Speed Up Quantum Transport Device Simulation on Ferroelectric Tunnel Junction With Machine Learning Methods

Tong Wu and Jing Guo, *Senior Member, IEEE*

***Abstract*— As the device size scales down to the nanometer regime, quantum effects play an important role in device characteristics and performance. Quantum transport device simulation based on the nonequilibrium Green's function (NEGF) has been extensively applied to simulate the nanoscale devices. The NEGF simulations, however, can be computationally expensive, especially in the presence of scattering. In this study, a machine learning (ML)-based framework is developed, targeting on replacing the computationally intensive NEGF simulations. This framework first learns a sparse representation of a quantum transport property of interest and then trains a model to describe the quantitative mapping relation between the device parameters and properties. Also, the accuracy is further improved with the application of feature engineering. As an example, a graphene–ferroelectric–metal (GFM) ferroelectric tunnel junction (FTJ) is simulated. The results show that the ML-based framework allows circumventing the NEGF calculation and simultaneously maintaining high accuracy in quantum transmissions and tunneling *I–V* characteristics. This ML-based framework can be applied to speed up the quantum transport device simulations and enable efficient tunneling device design.**

***Index Terms*— Machine learning (ML), nanodevice, nonequilibrium Green's function (NEGF) simulations.**

I. Introduction

AS THE size of electron devices scales down to the nanoscale regime, quantum effects become increasingly important. In nanoscale field-effect transistors (FETs), quantum-mechanical tunneling limits ultimate scaling. On the other hand, quantum effects serve as the fundamental operation mechanism for nanodevices such as tunneling FETs [1], [2], magnetic tunnel junction (MTJ) [3], [4], and ferroelectric tunnel junction (FTJ) [5], [6]. To capture the quantum effects in those device studies, the nonequilibrium Green's function (NEGF) formalism has been developed and applied to computer-aided design and simulation of nanoscale devices [7]–[10]. To model a realistic device size with scattering, the NEGF simulations, however, can be computationally expensive, which hinders efficient device design [9], [11].

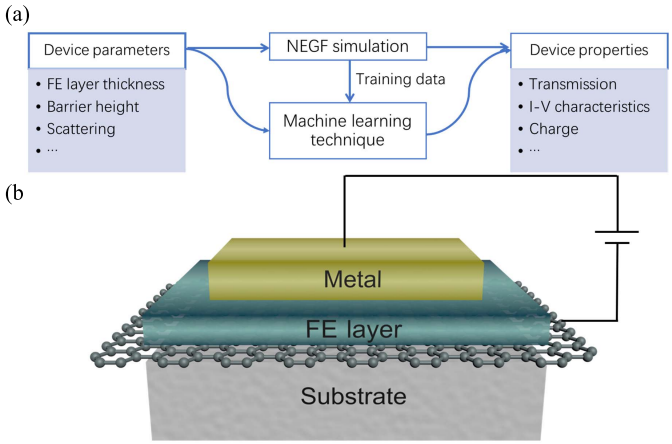

Fig. 1. (a) Diagram of the NEGF simulation speed up by ML techniques. The ML model is trained to link the device parameters to device properties and it can replace the NEGF simulation to accelerate the whole simulation. (b) Schematic of the GFM FTJ. The top contact (metal) and the bottom contact (graphene) are separated by a thin ferroelectric (FE) layer.

To address this issue, one approach is to develop a data-driven model that can replace the computationally most demanding part in the device simulation flow. Recently, machine learning (ML) methods have been developed and applied for design and simulation problems in the fields of quantum chemistry and quantum physics [11], [12], as well as computational materials science [13]–[15]. Compared to these more fundamental fields, the application of the ML method in the quantum device simulation remains much more limited.

In this work, an ML-based framework to enable efficient modeling of quantum transport in nanoscale devices is proposed, as shown in Fig. 1(a). In the quantum transport device simulations, the NEGF simulations, which simulate device properties from device parameters, can be computationally expensive in the presence of scattering for routine device design. Alternatively, an ML model can be trained, which maps device parameters to device properties based on the data obtained from the NEGF simulations, as shown in Fig. 1(a). Once the model is trained, the NEGF simulations path can be replaced with significantly improved computational efficiency. This ML-based framework is capable of speeding

Manuscript received August 17, 2020; accepted September 16, 2020. Date of publication October 7, 2020; date of current version October 22, 2020. This work was supported by the National Science Foundation Awards under Grant 1904580 and Grant 1610387. The review of this article was arranged by Editor S. Hong. *(Corresponding author: Jing Guo.)*

The authors are with the Department of ECE, University of Florida, Gainesville, FL 32611 USA (e-mail: guoj@ufl.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TED.2020.3025982

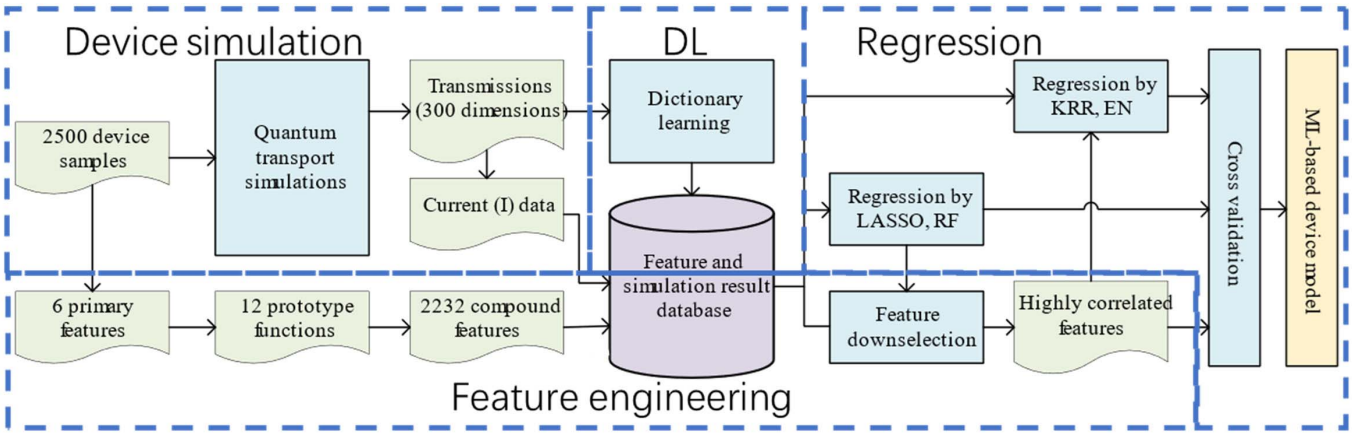

Fig. 2. Block diagram of the ML-based method in the quantum transport simulation. There are four main blocks in this approach: the device simulation block that uses the NEGF simulation to generate data, the DL block that reduces the learning target features; the regression block that implements the regression, prediction, and cross validation; and the feature engineering block that consists of feature generation and feature selection.

up the modeling quantum effects in the device simulation for enabling high-throughput, computationally efficient device design. Also, it can be further extended to a high-dimension data set or a complex device structure.

II. Approach

The overall purpose of the ML-based framework is to develop a data-driven model that can relate the device parameters to device properties in the quantum transport device simulations. As an example of illustration, a monolayer graphene–ferroelectric–metal (GFM) FTJ device, as shown in Fig. 1(b), is simulated [16]. The graphene contact has an energy-dependent density of state (DOS), which impacts the device I–V characteristics, and requires treatment beyond the Wentzel–Kramers–Brillouin (WKB) approximation calculation, which is incapable of modeling contacts and scattering. The NEGF calculation that can treat contact and scattering effect is necessary, and a time-efficient method that can speed up the NEGF device simulation is desired for efficient device design.

For the GFM device in Fig. 1(b), we focus on obtaining two of the device properties: transmission versus energy and I–V characteristics. The transmission provides physical insights into quantum mechanical behaviors of the FTJ device operations. Since the current can be straightforwardly computed from the transmissions, the learning target is only set as the transmission versus energy. Then the current will be computed from the ML-reconstructed transmissions. We take the logarithm of transmissions as a function of energy $\log(T(E))$ due to the large range of the order of magnitude of the transmission data.

Fig. 2 outlines the ML-based framework, which consists of four blocks: 1) device simulation block, 2) dictionary learning (DL) block, 3) regression block, and 4) feature engineering block, as indicated by the dashed line boxes. The simulation block performs the NEGF simulations for obtaining the training and testing data sets. We describe the NEGF simulations used to generate the training and testing data in Section II-A. The DL technique, which facilitates the representation of $\log(T(E))$, is described in Section II-B. Regression algorithms, which are the core of relating the device parameters to the device properties, are discussed in Section II-C. The feature engineering block is introduced in Section II-D. Finally, the overall workflow is described in Section II-E.

A. Quantum Transport Simulation

To study the carrier transport and device performance of the GFM FTJ device, which is shown in Fig. 1(b), the NEGF on a finite-difference grid device simulation is used [7], [8]. The potential profile of the device is calculated by solving the electrostatic Poisson equation in 1-D along the vertical direction self-consistently with the contact equilibrium carrier statistics [16]. Dirichlet boundary condition is taken for the last node.

Since the tunneling barrier thickness is small in the FTJ device, the ballistic transport is assumed for simplicity. The Fermi level pinning is neglected as it can be weak at the graphene–FE insulator interface [16]. The Green's function can be written as

$$G(E)=\left[(E+i0^{+})I_0-H-E_c-\Sigma_1(E)-\Sigma_2(E)-\Sigma_{\mathrm{Sca}}(E)\right]^{-1} \tag{1}$$

where I_0 is the unit matrix, H is the device Hamiltonian obtained by the finite difference approximation, E_c is the conductance band edge, Σ_1 and Σ_2 are the self-energies for the graphene and metal contacts, respectively, and $\Sigma_{\mathrm{Sca}}(E)$ is the scattering self-energy [17]. The FTJ is assumed to be uniform in the transverse direction, and the NEGF transport equation is solved in the 1-D transport direction by using an effective mass Hamiltonian in the energy range we interested in.

To provide a capability to phenomenologically model elastic scattering due to defects and impurities in the device simulations, the self-consistent Born approximation (SCBA) is used in the NEGF transport simulations [17], [18]. The scattering self-energy is related to the retarded Green's function

as $\Sigma_{\mathrm{Sca}}(i,j) = \bar{D}(i,j)G(i,j)$, and the in-scattering (out-scattering) function $\Sigma^{\mathrm{in,out}}$ is related to the electron (hole) correlation function $G^{\mathrm{n,p}}$ as $\Sigma^{\mathrm{in,out}}(i,j) = \bar{D}(i,j)G^{\mathrm{n,p}}(i,j)$, where $\bar{D}(i,j) = S\delta_{ij}$, where S is a constant factor describing the strength of scattering, δ_{ij} is the Dirac delta function, and i and j are the point position indices [17].

Contact effects can play an important role in nanoscale devices. As for the GFM structure, self-consistent electrostatics modulate the Dirac energy point of graphene contact with regard to its Fermi level. In the NEGF device simulation, a phenomenological description is used for the self-energy of a contact [19], which can be expressed as

$$\Sigma_1(E) = \tau g_{\mathrm{s}}(E)\tau^{+} \tag{2}$$

where $g_{\mathrm{s}}(E)$ is the surface Green's function of the contact, and τ is the coupling between the device node and the surface node of the contact. The contact broadening can be expressed as

$$\Gamma_1(E) = i\left(\Sigma_1(E) - \Sigma_1(E)^{+}\right) = \tau A_{\mathrm{s}}(E)\tau^{+} \tag{3}$$

where $A_{\mathrm{s}}(E) = i(g_{\mathrm{s}}(E) - g_{\mathrm{s}}(E)^{+}) = 2\pi D_{\mathrm{S}}(E)$ is the surface spectral function and $D_{\mathrm{S}}(E)$ is the DOS at the surface node of the contact. The imaginary part of the contact self-energy is proportional to the DOS of the contact material. For a monolayer graphene contact, the surface node of the contact is essentially the graphene layer. By assuming τ is energy independent, the self-energy of graphene contact can be expressed as

$$\Sigma_1(E) = -i\alpha_0(|E - E_{\mathrm{d}}| + \sigma_0) \tag{4}$$

where α_0 is a proportional constant determined by the coupling strength between the contact and device, E_{d} is the Dirac energy, the first term describes the linear dependence of the graphene DOS on energy, and the second term describes a nonzero residue DOS at the Dirac point due to fluctuations and defects in graphene [20], [21].

In the transport energy range of interest, the DOS of metal contact is approximately constant, and its contact self-energy can be expressed as

$$\Sigma_2(E) = -it_0 \tag{5}$$

where $t_0 = \hbar^2/(2m_{\mathrm{e}}\Delta x^2)$, m_{e} is the effective mass, and Δx is the grid spacing.

The transmissions can be computed as [8]

$$T(E) = \mathrm{trace}\left(G\Gamma_1 G^{+}\Gamma_2\right) \tag{6}$$

where $\Gamma_{1,2} = -2\mathrm{Im}(\Sigma_{1,2})$ is the broadening function of the electrodes. In the presence of elastic scattering or at the ballistic limit, the current can be computed from the Landauer–Büttiker formula [8]

$$I = \int dE \cdot T(E)\left[f_{2\mathrm{D},1}(E) - f_{2\mathrm{D},2}(E)\right] \tag{7}$$

where $f_{2\mathrm{D},i}(E) = N_0 \ln(1+\exp(-(E-E_{F,i})/k_{\mathrm{B}}t))$ is derived by summing over the transverse modes, where $i = 1, 2$ is for graphene contact and metal contact, $N_0 = m_{\mathrm{e}}k_{\mathrm{B}}t/(2\pi\hbar^2)$ is the density constant, where k_{B} is the Boltzmann's constant, and t is the temperature [19].

For the GFM FTJ device as shown in Fig. 1(b), the device parameters of interest in this work include six most important parameters, the barrier heights Φ_1, Φ_2, the thickness of the FE layer t_{fe}, the scattering strength S, the Dirac energy of graphene E_{d}, and the effective mass m_{e}. Although the ML-based framework only takes six parameters for demonstration, it is extendible to higher dimensions of the device parameter space.

B. DL for Dimensionality Reduction of Output Device Properties

As described above, the target of the ML method is to learn $\log(T(E))$. A straightforward choice of the basis function of the transmissions is the numerical grid basis. For a typical simulated energy range of ~1 eV and a grid spacing of ~1 meV, the size of the basis set determined by the number of energy grid points is ~1000. One issue that needs to be addressed is that since $\log(T(E))$ is discretized as a vector, the large size of the basis set leads to the high dimensionality of the learning target. To reduce the dimensionality of the learning target for computationally efficient ML, it is preferred to describe the target function, which is the transmission versus energy here, in a feature space.

Hence, we investigate the DL as an efficient way to reduce the feature size of the transmissions, so that the $\log(T(E))$ can be replaced with a sparse representation learned from DL [22], [23]. DL can efficiently generate a dictionary of the target data, which can map the high-dimension original data to a low-dimension processed data. With this dictionary, the transform between the original data and the new data can be simply a sparse coding step sharing the same implementation. In DL, the target function is represented by a linear combination of a feature basis set which can be learned from the data set, $X = D\alpha$, where D is an $m \times k$ dictionary matrix, m is the length of a data set vector, k is the size of the dictionary basis set, and $\alpha = [\alpha_{\mathbf{1}}, \alpha_{\mathbf{2}}, \ldots, \alpha_{\boldsymbol{n}}]$ are the coefficients that represent the data set of $X = [X_{\mathbf{1}}, X_{\mathbf{2}}, \ldots, X_{\boldsymbol{n}}]$ in the sparse dictionary representation, where n is the length of the data set. Here, the coefficient matrix α has a dimension of $k \times n$, and the data X matrix has a size of $m \times n$. In DL, an empirical cost function is optimized

$$C(D) = \frac{1}{n}\sum_{j=1}^{n} l\left(X_j, X_{\mathrm{D}j}\right) \tag{8}$$

where the cost function can be defined as the Euclidean distance between a data vector X_j and its corresponding vector constructed from the dictionary $X_{\mathrm{D}j} = D\alpha_j$.

To perform DL, the cost function $C(D)$ is iteratively minimized with the step-by-step procedure.

1) For a given dictionary matrix D, the orthogonal matching pursuit (OMP) algorithm is used to determine the coefficient matrix α. The OMP algorithm is a sparse coding algorithm that finds the closest matching project for a given data set X onto a dictionary.
2) For a given coefficient matrix α of a data set X, use DL to obtain the dictionary D by optimizing the cost function $C(D)$.

TABLE I
COMPARISON OF THE MAIN PROPERTIES OF THE REGRESSION ALGORITHMS

Regression algorithm	LASSO	EN	KRR	RF
Linear regression	×	×	×	
Non-Linear			×	×
Good for feature selection	×			×
Time efficiency	×	×	×	
Ensemble method				×

The above two steps are iterated until a specified condition, such as a threshold value for the cost function, is met. The iterative procedures output a dictionary D which is essentially a basis set and the coefficient matrix α which is essentially the sparse representation of the data set X in the dictionary basis set.

Specifically, the DL is accelerated with a minibatch approach. This approach divides the data into minibatches and optimizes in an online manner with cycling over the minibatches for the specified number of iterations.

C. Regression Algorithms to Link Device Parameters to Device Properties

The regression block, as shown in Fig. 2, establishes a regression relation between the device parameters and device properties. Regression algorithms focus on the analysis of the relationship between the variables which iteratively refine the modeling with the error in the prediction based on the model [24]. There are several different classes of regression algorithms, for example, kernel regression, regularization method regression, and decision tree regression. Four popular regression algorithms, least absolute shrinkage and selection operator (LASSO) regression (regularization algorithm) [25], [26], elastic net (EN) [26], kernel ridge regression (KRR) (regularization with kernel algorithm) [27], and random forest (RF) regression (decision tree algorithm) [28] are implemented to establish a mapping relation between the device parameters and device properties. The comparison of the main properties of the regression algorithms is listed in Table I.

The first learning scheme adopted to learn the mapping relation between the device representation and the $\log(T(E))$ is LASSO regression. LASSO regression is a type of linear regression trying to shrink the data values toward a central point, like the mean. It performs the L1 regularization, which gives a penalty on the absolute value of coefficients. This will return a sparse model with relatively few coefficients. If there is a group of highly correlated variables, LASSO regression has the trend to only select one of them. This property makes LASSO well suited for feature elimination and selection. LASSO, however, has limitations in the scenarios that the feature dimension is much larger than the data points.

Another implemented learning method is EN. It considers the quadratic penalty part and the absolute value part, which is a combination of LASSO regression and ridge regression. EN is shown that it often outperforms the LASSO in terms of prediction accuracy [29], and it suits better if only a few training data points are provided.

TABLE II
INPUT DEVICE PARAMETER SEARCH SPACE

Parameter	Φ_1	Φ_2	t_{fe}	S	E_d	m_e
Meaning	Barrier height 1	Barrier height 2	FE thickness	Scattering parameter	Graphene Dirac point	Effective mass
Unit	eV	eV	nm	eV^2	eV	m_0
Lower limit	0.8	0.7	1.5	0	-0.5	0.3
Upper limit	2.6	2.5	3.3	0.8	0.2	0.9

m_0 is the free electron mass.

To go beyond linear regression, KRR is implemented, which is a special ridge regression with kernel-based tricks. KRR can learn the linear function in the space of the kernel feature, which could be the nonlinear function on the original space if a nonlinear kernel is adopted. In this case, the radial basis function (RBF) kernel is used to describe the mapping relation. Combined with L2 regularization, the squared loss is taken as the loss function in KRR. Unlike the L1 regularization which trends to return to a sparse model, the L2 regularization trends to set all the coefficients to a relatively small value. This characteristic makes the ridge regression not suitable for feature selection.

The last adopted learning method is RF. RF regression is an ensemble learning model. It is more generalizable and more robust than other methods. It operates by combining a host of decision trees when training and outputting the mean prediction of the individual decision tree. As for now, it is still one of the most accurate learning algorithms available. Another advantage is that the RF can directly handle thousands of input features, which makes it a good choice for operating on our data set with constructed features. Furthermore, RF can also be implemented in feature selection as it provides the importance of each input feature.

D. Feature Engineering of Device Parameters

As discussed before, the goal of the ML-based framework is to discover a general relationship between the properties of interest and the device parameters. The feature engineering block in Fig. 2 targets on identifying more compact or even analytical relations by focusing on the input device parameter space.

The primary features are set as the six input device parameters, that is, $\{\Phi_1, \Phi_2, t_{\mathrm{fe}}, S, E_{\mathrm{d}}, m_{\mathrm{e}}\}$, as listed in Table II. Based on the above six primary features, a total of 2232 data features are constructed. Applying the 12 prototype functions, $x, x^{-1}, x^{0.5}, x^{-0.5}, x^2, x^{-2}, x^3, x^{-3}, \ln x, (\ln x)^{-1}, e^x$, and e^{-x}, 72 new one-term features are immediately generated considering x as each of the primary features in order. Similarly, choosing any two of the six primary features, and implementing the 12 prototype functions on both of the chosen features, can lead to 2160 ($C_6^2 \times 12 \times 12$) new two-term features. Thus, there are 2232 one-term and two-term features in total.

A feature selection algorithm, recursive feature elimination (RFE), is applied to the newly composed data set for feature down selection. RFE can be used along with any estimator that

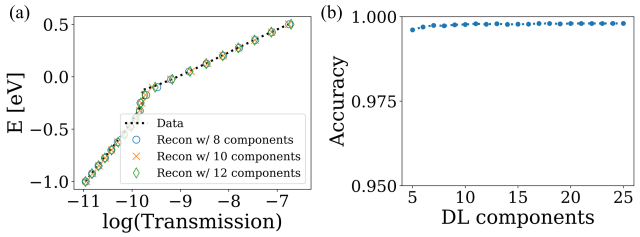

Fig. 3. Accuracy of DL. (a) Logarithmic transmission versus energy reconstructed from the DL with different number of features (DL components), compared to the data from the NEGF simulations, for a GFM FTJ as shown in Fig. 1(b). (b) DL reconstructed $\log(T(E))$ versus the DL components. The simulated FTJ device is shown in Fig. 1(b), with a polarization of $P = 10\ \mu C/cm^2$, the barrier heights of $\phi_1 = 1.40$ eV and $\phi_2 = 2.00$ eV, an FE layer thickness of $t_{FE} = 2.30$ nm, the effective mass of $m_e = 0.40m_0$ (m_0 is the free electron mass), and biased at $V_D = 0.4$ V. The screening length of metal contact is $a_2 = 1$ Å. The scattering strength is $S = 0.3\ eV^2$.

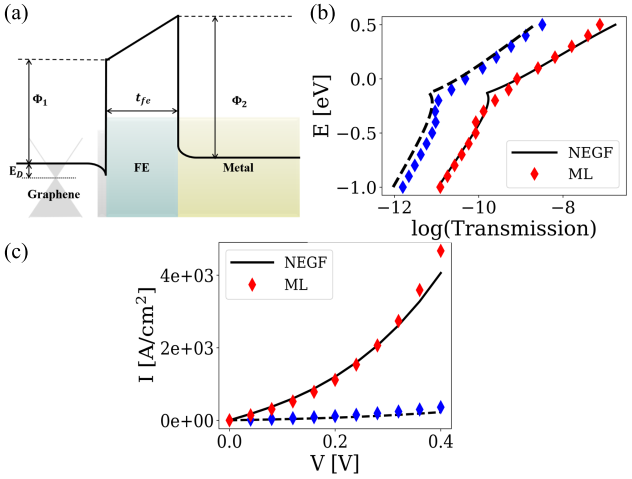

Fig. 4. (a) Band diagram of the GFM FTJ device at ON-state. (b) Transmission versus energy in the presence of scattering obtained by the ML-based model compared to the NEGF simulation result. The lines are from the NEGF simulation for the ON FE polarization state (solid line) and the OFF-state (dashed line), and the diamonds are the transmission predicted from the ML model for the ON-state (red) and OFF-state (blue) with DL basis. (c) ON and OFF current versus the applied voltage from the NEGF simulation and the ML-based model. The simulated FTJ device is the same as that in Fig. 3.

returns a feature-importance coefficient, for example, LASSO and RF. Based on a trained ML model, the unimportant features are eliminated, determined by comparing the threshold with the corresponding feature importance. Next, a new learning model is trained based on the remaining features. This recursive process will iteratively continue until the remaining features reduced to a specified value. The selected features will be tested and analyzed together with the data and the trained ML model.

E. Workflow of the Device Simulation With ML

The ML-based framework supported by the NEGF simulations works as follows.

1) Generate the training and testing data sets of device properties from device parameters by using the NEGF simulations.
2) Adapt DL for feature reduction of the device properties.
3) Train the ML model by using a regressor with the training data set.
4) Generate new features based on the original parameters and perform feature selection to get the most important features.
5) Analyze and test the selected features with the original ML model and the data set.

It is worth mentioning that the LASSO regressor is implemented not only in the regression part but also in the feature engineering part.

III. Results and Discussions

We first examine how to represent the output device property of interest, $\log(T(E))$, by using the DL method. While for the NEGF simulations of the FTJ device, $\log(T(E))$ is a vector of >100 elements, the goal of the DL is to reduce the feature size of the device property, so that it facilitates subsequent regression processes. Fig. 3(a) shows the transmissions computed by NEGF, compared to those reconstructed from DL with 8, 10, and 12 components. The $\log(T(E))$ vector has 300 elements, and the DL procedure essentially reduces the property vector size to 8, 10, and 12. The results show that despite much smaller vector size, the DL reproduces the transmission vector with high accuracy. In order to further understand the dependence of the accuracy on the DL vector size, Fig. 3(b) shows the accuracy of $\log(T(E))$ reconstruction as a function of the dictionary components. The results indicate that even for a dictionary size down to 5, the reconstruction accuracy is sufficiently high with >0.99. In this work, a dictionary size of ten components is implemented. By this way, the dimensionality of the learning target is reduced by 30 times from the 300 elements of $\log(T(E))$ to ten components of DL, which renders a high accuracy of approximately 0.995.

Next, we examine the results of different regression algorithms in the regression block with the ten-component DL. The device parameter space $\{\Phi_1, \Phi_2, t_{fe}, S, E_d, m_e\}$, as mentioned in Section II-D, has the range as listed in Table II. A total of 2500 device samples, which follow a uniform distribution in the device parameter space, are simulated. Fifty percent of the data are used as training data and 50% of the data are used as the testing data and cross-validation is implemented. Fig. 4 shows the comparison of the results computed by NEGF and those predicted by the ML model. The band diagram of the GFM device at ON-state and OFF-state is shown in Fig. 4(a). Fig. 4(b) compares the predicted transmissions and the NEGF simulation results. The current computed by the two sets of transmission data is shown in Fig. 4(c). It demonstrates that the ML model trained based on 1250 data points is applicable for the prediction of the transmissions and the corresponding current of the device.

It is desirable to be able to achieve high accuracy with a small training data set. The relationship between the accuracy and the size of the training data set is examined next. Fig. 5(a) and (b) shows the mean absolute relative error (MARE) and the root-mean-square error (RMSE) of the learning results based on training sets with 12–1250 datapoints. Here, MARE $= 1/(N_d \times N_e)$

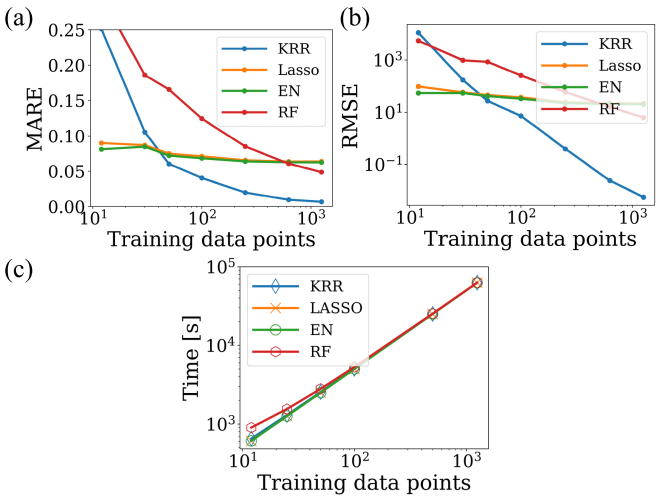

Fig. 5. Comparison of accuracy and time for four regression methods. (a) MARE and (b) RMSE of the ML result with different regressors versus the number of data points in training data set. (c) Total simulation time versus the number of data points in training data set with different regressors. The time is the sum of the NEGF simulation time for obtaining the training data and model training time.

$\Sigma_1^{N_d}\Sigma_1^{N_e}|(\log(T_{NEGF}) - \log(T_{pred}))/\log(T_{NEGF})|$, where N_d is the number of test datapoints, N_e is the number of energy gird, T_{NEGF} and T_{pred} are the transmission from the simulation and ML model, respectively, and RMSE $= (1/(N_d \times N_e)(\Sigma_1^{N_d}\Sigma_1^{N_e}(\log(T_{NEGF}) - \log(T_{pred}))^2))^{1/2}$. The results indicate that a prediction of $\log(T(E))$ with a low MARE (<0.1) can be achieved by all the four regressors with a training data set larger than 1000, but the increase of the training data set size adds computational costs.

To understand the computational resource cost, the total time cost, which consists of the time for the sum of time for obtaining the training data and time for training the ML model, is shown in Fig. 5(c), as a function of the training data points. Fig. 5(c) shows that increasing the training data size comes with a proportional increase of the total time, and an ML model that can be accurate for a smaller data training data set, therefore, is more computationally preferable. To achieve a small error of MARE < 0.05, the KRR regressor performs best and requires the smallest data set size of larger than 50, as shown in Fig. 5(a) and (b), which requires the lowest computational cost for training the model among the four regression methods studied.

Once the ML model is trained, it can offer significant improvement of computational time compared to the NEGF simulations for predicting the quantum transmission. For example, obtaining the transmissions of 1250 points with NEGF requires about 10^5 s on an Intel I7-3770K, which indicates that the direct NEGF simulations for extensive device design will be computationally intensive. In contrast, once the ML model is trained, the prediction from the model is <10 s, which is at least 10^4 times faster. The ML-based method, therefore, is suitable for fast, high-throughput device design and optimization in a large design space.

To further discover the impact and importance of each parameter on the device properties, a feature construction and selection procedure is used. The selected features can not only improve the regression in terms of time efficiency and accuracy but also provide the inner link or even an analytical expression between the device parameters and properties. The new one-term and two-term features are generated as stated in Section II-D. The LASSO estimator is tested in the RFE for its capability to return the feature importance coefficients. The procedures, as described in detail in Section II-D, identify that in addition to the original six-parameter feature set, the most important one-term and two-term features ranked in order include $\sqrt{\Phi_1}$, $\sqrt{\Phi_2}$, E_d^2, $(\Phi_1 m_e)^{1/2}$, and $(\Phi_2 m_e)^{1/2}$. By taking advantage of the features identified, it is possible to predict the transmission with higher accuracy.

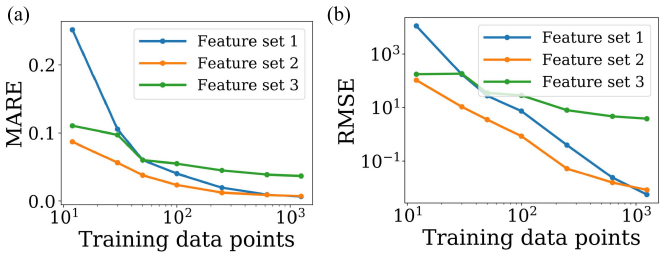

Fig. 6. Comparison of the feature sets. (a) MARE. (b) RMSE of the logarithm quantum transmission of the simulated FTJ device versus the number of data points of training data sets for three feature sets as described in text.

To examine the choice of the feature set, the following three feature sets are compared with the same regressor, KRR, and the same original learning data set in Fig. 6:

Feature set 1: $\{\Phi_1, \Phi_2, t_{fe}, S, E_d, m_e\}$;
Feature set 2: $\{\sqrt{\Phi_1}, \sqrt{\Phi_2}, t_{fe}, S, E_d^2, \sqrt{m_e}\}$;
Feature set 3: $\{\sqrt{\Phi_1 m_e}, \sqrt{\Phi_2 m_e}, t_{fe}, S, E_d^2\}$,

where Feature set 1 is the original parameter set, Feature set 2 and Feature set 3 include the identified important features with the rest of the original features. For a new feature set, first, the device parameters are processed, calculating the values of the corresponding features, to obtain the data for training and testing. Then the DL is applied, and an ML model is trained based on it. The performance of a feature set can be quantified in the same way by plotting the MARE and RMSE as a function of the size of the training data points.

Fig. 6 compares the accuracy of the three different feature sets. The results show that Feature set 2 can reach the MARE of 0.05 by only 50 datapoints, which is comparable to the result of Feature set 1 with 100 datapoints. Mapping the results to the I–V characteristics, based on Feature set 2, the MARE of current could be smaller than 0.09 on the 1250 tested device samples. Benefited from the more complex features, Feature set 3 also shows the improvement on a small data set, but the improvement is smaller than Feature set 2. The results indicate that a feature down selection procedure by using RFE can help to improve the regression and unveil the inner or even analytical relation between parameters and properties of interest.

Although a simple FTJ device is studied for illustration, the ML framework for modeling quantum transport in devices is built based on a general procedure by first reducing the dimensionality of the device properties by DL, then linking the input device parameters to device properties by ML regression

methods. Feature engineering further allows simplifying the input feature space. The framework can be further extended to a more complex device data set for more complex device structures in future, as DL has been proved to be capable of handling a larger and higher dimensional data set [22]. With the high-accuracy learned dictionary of transmissions, the regression and feature engineering modules, as shown in Fig. 2, can train a model whose accuracy can be improved by enlarging the DL components, expanding the feature space, and increasing the learning precision. Thus, the device properties of interest, such as the transmissions and I–V characteristics, can be predicted with good accuracy from the input device parameters.

IV. Conclusion

An ML-based framework for replacing the computationally intensive part of the NEGF device simulation is proposed. The results show as follows.

1) By using DL, the dimensionalities of the device properties of interest can be significantly reduced remaining the main information.
2) Four regression algorithms are investigated and compared in terms of accuracy and computation time, indicating that high prediction accuracy can be achieved.
3) The RFE procedures can be used to further improve the accuracy and discover the inner relation from the device parameters to device properties.

The framework presented here demonstrates the potential of the ML techniques for fast and high-throughput simulation and design of quantum-effect devices.

Acknowledgment

The authors would like to thank Prof. Han Wang of the University of Southern California for helpful technical discussions.

References

[1] K.-T. Lam, X. Cao, and J. Guo, "Device performance of heterojunction tunneling field-effect transistors based on transition metal dichalcogenide monolayer," *IEEE Electron Device Lett.*, vol. 34, no. 10, pp. 1331–1333, Oct. 2013.

[2] L. L. Chang and L. Esaki, "Tunnel triode—A tunneling base transistor," *Appl. Phys. Lett.*, vol. 31, no. 10, pp. 687–689, Nov. 1977, doi: 10.1063/1.89505.

[3] S. Ikeda *et al.*, "A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction," *Nature Mater.*, vol. 9, no. 9, pp. 721–724, Jul. 2010, doi: 10.1038/nmat2804.

[4] W. J. Gallagher, J. H. Kaufman, S. S. P. Parkin, and R. E. Scheuerlein, "Magnetic memory array using magnetic tunnel junction devices in the memory cells," U.S. Patant 5 640 343, Jun. 17, 1997. [Online]. Available: http://www.freepatentsonline.com/5640343.html

[5] G. Gerra, A. K. Tagantsev, N. Setter, and K. Parlinski, "Ionic polarizability of conductive metal oxides and critical thickness for ferroelectricity in $BaTiO_3$," *Phys. Rev. Lett.*, vol. 96, no. 10, Mar. 2006, Art. no. 107603, doi: 10.1103/PhysRevLett.96.107603.

[6] M. Y. Zhuravlev, R. F. Sabirianov, S. S. Jaswal, and E. Y. Tsymbal, "Giant electroresistance in ferroelectric tunnel junctions," *Phys. Rev. Lett.*, vol. 94, no. 24, Jun. 2005, Art. no. 246802, doi: 10.1103/PhysRevLett.94.246802.

[7] Z. Dong, X. Cao, T. Wu, and J. Guo, "Tunneling current in HfO_2 and $Hf_{0.5}Zr_{0.5}O_2$-based ferroelectric tunnel junction," *J. Appl. Phys.*, vol. 123, no. 9, Mar. 2018, Art. no. 094501, doi: 10.1063/1.5016823.

[8] S. Datta, *Quantum Transport: Atom to Transistor*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[9] S. Datta, "The non-equilibrium green's function (NEGF) formalism: An elementary introduction," in *IEDM Tech. Dig.*, Jan. 2002, pp. 703–706.

[10] R. Venugopal, Z. Ren, S. Datta, M. S. Lundstrom, and D. Jovanovic, "Simulating quantum transport in nanoscale transistors: Real versus mode-space approaches," *J. Appl. Phys.*, vol. 92, no. 7, pp. 3730–3739, Oct. 2002, doi: 10.1063/1.1503165.

[11] B. Sanchez-Lengeling and A. Aspuru-Guzik, "Inverse molecular design using machine learning: Generative models for matter engineering," *Science*, vol. 361, no. 6400, pp. 360–365, Jul. 2018, doi: 10.1126/science.aat2663.

[12] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Bypassing the kohn-sham equations with machine learning," *Nature Commun.*, vol. 8, no. 1, pp. 1–10, Oct. 2017, doi: 10.1038/s41467-017-00839-3.

[13] E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev, and A. R. Oganov, "Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning," *Phys. Rev. B, Condens. Matter*, vol. 99, no. 6, Feb. 2019, Art. no. 064114, doi: 10.1103/PhysRevB.99.064114.

[14] C. Kim, G. Pilania, and R. Ramprasad, "From organized high-throughput data to phenomenological theory using machine learning: The example of dielectric breakdown," *Chem. Mater.*, vol. 28, no. 5, pp. 1304–1311, Feb. 2016, doi: 10.1021/acs.chemmater.5b04109.

[15] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, "Machine learning in materials informatics: Recent applications and prospects," *NPJ Comput. Mater.*, vol. 3, no. 1, pp. 1–13, Nov. 2017, doi: 10.1038/s41524-017-0056-5.

[16] J. Wu *et al.*, "High tunnelling electroresistance in a ferroelectric van der Waals heterojunction via giant barrier height modulation," *Nature Electron.*, vol. 3, no. 8, pp. 466–472, Aug. 2020, doi: 10.1038/s41928-020-0441-9.

[17] R. Golizadeh-Mojarad and S. Datta, "Nonequilibrium Green's function based models for dephasing in quantum transport," *Phys. Rev. B*, vol. 75, no. 8, Feb. 2007, Art. no. 081301, doi: 10.1103/PhysRevB.75.081301.

[18] D. Nikonov, H. Pal, and G. Bourianoff, "Scattering in NEGF: Made simple," Intel/Purdue, Tech. Rep., 2009. [Online]. Available: https://nanohub.org/resources/7772

[19] S. Datta, "Quantum transport," in *Quantum Transportation*, vol. 418, S. Datta, Ed. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[20] M. F. Borunda, J. Berezovsky, R. M. Westervelt, and E. J. Heller, "Imaging universal conductance fluctuations in graphene," *ACS Nano*, vol. 5, no. 5, pp. 3622–3627, May 2011, doi: 10.1021/nn103450d.

[21] S. Barraza-Lopez, M. Vaneviá, M. Kindermann, and M. Y. Chou, "Effects of metallic contacts on electron transport through graphene," *Phys. Rev. Lett.*, vol. 104, no. 7, Feb. 2010, Art. no. 076807, doi: 10.1103/PhysRevLett.104.076807.

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.

[23] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2011, doi: 10.1109/TPAMI.2011.156.

[24] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2006.

[25] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ_1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 1, no. 4, pp. 606–617, Jul. 2007.

[26] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, p. 1, Oct. 2010, doi: 10.18637/jss.v033.i01.

[27] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[28] G. Louppe, "Understanding random forests: From theory to practice," 2014, *arXiv:1407.7502*. [Online]. Available: http://arxiv.org/abs/1407.7502

[29] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc., Ser. B Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, Apr. 2005.

Preflight Results

Document Overview

Title: Speed Up Quantum Transport Device Simulation on Ferroelectric Tunnel Junction with Machine Learning Methods
Author:
Creator: Aspose Ltd.
Producer: Aspose.Pdf for .NET 8.3.0; modified using iText® 5.5.6 ©2000-2015 iText Group NV (AGPL-version)

Preflight Information

Profile: Convert to PDF/A-2b
Version: Qoppa jPDFPreflight v2020R2.01
Date: Jul 4, 2021 4:23:48 PM

Legend: (X) - Can NOT be fixed by PDF/A-2b conversion.
(!X) - Could be fixed by PDF/A-2b conversion. User chose to be warned in PDF/A settings.

Document Results

(X) Embedded file found but it isn't PDF/A-1b or PDF/A-2b compliant