Convolutional Machine Learning Method for Accelerating Nonequilibrium Green's Function Simulations in Nanosheet Transistor

Preslav Aleksandrov, Ali Rezaei, Tapas Dutta, Nikolas Xeni, Asen Asenov, *Fellow, IEEE*, and Vihar Georgiev, *Senior Member, IEEE*

***Abstract*— This work describes a novel simulation approach that combines machine learning (ML) and device modeling simulations. The device simulations are based on the quantum mechanical nonequilibrium Green's function (NEGF) approach, and the ML method is an extension of a convolutional generative network. We have named our new simulation approach ML-NEGF. It is implemented in our in-house simulator called Nano-Electronics Simulation Software (NESS). The reported results demonstrate the improved convergence speed of the ML-NEGF method in comparison to the "standard" NEGF approach. The trained ML model effectively learns the underlying physics of nanosheet transistor behavior, resulting in faster convergence of the coupled Poisson-NEGF self-consistency simulations. Quantitatively, our ML-NEGF approach achieves an average convergence speedup of 60%, substantially reducing the computational time while maintaining the same accuracy.**

***Index Terms*— Autoencoder (AE), convergence acceleration, nano-sheet transistors, nonequilibrium Green's function (NEGF), quantum transport, silicon (Si) nanowire.**

I. Introduction

THE silicon (Si) nanowire and nanosheet transistors have a wide spectrum of promising applications [1], such as current field-effect transistors [2] and photovoltaics [3]. Moreover, the state-of-the-art CMOS technologies are based on single or stacked configurations of nanosheet or nanowire architectures [4]. Despite the recent advances in fabrication technology, technology computer-aided design (TCAD) simulations remain the main method through which the design of these devices is performed. To achieve useful results in the ever-shrinking size domain, more complex simulation methods, such as the nonequilibrium Green's function (NEGF) method, are being adopted by the industry. However, major bottlenecks for the wide adoption of these more complex methods are the increase in simulation time and the difficulty of achieving convergence. Therefore, this article aims to tackle these challenges by introducing a machine learning (ML) model that can predict device performance. The proposed method aims to decrease computation time and improve convergence without sacrificing the physical accuracy and device output characteristics.

The main aim of this work is to investigate the possibility of significantly improving or even replacing numerical TCAD device simulations with a convolutional autoencoder (CAE) [5], [6], [7]. To test our idea, we have developed a new simulation approach based on the combination of TCAD and ML. The current state of the art of TCAD simulations are based on the NEGF formalism, which can capture quantum mechanical effects, such as confinement and carrier tunneling in ultrascaled transistors (with channel lengths that are shorter than 10 nm) [8]. To enhance the capabilities of the NEGF method and decrease the computational time of our in-house Nano-Electronics Simulation Software (NESS) [9], [10] we combine ML with our existing effective-mass NEGF simulator implemented in NESS. We have called the new simulation approach ML-NEGF.

Our results show the potential of using the ML-NEGF methodology to significantly reduce the computational cost of device simulations without compromising the accuracy of physical results obtained from "standard TCAD" simulations.

II. Device Structure

To test our new simulation methodology, ML-NEGF, we have designed a device transistor structure that corresponds to the most advanced technologies of the 3 nm node and beyond. Fig. 1 shows the nanosheet transistor geometry created using the NESS structure generator. The gate is all around the channel, and the channel length is 16 nm, with source and drain lengths of 3 nm each. Hence, the total length of the device is 22 nm. The channel cross section is rectangular with dimensions of 3×12 nm, the oxide material is SiO_2, and its thickness is 1 nm. The channel doping is 1×10^{15} cm^{-3}, and the source and drain regions have a doping of 1×10^{20} cm^{-3}.

Manuscript received 11 July 2023; revised 6 August 2023; accepted 14 August 2023. Date of publication 28 August 2023; date of current version 22 September 2023. This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S001131/1 and Grant EP/P009972/1 and in part by the EPSRC Impact Acceleration Account Scheme under Grant EP/R511705/1. The review of this article was arranged by Editor T. Grasser. *(Corresponding author: Vihar Georgiev.)*

The authors are with the Device Modelling Group, James Watt School of Engineering, University of Glasgow, G12 8LT Glasgow, U.K. (e-mail: vihar.georgiev@glasgow.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TED.2023.3306319.

Digital Object Identifier 10.1109/TED.2023.3306319

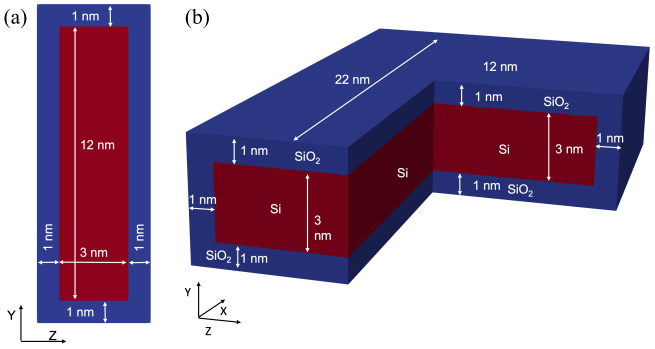

Fig. 1. Simulated device structures of the n-type Si nanosheet transistor used in this work. (a) Channel cross section is 3 × 12 nm and (b) length of the full device is 22 nm. The gate length is 16 nm, and the source and drain have 3 nm lengths each. The channel doping is 1×10^{16} cm^{-3} and the contacts (source and drain) are 1×10^{20} cm^{-3} doped. The oxide is SiO_2 with a thickness of 1 nm everywhere around the device.

III. Model Development and Simulations Methodology

All numerical simulations in the work are performed by utilizing the NEGF simulator implemented in our in-house code NESS. The NEGF implementation is based on the effective mass approximation (EMA). Our NEGF solver can compute ballistic and scattering transport in various devices. In this article, we have used the ballistic version of the NEGF solver to test our ML model. However, we would like to emphasize that our methodology is valid even if we use simulations that include the electron-phonon and surface roughness scattering mechanisms in the active region of the device.

We use an effective mass Hamiltonian, which is discretized using the finite difference approximation within the simulation domain, resulting in a tri-diagonal representation for the one-particle Hamiltonian $h(x)$ [16]. The retarded Green's function in the active region device-contact-interface and the corresponding self-energies can be further determined by employing the Dyson equation [17]. The retarded G^R, advanced G^A, and lesser/greater Green's function $G^{\lessgtr}$ may be expressed as

$$G^R(x,E) = \left[(E+i\eta)\cdot \mathbb{1} - h(x) - \Sigma^R(x,E)\right]^{-1} \quad (1)$$

$$G^A(x,E) = \left[G^R(x,E)\right]^{\dagger} \quad (2)$$

$$G^{\lessgtr}(x,E) = G^R(x,E)\cdot\Sigma^{\lessgtr}(x,E)\cdot G^A(x,E) \quad (3)$$

where E, $\mathbb{1}$, η, and Σ^R ($\Sigma^{\lessgtr}$) are the energy, identity matrix, an infinitesimal positive real number, and retarded (lesser/greater) self-energies, respectively. Here, for the ballistic case, these self-energies merely present the electron interactions with the contacts. However, in general, they can also take into account electron scattering ($\Sigma = \Sigma_{\mathbf{lead}} + \Sigma_{\mathbf{scattering}}$). Employing the retarded Green's function component, we can obtain the charge at position x

$$n(x) = \frac{-i}{2\pi}\int dE\, G^{<}(x,x';E) \quad (4)$$

and the current through the lth layer

$$j(l) = \frac{2|q|}{\hbar}\int \frac{dE}{2\pi}\mathrm{Tr}\left[2\Re\left(h_{l+1,l}\cdot G^{<}_{l,l+1}\right)\right] \quad (5)$$

where $h_{l+1,l}$ ($G^{<}_{l,l+1}$) are the matrix elements of the Hamiltonian ($G^{<}$) between the basis states on layers $l+1$ (l) and l ($l+1$).

To perform simulations in coupled-mode space using the EMA NEGF, we first solve the 2-D Schrödinger equation at each slice of the device to find the eigenvalues and eigenfunctions of the confinement cross section. These slices are then linked along the transport direction, and the carrier transport is estimated using a 1-D NEGF, using the recursive Green's function approach. Moreover, the NEGF solver and the 3-D nonlinear Poisson solver are linked in a self-consistent loop. The Poisson solver provides potential as an input to the effective mass Hamiltonian and, thus, the NEGF solver. In a similar way, the Poisson solver requires charge as input for quasi-Fermi level calculation, which is provided by the NEGF solver.

The NEGF formalism allows us to compute device characteristics, such as transfer and output current–voltage curves (I_D–V_G and I_D–V_D). From the I_D–V_G curve, we can extract important figures of merit (FoMs), such as OFF-current (I_{OFF}) and ON-current (I_{ON}), subthreshold slope (SS) and voltage threshold (V_{T}). In previous papers, we have shown that it is possible to train a neural network (NN) by using as input the key FoMs, such as SS, drive current, and leakage current to predict other key parameters, such as threshold voltage [15].

In this work, we investigate the interaction of our NEGF simulator with a novel ML model to accelerate the convergence of simulations. The main concept behind our approach is to use 3-D charge and potential distribution that are generated by our ML method as an initialization point for the NEGF-Poisson self-consistency loop (SCL), as shown in Fig. 2. The ML model can either produce a 3-D charge profile, which the Poisson solver uses, or a 3-D potential distribution, which the NEGF solver uses, depending on the configuration. The aim of the ML model is to learn and then predict the change in charge and potential between the first and last steps in the Poisson-NEGF SCL. Our hypothesis is that a device's charge and potential distributions change in a systematic way between the first and the last steps in the SCL. Our approach can be described as a similar method used in face age recognition or the removal of the noise from images [18], [19]. If our hypothesis is correct, an auto-encoder (AE) [11] NN architecture, and more specifically a subtype known as a denoising AE (DAE) [12], [13], [14], can be used to learn the systematic changes in the charge and the potential. Hence, such DAE NN will be able to predict main device characteristics, such as charge and potential distributions, but it can also be adapted to predict other parameters.

For the purpose of the discussions in this article, we introduce two terms: "classical" NEGF and ML-NEGF simulations. The former term corresponds to simulations where we executed only Poisson-NEGF SCL simulations, and the latter term is a combination of DAE NN with NEGF simulations.

In this section, we will discuss in detail our model development approach. Model architecture is based on a convolutional DAE network augmented by methods stemming from transformer networks. The basic structure was chosen

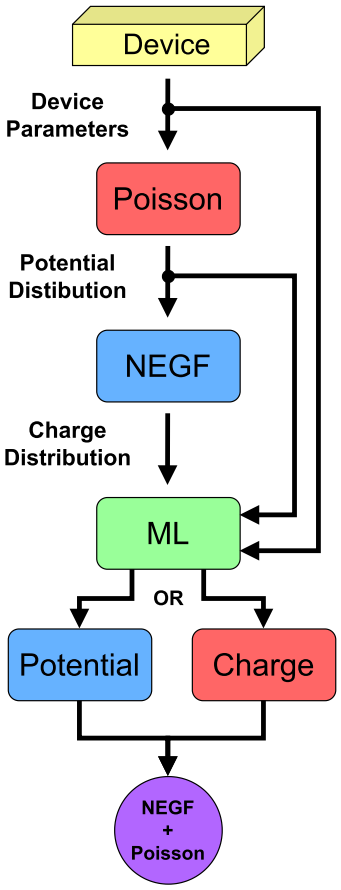

Fig. 2. Schematic representation of the simulation flow in the ML-NEGF method. The simulation commences with a conventional device structure and parameters, which are utilized to determine the initial potential distribution employing a Poisson solver (indicated in red). The resulting potential distribution serves as input for the NEGF solver, enabling the determination of the charge distribution. The charge distribution, potential distribution, and other device parameters are subsequently input into a trained ML model. The model's output, normalized with respect to the inputted, can represent either a charge or a potential distribution, depending on the desired training objective. The output is then de-normalized and reintroduced into the self-consistent Poisson-NEGF loop.

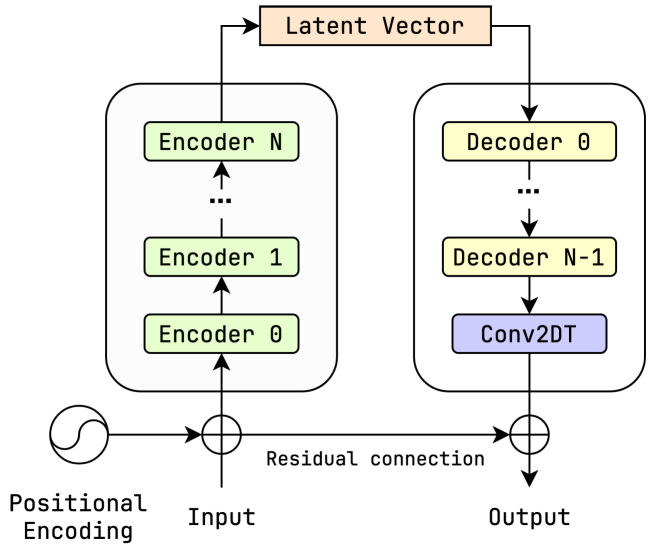

Fig. 3. Architecture of the ML-NEGF model. The model consists of two divisions: encoding and decoding. The encoding section operates on a nine-channel image, which contains information from an initial Poisson-NEGF loop. The residual connection is used to add the initial condition (potential or charge) to the output. In this way, the model learns to predict change rather than magnitudes. The residual connection is, the only way to implement this without having access to the target data.

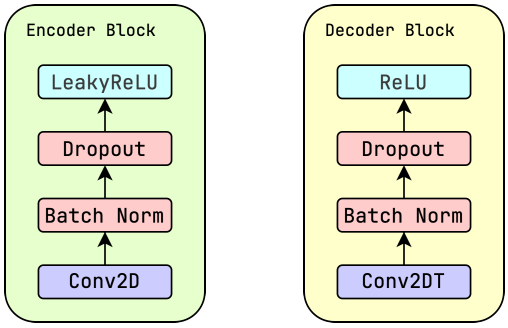

Fig. 4. Encoder and decoder blocks consist of four sections. The sections in red are active during the training process alone. During inference, batch norm is used, but its standardization parameters are not updated. Conv2D and Conv2DT are the dimensional convolution and its transpose, respectively. Our research finds the reconstruction stage suffers less from diminishing gradients, our research finds. Therefore, the ReLu activation has been used.

to be fully convolutional, as this guarantees model generality and improves robustness. The augmentations, borrowed from transformer networks, are the inclusion of location encoding in the initial input and the residual connection between the input and output of the model. The full model diagram is shown in Fig. 3.

The model can be constructed using N number of encoder and decoder blocks to extract a latent space representation of the input and apply relevant transformations in the decoder section. Each block consists of a convolution, batch normalization, dropout, and activation function. Leaky Relu was chosen as the activation function in the encoder section as it has a high gradient. The choice of this function was a compromise because the depth of the network was relatively low ($N = 3$) to justify the use of residual connections within the network. Therefore, to compensate for diminishing gradients, a high-gradient activation function was chosen. The structures of the encoder and decoder blocks are shown in Fig. 4.

In our approach, we introduced location encodings. They are used to point out the location of the kernel in the model. This was done after preliminary tests revealed that systematic similarities between the first and last Poisson-NEGF iterations exist in different regions of the device. The main reason for such systematic similarities comes from the values of the charge in the middle of the device and in the source and drain regions. For example, after solving the "classical" Poisson-NEGF simulations, the charge magnitude in the source and drain regions is five to six orders of magnitude higher than that in the middle of the device (the channel). Such a severe difference was not present in the initial iteration. Therefore, to allow the model to distinguish between the different device regions, location encodings were introduced.

The inclusion of location mappings aids the model in learning multiple transformations. The location encoding values were artificially generated, making a gradient map $M \in [0, 1]$ in each of the basis directions (X, Y, Z). The

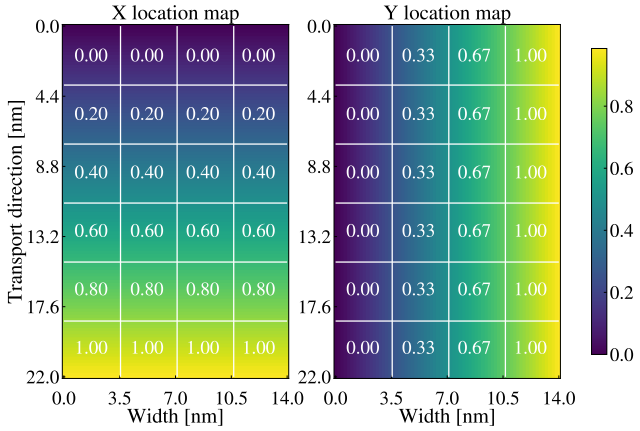

Fig. 5. Location maps in X- and Y-directions. X is the transport direction with a length of 22 nm and Y is the longer cross section with a length of 14 nm (including oxide). The color map shows the location of the kernel in relation to the model. Numbers and lines show the rough splitting of the data.

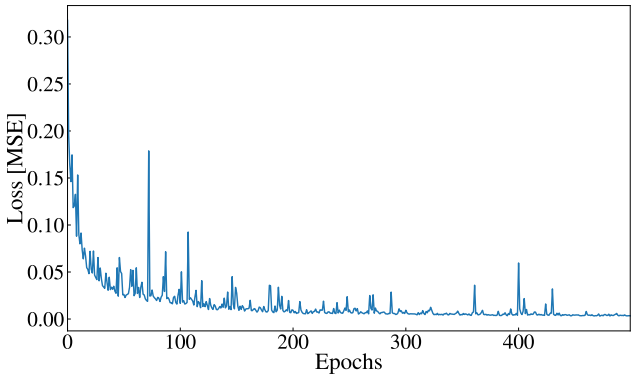

Fig. 6. Evolution of the mse as a function of the epochs (training steps). This is also known as a validation loss function. The loss function experiences oscillations that are characteristic the mini-batch gradient descent in Adam [20].

range $[0, 1]$ was chosen in order to maintain the generality and transferability of the DAE's knowledge to other device structures. A sample of two of the maps is shown in Fig. 5.

The fully trained DAE model can be examined as a kernel-based analytical representation of the "classical" NEGF solver, the solution of which is the forward pass of the ML model. Computationally, the cost of this is negligible compared to the cost of utilizing the NEGF solver. For example, the computation cost of the forward ML pass is orders of magnitude quicker than a single "classical" NEGF iteration. Therefore, if the number of iterations is reduced by 60% the corresponding reduction in computational time will also be 60%. Thus, this method, once trained, is computationally efficient and it can be used to accelerate "classical" NEGF simulations.

Model training is performed with the help of the "classical" simulator. We use the initial and final iterations as the input and target for ML training. Once the model is trained, we follow a similar procedure for inference. We obtain an initial iteration, use it for the ML input and normalization and later use the ML output instead of the initial iteration.

IV. Results and Discussions

In order to validate our ML-NEGF model, input and target data were generated by our "standard" NEGF. The data was divided into two sets. The training set encompassed 70% of the total data available; the remaining 30% was used as a testing set. The training set is used to train the ML model and the relevant loss is shown in Fig. 6. The output of the DAE consists of a single-channel image that represents a standardized 2-D distribution of the potential or charge fields. The output is standardized with respect to the mean and deviation of the input.

In order to validate the training process, we ran the training loop until the mean square error (mse) loss reached saturation, or, in other words, the point of training at which the model had extracted all the available knowledge. The loss function characteristic (mse versus epochs) can be considered to have reached saturation when the slope of the curve is close to zero.

Fig. 6 shows the evolution of the mse as a function of the epochs (training steps). The model was trained for 500 epochs. The number of 500 epochs was discovered empirically. However, any number of epochs is sufficient as long as saturation is achieved. The reason for this conclusion is that the training loss determines the change to the model parameters. When the loss reduction decelerates, the model parameters have converged to an optimal configuration. There is a potentially infinite number of optimal points, and the final loss achieved varies depending on the device type, size of the device, and initial parameters of the ML model. The loss graph shows significant oscillation and an exponential decrease in around 100 epochs, followed by a reduction in training speed between 100 and 400 epochs. After 400 epochs, the mse value is saturated (0.02 mse). Indeed, such saturation of the mse value proves that the NN is well trained.

Figs. 7 and 8 are showing a comparison between model the output and the target scalar fields for potential and charge in the middle of the channel, correspondingly. The top row compares the potential distribution of a slice along the transport direction. In this case, the transport direction is along the X-axis (vertical axes in the figure), and the Y-axis represents the height of the device (horizontal axes in the figure). The bottom row shows the 2-D potential distribution in the middle of the channel. In this case, the Y-axis (vertical axes in the figure) is the height of the device and the X-axis (horizontal axes in the figure) is the width of the device. The output from the DAE is labeled as ML-NEGF, and it is always in the left column. The output of the "standard" NEGF is labeled as NEGF and is in the right column.

Fig. 7 shows the comparison between the 2-D potential distribution in the middle of the channel, along the transport direction, for ML-NEGF simulations and the target scalar fields produced by the "standard" NEGF method. Consistent with the device structure and the doping profile along the device, the value of the potential is the lowest (-0.2 V) in the source and drain regions and the highest (0.3 V) in the middle of the channel of the device. The gate bias is set to 0.7 V. This is consistent with the device physics, where the potential has the highest value in the middle of the channel.

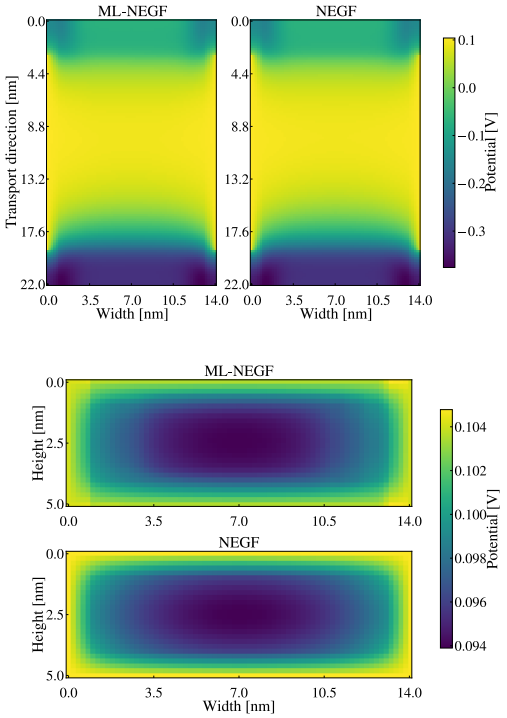

Fig. 7. (Top row) Comparison of potential distribution along the *XY* plane along the transport direction between the ML-NEGF and "standard" NEGF simulations. The potential has the highest value in the source and drain regions and the highest potential in the middle of the channel. (Bottom row) The same comparison of the potential but in a *YZ* plane that is perpendicular to the transport direction, between the ML-NEGF and "standard" NEGF simulations. Consistently with the physical picture, the potential distribution is localized in the middle of the channel.

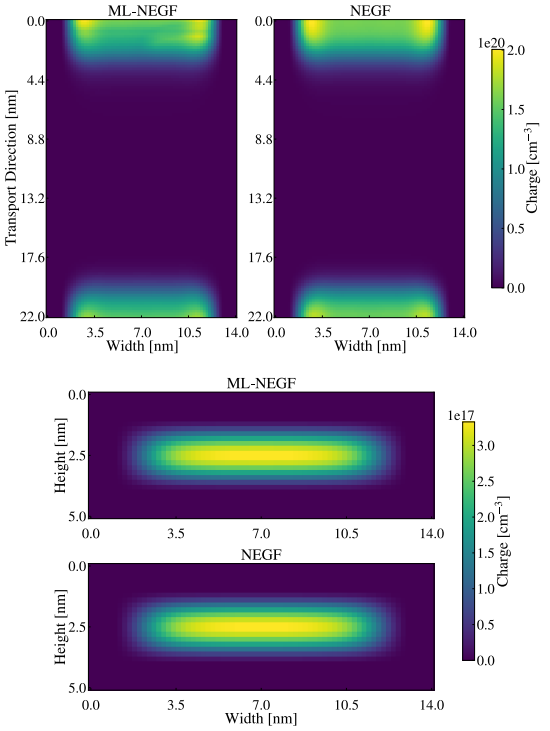

Fig. 8. (Top row) Comparison of charge distribution along the *XY* plane along the transport direction between the ML-NEGF and "standard" NEGF simulations. The charge has the highest value in the source and drain regions and the highest potential in the middle of the channel. (Bottom row) The same comparison of the charge but in a *YZ* plane that is perpendicular to the transport direction, between the ML-NEGF and "standard" NEGF simulations. Consistently with the physical picture, the charge distribution is localized in the middle of the channel.

Fig. 8 shows the comparison between the 2-D charge distribution in the middle of the channel, along the transport direction (X-axis), for the ML-NEGF simulations and the target scalar fields produced by the "standard" NEGF method. Consistent with the device structure and the doping profile along the device, the value of the charge is the highest in the source and drain regions and the lowest in the middle of the channel of the device. The gate bias is set to 0.7 V.

From the results in Figs. 7 and 8, it can be concluded that the charge and the potential distribution obtained from the ML-NEGF method are identical to those extracted from the NEGF. Hence, it can be concluded that our DAE is indeed well trained.

Once the ML model was trained, we wanted to evaluate and compare the convergence behavior in both cases. Fig. 9 shows the number of self-consistent interactions as a function of gate voltage (V_G), at low ($V_D = 0.05$ V) and high ($V_D = 0.7$ V) drain bias. From the data in Fig. 9, it can be concluded that overall, the ML-NEGF method requires a smaller number of iterations in comparison to the NEGF method. Specifically, at V_G up to 0.2 V, both methods have almost identical iterations. However, when the V_G has values above 0.2 V, the ML-NGEF simulations (see the red and orange curves in Fig. 9) show a consistently lower iteration number than the "standard" NEGF method. For example, the difference between both methods is very well pronounced at $V_G = 0.8$ V. At low drain bias ($V_D = 0.05$ V) ML-NEGF (orange curve)

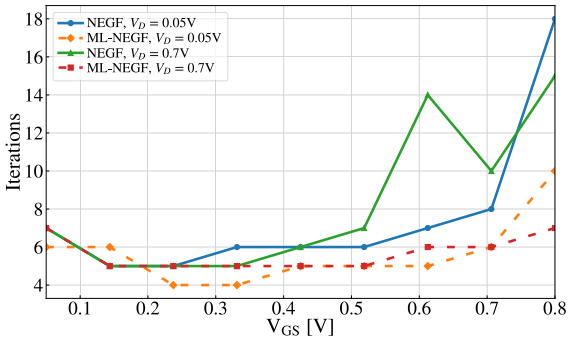

Fig. 9. Comparison of the simulations of self-consistent iterations between ML-NEGF and NEGF (ballistic), as a function of the gate bias (V_G).

converges in ten iterations, while the conventional NEGF method (blue curve) needs 18 iterations. At high ($V_D = 0.7$ V) drain bias, ML-NEGF (red curve) requires seven steps, and the NGEF method (green curve) converges after 15 steps. Hence, in both cases, the ML-NEGF approach achieves an average convergence speedup of 60%, substantially reducing the computational time while maintaining the same accuracy.

In Fig. 10, we have plotted and compared the I_D–V_G curves for both ML-NEGF and "standard" NEGF approach at low ($V_D = 0.05$ V) and high ($V_D = 0.7$ V) drain bias. From the result, in Fig. 10, it is evident that both methods produced identical I_D–V_G curves, hence, transistor's behavior and it is also identical. The results presented in Fig. 10 show that our

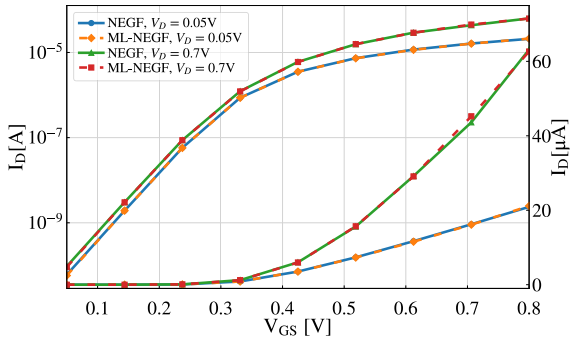

Fig. 10. Comparison of the current–voltage characteristic (I_D–V_G curves) for both the ML-NEGF and NEGF methods as a function of the gate bias (V_G) at fixed drain bias (low = 0.05 V and high = 0.7 V).

DEA NN used in the ML-NEGF method can reproduce not only physical properties but also key device characteristics.

V. Conclusion

In this work, we have reported a combined ML and device simulation computational approach that allows us to simulate the device characteristics (current–voltage) of Si-based nanosheet transistors. Our ML method is based on a convolutional NN and an AE architecture. Results obtained from the ML-NEGF approach lead to the following conclusions.

Firstly, using the AE-accelerated ML-NEGF method instead of the standard TCAD (NEGF) simulations could, in principle, significantly decrease the computational time and shorten the research and development process. For example, the ML-NEGF approach achieves an average convergence speedup of 60%, while maintaining the same accuracy. Secondly, our AE-accelerated ML-NEGF method can reproduce not only the device characteristics but also the 3-D charge density and potential distribution in the whole device. Lastly, a similar ML-based approach can be used to describe material properties, such as resistance in metal nanowires, that cannot be described by nonparametric methods, such as a general linear model. However, it needs to be noted that the predictive power of the ML-NEGF method can be improved even further by providing more data, using different preprocessing schemes, and attempting alternative network architectures. Indeed, all these options are currently under investigation.

In summary, the integration of ML methods (DAE) with the quantum mechanical method (NEGF) presents a promising pathway for efficiently simulating nano-sheet transistors. The combination of accurate predictions and reduced computational cost achieved through ML-based acceleration offers a valuable tool for researchers and engineers in the field of nanoelectronics.

References

[1] J. Appenzeller, J. Knoch, M. T. Bjork, H. Riel, H. Schmid, and W. Riess, "Toward nanowire electronics," *IEEE Trans. Electron Devices*, vol. 55, no. 11, pp. 2827–2845, Nov. 2008, doi: 10.1109/TED.2008.2008011.

[2] V. P. Georgiev et al., "Experimental and simulation study of silicon nanowire transistors using heavily doped channels," *IEEE Trans. Nanotechnol.*, vol. 16, no. 5, pp. 727–735, Sep. 2017, doi: 10.1109/TNANO.2017.2665691.

[3] B. Tian et al., "Coaxial silicon nanowires as solar cells and nanoelectronic power sources," *Nature*, vol. 449, no. 7164, pp. 885–889, Oct. 2007, doi: 10.1038/nature06181.

[4] A. Asenov et al., "Nanowire transistor solutions for 5 nm and beyond," in *Proc. 17th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2016, pp. 269–274.

[5] S. Woo, J. Jeon, and S. Kim, "Prediction of device characteristics of feedback field-effect transistors using TCAD-augmented machine learning," *Micromachines*, vol. 14, no. 3, p. 504, 2023, doi: 10.3390/mi14030504.

[6] R. Singh, A. Gehlot, P. S. Ranjit, and D. Sharma, *Futuristic Sustainable Energy and Technology: Proceedings of the International Conference on Futuristic Sustainable Energy & Technology (ICFSE, 2021)*. Boca Raton, FL, USA: CRC Press, Sep. 2021, pp. 19–20.

[7] H.-C. Ruiz Euler et al., "A deep-learning approach to realizing functionality in nanoelectronic devices," *Nature Nanotechnol.*, vol. 15, no. 12, pp. 992–998, Dec. 2020, doi: 10.1038/s41565-020-00779-y.

[8] V. P. Georgiev, E. A. Towie, and A. Asenov, "Impact of precisely positioned dopants on the performance of an ultimate silicon nanowire transistor: A full three-dimensional NEGF simulation study," *IEEE Trans. Electron Devices*, vol. 60, no. 3, pp. 965–971, Mar. 2013, doi: 10.1109/TED.2013.2238944.

[9] S. Berrada et al., "Nano-electronic simulation software (NESS): A flexible nano-device simulation platform," *J. Comput. Electron.*, vol. 19, no. 3, pp. 1031–1046, Sep. 2020, doi: 10.1007/s10825-020-01519-0.

[10] D. Nagy et al., "Hierarchical simulation of nanosheet field effect transistor: NESS flow," *Solid-State Electron.*, vol. 199, Jan. 2023, Art. no. 108489, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003811012200260X, doi: 10.1016/j.sse.2022.108489.

[11] E. F. Franco et al., "Performance comparison of deep learning autoencoders for cancer subtype detection using multi-omics data," *Cancers*, vol. 13, no. 9, p. 2013, Apr. 2021, doi: 10.3390/cancers13092013.

[12] W. Alvarado et al., "Denoising autoencoder trained on simulation-derived structures for noise reduction in chromatin scanning transmission electron microscopy," *ACS Central Sci.*, vol. 9, no. 6, pp. 1200–1212, Jun. 2023, doi: 10.1021/acscentsci.3c00178.

[13] K. J. Geras and C. Sutton, "Composite denoising autoencoders," in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science), P. Frasconi, N. Landwehr, G. Manco, and J. Vreeken, Eds. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-46128-1_43.

[14] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, New York, NY, USA, 2008, pp. 1096–1103.

[15] H. Carrillo-Nuñez, N. Dimitrova, A. Asenov, and V. Georgiev, "Machine learning approach for predicting the effect of statistical variability in Si junctionless nanowire transistors," *IEEE Electron Device Lett.*, vol. 40, no. 9, pp. 1366–1369, Sep. 2019, doi: 10.1109/LED.2019.2931839.

[16] G. Stefanucci and R. Van Leeuwen, *Nonequilibrium Many-Body Theory of Quantum Systems: A Modern Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 2013, doi: 10.1017/CBO9781139023979.

[17] S. Datta, "Nanoscale device modeling: The green's function method," *Superlattices Microstructures*, vol. 28, no. 4, pp. 253–278, Oct. 2000, doi: 10.1006/spmi.2000.0920.

[18] L. Boussaad and A. Boucetta, "Deep-learning based descriptors in application to aging problem in face recognition," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2975–2981, Jun. 2022, doi: 10.1016/j.jksuci.2020.10.002.

[19] L. Fan et al., "Brief review of image denoising techniques," *Vis. Comput. Ind. Biomed. Art*, vol. 2, p. 7, Jul. 2019, doi: 10.1186/s42492-019-0016-7.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015, *arXiv:1412.6980*.