster prediction speed.

### C. Geometry Dependence

To assess the validity of our model in handling geometric variations, we expanded the input dimension of the SCP-NET as illustrated in Fig. 12(a). To handle varying channel lengths, real space coordinates are approximately normalized. The SCP-NET was trained using potential data for 25 bias conditions shown in Fig. 1(e), while systematically varying the channel length ( $L_{ch}$ ) between 10, 15, and 20 nm. We then evaluated the extrapolation accuracy of the SCP-NET when further varying  $L_{ch}$  to 8 and 22 nm. Remarkably, our SCP-NET consistently provided accurate predictions of potential values, despite being trained with only three  $L_{ch}$  conditions. In comparison to the previous work that reported MAPE and maximum APE values of 0.73% and 15.5%, respectively, our model exhibited superior performance with MAPE and maximum APE values of 0.20% and 1.5%, respectively, for predicting potential in devices with shorter  $L_{ch}$  than the training conditions [2]. Notably, the previous work trained the NN by varying  $L_{ch}$  from 24 to 30 nm with a 2-nm spacing, while our NN was trained using  $L_{ch}$  ranging from 10 to 20 nm with a 5-nm spacing.

Although we have only tested the SCP-NET for geometry variation in this study, we intend to extend our model to predict multiple physical quantities for various geometries in future studies.

### IV. CONCLUSION

In this work, the state-of-the-art PIML techniques of PINN and PI-DeepONet are used to construct an NN device model that can predict device performance while being physically interpretable. As the governing equations (PE, SE, BTE) are directly referenced to provide the correct physics, our PIML-based modeling offers a good strategy to boost interpolation and extrapolation accuracy for predicting spatial physical quantities. Although our model is trained with a small number of bias conditions, it exhibits high extrapolation accuracy; an MAPE of 0.19% and 2.6% for predicting potential and charge density, respectively. We have also demonstrated

that as an initial step toward full device optimization, the input dimension to the NN can be expanded to include the channel length. We believe our model can be used for device optimization by enabling data-efficient NN modeling and real-time physics analysis for semiconductor devices.

### ACKNOWLEDGMENT

The authors would like to thank Dr. Jae Yong Lee for providing valuable discussions about physics-informed machine learning.

### REFERENCES

- [1] S.-C. Han, J. Choi, and S.-M. Hong, "Acceleration of semiconductor device simulation with approximate solutions predicted by trained neural networks," *IEEE Trans. Electron Devices*, vol. 68, no. 11, pp. 5483–5489, Nov. 2021, doi: [10.1109/TED.2021.3075192](https://doi.org/10.1109/TED.2021.3075192).
- [2] W.-J. Lee, W.-T. Hsieh, B.-H. Fang, K.-H. Kao, and N.-Y. Chen, "Device simulations with a U-Net model predicting physical quantities in two-dimensional landscapes," *Sci. Rep.*, vol. 13, no. 1, p. 731, Jan. 2023, doi: [10.1038/s41598-023-27599-z](https://doi.org/10.1038/s41598-023-27599-z).
- [3] M.-Y. Kao, H. Kam, and C. Hu, "Deep-learning-assisted physics-driven MOSFET current-voltage modeling," *IEEE Electron Device Lett.*, vol. 43, no. 6, pp. 974–977, Jun. 2022, doi: [10.1109/LED.2022.3168243](https://doi.org/10.1109/LED.2022.3168243).
- [4] 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, doi: [10.1109/TED.2020.3048918](https://doi.org/10.1109/TED.2020.3048918).
- [5] C.-T. Tung, M.-Y. Kao, and C. Hu, "Neural network-based and modeling with high accuracy and potential model speed," *IEEE Trans. Electron Devices*, vol. 69, no. 11, pp. 6476–6479, Nov. 2022, doi: [10.1109/TED.2022.3208514](https://doi.org/10.1109/TED.2022.3208514).
- [6] C.-T. Tung and C. Hu, "Neural network-based BSIM transistor model framework: Currents, charges, variability, and circuit simulation," *IEEE Trans. Electron Devices*, vol. 70, no. 4, pp. 2157–2160, Apr. 2023, doi: [10.1109/TED.2023.3244901](https://doi.org/10.1109/TED.2023.3244901).
- [7] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Rev. Phys.*, vol. 3, no. 6, pp. 422–440, May 2021, doi: [10.1038/s42254-021-00314-5](https://doi.org/10.1038/s42254-021-00314-5).
- [8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019, doi: [10.1016/j.jcp.2018.10.045](https://doi.org/10.1016/j.jcp.2018.10.045).
- [9] S. Wang, H. Wang, and P. Perdikaris, "Learning the solution operator of parametric partial differential equations with physics-informed DeepONets," *Sci. Adv.*, vol. 7, no. 40, Oct. 2021, Art. no. eabi8605, doi: [10.1126/sciadv.abi8605](https://doi.org/10.1126/sciadv.abi8605).
- [10] S. Myung, W. Jang, S. Jin, J. M. Choe, C. Jeong, and D. S. Kim, "Restructuring TCAD system: Teaching traditional TCAD new tricks," in *IEDM Tech. Dig.*, Dec. 2021, pp. 18.2.1–18.2.4, doi: [10.1109/IEDM19574.2021.9720616](https://doi.org/10.1109/IEDM19574.2021.9720616).
- [11] S. Jin, T.-W. Tang, and M. V. Fischetti, "Simulation of silicon nanowire transistors using Boltzmann transport equation under relaxation time approximation," *IEEE Trans. Electron Devices*, vol. 55, no. 3, pp. 727–736, 2008, doi: [10.1109/TED.2007.913560](https://doi.org/10.1109/TED.2007.913560).
- [12] S. Scaldaferri, G. Curatola, and G. Iannaccone, "Direct solution of the Boltzmann transport equation and Poisson-Schrödinger equation for nanoscale MOSFETs," *IEEE Trans. Electron Devices*, vol. 54, no. 11, pp. 2901–2909, Nov. 2007, doi: [10.1109/TED.2007.906927](https://doi.org/10.1109/TED.2007.906927).
- [13] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," *J. Comput. Phys.*, vol. 404, Mar. 2020, Art. no. 109136, doi: [10.1016/j.jcp.2019.109136](https://doi.org/10.1016/j.jcp.2019.109136).
- [14] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.
- [15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [16] Google. (2018). *JAX: Composable Transformations of Python+Numpy Programs*. [Online]. Available: <https://github.com/google/jax>
- [17] A. D. Jagtap and G. E. Karniadakis, "Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations," *Commun. Comput. Phys.*, vol. 28, no. 5, pp. 2002–2041, Jun. 2020, doi: [10.4208/cicp.2020-0164](https://doi.org/10.4208/cicp.2020-0164).