ELectronics EXpress

IEICE Electronics Express, Vol.17, No.4, 1–6

LETTER

# Acceleration of nonequilibrium Green's function simulation for nanoscale FETs by applying convolutional neural network model

**Satofumi Souma**[1a)] **and Matsuto Ogawa**[1]

**Abstract** We investigate the application of convolutional neural networks (CNNs) to accelerate quantum mechanical transport simulations (based on the nonequilibrium Green's function (NEGF) method) of double-gate MOSFETS. In particular, given a potential distribution as input data, we implement the convolutional autoencoder to train and predict the carrier density and local quantum capacitance distributions. The results indicate that the use of a single trained CNN model in the NEGF self-consistent calculation along with Poisson's equation produces accurate potentials for a wide range of the gate lengths, and all within a significantly shorter computational time than the conventional NEGF calculations.

**Keywords:** nanoscale MOSFET, simulation, nonequilibrium Green's function, convolutional neural network, convolutional autoencoder

**Classification:** Electron devices, circuits and modules

## 1. Introduction

Downscaling of MOSFETs have been one of the key issues in the technological evolution of the semiconductor industry [1, 2]. Multigate structures such as FinFET, which are already available on cutting-edge chips, are expected to reduce power consumption because of their efficient gate electrostatic control of channel carriers over the thin semiconductor layer [3, 4]. However, for continued development of semiconductor technology, the power consumption must be further reduced; for example, by introducing new structures such as the gate-all-around structure or the quantum sheet structure, and new materials such as III–V semiconductors and intrinsically two-dimensional materials (e.g., graphene [5, 6, 7], phosphorene [8] and $MoS_2$ [9]). The investigation of such lower-power consumption devices can be significantly advanced using modeling and simulation studies. Among the various methods to simulate semiconductor devices, one of the most reliable schemes, especially for nanoscale devices, is the fully-quantum-mechanical simulation using non-equilibrium Green's functions (NEGF) [10]. However, NEGF simulation requires repeated matrix operations and energy integrations to calculate the carrier density, which must be self-consistently calculated with Poisson's equation. This procedure is computationally expensive, which hinders the research and development efforts.

[1]Department of Electrical and Electronic Engineering, Kobe University, Kobe 657–8501, Japan

a) ssouma@harbor.kobe-u.ac.jp

DOI: 10.1587/elex.17.20190739
Received December 16, 2019
Accepted December 20, 2019
Publicized January 9, 2020
Copyedited February 25, 2020

Therefore it is important to reduce the computational time required for NEGF device simulations; for instance, by applying information-scientific viewpoint [11, 12, 13, 14]. As a representative example, consider double-gate MOSFET (DG-MOSFET) [15, 16, 17, 18, 19, 20, 21]. The precise two-dimensional distribution pattern of the electrostatic potential and the carrier density in the $x$-$z$ plane (see Fig. 1) essentially influences the gate electrostatic control. Given that such two-dimensional distributions of physical quantities are equivalent to black-and-white images (where positional grid elements correspond to pixels), convolutional neural networks (CNNs), which have recently been successfully applied to image recognition [22, 23, 24, 25] should be useful for device simulations and provide insight into the physical analyses of device operation.

With this motivation, we investigate the use of a CNN model to accelerate a NEGF-based quantum-mechanical transport simulation of DG-MOSFETs.

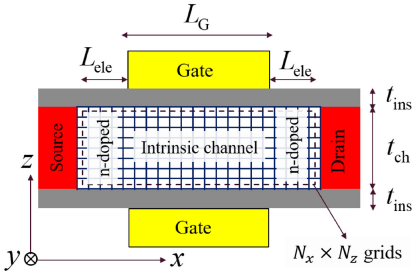


**Fig. 1.** Schematic illustration of DG-MOSFET structure studied in this work. See text for details.

## 2. Model and method

### 2.1 Model structure and NEGF equations

Fig. 1 shows a schematic illustration of a DG-MOSFET, where the gate length along the transport direction $x$ is $L_G$ and the length of the n-doped source and drain regions included in the simulation region is $L_{ele}$. The channel thickness along the confinement direction $z$ is $t_{ch}$, and the insulator thickness is $t_{ins}$. Electrons freely move in the $y$ direction. The simulation region with area $(L_G + 2L_{ele}) \times t_{ch}$ enclosed by the dashed line in Fig. 1 is discretized into $N_x \times N_z$ grid points spanned by $i_x = 1 \sim N_x$ and $i_z = 1 \sim N_z$.

We use the standard mode-space NEGF simulation for DG-MOSFETs [16]. Fig. 2 shows a flowchart describing

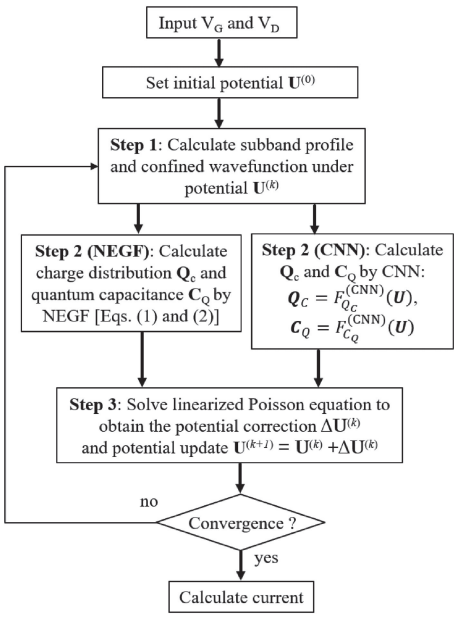


**Fig. 2.** Flowchart of mode-space NEGF simulation of DG-MOSFET.

the process of device simulation based of NEGFs. Here, the potential $\mathbf{U}$, charge distribution $\mathbf{Q}_C$, and the local quantum capacitance distribution $\mathbf{C}_Q$ are all vector quantities with elements $N_xN_z$. In all self-consistent (SC) calculations, we assume the simplest square initial potential $\mathbf{U}_0$, which is constant within the source, channel, and drain regions, with the respective values determined by applying charge neutrality. Given the two-dimensional potential distribution $U_i$ ($i = 1 \sim N_xN_z$), **step 1** calculates the subband (mode) energy $\varepsilon_{i_x,n}$ due to confinement along in the $z$ direction and the corresponding wave function $\chi_{i_x,n}(i_z)$ at the each longitudinal element $i_x$th by solving the one-dimensional Schrödinger equation in the $z$ direction.

**Step 2** then calculates the each $i$th [$i = (i_x - 1)N_z + i_z = 1 \sim N_xN_z$] element of $\mathbf{Q}_C$ and $\mathbf{C}_Q$ as

$$Q_{Ci} = e\tilde{N}_{di}^{+} - g_s g_v \frac{e}{2\pi}\int dE \int dk_y [\rho_i^{S}(k_y,E) f(E-E_{FS}) + \rho_i^{D}(k_y,E) f(E-E_{FD})], \quad (1)$$

$$C_{Qi} = -g_s g_v \frac{e^2}{2\pi}\int dE \int dk_y \left[\rho_i^{S}(k_y,E)\frac{f(E-E_{FS})}{dE} + \rho_i^{D}(k_y,E)\frac{f(E-E_{FD})}{dE}\right], \quad (2)$$

per unit length in the $y$ direction, where $e$ ($> 0$) is the elementary charge, $\tilde{N}_{di}^{+} \equiv N_d^{+}(x_i)\Delta_x\Delta_z$ with $N_d^{+}(x)$ being the ionized donor density at the position $x$ and $\Delta_{x(z)}$ being the grid spacing in the $x(z)$-direction, $k_y$ is the wavenumber in the $y$ direction, and $f(E - E_{FS}(E_{FD}))$ is the Fermi distribution function with $E_{FS} = 0$ and $E_{FD} = -eV_D$ being the Fermi energies in the source and the drain electrodes, respectively (the source is assumed to be grounded and $V_D$ is the applied drain voltage). The spin degeneracy $g_s = 2$, and $g_v = 2$ is the valley degeneracy corresponding to two valleys with heavier effective mass along the confinement direction. $\rho_i^{S}(E,k_y)[\rho_i^{D}(E,k_y)]$ is the local density of states at the $i$th site for electrons incoming from the source (drain) electrode, which we calculate as

$$\rho_i^{S/D}(k_y,E) = \frac{1}{2\pi}\sum_n [G_n(E,k_y)\Gamma_{n,S/D}(E,k_y)G_n^{\dagger}(E,k_y)]_{i_xi_x} \times |\chi_{i_x,n}(i_z)|^2, \quad (3)$$

where $G_n(k_t,E)$ is the retarded Green's function given by

$$G_n(k_y,E) = [E - \mathcal{H}_n(k_y) - \Sigma_n^{(S)}(k_y,E) - \Sigma_n^{(D)}(k_y,E)]^{-1}. \quad (4)$$

Here $\mathcal{H}_n(k_y)$ is the $N_x \times N_x$ Hamiltonian matrix with finite-derivative describing motion in the $x$-direction and including the subband energy $\varepsilon_{i_x,n}$ and the transverse ($y$ direction) kinetic energy given by $[\mathcal{H}_n(k_y)]_{i_xj_x} = (E_C + 2t_x + \varepsilon_{i_x,n} + \hbar^2k_y^2/2m_y^*)\delta_{i_xj_x} - t_x(\delta_{i_x,j_x-1} + \delta_{i_x,j_x+1})$, where $E_C$ is the conduction-band edge, $t_x = \hbar^2/(2m_x^*\Delta_x^2)$ is the hopping energy in the $x$-direction with $\Delta_x$ being the grid spacing, $m_x^*(m_y^*)$ is the effective mass along the $x(y)$ direction. The quantity $\Sigma_n^{(S/D)}(k_y,E)$ is the retarded self-energy due to the coupling to the source (drain) electrode [10], and $\Gamma_n^{(S/D)}(k_y,E)$ is the corresponding broadening function, which is given by

$$\Gamma_n^{(S/D)}(k_y,E) = i[\Sigma_n^{(S/D)}(k_y,E) - \Sigma_n^{(S/D)\dagger}(k_y,E)]. \quad (5)$$

### 2.2 Updating the potential by solving the linearized Poisson equation

Following the standard Newtonian scheme, Poisson's equation is linearized to obtain the potential correction $\Delta\mathbf{U}^{(k)}$ as

$$[J(\mathbf{U}^{(k)})]\Delta\mathbf{U}^{(k)} = -\mathbf{f}(\mathbf{U}^{(k)}), \quad (6)$$

under the potential $\mathbf{U}^{(k)}$ and at the $k$th iteration. The solution to Eq. (6) is used to update the potential as $\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} + \Delta\mathbf{U}^{(k)}$, as per **step 3** in Fig. 2. In Eq. (6) the Jacobi matrix $[J(\mathbf{U})]$ is defined by $[J(\mathbf{U})] \equiv [C] + [C_Q(\mathbf{U})]$ with the local quantum capacitance matrix $[C_Q(\mathbf{U})]_{ij} = C_{Qi}\delta_{ij}$ given by Eq. (2), and the vector $\mathbf{f}(\mathbf{U})$ defined by $\mathbf{f}(\mathbf{U}) \equiv [C]\mathbf{U} - \mathbf{Q}_C'(\mathbf{U})$. Here $[C]$ is the electrostatic capacitance matrix obtained by the finite difference discretization of Poisson's equation, which is given by $[C] = [C^{(x)}] + [C^{(z)}] + [C^{(\mathrm{gate})}]$ with $C_{ij}^{(x)} = [2C_x - C_x(\delta_{i_x1} + \delta_{i_xN_x})]\delta_{ij} - C_x(\delta_{i_x,j_x-1} + \delta_{i_x,j_x+1})\delta_{i_z,j_z}$, $C_{ij}^{(z)} = [2C_z - C_z(\delta_{i_z1} + \delta_{i_zN_z})]\delta_{ij} - C_z(\delta_{i_z,j_z-1} + \delta_{i_z,j_z+1})\delta_{i_x,j_x}$, and $C_{ij}^{(\mathrm{gate})} = C_G\delta_{ij}(\delta_{i_z1} + \delta_{i_zN_z})\delta_{i_x,i_x\in L_G}$, with Neumann boundary condition (zero electric field) assumed at all boundaries except for those adjacent to the gate electrodes. In addition $C_x = \kappa_{ch}\varepsilon_0\Delta_z/\Delta_x$, $C_z = \kappa_{ch}\varepsilon_0\Delta_x/\Delta_z$, and $C_G = 2\kappa_{ox}\varepsilon_0\Delta_x/t_{oc}$ are the channel capacitances in the $x$ and $z$ directions and the gate capacitance, respectively, all per unit transverse ($y$) length. The quantity $\Delta_x$ ($\Delta_z$) is the grid spacing in the $x$ ($z$) direction, $\kappa_{ch}$ ($\kappa_{ox}$) is the dielectric constant in the channel (insulator), and $t_{ox}$ is the thickness of the insulating layer. In the definition of $\mathbf{f}(\mathbf{U})$ introduced above, $\mathbf{Q}_C'$ is defined by the element $Q_{Ci}' = Q_{Ci} + C_GV_G(\delta_{i_z1} + \delta_{i_zN_z})\delta_{i_x,i_x\in L_G}$ with $Q_{Ci}$ given by Eq. (1).

As indicated in Fig. 2, the NEGF calculation presented above is to be repeated until the convergence criterion for $\Delta\mathbf{U}^{(k)}$ is satisfied. We introduce the convergence factor $0 < \alpha_{conv} < 1$ as $\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} + \alpha_{conv}\Delta\mathbf{U}^{(k)}$ to reduce the amount of potential update and stabilize the convergence. After convergence the electronic current is calculated by using the standard NEGF equation.

Note that, in the Newtonian scheme presented above, the computational load for the calculation of $\mathbf{Q}_C$ and $\mathbf{C}_Q$ is much heavier than that for the linear equation (6) because of the numerical integrations.

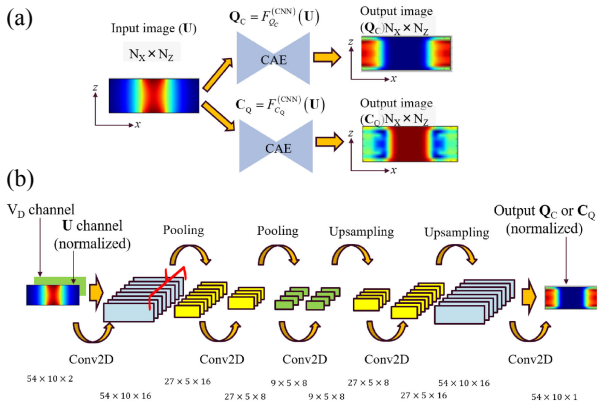


**Fig. 3.** (a) Conceptual explanation of the application of CNN to calculate $\mathbf{Q}_C$ and $\mathbf{C}_Q$ from $\mathbf{U}$. Each CNN (CAE) is trained so that the output data agree with those calculated by the NEGF. (b) Illustration of CAE network used in this study.

### 2.3 CNN model to train and predict charge density and quantum capacitance from the input potential

This subsection describes the proposed CNN model. Our goal is to construct two CNNs: $\mathbf{Q}_C = F_{Q_C}^{(CNN)}(\mathbf{U})$ and $\mathbf{C}_Q = F_{C_Q}^{(CNN)}(\mathbf{U})$, where the input data are contained in $\mathbf{U}$, the output data are contained in $\mathbf{Q}_C$ in the first CNN and in $\mathbf{C}_Q$ in the second CNN, and $F_{Q_C}^{(CNN)}$ and $F_{C_Q}^{(CNN)}$ must be trained to reproduce the NEGF results. The trained CNN functions are then destined to replace the time-consuming NEGF calculation in **step 2** of Fig. 2 [13].

Since our purpose in using CNN is to obtain the output image ($\mathbf{Q}_C$ or $\mathbf{C}_Q$) with $N_x \times N_z$ pixels from the input image ($\mathbf{U}$) with the same number of pixels, we use a convolutional autoencoder (CAE) scheme [Fig. 3(a)] which has been used in various image-recognition (e.g., denosing and super-resolution) [26, 27]. Fig. 3(b) shows the CAE network structure used in this study. Keeping in mind that in the NEGF calculation of $\mathbf{Q}_C$ and $\mathbf{C}_Q$ in Eqs. (1, 2) the drain voltage $V_D$ is explicitly used through the Fermi distribution function (whereas the source voltage $V_S$ is assumed to be zero), so we include the drain voltage $V_D$ as an additional input layer (additional "channel" in the tensor), where the value of $V_D$ is stored in all $N_x \times N_z$ pixels (see Fig. 3). The effective masses and the valley degeneracy are also explicitly used to calculate $\mathbf{Q}_C$ and $\mathbf{C}_Q$, but they are not included as the CNN input parameters because we restrict out attention to the silicon channel confined in the [001] direction and transport along the [100] direction. Note also that $L_G$ and $L_{ele}$ are not included explicitly as input parameters. Parameters related to solve Poisson's equation (doping concentration, $\kappa_{ch}$, $\kappa_{ox}$, $t_{ox}$) are taken into account essentially through Poisson's equation, and so are also not included as CNN input parameters.

We now summarize CNN structure in detail. In the present study we assume a grid size $N_x = 54$ and $N_z = 10$ (see next section for detail). The kernel size in all convolution (Conv2D) is $(3, 3)$, and the number of filters is 16, 8, 8, 16, and 1 in the five Conv2D, from the left to right, respectively. We use the Relu activation function in all Conv2D and pooling (MaxPooling2D) except for the final Conv2D, where we use sigmoid activation. Zero padding without changing the image size is used in Conv2D and MaxPooling2D. Pooling sizes are $(2, 2)$ and $(3, 1)$ in the first and second pooling, respectively, and up-sampling sizes are $(3, 1)$ and $(2, 2)$ in the first and second up-sampling, respectively. Note that, when used in a CNN, $\mathbf{U}$, $\mathbf{Q}_C$, and $\mathbf{C}_Q$ have to be normalized to $[0, 1]$ by introducing their respective minimum and maximum values. When a trained CNN is used in **step 2** of Fig. 2, the input of $\mathbf{U}$ to the CNN has to be normalized to $[0, 1]$ and the CNN outputs $\mathbf{Q}_C$ and $\mathbf{C}_Q$ have to be denormalized to their original ranges when used in Poisson's equation. For the actual implementation we used the Keras API with the TensorFlow backend engine [28].

## 3. Results and discussion

### 3.1 Potential calculation by CNN-SC process

In this work, we consider that the silicon channel is confined in the [001] direction and that transport is in the [100] direction, so that the effective masses are $m_x^* = m_y^* = 0.19m_0$, $m_z^* = 0.98m_0$, where $m_0$ is the free electron mass, and $\kappa_{ch} = 11.7$ is channel dielectric constant. For the gate insulator we assume a dielectric constant $\kappa_{ox} = 3.8$ ($SiO_2$) with the thickness $t_{ox} = 1$ nm. Doping concentration for the source and drain electrodes is $N_D^+ = 1 \times 10^{26}\,\mathrm{m}^{-3}$, and the temperature is $T = 300$ K. We assume a difference in work function of 0.36 eV between the gate metal and silicon. All of the simulations and trainings presented below have been performed by using a machine with an Intel Core i7 CPU (2.80 GHz) and 8 GB memory.

We first consider a DG-MOSFET model with $L_G = 8$ nm, $L_{ele} = 10$ nm, and $t_{ch} = 5$ nm (see Fig. 1). The rectangular area with $(L_{gate} + 2L_{ele}) \times t_{ch}$ is discretized into $(N_x = 54) \times (N_z = 10)$ elements, so that $\Delta_x \simeq 0.31$ and $\Delta_z \simeq 0.45$ nm.

For the CNN training, we prepared 3600 datasets [(6 values of $V_D = 0 \sim 0.5$) × (6 values of $V_G = 0 \sim 0.5$) × (100 SC iterations performed in NEGF simulator)], where the voltage increment is 0.1 V. In the training process, 80% of the data were used for training, with the remaining 20% used for validation, where we used the mean squared error as loss function and an adaptive moment estimation method (Adam) as optimizer [29]. In the data presented in this subsection the batch size is 30 and carried out 100 epochs, yielding the final validation loss around $\sim 10^{-4}$.

We then use the trained CNN in **step 2** of the SC calculation (see Fig. 2). To fully benefit from the performance of CNN model itself in the SC iterative process, we use the `frugally-deep` library [30], which allows us to use the trained Keras models directly in our device simulator (written in C++ to predict $\mathbf{Q}_C$ and $\mathbf{C}_Q$ without requiring Python interface).

Fig. 4 shows the RMSE as a function of SC iteration for the potential obtained by the CNN-SC process with respect to the converged NEGF-SC potential, all evaluated

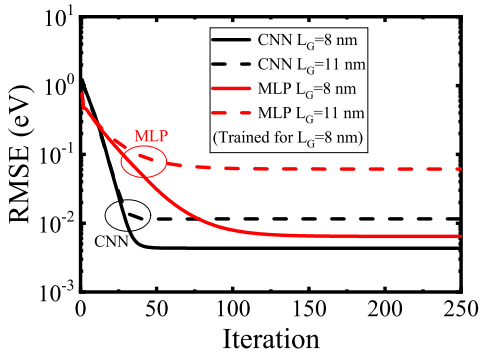


**Fig. 4.** Root-mean-squared error (RMSE) of potentials updated by using CNN-SC and MLP-SC processes with respect to the "converged" potential calculated by NEGF, plotted as a function of SC step for $V_G = V_D = 0.3$ eV. Here the neural network training was done for $L_G = 8$ nm and the trained models are tested for $L_G = 8$ and 11 nm.

for the same model as for the trained Model 1 at $V_G = V_D = 0.3$ V. We also plot the RMSE evaluated for Model 2 ($L_{gate} = 11$ nm; the other parameters including $N_x$ and $N_z$ are the same as for Model 1) from the trained Model 1 at $V_G = V_D = 0.3$ V. Note that the maximum values of $\mathbf{Q}_C$ and $\mathbf{C}_Q$ (the minimum values are always set to zero) required for normalization to $[0, 1]$ are scaled by the multiplication factor $\gamma^{(M,M1)} \equiv (\Delta_x^{(M)}/\Delta_x^{(M1)})(\Delta_z^{(M)}/\Delta_z^{(M1)})$, where $\Delta_{x,z}^{(M1)}$ and $\Delta_{x,z}^{(M)}$ are the grid spacings in Model 1 and in the target model, respectively.

For comparison, the above RMSE are examined by using $F_{Q_C}^{(MLP)}$ and $F_{C_Q}^{(MLP)}$ trained by a conventional NN [i.e., multi-layer perceptron (MLP)], where we use two intermediate (hidden) layers with 100 nodes with Relu activation for hidden layers and sigmoid activation for the output layer (adjusting the number of layers and nodes does not change the qualitative tendencies shown in Fig. 4). We used the convergence factor $\alpha_{conv} = 0.5$ for all data in Fig. 4.

Both the CNN-SC and MLP-SC processes reproduce the NEGF potential with high accuracy when applied to the same model as the training model. However, the MLP-SC converges slower than the CNN-SC process. Moreover, the CNN-SC process is more accurate than the MLP-SC process when applied to the model with different $L_G$ from the training model. These overall tendencies suggest that the CNN-SC process converges faster and can be more widely generalized than the MLP-SC process. The advantage of the CNN process over the MLP process in the present study is the ability of the CNN to learn two-dimensionally precise distributions of $\mathbf{Q}_C$ and $\mathbf{C}_Q$ in the $x$-$z$ plane of a DG-MOSFET. The reduced effort required by the training process and arising from the wide generalizability of the process is a significant advantage of using CNN-based model for device simulation.

### 3.2 Comparison of $I_D$-$V_G$ and $I_D$-$V_D$ curves obtained by NEGF and CNN models

Given the confirmed reliability and the generalizability of the CNN-SC process, we next discuss the use of the CNN-SC process to reproduce the $I_D$-$V_G$ and $I_D$-$V_D$ curves calculated by the NEGF-SC process for various device parameters. To clarify the usefulness of the CNN-SC process for practical use case, we prepared three DG-MOSFET models for CNN training, with $L_G = 8$, 11, and 15 nm for Model 1–3, respectively, and performed a single set of training covering these three models. The other parameters are same for all three models; namely, $L_{ele} = 10$ nm, $t_{ch} = 5$ nm, $N_x = 54$, and $N_z = 10$.

In other word, we performed the CNN training over 10800 datasets [(3 models) × (6 values of $V_D$) × (6 values of $V_G$) × (100 SC iterations in the NEGF simulator)] for three above-mentioned models, generating only two Keras model files (HDF5 format files) corresponding to $F_{Q_C}^{(CNN)}$ and $F_{C_Q}^{(CNN)}$. Here, we used a batch size of 50 and 800 epochs in the results shown in this subsection. Here the pure CNN training time (excluding the dataset preparation time) for the above datasets is around $T_{training} \sim 100$ min. The datasets preparation time by full-NEGF simulations is $T_{datasets} \sim 270$ min.

Fig. 5(a) plots the first subband profiles calculated in **step 1** of the flowchart (Fig. 2) by using the converged CNN-SC potentials. The results obtained by using the converged NEGF-SC potentials are also plotted to check the accuracy. The CNN model with SC calculations reproduces the NEGF results with good accuracy for all drain voltages. Although the potentials and the resulting subband profiles in the CNN results deviate slightly from the NEGF results, the discrepancy can be quickly improved by implementing several additional NEGF-SC steps after convergence of CNN-SC calculations. This is shown in

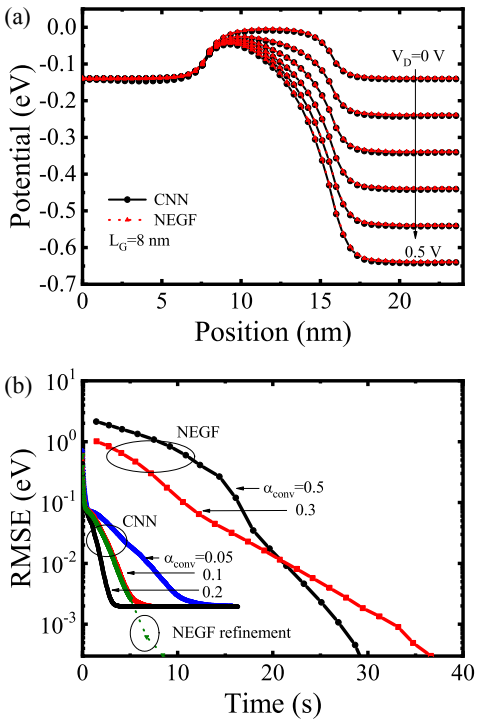


**Fig. 5.** (a) First subband profiles calculated by using converged CNN-SC potentials and NEGF-SC potentials, for $L_G = 8$ nm. (b) RMSE of the potentials obtained by using CNN-SC and NEGF processes with respect to the accurate converged potential obtained by NEGF, plotted as a function of time for the SC process with $V_G = V_D = 0.3$ V. Convergence is unstable for $\alpha_{conv} > 0.5$ and $\alpha_{conv} > 0.2$ for the NEGF-SC and CNN-SC process, respectively. Also plotted is the potential obtained by applying an additional NEGF after the CNN-SC process with $\alpha_{conv} = 0.1$. In the refinement process, $\alpha_{conv} = 1$ leads to stable convergence.

Fig. 5(b), which plots the RMSE during the SC process as a function of time for the CNN-SC and NEGF-SC processes. Fig. 5(b) also plots the refinement process by the additional NEGF-SC calculation. Note that, in the NEGF refinement process, the convergence is stable, so the full update ($\alpha_{\mathrm{conv}} = 1$) may be safely employed. We also emphasize that the use of the CNN-SC process gives high accuracy (RMSE $\sim 10^{-7}$) in significantly less computational time than the full NEGF-SC process.

Figs. 6(a) and 6(b) compare the $I_{\mathrm{D}}$-$V_{\mathrm{G}}$ and $I_{\mathrm{D}}$-$V_{\mathrm{D}}$ curves obtained by using the CNN-SC and NEGF-SC processes for various values of $L_{\mathrm{G}}$ and $V_{\mathrm{G}}$, respectively, where, to clarify the usefulness of the CNN-SC process itself, the additional NEGF-SC refinement process was not applied. We use $\alpha_{\mathrm{conv}} = 0.1$ in both plots of Fig. 6. As shown by these figures, the proposed CNN model reproduces the NEGF results with good accuracy through the SC calculations for all values of $L_{\mathrm{G}}$, allowing it to reproduce the larger subthreshold slope (SS) for shorter $L_{\mathrm{G}}$ and the shift in $V_{\mathrm{th}}$ due to drain-induced barrier lowering (DIBL). The agreement between CNN and NEGF is particularly noteworthy in the linear $I_{\mathrm{D}}$-$V_{\mathrm{D}}$ regime and for $I_{\mathrm{D}}$-$V_{\mathrm{G}}$ curves everywhere except for high $V_{\mathrm{G}}$. Note that high accuracy is achieved despite applying the generated $F_{Q_{\mathrm{C}}}^{(\mathrm{CNN})}$ and $F_{C_{\mathrm{Q}}}^{(\mathrm{CNN})}$ models to the case of $L_{\mathrm{G}}$, which differs from any of the training models. Benefited by the superior generalizability clarified above, once the CNN training as above is done, subsequent simulations can be performed quickly by using CNN-SC process (optionally with quick NEGF-SC refinement process) without requiring additional training as far as $L_{\mathrm{G}}$ is around in the range of 5∼15 nm.

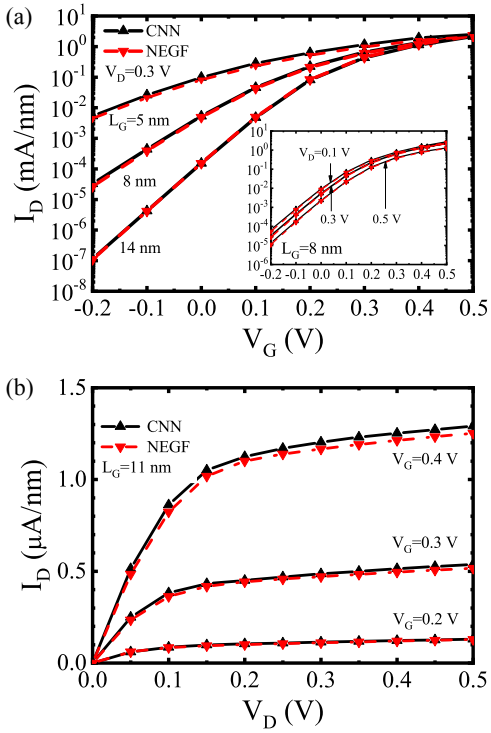


**Fig. 6.** (a) $I_{\mathrm{D}}$-$V_{\mathrm{G}}$ curve obtained by using CNN-SC and NEGF-SC processes for various values of $L_{\mathrm{G}}$ at $V_{\mathrm{D}} = 0.3$ V. Inset shows how $V_{\mathrm{D}}$ affects the $I_{\mathrm{D}}$-$V_{\mathrm{G}}$ curve through the DIBL. (b) $I_{\mathrm{D}}$-$V_{\mathrm{D}}$ curve obtained by using CNN-SC and NEGF-SC processes for various values of $V_{\mathrm{G}}$, where $L_{\mathrm{G}} = 11$ nm.

Thus far, we have used the CNN trained for a channel thickness $t_{\mathrm{ch}} = 5$ nm to make predictions about models with the same thickness. However, upon applying the trained CNN to models with thicknesses differing by several nanometers with respect to the training models, the converged potential deviates by at most about 0.01 eV from the NEGF result. This result indicates that the CNN prediction remains useful for quantitative simulations and that quick refinement by applying an additional NEGF process is also possible, which suggests that the CNN model may be more widely generalized.

Finally, we remark on the role of batch size in determining convergence stability and accuracy. A smaller batch size in the CNN training produces more accurate current predictions from the CNN-SC process but also results in unstable convergence, requiring us to use a smaller value of $\alpha_{\mathrm{conv}}$. Conversely, a larger batch size produces stable convergence, allowing us to use a larger $\alpha_{\mathrm{conv}}$ and enabling quicker convergence but also resulting in lower accuracy.

## 4. Conclusion

We present herein a study on the application of CNNs to accelerate NEGF-based quantum-mechanical transport simulations for DG-MOSFET. Given a potential distribution as input data, the CAE model is used to train and predict the carrier density and local quantum capacitance distributions as output data, where the input and output data are treated as two-dimensional images in the $x$-$z$ plane. The use of the CNN model in the SC calculation along with Poisson's equation helps to obtain accurate potential and currents over a wide range of gate length and with significantly less computational time than conventional NEGF calculations, where including $V_{\mathrm{D}}$ as an additional input image layer is crucial for generalizing this approach over wide range of $V_{\mathrm{D}}$. The more simulations users require to perform, the more benefited by the proposed scheme. The proposed scheme is especially beneficial for simulator users if simulation software vendors perform the CNN training and provides the trained CNN model files to users. Then the users can be fully benefited by the advantage of proposed CNN-SC process without taking time for training.

## Acknowledgments

This work was partially supported by JSPS KAKENHI Grant No. 19H04546. The authors would like to thank Enago (www.enago.jp) for the English language review.

### References

[1] International Roadmap for Devices and Systems (2019) https://irds.ieee.org/.
[2] E. Pop: "Energy dissipation and transport in nanoscale devices," Nano Res. **3** (2010) 147 (DOI: 10.1007/s12274-010-1019-z).
[3] R. S. Pal, *et al.*: "Recent trend of FinFET devices and its challenges: A review," 2017 Conference on Emerging Devices and Smart Systems (ICEDSS) (2017) 150 (DOI: 10.1109/ICEDSS.2017.8073675).
[4] S. Verma, *et al.*: "Performance analysis of FinFET device using qualitative approach for low-power applications," 2019 Devices for Integrated Circuit (DevIC) (2019) 84 (DOI: 10.1109/DEVIC.2019.

8783754).
[5] K. S. Novoselov, *et al.*: "A roadmap for graphene," Nature **490** (2012) 192 (DOI: 10.1038/nature11458).
[6] S. Souma, *et al.*: "Simulation-based design of a strained graphene field effect transistor incorporating the pseudo magnetic field effect," Appl. Phys. Lett. **104** (2014) 213505 (DOI: 10.1063/1.4880579).
[7] S. Souma, *et al.*: "Effect of lateral strain on gate induced control of electrical conduction in single layer graphene device," J. Comput. Electron. **12** (2013) 170 (DOI: 10.1007/s10825-013-0451-1).
[8] A. Carvalho, *et al.*: "Phosphorene: From theory to applications," Nat. Rev. Mat. **1** (2016) 16061 (DOI: 10.1038/natrevmats.2016.61).
[9] R. Ganatra and Q. Zhang: "Few-layer MoS2: A promising layered semiconductor," ACS Nano **8** (2014) 4074 (DOI: 10.1021/nn405938z).
[10] S. Datta: *Electronic Transport in Mesoscopic Systems* (Cambridge University Press, UK, 1997).
[11] F. Djeffal, *et al.*: "An approach based on neural computation to simulate the nanoscale CMOS circuits: Application to the simulation of CMOS inverter," Solid-State Electron. **51** (2007) 48 (DOI: 10.1016/j.sse.2006.12.004).
[12] F. Djeffal, *et al.*: "A neural approach to study the scaling capability of the undoped double-gate and cylindrical gate all around MOSFETs," Mater. Sci. Eng. B **147** (2008) 239 (DOI: 10.1016/j.mseb.2007.08.034).
[13] K. Tamersit and F. Djeffal: "A computationally efficient hybrid approach based on artificial neural networks and the wavelet transform for quantum simulations of graphene nanoribbon FETs," J. Comput. Electron. **18** (2019) 813 (DOI: 10.1007/s10825-019-01350-2).
[14] J. Zhang, *et al.*: "Prior knowledge input neural network method for GFET description," J. Comput. Electron. **15** (2016) 911 (DOI: 10.1007/s10825-016-0842-1).
[15] Z. Rajabi, *et al.*: "The non-equilibrium Green's function (NEGF) simulation of nanoscale lightly doped drain and source double gate MOSFETs," 2012 International Conference on Devices, Circuits and Systems, ICDCS (2012) 25 (DOI: 10.1109/ICDCSyst.2012.6188669).
[16] Y. M. Sabry, *et al.*: "Uncoupled mode-space simulation validity for double gate MOSFETs," Proc. of International Conference on Microelectronics (2007) 351 (DOI: 10.1109/ICM.2007.4497727).
[17] R. Venugopal, *et al.*: "Simulating quantum transport in nanoscale transistors: Real versus mode-space approaches," J. Appl. Phys. **92** (2002) 3730 (DOI: 10.1063/1.1503165).
[18] S. Datta: "Nanoscale device modeling: The Green's function method," Superlattices Microstruct. **28** (2000) 253 (DOI: 10.1006/spmi.2000.0920).
[19] R. Hosseini: "Uncoupled mode space approach for analysis of nanoscale strained junctionless double-gate MOSFET," J. Comput. Electron. **15** (2016) 787 (DOI: 10.1007/s10825-016-0826-1).
[20] H. Fitriawan, *et al.*: "Quantum electron transport modeling in uniaxially strained silicon channel of double-gate MOSFETs," Phys. Stat. Sol. (c) **5** (2008) 74 (DOI: 10.1002/pssc.200776542).
[21] H. Fitriawan, *et al.*: "Multiband simulation of nano-scale MOSFETs based on a non-equilibrium Green's function method," IEICE Trans. Electron. **E91-C** (2008) 105 (DOI: 10.1093/ietele/e91-c.1.105).
[22] Y. LeCun, *et al.*: "Deep learning," Nature **521** (2015) 436 (DOI: 10.1038/nature14539).
[23] J. Schmidhuber: "Deep learning in neural networks: An overview," Neural Netw. **61** (2015) 85 (DOI: 10.1016/j.neunet.2014.09.003).
[24] A. Krizhevsky, *et al.*: "ImageNet classification with deep convolutional neural networks," Commun. ACM **60** (2017) 84 (DOI: 10.1145/3065386).
[25] J. Gu, *et al.*: "Recent advances in convolutional neural networks," arXiv preprint (2015) arXiv:1512.07108.
[26] M. Tschannen, *et al.*: "Recent advances in autoencoder-based representation learning," arXiv preprint (2018) arXiv:1812.05069.
[27] V. Turchenko, *et al.*: "A deep convolutional auto-encoder with pooling-unpooling layers in Caffe," arXiv preprint (2017) arXiv:1701.0494.
[28] Keras: Deep learning library for Theano and TensorFlow (2015) https://keras.io/.
[29] D. P. Kingma and J. Ba: "Adam: A method for stochastic optimization," arXiv preprint (2014) arXiv:1412.6980.
[30] Frugally-deep library, https://github.com/Dobiasd/frugally-deep.