Deep-Learning-Assisted MOSFET Current-Voltage Model with Symmetric Fourier Features

Hikaru A. Kaneshiro[1]

Abstract— **In this work, we utilize symmetric Fourier features to improve the accuracy of deep learning-augmented MOSFET current-voltage model. We first apply the Fourier analysis on the cost function and discuss the spectral bias of neural network. The benefits of augmenting the neural network with symmetric Fourier features, designed to satisfy the Gummel symmetry, are then discussed. Using TCAD-simulated FinFET as our dataset, our improved model, augmented with 8 Fourier features, achieves outstanding 3-sigma errors of 0.15% for IV characteristics, 4.14% for output conductance, and 0.33% for transconductance. This represents a cost reduction of 13 times compared to the scenario without the Fourier features, highlighting the benefits gained from their inclusion.**

Index Terms—**MOSFET, deep learning, compact model, Fourier features**

I. Introduction

Deep-learning approaches have recently gained significant traction in semiconductor device modeling, particularly for MOSFET models [1-5]. These techniques aim to improve computational efficiency and enhance accuracy. Recent studies have demonstrated that neural network-based models can achieve a root mean square error of less than 1% for MOSFET current-voltage (*IV*) characteristics [2], while providing a notable 7-fold increase in speed compared to traditional methods [4].

However, it has been noticed that many proposed models suffer from significant inaccuracies in output conductance *gds* and transconductance *gm*, with maximum errors surpassing the range of 30%. These inaccuracies hinder the application of the model in analog circuit designs.

To understand this observation, Fourier analysis is first applied to the cost function associated with the neural network, as detailed in Section II. Utilizing the simulated 12nm FinFET detailed in Section III as the dataset, Section IV illustrates that neural networks exhibit slow convergence rates at specific frequencies in the Fourier domain. This phenomenon is commonly referred to as the spectral bias. It hinders model accuracy, particularly for the derivatives.

Building upon the Fourier features previously proposed in [6], Section V introduces symmetric Fourier features to address the issue of spectral bias. Section VI then presents the modeling results using simulated *IV* characteristics as an example. Finally, section VII concludes this paper.

II. Fourier Analysis on Neural Network

Deep learning augmented models incorporate a physics-based *I-V* model $I_{D,M}$, which is bolstered by the inclusion of a bias-dependent correction function $\varepsilon(V_{GS},V_{GD}) \equiv \exp(h(V_{GS},V_{GD}))$ [1-4]:

$$I_D(V_{GS},V_{GD}) = I_{D,M}(V_{GS},V_{GD}) \times exp\big(h(V_{GS},V_{GD})\big) \tag{1}$$

where h is to be trained by the neural network.

Cost function J is the sum of the error in I_D and I_D sensitivities to the terminal voltages. Since I_D varies across a wide range, it is more convenient to express the cost in terms of both I_D $\log(I_D)$:

$$J = \frac{1}{I_{MEAN}^2}\left[MSE(I_D) + MSE\left(\frac{dI_D}{dV_{GS}}\right) + MSE\left(\frac{dI_D}{dV_{DS}}\right)\right] + MSE\big(log(I_D)\big) + MSE\left(\frac{dlog(I_D)}{dV_{GS}}\right) + MSE\left(\frac{dlog(I_D)}{dV_{DS}}\right) \tag{2}$$

where I_{MEAN} is a normalization factor and represents the mean current for the training dataset. MSE is the mean square error.

Substitute (1) into (2), we get:

$$J = \frac{1}{I_{MEAN}^2}\left[MSE(I_D) + MSE\left(\frac{dI_D}{dV_{GS}}\right) + MSE\left(\frac{dI_D}{dV_{DS}}\right)\right] + MSE(h) + MSE\left(\frac{dh}{dV_{GS}}\right) + MSE\left(\frac{dh}{dV_{DS}}\right) \tag{3}$$

Assume the MOSFET *IV* characteristics are sampled at a uniform grid in V_{GS} and V_{DS}, with dimensions m and n, respectively, the MSE terms in (3) can be expressed by:

$$MSE(x) = \frac{1}{m\times n}\sum_{j=1}^{n}\sum_{i=1}^{m}\left(x_{net}^{(i,j)} - x_{data}^{(i,j)}\right)^2 \tag{4}$$

where x is the variable of interest, the subscript *net* refers to the neural network output, while *data* corresponds to the measured data.

By utilizing Parseval's theorem [7], the mean square error can be expressed through the Fourier transform:

$$MSE(I_D) = \sum_{j=1}^{n}\sum_{i=1}^{m}\left|\mathfrak{I}_{net}^{(i,j)} - \mathfrak{I}_{data}^{(i,j)}\right|^2$$
$$MSE(h) = \sum_{j=1}^{n}\sum_{i=1}^{m}\left|\mathrm{H}_{net}^{(i,j)} - H_{data}^{(i,j)}\right|^2 \tag{5}$$

where $\mathfrak{I}(k_{GS},k_{DS})$ and $H(k_{GS},k_{DS})$ are the-two dimensional Fourier Transform of $I_D(V_{GS},V_{DS})$ and $h(V_{GS},V_{DS})$ respectively, and k_{GS},k_{DS} are the Fourier basis frequencies:

$$k_{GS} = \frac{2\pi i}{range(V_{GS})} \qquad \text{for i = -(m-1)/2 to } (m\text{-}1)/2$$
$$k_{DS} = \frac{2\pi j}{range(V_{DS})} \qquad \text{for j = -(n-1)/2 to } (n\text{-}1)/2 \tag{6}$$

Note that the dimension of $\mathfrak{I}$ and H are both $m\times n$.

Next, since differentiation involves multiplying the Fourier transform of a signal by its frequency [7], the mean square errors of the gradients are:

[1] Corresponding Author: Hikaru A. Kaneshiro (hikaru.a.kaneshiro@gmail.com)

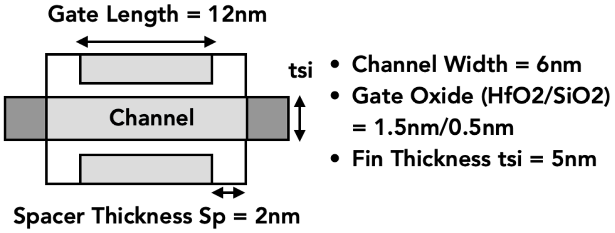
Gate Length = 12nm
tsi
Channel
• Channel Width = 6nm
• Gate Oxide (HfO2/SiO2) = 1.5nm/0.5nm
• Fin Thickness tsi = 5nm
Spacer Thickness Sp = 2nm

Fig 1. Cross-sectional schematic of the simulated FinFET structure.

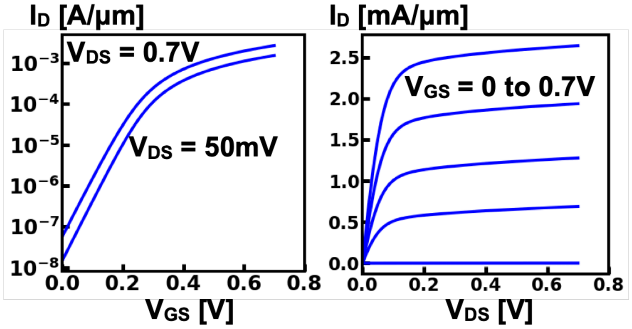

Fig 2. Simulated transfer and output *IV* characteristics.

$$MSE\left(\frac{\partial I_D}{\partial V_{GS}}\right) = \sum_{j=1}^{n}\sum_{i=1}^{m} k_{GS}^2\left|\left(\Im_{net}^{(i,j)} - \Im_{data}^{(i,j)}\right)\right|^2$$
$$MSE\left(\frac{\partial h}{\partial V_{GS}}\right) = \sum_{j=1}^{n}\sum_{i=1}^{m} k_{GS}^2\left|\left(\mathrm{H}_{net}^{(i,j)} - H_{data}^{(i,j)}\right)\right|^2$$
$$MSE\left(\frac{\partial I_D}{\partial V_{DS}}\right) = \sum_{j=1}^{n}\sum_{i=1}^{m} k_{DS}^2\left|\left(\Im_{net}^{(i,j)} - \Im_{data}^{(i,j)}\right)\right|^2$$
$$MSE\left(\frac{\partial h}{\partial V_{DS}}\right) = \sum_{j=1}^{n}\sum_{i=1}^{m} k_{DS}^2\left|\left(\mathrm{H}_{net}^{(i,j)} - H_{data}^{(i,j)}\right)\right|^2 \qquad (7)$$

Substitute (5) and (7) into (3), the cost function can be expressed in a compact manner:

$$J = \sum_{j=1}^{n}\sum_{i=1}^{m}\left(\left(\frac{1}{I_{MEAN}^2}\left|\Im_{net}^{(i,j)} - \Im_{data}^{(i,j)}\right|^2 + \left|\mathrm{H}_{net}^{(i,j)} - H_{data}^{(i,j)}\right|^2\right)\cdot \left(1 + k_{GS}^2 + k_{DS}^2\right)\right) \qquad (8)$$

which is the weighted sum of the squared magnitude errors in the Fourier coefficients ℑ and H. Note that, given k_{GS} and k_{DS} are greater than 1, the costs in predicting derivatives are larger than those in predicting the function itself. This not only explains the observations in previous studies [1-4], but also underscores the importance of accurate Fourier coefficient prediction in minimizing the overall cost. Before doing so, we will first discuss the *IV* data collection and spectral bias in neural networks.

III. Dataset and Neural Network

Fig. 1 shows the device structure for the 12nm gate length FinFET transistor used in this work [8]. Physics models such as the Fermi Dirac Distribution, Philips unified mobility model [9], and Fermi statistics and quantization effect are included in the device simulation.

Fig. 2 shows the simulated *IV* characteristics for the transistor. In this study, we use a 3-parameter model [10] for $I_{D,M}$:

$$I_{D,M}(V_{GS}, V_{GD}) = P[\phi(V_{GS})^2 - \phi(V_{GD})^2],$$

where $\phi(V) = V_{SS}log\left(1 + exp\left(\frac{V - V_T}{V_{SS}}\right)\right) \qquad (9)$

with P =33.7mA/V², V_{SS}= 57.5mV and V_T= 0.25V. These parameter values are extracted from the simulated I_D-V_{GS} data with a gate voltage step size of 10 mV, ranging from 0 to 0.7 V, and a V_{DS} of 50 mV.

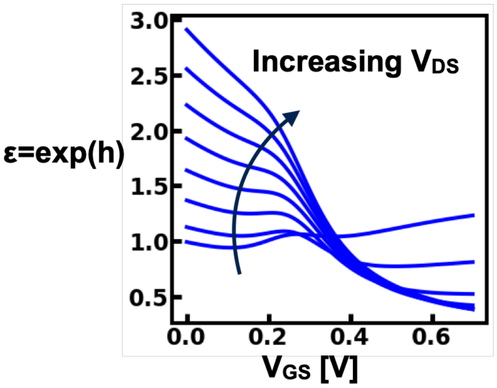

Fig 3. Computed error correction function ε=exp(h).

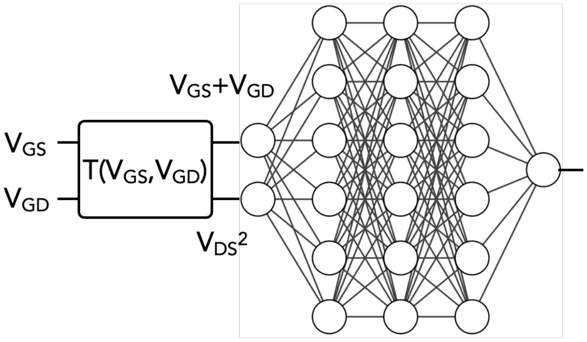

Fig. 4. Reference neural network. Input (V_{GS},V_{GD}) is first transformed to satisfy the Gummel symmetry [2].

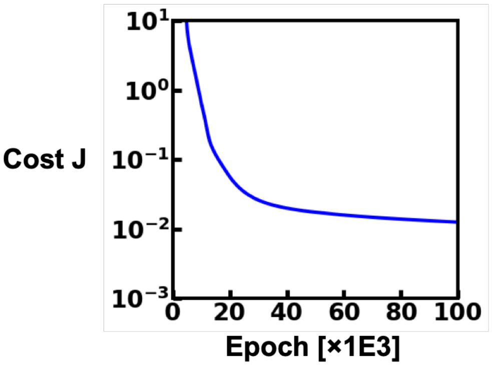

Fig. 5: Cost vs. Epoch for 100,000 epochs of training.

For training, we use V_{GS} and V_{DS} ranging from 0V to 0.7V with a step size of 20mV as the dataset. For the test set, we use V_{GS} and V_{DS} within the same range, each with a 10mV step size. This corresponds to a training and test dataset of size 1296 and 5041, respectively. The mean current for the train dataset, I_{MEAN}, is 684μA. Fig. 3 shows the computed ε(V_{GS},V_{DS}) using the dataset and $I_{D,M}$. As shown in the figure, ε ranges from 0.39 to 2.91.

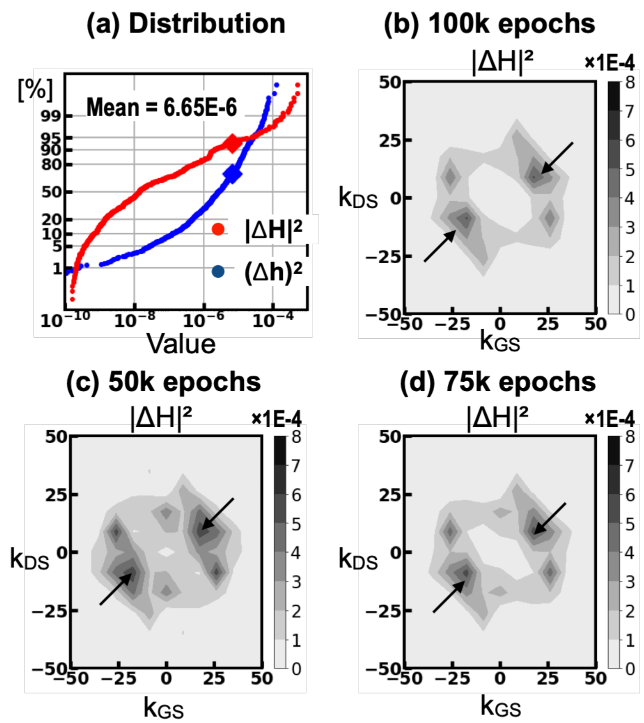

Fig. 6(a): Distribution of $(\Delta h)^2$ (input) and $|\Delta H|^2$ (output) after 100k epochs of training. (b-d): Contour of $|\Delta H|^2$ after 100k,50k and 75k epochs of training, with the peaks all the same at $(k_{GS},k_{DS}) = \pm(17.45V^{-1}, 8.73V^{-1})$ as highlighted by the arrows.

Fig. 4 shows the reference neural network used for both training and inferring *h*. Our starting architecture is a 3-layer neural network with 6 neurons per layer. The hyperbolic tangent is used as the activation. As was discussed in [2], the transistor voltage biases (V_{GS}, V_{GD}) are first transformed into $(V_{GS} + V_{GD}, V_{DS}^2)$ so that the correction function is symmetric, i.e. $\varepsilon(V_{GS}, V_{GD}) = \varepsilon(V_{GD}, V_{GS})$. This ensures the *IV* model satisfies the Gummel symmetry. TensorFlow with Adam optimization (with learning rate=1E-4) is used for training. A single batch is fed into the neural network due to the small training size. With this setup, the model's accuracy limit due to the spectral bias will be discussed in the next section.

IV. Spectral Bias of Neural Work

The cost versus epoch for the networks after 100k training is depicted in Fig. 5. As evident from the figure, the cost reaches a plateau after 40k training. To understand this, Fig. 6(a) compares the distributions of random variables $(\Delta h)^2$ and $|\Delta H|^2$ at 100k epochs, where $(\Delta h)^2 \equiv (h_{net} - h_{data})^2$ and $|\Delta H|^2 \equiv m \times n \times\ |H_{net} - H_{data}|^2$. The calculated mean of $(\Delta h)^2$ and $|\Delta H|^2$ is equal, as expected by Parseval's theorem.

However, the shapes of the two distributions are noticeably different. $(\Delta h)^2$ has a tighter spread, with a standard deviation of 1.12E-5 and its maximum value below ~10 times the mean. In contrast, distribution $|\Delta H|^2$ has a wider spread characterized by a standard deviation of 3.75E-5. It is notably skewed by a few pronounced highfliers, reaching up to 100 times the mean.

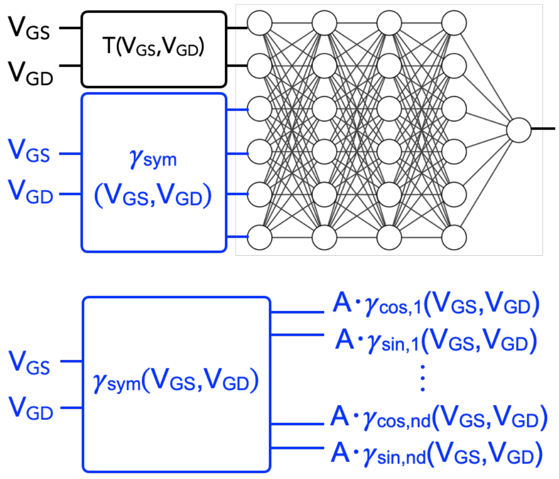

Fig. 7. Neural Network with *nd* pairs of symmetric Fourier features.

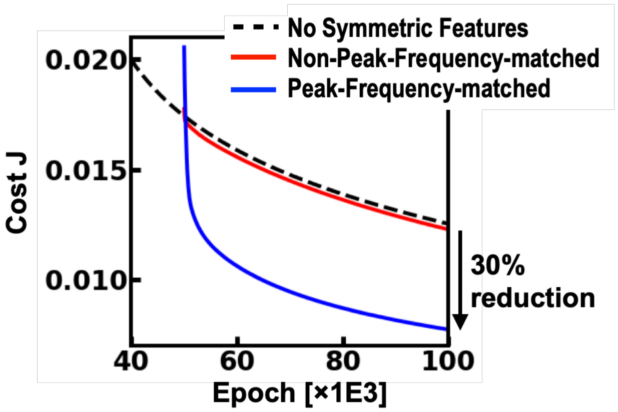

Fig. 8: Cost vs. epoch for a neural network with and without the symmetric Fourier features.

The two-dimensional contour plot for $|\Delta H|^2$ shown in Fig. 6(b) highlights these highfliers. As depicted in the figure, the maximum value of $|\Delta H|^2$ is located at $(k_{GS}, k_{DS}) = \pm(17.45/V , 8.73/V)^2$. Furthermore, this frequency peak remains stable as the epochs progress from 50k to 100k, as shown in Fig. 6(c) and (d). This suggests that the neural network is unable to capture the frequency feature at that specific frequency, resulting in a lower bound in the mean square error. Further training does not significantly reduce cost. Spectral analysis is similarly performed on $\Im$, and the same conclusion is drawn.

The phenomenon of slow convergence and the presence of frequency peaks in the Fourier domain is recognized as the spectral bias of neural networks [11-12]. Neural networks with Fourier features were previously proposed to eliminate these high error peaks [6], which is the focus of the next section.

V. Symmetric Fourier Features

A neural network with Fourier features maps input (V_{GS},V_{GD}) to a higher-dimensional space using sinusoids [6]. Integrating these features allows the model to learn information associated with specific frequencies. To satisfy Gummel symmetry, we

[2] Since $h(V_{GS},V_{GD})$ is a real function, $H(k_{GS},k_{GD})$ is conjugate symmetric, and therefore $|\Delta H|^2$ is symmetric with $|\Delta H(k_{GS}, k_{DS})|^2=|\Delta H(-k_{GS}, -k_{DS})|^2$.

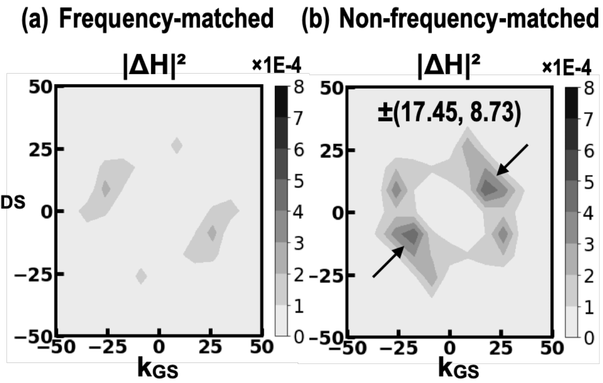

Fig. 9. Contour of $|\Delta H|^2$ after 100k epochs of training trained with (non) peak-frequency-matched Fourier feature.

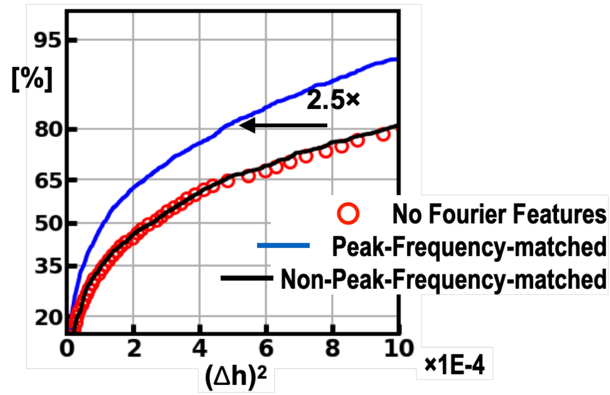

Fig. 10. Distribution plot for square error $(\Delta h)^2$, comparing neural network with (non) peak-frequency-matched features vs. without features.

extend such concept and introduce symmetric Fourier features γ_{sym} (V_{GS}, V_{GD}), as depicted in Fig. 7:

$$\gamma_{sym}(V_{GS},V_{GD}) = \begin{bmatrix} A\cdot\gamma_{cos,1}(V_{GS},V_{GD}) \\ A\cdot\gamma_{sin,1}(V_{GS},V_{GD}) \\ \vdots \\ A\cdot\gamma_{cos,nd}(V_{GS},V_{GD}) \\ A\cdot\gamma_{sin,nd}(V_{GS},V_{GD}) \end{bmatrix} \qquad (10)$$

where A is a prefactor, nd is the number of sinusoid pairs,

$$\gamma_{cos,i}(V_{GS},V_{GD}) \equiv \sigma(V_{DS}/V_0)\cdot\cos(\Omega_{GSi}V_{GS}+\Omega_{DSi}V_{DS}) + \left(1-\sigma(V_{DS}/V_0)\right)\cdot\cos(\Omega_{GSi}V_{GD}-\Omega_{DSi}V_{DS}) \qquad (11)$$

$$\gamma_{sin,i}(V_{GS},V_{GD}) \equiv \sigma(V_{DS}/V_0)\cdot\sin(\Omega_{GSi}V_{GS}+\Omega_{DSi}V_{DS}) + \left(1-\sigma(V_{DS}/V_0)\right)\cdot\sin(\Omega_{GSi}V_{GD}-\Omega_{DSi}V_{DS}) \qquad (12)$$

are the i-th user-defined Fourier feature with frequencies (Ω_{GSi}, Ω_{DSi}) and

$\sigma(V_{DS}/V_0) = \frac{1}{1+exp(V_{DS}/V_0)}$ is the sigmoid function. V_0 is a constant that determines the transition steepness. In this work V_0 is set to be 0.0256V. The proof for symmetry is detailed in the Appendix. For simplicity, we'll term these features 'Fourier features' in subsequent discussions. In the next section, we'll demonstrate how the systematic selection of these features enhances model prediction accuracy. Three examples will be presented: a neural network with one, four, and eight feature pairs.

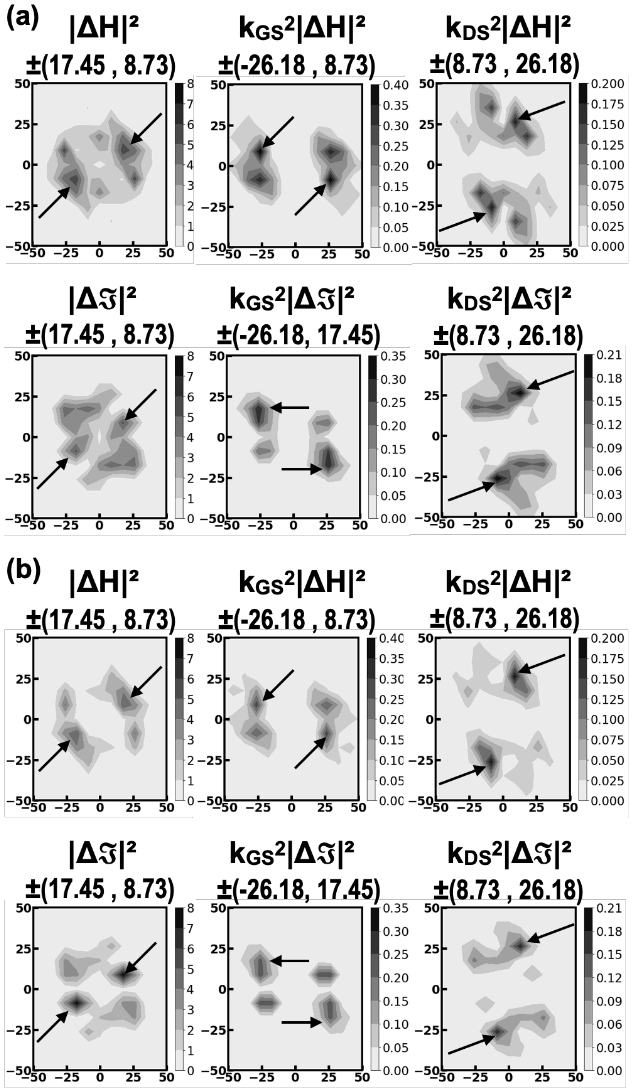

Fig. 11. Contour of the six cost components after (a) 50k and (b) 100k epochs of training, without Fourier features. The corresponding peak frequencies are labelled and denoted by the arrows in the figure.

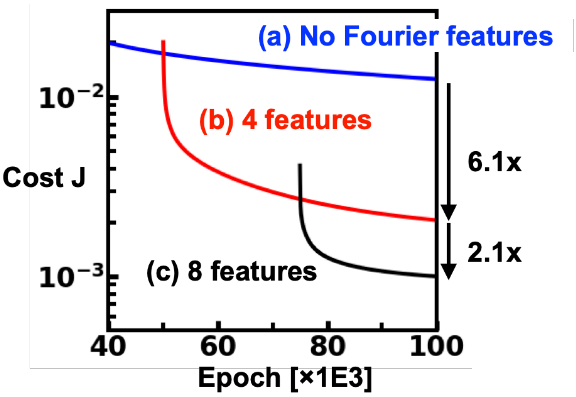

Fig. 12. Comparison of cost vs. epoch for (a) no Fourier features, (b) 4 pairs of features added after 50k epochs, and (c) an additional 4 pairs (total 8 features) added at 75,000 epochs.

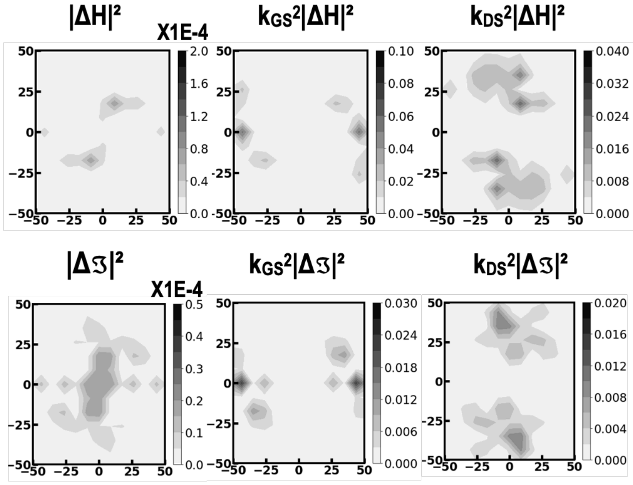

Fig. 13. Contour of the six cost components after 100k epochs of training, with 4 pairs of Fourier features.

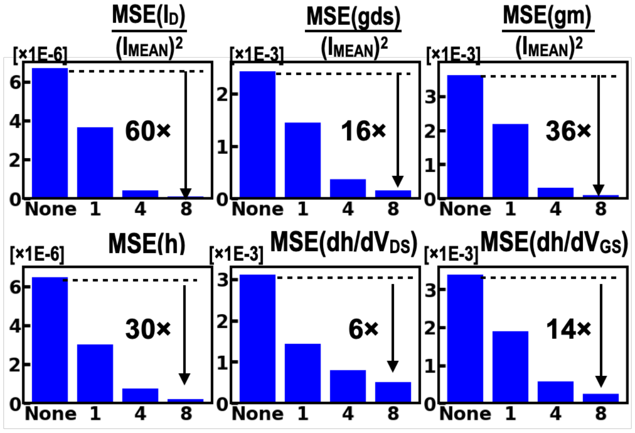

Fig. 14. Cost component breakdown for a neural network with varying pairs of Fourier features (None, 1, 4, and 8) after 100k epochs

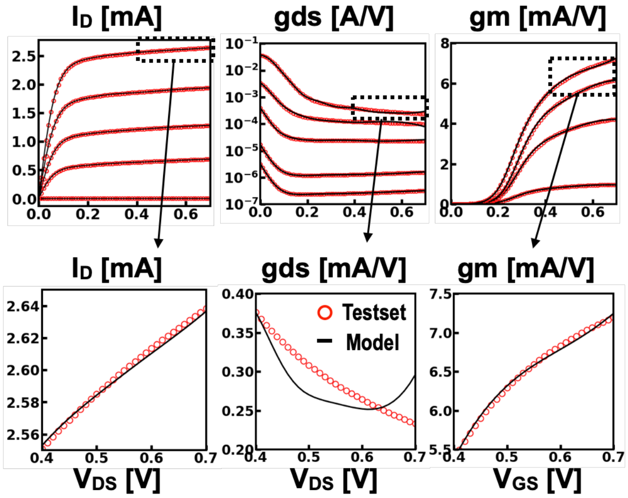

Fig. 15. Comparison between I_D test data and model prediction without Fourier features

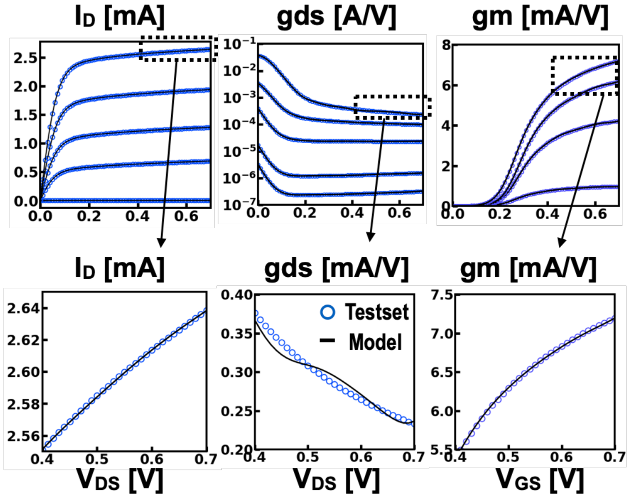

Fig. 16. Comparison between I_D test data and model prediction with 8 pairs of Fourier features

VI. Results and Discussions

We begin by adding a single frequency feature pair as an example. Our starting point is the neural network without Fourier features after 50k training. As shown in Fig. 6, the peak frequency of $|\Delta H|^2$ at 50k epochs is ± (17.45V^{-1}, 8.73V^{-1}). Given that $|\Delta H|^2$ is inherently symmetric, either peak can be selected for the Fourier feature:

$$\gamma_{sym}(V_{GS}, V_{GD}) = \begin{bmatrix} A \cdot \gamma_{cos}(V_{GS}, V_{GD}) \\ A \cdot \gamma_{sin}(V_{GS}, V_{GD}) \end{bmatrix} \quad (13)$$

where (Ω_{GS}, Ω_{DS}) = (17.45V^{-1}, 8.73V^{-1}) is peak frequency matched. For comparison, a non-peak-frequency matched Fourier feature at (Ω_{GS}, Ω_{DS}) = (40 V^{-1}, 40 V^{-1}) is utilized.

The half-trained network is then further trained for an additional 50k epochs. Different values of A were experimented, and it was found that A = 1E-3 strikes the right balance between small perturbation and fast convergence speed. The cost vs. epoch comparison in Fig. 8 shows that the peak-frequency-matched Fourier feature results in a 30% cost reduction compared to no Fourier feature, while the non-peak-frequency matched Fourier feature shows no improvement. Furthermore, as shown in Fig. 9, the frequency-matched Fourier feature effectively eliminates the peak at (k_{GS}, k_{DS}) = ± (17.45V^{-1}, 8.73V^{-1}). This, in turn, results in a 2.5× reduction in the mean square error for log(I_D), as illustrated in Fig. 10. The non-frequency-matched feature shows no effect, highlighting the significance of the improvement.

With a methodology established for utilizing Fourier features to minimize cost, the same method is now applied to improve the accuracies of derivative terms in the cost function. Fig. 11(a) shows the two-dimensional contour plot for the six cost components, $|\Delta H|^2$, $k_{GS}^2|\Delta H|^2$, $k_{DS}^2|\Delta H|^2$, $|\Delta\Im|^2$, $k_{GS}^2|\Delta\Im|^2$ and $k_{DS}^2|\Delta\Im|^2$, where $\Delta\Im \equiv \frac{m\times n}{I_{MEAN}^2} \times |\Im_{net} - \Im_{data}|^2$, after the same 50k epochs of training without the Fourier features. The corresponding peak frequencies are also in figure.

With these, four unique pairs of Fourier features are introduced, each intentionally matched with the peak frequencies:

$$(\Omega_{GS1}, \Omega_{DS1}) = (17.45\text{V}^{-1}, 8.73\text{V}^{-1}),$$
$$(\Omega_{GS2}, \Omega_{DS2}) = (-26.18\text{V}^{-1}, 8.73\text{V}^{-1}),$$
$$(\Omega_{GS3}, \Omega_{DS3}) = (-26.18\text{V}^{-1}, 17.45\text{V}^{-1}),$$

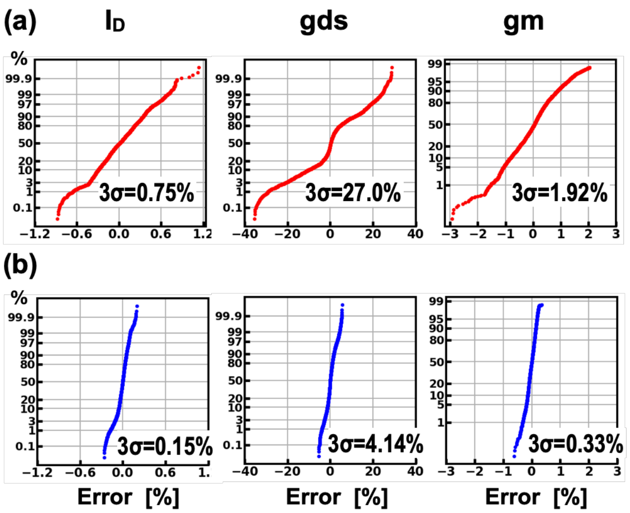

Fig. 17: Distribution of percentage error in I_D, gm, and gds test dataset for (a) without, (b) with 8 pairs of symmetric Fourier features.

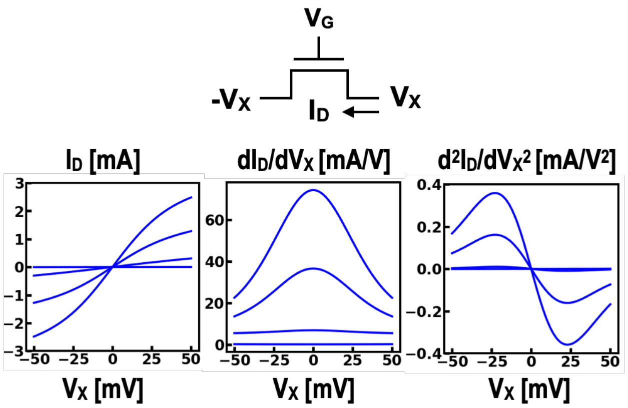

Fig. 18 Gummel symmetry test.

$(\Omega_{GS4}, \Omega_{DS4}) = (8.73\mathrm{V}^{-1}, 26.18\mathrm{V}^{-1})$ and A=1E-3. The neural network training is then continued for an additional 50k epochs.

The cost comparison in Fig. 12 shows that after 100k training epochs, these features result in a remarkable 6.1× cost reduction compared to the scenario without these features. Furthermore, by comparing Fig.11(b) and Fig. 13, they also effectively eliminate the error peak at the selected frequencies. This, in turn, leads to a remarkable cost reduction across the six cost components, as shown in Fig.14.

With the elimination of the highest error peaks, secondary error peaks can be iteratively suppressed by incorporating additional Fourier features. For example, utilizing the same neural network with 4 Fourier features, we conducted spectral analysis on the six cost terms at 75k epochs to identify the next peak frequencies. Subsequently, four additional, unique pairs (making a total of 8 pairs) frequency-matched features were incorporated. As illustrated in Fig. 12, the total cost is further reduced by 2.1 times after 100k training epochs, representing an overall ~13 times cost reduction compared to the case without the features. The detail breakdown of the cost reduction is summarized in fig. 14.

Fig. 15 and 16 compare the predicted I_D, gds, and gm accuracies for the test dataset for neural network with and without the eight symmetric features. As shown in the figure, Fourier features reduce error of the trained models, especially for gds and gm. This highlights the effectiveness of the Fourier features in capturing the frequency characteristics of the test dataset. As shown in Fig. 17, a neural network with 8 pairs of Fourier features results in a 3σ percentage error of 0.15% for I_D, 4.14% for gds, and 0.33% for gm. This represents more than a fivefold improvement compared to the scenario without Fourier features.

Furthermore, as depicted in Fig. 18, the trained model satisfies the Gummel symmetry. This aligns with expectations since ε and the Fourier features are symmetric by design, as previously explained.

VII. Conclusions

In this work, a highly accurate MOSFET *IV* model enhanced with deep-learning and symmetric Fourier features is proposed. Fourier analysis and spectral bias in conventional neural networks is discussed. By integrating Fourier features into the network architecture, our model achieves enhanced accuracies in predicting I_D, gm, and gds. Using simulated 12nm FinFET transistor as an example, our model achieves an exceptional 3-sigma error of 0.15% for I_D, 4.14% for gds, and 0.33% for gm, and results in a thirteen-fold cost reduction. To conclude, the combination of deep neural networks and Fourier features holds great promise for modeling next-generation MOSFETs and beyond.

Appendix

As discussed in [2], h(V_{GS}, V_{GD}) must be symmetric, i.e., h(V_{GS}, V_{GD}) = h(V_{GD}, V_{GS}), to ensure the model satisfies the Gummel symmetry. This implies $\gamma_{sym}(V_{GS}, V_{GD})$ is symmetric. To prove this, we swap the source and drain voltages:

$$\gamma_{cos,i}(V_{GD}, V_{GS}) = \sigma(V_{SD}/V_0)\cdot\cos(\Omega_{GSi}V_{GD} + \Omega_{DSi}V_{SD}) + \big(1-\sigma(V_{SD}/V_0)\big)\cdot\cos(\Omega_{GSi}V_{GS} - \Omega_{DSi}V_{SD})$$

Since $\sigma(-x) = 1 - \sigma(x)$,

$$\begin{aligned}\gamma_{cos,i}(V_{GD}, V_{GS}) &= \big(1-\sigma(V_{DS}/V_0)\big)\cdot\cos(k_{GSi}V_{GD} - k_{DSi}V_{DS}) \\ &\quad + \sigma(V_{DS}/V_0)\cdot\cos(k_{GSi}V_{GS} + k_{DSi}V_{DS}) \\ &= \gamma_{cos,i}(V_{GS}, V_{GD})\end{aligned}$$

Similarly, it can be proved that $\gamma_{sin,i}(V_{GD}, V_{GS}) = \gamma_{sin,i}(V_{GS}, V_{GD})$. Therefore $\gamma_{sym}(V_{GS}, V_{GD}) = \gamma_{sym}(V_{GD}, V_{GS})$.

References

[1] J. Wang, Y. -H. Kim, J. Ryu, C. Jeong, W. Choi and D. Kim, "Artificial Neural Network-Based Compact Modeling Methodology for Advanced Transistors," in *IEEE Transactions on Electron Devices*, vol. 68, no. 3, pp. 1318-1325, March 2021, doi: 10.1109/TED.2020.3048918.

[2] M.-Y. Kao, H. Kam and C. Hu, "Deep-Learning-Assisted physics-driven MOSFET current-voltage modeling", *IEEE Electron Device Lett.*, vol. 43, no. 6, pp. 974-977, Jun. 2022.

[3] C. -T. Tung and C. Hu, "Neural Network-Based BSIM Transistor Model Framework: Currents, Charges, Variability, and Circuit Simulation," in *IEEE Transactions on Electron Devices*, vol. 70, no. 4, pp. 2157-2160, April 2023, doi: 10.1109/TED.2023.3244901.

[4] C. -T. Tung, M. -Y. Kao and C. Hu, "Neural Network-Based and Modeling With High Accuracy and Potential Model Speed," in *IEEE Transactions on Electron Devices*, vol. 69, no. 11, pp. 6476-6479, Nov. 2022, doi: 10.1109/TED.2022.3208514.

[5] L. Zhang and M. Chan. "Artificial neural network design for compact modeling of generic transistors," in *Journal of Computational Electronics*, vol. 16, no. 3, pp. 825-832, Sep. 2017, doi: 10.1007/s10825-017-0984-9

[6] M. Tancik, et al. "Fourier features let networks learn high frequency functions in low dimensional domains." *Advances in Neural Information Processing Systems* 33 (2020): 7537-7547.

[7] A.V. Oppenheim, et al. *Signals and systems*. Vol. 2. Upper Saddle River, NJ: Prentice hall, 1997.

[8] IRDS 2020 Edition, 2020. [https://irds.ieee.org/editions/2020]

[9] D. B. M. Klaassen, "A unified mobility model for device simulation," *International Technical Digest on Electron Devices*, 1990, pp. 357-360, doi: 10.1109/IEDM.1990.237157.

[10] C. C. Enz and E. A. Vittoz, “Charge-Based MOS Transistor Modeling: The EKV Model for Low-Power and RF IC Design." *John Willey, New York* (2006).

[11] Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu. Towards understanding the spectral bias of deep learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

[12] N. Rahaman, et al. "On the spectral bias of neural networks." *International Conference on Machine Learning*. PMLR, 2019.